

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E
ELETRÔNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

Lucas Bez Rocha

**SISTEMA DE MONITORAÇÃO, CONTROLE E
AUTOMAÇÃO DE PARÂMETROS AMBIENTAIS EM
ESTUFAS AGRÍCOLAS UTILIZANDO CONCEITOS DE IOT**

**FLORIANÓPOLIS
2023**

LUCAS BEZ ROCHA

SISTEMA DE MONITORAÇÃO, CONTROLE E
AUTOMAÇÃO DE PARÂMETROS AMBIENTAIS EM
ESTUFAS AGRÍCOLAS UTILIZANDO CONCEITOS DE IOT

**Trabalho de Conclusão do Curso de Gra-
duação em Engenharia Elétrica do Cen-
tro Tecnológico da Universidade Federal
de Santa Catarina como requisito para
a obtenção do título de Bacharel em
Engenharia Elétrica**

Orientador: Prof. Joceli Mayer, Ph.D.

Florianópolis

2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Bez Rocha, Lucas
SISTEMA DE MONITORAÇÃO, CONTROLE E AUTOMAÇÃO DE
PARÂMETROS AMBIENTAIS EM ESTUFAS AGRÍCOLAS UTILIZANDO
CONCEITOS DE IOT / Lucas Bez Rocha ; orientador, Joceli
Mayer, 2023.
99 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Engenharia Elétrica, Florianópolis, 2023.

Inclui referências.

1. Engenharia Elétrica. 2. Automação. 3. Estufas
agrícolas. 4. Internet das Coisas. 5. Sistema de
monitoração. I. Mayer, Joceli. II. Universidade Federal de
Santa Catarina. Graduação em Engenharia Elétrica. III.
Título.

Lucas Bez Rocha

Sistema de Monitoração, Controle e Automação de Parâmetros Ambientais em Estufas Agrícolas Utilizando Conceitos de IoT

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Engenharia Elétrica” e aceito, em sua forma final, pelo Curso de Graduação em Engenharia Elétrica.

Florianópolis, 11 de dezembro de 2023.



Documento assinado digitalmente
Miguel Moreto
Data: 14/12/2023 08:32:31-0300
CPF: ***.850.100-**
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Miguel Moreto, Dr.
Coordenador do Curso de Graduação em Engenharia Elétrica

Banca Examinadora:



Documento assinado digitalmente
JOCELI MAYER
Data: 13/12/2023 15:48:48-0300
CPF: ***.833.519-**
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Joceli Mayer, Ph.D.
Orientador
Universidade Federal de Santa Catarina



Documento assinado digitalmente
Miguel Moreto
Data: 14/12/2023 08:32:49-0300
CPF: ***.850.100-**
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Miguel Moreto, Ph.D.
Universidade Federal de Santa Catarina



Documento assinado digitalmente
Richard Demo Souza
Data: 14/12/2023 07:05:16-0300
CPF: ***.267.379-**
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Richard Demo Souza, Ph.D.
Universidade Federal de Santa Catarina

Dedicado às maiores influências

Agradecimentos

Gostaria de expressar minha sincera gratidão ao meu orientador, Joceli Mayer, pela orientação paciente, pelo conhecimento valioso e pelo constante apoio durante este projeto. Suas sugestões e críticas foram fundamentais para o desenvolvimento deste trabalho.

Agradeço também aos professores e funcionários do departamento de engenharia elétrica da UFSC pela assistência e pelas valiosas contribuições ao longo de minha jornada acadêmica.

Um agradecimento especial aos meus colegas dentro e fora do curso, como Artur, Martin, Hugo e Otávio, pela amizade, pelo provimento de risadas não intencionadas, pelo incentivo e pelas discussões enriquecedoras que muito contribuíram para o meu crescimento profissional e pessoal.

À minha família, por seu amor incondicional, apoio emocional e por acreditarem em mim mesmo nos momentos mais desafiadores. Vocês foram minha força e inspiração.

Por fim, agradeço a todos os que, direta ou indiretamente, fizeram parte desta jornada e contribuíram para a realização deste trabalho.

Resumo

Este projeto é dedicado à justificativa e criação de um modelo e protótipo com potencial escalável de um sistema inteligente de monitoração e controle automatizado de uma estufa agrícola. O objetivo principal é fazer com que seja possível manter as condições ambientais otimizadas para a prevalência de uma cultura saudável usando componentes comerciais e os princípios IoT, proporcionando uma reprodutibilidade e utilização intuitiva e de baixo custo para o usuário. Todas as informações coletadas serão exibidas em uma interface gráfica, que será acessível com qualquer celular ou computador.

Palavras-chave: Automação. Agricultura de precisão. Estufas agrícolas. Internet das Coisas. Sistema de monitoração.

Abstract

This project is dedicated to the justification and creation of a model and prototype with scalable potential of a smart monitoring and automated control system of a greenhouse. The main objective is to make it possible to maintain optimized environmental conditions for the prevalence of a healthy cultivation using commercial components and IoT principles, providing an intuitive and low-cost reproducibility and utilization for the user. All the collected information will be displayed in a graphical interface, which will be accessible with any phone or computer.

Keywords: Agricultural greenhouses. Automation. Internet of Things. Monitoring system. Precision farming.

Lista de ilustrações

Figura 1 – Exemplo de uma estufa agrícola. Fonte: Sotrafa, 2021	28
Figura 2 – Modelo do sistema proposto. Fonte: [Zhang et al. 2007]	33
Figura 3 – Modelo e lógica do sistema proposto. Fonte: [Ardiansah et al. 2020]	34
Figura 4 – Representação do sistema NFT. Fonte: AGrowTonics, 2021	34
Figura 5 – Classificação de estufas de acordo com o formato do telhado. Fonte: [Kumar et al. 2006]	38
Figura 6 – Gráfico de dispersão de temperatura em Celsius (X) e umidade relativa em porcentagem (Y) com $R^2=0,7369$ de 28.906 dados de microclima. Fonte: [Ardiansah et al. 2021]	40
Figura 7 – Relação do espectro de frequências de radiação com a produtividade na atividade fotossintética. Fonte: [Timmerman e Kamp 2003]	41
Figura 8 – Arduino Uno Rev3. Fonte: Arduino, 2023	41
Figura 9 – ESP8266. Fonte: Espressif Systems, 2020	42
Figura 10 – Raspberry Pi 3 Model B+. Fonte: Adaptado de [Raspberry Pi Documentation 2023]	43
Figura 11 – DHT11. Fonte: [DHT11 Humidity and Temperature Sensor 2023]	44
Figura 12 – <i>Node</i> do sensor de temperatura e umidade. Fonte: [Shah e Bhatt 2017]	44
Figura 13 – YL-38 + YL-69. Fonte: Autor, 2023	45
Figura 14 – Sensor LDR. Fonte: Autor, 2023	45
Figura 15 – Relé de 5V. Fonte: Autor, 2023	46
Figura 16 – Exemplo de uma ventoinha de 120mm. Fonte: Autor, 2023	46
Figura 17 – Sinais PWM com 50% e 75% de <i>duty cycle</i> . Fonte: [Ullah, Gunawan e Muhida 2012]	47
Figura 18 – Mini bomba d’água utilizada. Fonte: Autor, 2023	47
Figura 19 – Lâmpada <i>Yeelight E27</i> que será utilizada. Fonte: Autor, 2023	48
Figura 20 – Gráfico de distribuição de aplicações das tecnologias de comunicação em sistemas de monitoramento de estufas agrícolas. Fonte: [Li et al. 2021]	49
Figura 21 – Estruturas de configuração de rede ZigBee. Fonte: ECSTUFF4U, 2023	50
Figura 22 – Funcionamento do sistema de QoS do MQTT. Fonte: Devopedia, 2022	51
Figura 23 – Fonte ATX de 200W. Fonte: Autor, 2023	56
Figura 24 – Diagrama de blocos com o fluxo de informações entre os dispositivos e camadas. Fonte: Autor, 2023	57
Figura 25 – Representação do circuito. Fonte: Autor, 2023 utilizando o <i>software</i> Fritzing	58
Figura 26 – Circuito do protótipo montado. Fonte: Autor, 2023	58

Figura 27 – Tela de terminal do Raspberry Pi acessado via chave SSH com o <i>software Putty</i> . Fonte: Autor, 2023	61
Figura 28 – Interface web da instalação Portainer. Fonte: Autor, 2023	62
Figura 29 – Configuração de domínio e <i>host</i> no Nginx. Fonte: Autor, 2023	64
Figura 30 – Fluxo <i>Node-RED</i> com gráficos e controle manual. Fonte: Autor, 2023	67
Figura 31 – <i>Dashboard</i> com gráficos e controle manual. Fonte: Autor, 2023	67
Figura 32 – Diagrama de blocos de um controlador PID. Fonte: [Santos, Júnior e Neves 2017]	70
Figura 33 – Resposta ao degrau do sistema com o sinal PWM máximo δ . Fonte: [Santos, Júnior e Neves 2017]	71
Figura 34 – Resposta do sistema com o controlador PID. Fonte: [Santos, Júnior e Neves 2017]	71
Figura 35 – Fluxograma da lógica de automatização dos atuadores. Fonte: Autor, 2023	72
Figura 36 – Intensidade de atuação para valores entre T_i e a soma de T_i com T_t . Fonte: Autor, 2023	73
Figura 37 – Intensidade de atuação para valores entre L_i e a soma de L_i com L_t . Fonte: Autor, 2023	77
Figura 38 – Exemplo de alerta enviado automaticamente via <i>e-mail</i> . Fonte: Autor, 2023	78
Figura 39 – Representação visual do sistema completo montado no Node-RED. Fonte: Autor, 2023	79
Figura 40 – Visão geral do protótipo. Fonte: Autor, 2023	81
Figura 41 – <i>Dashboard</i> com opções de controle e dados. Fonte: Autor, 2023	82
Figura 42 – Resultado do aumento gradativo de brilho de uma lâmpada a cerca de 25cm do LDR em um ambiente com pouca iluminação. Fonte: Autor, 2023	82
Figura 43 – Simulação de regulação dos atuadores. Fonte: Autor, 2023	83
Figura 44 – Representação das configurações possíveis utilizando múltiplos micro-controladores. Fonte: Autor, 2023	86
Figura 45 – Representação de um sistema de <i>intake</i> e <i>exhaust</i> . Fonte: Adaptado de AC Infinity	88

Lista de tabelas

Tabela 1 – Comparação de características do ESP32 e ESP8266. Fonte: [Umarov et al. 2020]	42
--	----

Lista de abreviaturas e siglas

AC	Alternating Current
ADC	Analog-to-Digital Converter
AI	Artificial Intelligence
API	Application Programming Interface
ATX	Advanced Technology Extended
CPU	Central Processing Unit
CSS	Cascading Style Sheets
DC	Direct Current
DNS	Domain Name System
DDNS	Dynamic Domain Name System
EC	Electrical Conductivity
GPG	GNU Privacy Guard
GPIO	General Purpose Input Output
GUI	Graphical User Interface
HDMI	High Definition Multimedia Interface
HID	High Intensity Discharge
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
I/O	Input/Output
I ² C	Inter-Integrated Circuit
IBM	International Business Machines Corporation
IC	Integrated Circuit
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers

IoT	Internet of Things
IP	Internet Protocol
JSON	JavaScript Object Notation
LAN	Local Area Network
LDR	Light Dependent Resistor
LNA	Low-Noise Amplifier
LoRa	Long Range
M2M	Machine to Machine
MAC	Media Access Control
MCU	Microcomputer
MQTT	MQ Telemetry Transport
NDIR	Nondispersive Infrared
NB-IoT	NarrowBand-Internet of Things
NFT	Nutrient Film Technique
PID	ProportionalIntegralDerivative
PWM	Pulse Width Modulation
OS	Operating System
QoS	Quality of Service
RAM	Random-Access Memory
RFID	Radio-Frequency Identification
SAR	Successive-Approximation
SD	Secure Digital
SDK	Software Development Kit
SNS	Social Networking Service
SoC	System on a Chip
SPI	Serial Peripheral Interface

SSL	Secure Sockets Layer
TLS	Transport Layer Security
UFSC	Universidade Federal de Santa Catarina
USB	Universal Serial Bus
VCC	Voltage Common Collector
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network

Lista de símbolos

C	Celsius
CO ₂	Dióxido de Carbono
cm	Centímetro
RH	Relative Humidity
GB	Gigabyte
GHz	Gigahertz
kbps	Kilobit per second
L/h	Liters per Hour
Li	Luminosidade ideal
lm	Lumen
Lt	Luminosidade de tolerância
Mbps	Megabit per second
mA	Milliampere
MHz	Megahertz
m	Meter(s)
mm	Millimeter
ms	Millisecond
RPM	Revolutions per Minute
Ti	Temperatura ideal
Tt	Temperatura de tolerância
V	Volt
W	Watt

Sumário

1	INTRODUÇÃO	27
1.1	Problemática	27
1.2	Motivação	28
1.3	Objetivos	30
1.3.1	Objetivo geral	30
1.3.2	Objetivos específicos	30
1.4	Organização do documento	30
2	ESTADO DA ARTE	31
2.1	Revisão de projetos realizados	31
3	METODOLOGIA	37
3.1	Tipo de estufa	37
3.2	Tipos de plantas	37
3.3	Parâmetros ambientais	38
3.3.1	Concentração de CO_2	38
3.3.2	Temperatura do ar e das folhas	38
3.3.3	Temperatura do solo	39
3.3.4	Umidade do ar e das folhas	39
3.3.5	Umidade do solo	39
3.3.6	Nível de pH, EC e temperatura d'água	39
3.3.7	Luminosidade	40
3.4	Opções de componentes	41
3.4.1	Microcontrolador	41
3.4.1.1	Arduino	41
3.4.1.2	ESP8266	42
3.4.2	Single Board Computer	43
3.4.2.1	Raspberry Pi	43
3.4.3	Sensores	43
3.4.3.1	Temperatura	43
3.4.3.1.1	DHT11	43
3.4.3.2	Umidade	44
3.4.3.2.1	YL-69 + YL-38	44
3.4.3.3	Luminosidade	45
3.4.3.3.1	LDR	45
3.4.4	Atuadores	45

3.4.4.1	Relé	45
3.4.4.2	Temperatura, circulação e umidade do ar	46
3.4.4.2.1	Ventoinha	46
3.4.4.3	Umidade e temperatura do solo	46
3.4.4.4	Luminosidade	47
3.5	Opções de software e tecnologias	48
3.5.1	Protocolos de comunicação e mensagem	48
3.5.1.1	Com fio	48
3.5.1.2	Sem fio	48
3.5.2	ZigBee	49
3.5.3	LoRa	49
3.5.4	Bluetooth	50
3.5.5	Wi-Fi	50
3.5.6	MQTT	50
3.5.7	Editores e compiladores	52
3.5.7.1	ThingSpeak	52
3.5.7.2	Node-RED / JavaScript	52
3.5.8	Software	52
3.5.8.1	Arduino IDE / C++	52
3.5.8.2	Raspberry Pi OS	52
3.5.8.3	Docker / Portainer	52
3.5.8.4	Nginx	53
4	CONSTRUÇÃO DO PROJETO	55
4.1	Lista de componentes	55
4.2	Modelo	56
4.2.1	Lógica de funcionamento	56
4.2.1.1	Camada de medições e atuações	56
4.2.1.2	Camada de processamento	56
4.2.1.3	Camada de aplicativo	57
4.2.2	Circuito	57
4.2.2.1	Sensores	57
4.2.2.1.1	Temperatura e umidade do ar	57
4.2.2.1.2	Umidade do solo	59
4.2.2.1.3	Luminosidade	59
4.2.2.2	Atuadores	60
4.2.2.2.1	Temperatura e umidade do ar	60
4.2.2.2.2	Umidade do solo	60
4.3	Configuração dos dispositivos	60
4.3.1	Programação do microcontrolador	60

4.3.2	Lâmpada	61
4.3.3	Configuração do Raspberry Pi	61
4.3.3.1	Raspberry Pi OS	61
4.3.4	Docker / Portainer	62
4.3.4.1	Eclipse Mosquitto	63
4.3.4.2	Node-RED	63
4.3.4.3	Nginx	64
4.3.4.4	Motion	64
4.4	Controle e monitoramento	65
4.4.1	Sensores	66
4.4.2	Controle manual	66
4.4.2.1	Ventoinha	66
4.4.2.2	Bomba d'água	66
4.4.2.3	Lâmpada	66
4.4.3	Controle automatizado	66
4.4.3.1	Sistemas de controle	69
4.4.3.1.1	Temperatura e circulação de ar	73
4.4.3.1.2	Umidade	74
4.4.3.1.3	Luminosidade	75
4.5	Sistemas de contingência	77
5	RESULTADOS	81
5.1	Testes e simulação	81
6	CONCLUSÃO	85
6.1	Síntese	85
6.2	Próximos passos	86
6.2.1	Alternativas e melhorias	86
6.2.2	Sugestões de possíveis modificações	87
	REFERÊNCIAS	89
	APÊNDICES	93
	APÊNDICE A – CÓDIGO EM C++ FINAL UTILIZADO NO ESP8266	95

1 INTRODUÇÃO

1.1 Problemática

O crescimento populacional exponencial, junto com uma demanda de maior qualidade de alimentos vegetais [Barbosa 2019] tornou a utilização de estufas (figura 1) um método popular, levando a fabricação de estufas a ser a indústria de crescimento mais rápido no mundo [Sushmitha, Jyothi e Sain 2022], com países como o Japão dedicando 42.000 hectares de terra à tecnologias similares [Ardiansah et al. 2020]. Estufas agrícolas (figura 1) são espaços fechados cobertos com um material transparente dedicadas ao cultivo de plantas, sendo possível ter maior controle sobre os parâmetros ambientais, proporcionando uma consistência de condições às plantas similares às da época de safra [Deus 2022], o que aumenta a produtividade e qualidade do que é produzido. No Brasil, o PIB nominal do agronegócio tem um valor estimado de R\$2,7 trilhões em 2023, o que representa 24,8% do PIB nominal brasileiro, contando com um aumento de 92,8% nos últimos 5 anos e uma tendência de aumento no setor primário quando comparado aos adjacentes [CEPEA 2023]. No país, se destaca a utilização de estufas para o cultivo de hortaliças, que representou um aumento na rentabilidade para os agricultores, já que o gerenciamento adequado das condições resulta em um desenvolvimento das plantas favorável, economia com água e energia e em descobertas de problemas mais cedo [Costa et al. 2021].

Devido à necessidade de constante monitoramento, variações imprevisíveis de tempo em diversos climas e pragas, é comum se deparar com catástrofes e casos de grandes perdas em plantações de comunidades de portes pequeno e médio, que são responsáveis pela maior parte de atividades agrícolas. Sendo a única fonte de renda de muitas pessoas ainda utilizando métodos de agricultura tradicionais, os problemas causados afetam indiretamente a população com demanda dos produtos e diretamente quem é dependente da comercialização deles, deixando-os para trás dentro de um mercado competitivo [Ardiansah et al. 2021] e, em diversos casos, causa um impossibilidade em os agricultores pagarem empréstimos feitos de volta, frequentemente levando-os ao suicídio [Arshad et al. 2020]. Além dos outros problemas apresentados, existe uma constante limitação de locais apropriados para a prática da agricultura, seja pelo encolhimento de terras adequadas, crescimento da urbanização ou condições climáticas e limitações impostas por temporadas [Ardiansah et al. 2020].

Com a elevação do impacto de intervenção humana na natureza nas últimas décadas, a mudança climática se tornou uma inevitabilidade [Anire, Cruz e Agulto 2017], e a emissão de gases de efeito estufa por produções agrícola atinge 30% do que é emitido mundialmente



Figura 1 – Exemplo de uma estufa agrícola. Fonte: Sotrafa, 2021

por atividades humanas, indicando uma urgência em diminuir o consumo de recursos pela indústria [Li et al. 2021].

1.2 Motivação

O controle de variáveis ambientais necessita de um sistema bem planejado, com a presença de equipamentos como sensores, aquecedores e lâmpadas para proporcionar um controle aproximadamente preciso e individual de cada um dos parâmetros medidos. A relevância do tópico deduzida a partir do crescente número de estudos na área [Ardiansah et al. 2020], impacto ambiental global e tendência de aumento de demanda da indústria agrícola reforça a importância do aperfeiçoamento de modelos competentes aplicáveis em diferentes casos e aumento de eficiência dos sistemas empregados. A utilização de um monitoramento manual de um sistema controlável depende de observações e ações com base em intuição, o que pode resultar em divergências dos parâmetros ideais, que resultam em problemas como redução no metabolismo e morte de plantas e surgimento de pragas e doenças.

As estufas da geração anterior utilizavam estruturas muito simples, com apenas um ponto de medidas e sem possibilidade de controle de parâmetros ambientais locais. Porém, as estufas agrícolas modernas são de escalas maiores, e provêm uma granularidade em seu

controle muito maior [Ahonen, Virrankoski e Elmusrati 2008]. Dentro das possibilidades de diferentes abordagens modernas, se destaca a utilização do conceito de *IoT*, que impõe uma comunicação entre dispositivos eletrônicos utilizados no cotidiano para seguir um objetivo comum [Costa et al. 2021]. No entanto, uma estufa que depende unicamente de monitoramento ainda estará sujeita à dedicação de tempo elevada e à erros humanos [Sushmitha, Jyothi e Sain 2022], criando a necessidade de uma automatização dos atuadores que influenciam os parâmetros ambientais seguindo um método científico. O emprego de uma série de sensores e atuadores no mesmo ambiente utilizando este princípio abre um leque de alternativas acessíveis para o controle de diversos parâmetros do espaço interno de uma estufa, protegendo os cultivos de grandes variações climáticas utilizando um consumo de energia baixo, resultando em um custo reduzido de recursos e energia ao usuário, confiabilidade maior no processo de colheita e gerenciamento de risco devido à previsibilidade de condições, velocidade maior de resposta aos problemas, economizando tempo e dinheiro, redução no uso de pesticidas e redução na emissão de gases de efeito estufa [Li et al. 2021].

Tendo seu início em aplicações militares, as tecnologias sem fio tiveram grandes avanços nas últimas décadas, sendo, hoje em dia, possível aplicá-las em situações de controle industrial de maneira simples e confiável utilizando produtos disponíveis comercialmente [Zhang et al. 2007]. O avanço e versatilidade dos protocolos de comunicação modernos e microcontroladores proporcionam uma adaptação e rapidez de aplicação no contexto da agricultura de precisão de pouco custo, tendo potencial de amenizar os problemas de disponibilidade de terra e mudanças climáticas mencionados anteriormente e ajudar na demanda crescente de alimentos. O *IoT* é considerado a terceira onda de tecnologia da informação, seguindo a *internet* e comunicações móveis, caracterizada por medidas precisas, interoperabilidade de função e inteligência, sendo que cada componente responsável por algum tipo de comunicação possui um *IP* designado para ser identificado na rede local [Danita et al. 2018].

A tentativa de individualizar e controlar parâmetros ambientais, é, no entanto, uma tarefa extremamente desafiante. Apesar de estufas agrícolas funcionarem em um ambiente fechado, os parâmetros são codependentes, resultando em um sistema não-linear e complexo. Portanto, não existe um modelo físico até então criado que indica a lei entre parâmetros de forma satisfatória, sendo necessário uma análise para cada tipo de aplicação [Li et al. 2021].

1.3 Objetivos

1.3.1 Objetivo geral

O objetivo principal do Trabalho de Conclusão de Curso a ser realizado é a modelagem de um sistema automatizado de monitoração e controle de uma estufa de plantas para que seja possível manter as condições ambientais e climáticas otimizadas para a prevalência de uma cultura saudável.

1.3.2 Objetivos específicos

- Um protótipo simples e de baixa escala com atuadores controláveis e sensores será montado utilizando componentes comerciais, tendo em vista analisar e colocar em prática a viabilidade de construção de um projeto com objetivos menos generalizados e de maiores proporções.
- Uma interface gráfica, acessível de qualquer local remotamente via *web* por qualquer aparelho celular ou computador, será disponibilizada de maneira segura, onde o usuário terá todas as informações de parâmetros ambientais relevantes.
- Com possibilidade de aplicações tanto em âmbitos residenciais quanto de grande escala, o projeto propõe uma maneira de reduzir os gastos e consumo do monitoramento de plantações enquanto automatiza o processo e, conseqüentemente, reduz a frequência de erros humanos.
- Através de simulações, o sistema será conceitualizado e testado para as aplicações definidas, devendo seguir o comportamento lógico de atuação determinado.

1.4 Organização do documento

Este texto será estruturado da seguinte forma:

Capítulo 2: Revisão de estudos e projetos similares e de áreas adjacentes.

Capítulo 3: Dedicado à pesquisa e justificativas de métodos, componentes e parâmetros relativos à execução do trabalho.

Capítulo 4: Modelagem, escolha de componentes e construção do projeto.

Capítulo 5: Resultados das medidas e automação dos parâmetros ambientais, bem como análise de comportamento.

Capítulo 6: Conclusão relativa ao projeto e à proposta, bem como sugestões de alterações possíveis para garantir uma melhoria e alternativas de execução.

2 ESTADO DA ARTE

Existem diversos projetos criados com o objetivo de controle de parâmetros ambientais com estrutura e objetivos similares aos principais pontos apontados no capítulo 1. Em sequência, uma revisão de diversos artigos com foco em monitoramento de ambientes de estufas agrícolas utilizando conceitos de *IoT* será feita para que seja possível tomar uma decisão de quais componentes e princípios de funcionamento serão utilizados para a realização deste projeto com base na praticidade de aplicação, utilidade e disponibilidade. Por definição, o termo 'atuador' será utilizado para descrever os componentes responsáveis por influenciar e modificar os parâmetros ambientais.

2.1 Revisão de projetos realizados

1. O *paper* de [Danita et al. 2018] utiliza o princípio de monitoramento de 3 parâmetros ambientais, sendo que o protótipo montado visa as condições ideais do vegetal espinafre:
 - Temperatura e umidade do ar a partir do sensor DHT11 instalado longe da influência dos atuadores.
 - Umidade do solo utilizando o sensor YL-69, instalado próximo às raízes das plantas.

Os dados são então coletados e guardados em uma base de dados na nuvem, acessível em qualquer lugar e momento, e a partir da análise das condições enviadas ao MCU *Raspberry Pi*, uma série de automações dos atuadores serão determinadas a partir dos limites definidos, ativando-os quando os valores lidos pelos sensores forem ultrapassados. Os atuadores selecionados são:

- Um tanque de água controlado por motor CC (IC L293D) conectado à tubos d'água que irá irrigar a plantação até que o valor de umidade do solo ideal de 50% seja atingido.
- Uma ventoinha em conjunto com janelas deslizantes instaladas nas paredes da estufa para manter o controle de temperatura e circulação de ar, sendo ativados quando os valores de temperatura e umidade do ar atingirem leituras maiores que o calculado para o tipo de plantação escolhido. O valor experimental máximo de 26,4 que será seguido como base de atuação para os valores de umidade de 14% e de temperatura de 25°C, representado pela variável *idx*, foi

determinado através da equação 2.1:

$$idx = Temperatura + (Umidade \cdot 0,1) \quad (2.1)$$

A estufa foi dividida em múltiplas seções, tendo um sensor YL-69 dedicado à cada seção, e os sensores são conectados diretamente ao GPIO do Raspberry Pi. Utilizando o módulo *Wi-Fi* presente no *Raspberry Pi*, todos os dados são sincronizados com o *ThingSpeak Cloud*, uma plataforma IoT que possibilita a coleta, processamento e análise de dados em tempo real, sendo transformados em gráficos disponíveis na página *web* desenvolvida em HTML e CSS.

2. O estudo de [Li et al. 2021] aponta para um sistema com 5 fatores ambientais principais interdependentes; temperatura, umidade relativa do ar, luz, umidade do solo e conteúdo de CO_2 , e 2 parâmetros de colheita; temperatura e umidade das folhas. Dentro das opções de central de processamento, os de maior relevância que foram citados são:

- MCU
- Arduino
- Plataforma TelosB
- Raspberry Pi

O artigo também ataca em detalhe a seleção de protocolo de comunicação, sendo dividido em comunicações com fio e sem fio. Dentro do segundo modo, as opções que devem ser consideradas são:

- ZigBee
- LoRa
- NB-IoT
- Bluetooth
- Protocolos privados

3. O protótipo criado por [Zhang et al. 2007] utiliza como base o padrão de comunicação *ZigBee* e emprega o uso de sensores sem fio de luminosidade, temperatura e umidade integrados através de um PIC 16F877. As informações são então transmitidas ao controlador, onde os dados são exibidos em uma tela LCD e um algoritmo toma as decisões de ativação dos atuadores (figura 2).

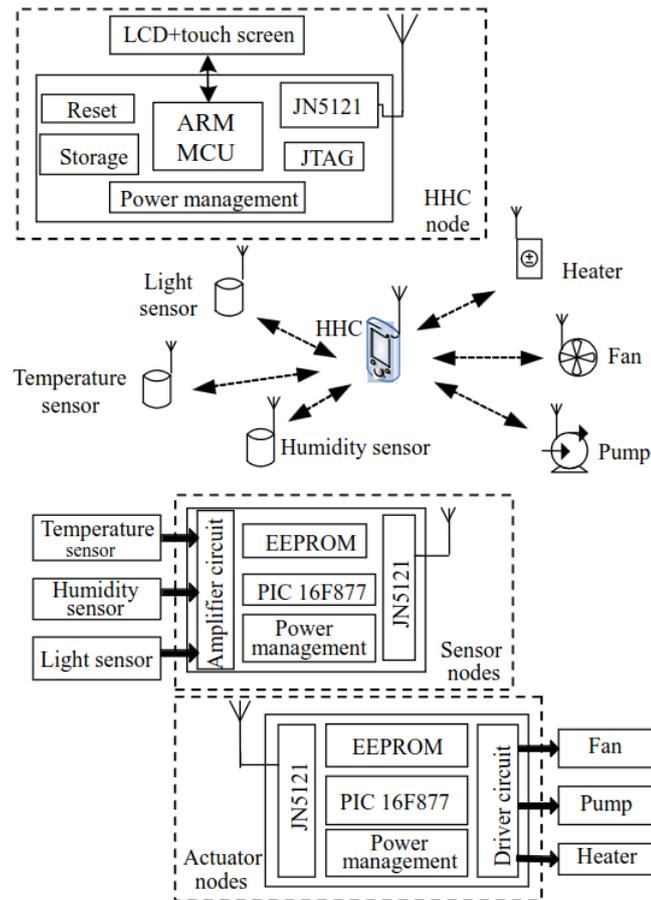


Figura 2 – Modelo do sistema proposto. Fonte: [Zhang et al. 2007]

4. O modelo criado em [Ardiansah et al. 2020] utiliza o microcontrolador Arduino conectado à 6 sensores ambientais, conforme a figura 3, incluindo sensores de chuva, pressão, qualidade e velocidade do ar e detector de granizo. Um dos atuadores incluídos foi a inserção de um sinal *buzzer* para o controle de pragas, que imita a frequência do som emitido pela *Diaphorina citri* fêmea para atrair a população macho à um *spray* pesticida. Para acompanhar o local e quantidade de *spray* utilizado, um sistema RFID foi implementado, que registra e comunica todos os dados necessários de dispositivos móveis equipados com um *spray* pesticida, incluindo a posição e direção.
5. O trabalho feito por [Barbosa 2019] utiliza a técnica da hidroponia, cultivando as plantas através do contato das raízes com um meio híbrido que contém os nutrientes essenciais para o desenvolvimento da planta, e aplica o sistema NFT, no qual uma bomba é utilizada para circular a solução nutritiva do reservatório através de canais (figura 4). O modelo necessita de sensores para realizar o monitoramento do pH e EC (*Electrical Conductivity*) da água, tendo atuadores para corrigir as variáveis automaticamente, junto com uma série de sensores meteorológicos para acompanhamento dos parâmetros ambientais.

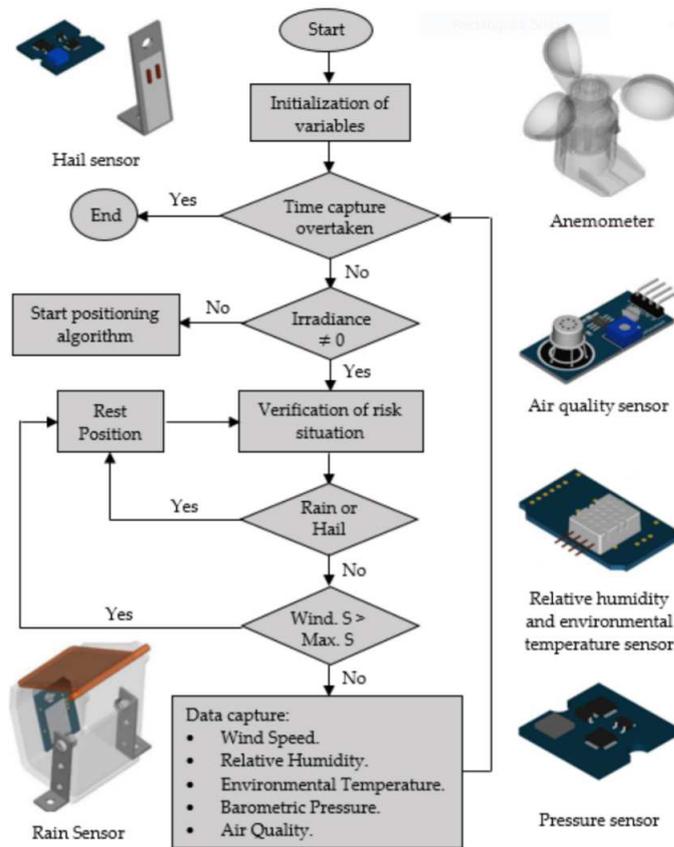


Figura 3 – Modelo e lógica do sistema proposto. Fonte: [Ardiansah et al. 2020]



Figura 4 – Representação do sistema NFT. Fonte: AGrowTonics, 2021

Os demais projetos utilizados de referência, como em [Ganesan et al. 2018], [Arshad et al. 2020] e [Ahonen, Virrankoski e Elmusrati 2008], usufruem de uma estrutura similar aos citados, com variações de sensores, protocolos e controladores escolhidos.

3 METODOLOGIA

Para que o projeto possa ser executado de maneira eficiente, diversos conceitos e classificações devem ser analisados para que seja possível tomar decisões de escolhas de estrutura e componentes congruentes com a proposta do trabalho.

3.1 Tipo de estufa

A classificação de estufas agrícolas se dá pelo formato do telhado, e são divididos em 8 tipos, conforme a figura 5:

- i Domo esférico
- ii Parabolóide hiperbólico
- iii *Quonset*
- iv *Quonset* modificado
- v Arco gótico
- vi Telhado de mansarda
- vii *Evenspan*
- viii *Unevenspan*

O formato mais comumente utilizado é o *Evenspan* [Ardiansah et al. 2020]. No entanto, devido à simplicidade de replicação, escala pequena e uniformidade da estrutura, o modelo utilizado no protótipo e otimizações terão como base o domo esférico.

3.2 Tipos de plantas

Estufas agrícolas são, usualmente, especializadas para o cultivo de plantas de período de curto crescimento e colheita, como nos casos de tomate, feijão, milho e pimenta [Ardiansah et al. 2020]. Um dos pontos principais da proposta do projeto é proporcionar uma versatilidade de aplicação, removendo a necessidade de localidade ou aplicação específica. O modelo será criado tendo em mente a alta customização no que diz respeito à seleção de parâmetros ambientais ideais, que irá cobrir uma vasta gama de vegetais comumente cultivados, servindo como uma fundação sólida e deixando espaço para modularidades e adaptações para uma aplicação otimizada em casos específicos.

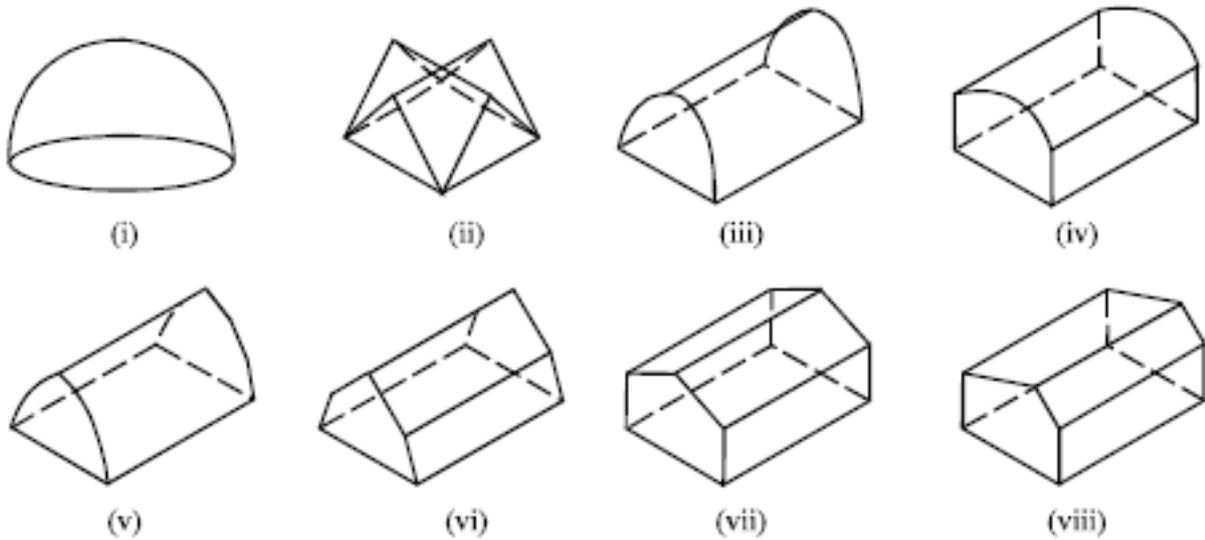


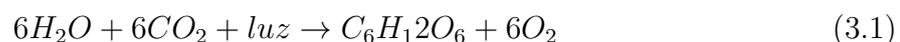
Figura 5 – Classificação de estufas de acordo com o formato do telhado. Fonte: [Kumar et al. 2006]

3.3 Parâmetros ambientais

Um dos aspectos mais importantes a ser decidido na criação de uma estufa agrícola se dá na escolha dos parâmetros ambientais que serão monitorados. A seleção neste projeto se dará através da análise e balanço de fatores como importância e relevância do parâmetro durante o cultivo, facilidade e precisão de leitura e controle através de atuadores, correlação local com outros parâmetros e custo de implementação e manutenção. A revisão bibliográfica auxiliou em revelar os critérios que são mais comumente considerados:

3.3.1 Concentração de CO_2

O dióxido de carbono é uma das matérias-primas utilizada pelas plantas para seu crescimento através da fotossíntese. No entanto, os métodos de aumento de concentração de CO_2 são caros e podem ser perigosos [Li et al. 2021], tornando uma aplicação do conceito em baixa escala inviável. O processo fotossintético se dá pela equação 3.1, indicando a importância do CO_2 correlacionando a presença de luz com a quantidade de dióxido de carbono [Ahonen, Virrankoski e Elmusrati 2008].



3.3.2 Temperatura do ar e das folhas

A temperatura é um dos parâmetros críticos que afetam diretamente o crescimento, rendimento e qualidade da colheita, fazendo com que o objetivo de manter uma faixa de temperatura aceitável seja de suma importância. Uma opção de monitoramento da

temperatura das plantas gira em torno do acompanhamento da temperatura das folhas, ajudando a mensurar a concentração de vapor de água presente, o que afeta a transpiração [Li et al. 2021], que é um processo de liberação de vapor d'água pelo estômato [Ahonen, Virrankoski e Elmusrati 2008].

3.3.3 Temperatura do solo

Afetando as enzimas, oligoelementos e crescimento das plantas e possuindo correlação com a temperatura ambiente, a elevação moderada de temperatura do solo pode levar ao aumento de biomassa e colheita em 21% [Noda, Kobayashi e Suda 2001], porém, uma temperatura acima da faixa adequada pode causar doenças [Li et al. 2021]. As temperaturas do solo e folhas serão indiretamente controladas através do monitoramento da temperatura do ar.

3.3.4 Umidade do ar e das folhas

A umidade relativa impacta o crescimento da colheita, doenças, pragas e volume, já que a mudança do parâmetro afeta a alteração de biomassa, mudando a área de folha e o peso seco das plantas. Um valor de umidade abaixo do ideal pode levar à um estresse hídrico, e problemas de transpiração e fotossíntese podem ser causados em valores altos. Se mostra necessário, também, a implementação de um intervalo aceitável dinâmico de umidade para evitar a presença de praga de insetos. Similarmente ao monitoramento de temperatura, sensores de maior precisão podem ser utilizados para acompanhar a umidade nas folhas [Li et al. 2021].

A umidade do ar pode ser indiretamente controlada através da regulação da temperatura devido à sua correlação negativa alta, conforme a figura 6.

3.3.5 Umidade do solo

A umidade do solo é a base dos processos físicos das plantas, mantendo as funções do ecossistema e afetando a qualidade de colheita e área foliar. Os valores ideais também são diferentes dependendo da temporada, e as estufas, por isolarem as plantas de qualquer contato com chuva, devem prover a quantidade correta de umidade ao solo para que o fornecimento de água não seja em excesso, levando à um aumento de gastos e apodrecimento da raiz, ou escasso, podendo levar à morte da planta [Li et al. 2021].

3.3.6 Nível de pH, EC e temperatura d'água

A absorção de nutrientes de proporções adequadas é influenciada pelos parâmetros de pH, EC e temperatura d'água, que aumentam a confiança de estabilidade no desenvolvimento das plantas. Uma granularidade e controle maior desses parâmetros é relevante

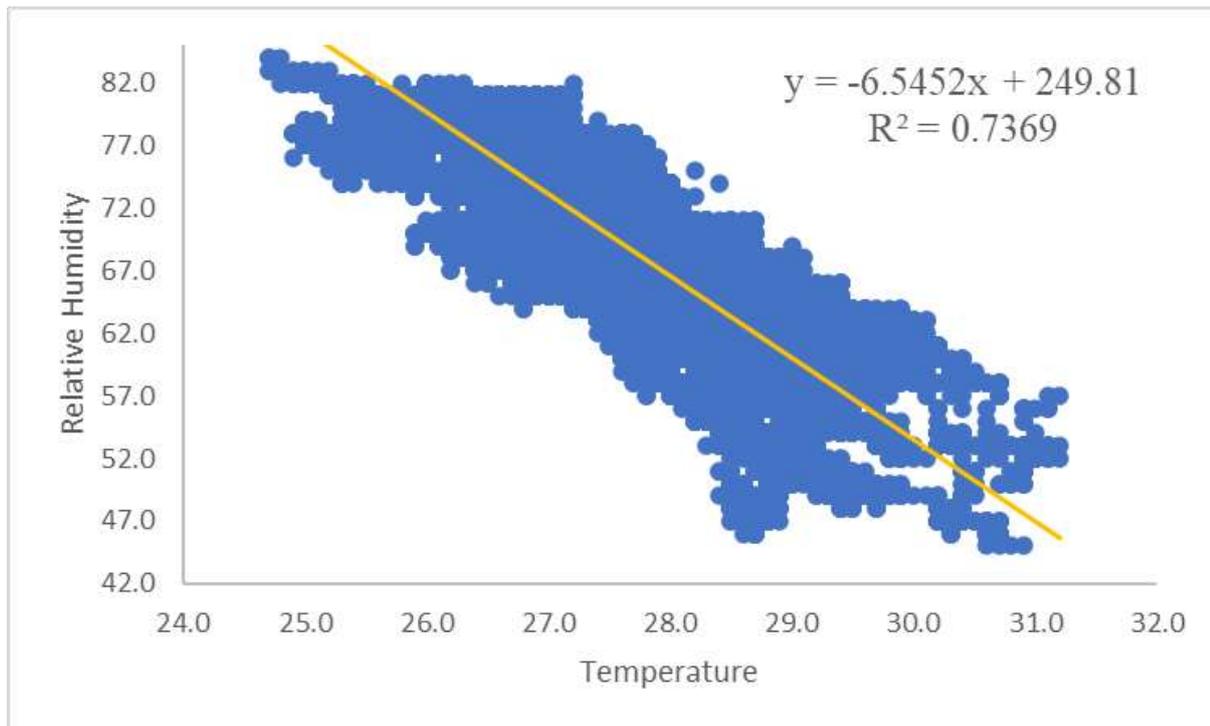


Figura 6 – Gráfico de dispersão de temperatura em Celsius (X) e umidade relativa em porcentagem (Y) com $R^2=0,7369$ de 28.906 dados de microclima. Fonte: [Ardiansah et al. 2021]

para aplicações mais específicas, como nos modelos de hidroponia [Barbosa 2019], e não será implementado no presente projeto devido à maior complexidade de monitoramento e regulação.

3.3.7 Luminosidade

A intensidade de luz é um fator essencial para a fotossíntese, aumentando, em média, de 0,5% a 1% a produção para cada 1% de ganho de incidência de luz, e pode afetar positivamente a qualidade dos frutos, como indicado por estudos feitos em morangos [Tang et al. 2020]. Os valores ideais de luminosidade alteram para cada planta, e possuem diferentes faixas de acordo com a temperatura local [Li et al. 2021], tipicamente necessitando de luz durante 10 a 12 horas por dia, e aumentando para 16 horas quando produzindo flores ou frutos [Ahonen, Virrankoski e Elmusrati 2008]. Também é necessário considerar o espectro de luz utilizado, dado que a cor da luz incidente afeta diretamente a atividade fotossintética da planta, conforme a figura 7.

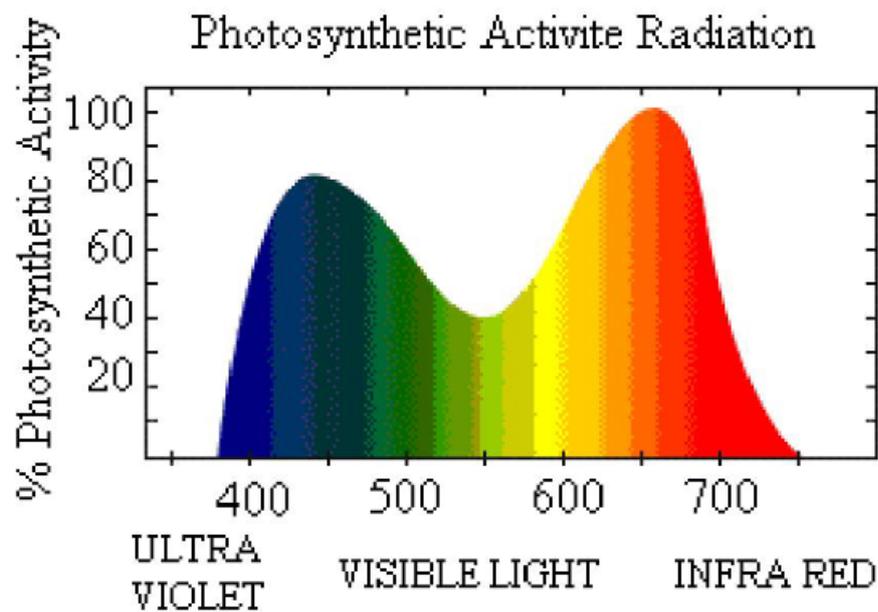


Figura 7 – Relação do espectro de frequências de radiação com a produtividade na atividade fotossintética. Fonte: [Timmerman e Kamp 2003]



Figura 8 – Arduino Uno Rev3. Fonte: Arduino, 2023

3.4 Opções de componentes

3.4.1 Microcontrolador

A escolha de um microcontrolador é necessária para ser utilizado como central de processamento e comunicação dos sensores e atuadores, e os principais pontos a serem considerados serão os recursos disponíveis e o custo.

3.4.1.1 Arduino

As placas *Arduino* são uma família de microcontroladores de código aberto, sendo o *Arduino UNO* (figura 8) uma das variações mais populares, possuindo o *chipset* ATmega328 [Ardiansah et al. 2021].

3.4.1.2 ESP8266

O ESP8266 (figura 9) é uma série microcontroladores de alimentação 3,3V DC com SoC de baixo custo e *Wi-Fi* integrado que utiliza de base a CPU *Tensilica Xtensa LX3*. Sua programação também pode ser feita através do *Arduino IDE* por ter um ecossistema estabelecido ao redor do SDK *Arduino* [Grokhotkov 2017], e sua escolha é favorável ao *Arduino* devido à conectividade sem fio integrada, já que o modelo do projeto necessita de diversos *nodes* (pontos de separação onde são feitas múltiplas conexões) de sensores, o que inviabilizaria conexões físicas ao microcomputador. É necessário considerar como possibilidade de utilização a sua versão atualizada e melhorada denominada ESP32, que pode ser mais otimizada e reduzir os custos e número de microcontroladores necessários dependendo da aplicação e escala, conforme demonstrado na tabela 1.

Tabela 1 – Comparação de características do ESP32 e ESP8266. Fonte: [Umarov et al. 2020]

ESP32	ESP8266
SAR-ADC de 16 canais e 12 bits com LNA	ADC de 10 bits sem LNA
2 interfaces I^2C	1 interface I^2C
16 canais para modulação de frequência de banda larga	8 canais para modulação de frequência de banda larga
36 GPIOs	17 GPIOs
4 interfaces SPI com <i>Quad-SPI</i> e com uma frequência máxima de 80 MHz	3 interfaces SPI com <i>Quad-SPI</i> e com uma frequência máxima de 80 MHz
CPU <i>dual-core</i> com frequência de 160MHz a 240MHz	CPU <i>single-core</i> com frequência de 80MHz

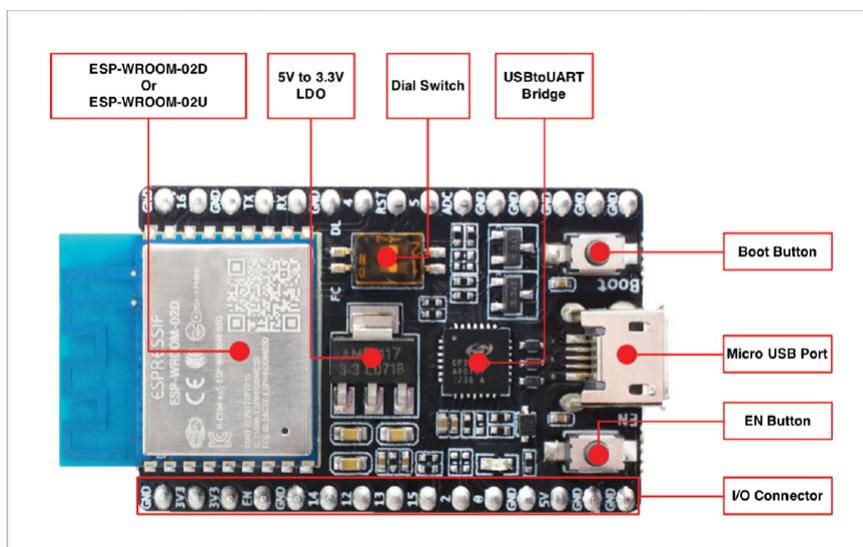


Figura 9 – ESP8266. Fonte: Espressif Systems, 2020

3.4.2 Single Board Computer

3.4.2.1 Raspberry Pi

O *Raspberry Pi* é uma família de computadores de baixo custo e placa única com SoC que utilizam processadores da arquitetura ARM. As versões consideradas para este projeto utilizam revisões igual ou superior ao Raspberry Pi 3 (figura 10), já que possuem características essenciais como conectividade *Wi-Fi* IEEE 802.11n integrada utilizando o *chip* Broadcom BCM43438 [Shah e Bhatt 2017], uma GPIO de 40 pinos, frequência da CPU de 1,2GHz e 1GB de RAM [Danita et al. 2018]. A placa é alimentada através de uma tensão de 12V DC e sua I/O inclui uma saída mini HDMI, 4 entradas USB, bem como uma entrada de cartão micro SD que é utilizada como armazenamento e disco de inicialização do OS [Raspberry Pi Documentation 2023].

3.4.3 Sensores

3.4.3.1 Temperatura

3.4.3.1.1 DHT11

O DHT11 (figura 11) é um dos sensores com saída digital mais populares e básicos utilizado para a medição da temperatura e umidade do ambiente, operando entre 3V e 5V DC e possuindo uma corrente máxima de 2,5mA. Seu funcionamento se dá através da utilização de resistores variáveis influenciados por um filme piroelétrico para identificação de temperatura e um higrômetro para a umidade, ambos passando por um sistema ADC (figura 12) em comum antes de serem enviados à saída [Shah e Bhatt 2017]. A faixa de temperatura reconhecida está entre 0°C e 50°C e a de umidade entre 20% RH e 80% RH com precisão de 2% [Danita et al. 2018] e, apesar de essas especificações serem inferiores



Figura 10 – Raspberry Pi 3 Model B+. Fonte: Adaptado de [Raspberry Pi Documentation 2023]

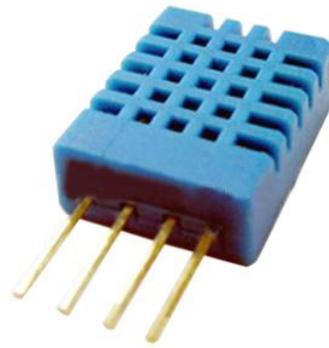


Figura 11 – DHT11. Fonte: [DHT11 Humidity and Temperature Sensor 2023]

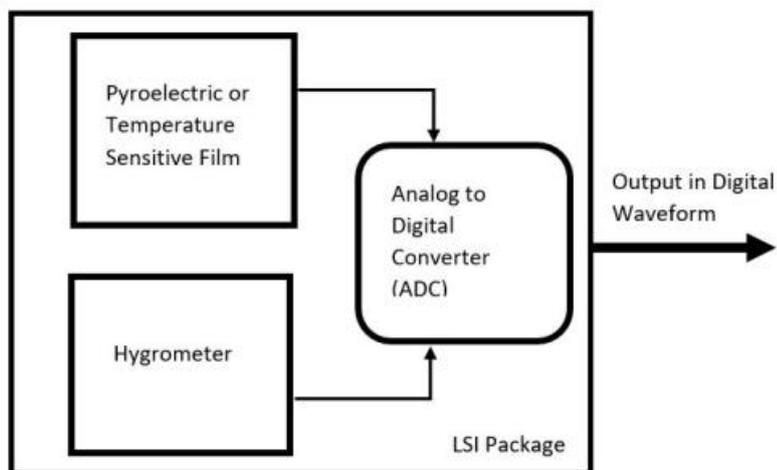


Figura 12 – *Node* do sensor de temperatura e umidade. Fonte: [Shah e Bhatt 2017]

ao sensor atualizado DHT22, seu custo reduzido e o fato de sua precisão e intervalos de medição serem o suficiente para o escopo do projeto justificam sua utilização.

3.4.3.2 Umidade

3.4.3.2.1 YL-69 + YL-38

Um sensor de baixo custo utilizado para medições de conteúdo de umidade no solo (13), sua operação de tensão é de 3,3V a 5V e possui corrente de 35mA [Danita et al. 2018]. Seu funcionamento se dá a partir de dois eletrodos capacitivos sensíveis à umidade do material dielétrico, sendo neste caso o solo [Shah e Bhatt 2017], que ajustam um potenciômetro de acordo com a capacitância, aumentando a tensão de saída na presença de um contato mais seco, sendo este parâmetro determinado através do conteúdo do vapor de água e temperatura ambiente [Cruz et al. 2019].



Figura 13 – YL-38 + YL-69. Fonte: Autor, 2023

3.4.3.3 Luminosidade

3.4.3.3.1 LDR

Também conhecido como fotoresistor, o LDR (figura 14) é um sensor que diminui sua resistência conforme o aumento da incidência de luz. Seu funcionamento se dá através da utilização de um semiconductor de alta impedância que, quando exposto à uma luz de alta frequência, transfere energia suficiente dos fótons absorvidos aos elétrons para pularem à camada de condução [Sushmitha, Jyothi e Sain 2022].

3.4.4 Atuadores

Com os dados coletados, é possível decidir os atuadores que serão escolhidos devido à sua praticidade de utilização e aquisição, capacidade de influência direta e isolada no parâmetro principal e relevância do parâmetro ambiental para a regulação de condições favoráveis às plantas.

3.4.4.1 Relé

O relé (figura 15), um *switch* com um eletroímã controlado por um sinal que fecha e abre o contato do circuito [Sushmitha, Jyothi e Sain 2022], será utilizado para a ativação de alguns atuadores.



Figura 14 – Sensor LDR. Fonte: Autor, 2023



Figura 15 – Relé de 5V. Fonte: Autor, 2023



Figura 16 – Exemplo de uma ventoinha de 120mm. Fonte: Autor, 2023

3.4.4.2 Temperatura, circulação e umidade do ar

3.4.4.2.1 Ventoinha

Uma ventoinha de *desktop* de 120mm e 12V (figura 16) pode ser utilizada para o controle da temperatura, umidade e circulação do ar, sendo o último o intuito principal do protótipo montado devido à falta de um espaço fechado, necessidade de múltiplas ventoinhas e de uma fonte de resfriamento. O movimento de ar em uma estufa é uma das técnicas de maior importância para manter uma uniformidade nas condições ambientais e melhora o fluxo de energia, vapor d'água e CO_2 entre as folhas e ar ao redor [Fernandez e Bailey 1994]. O controle da velocidade de rotação pode ser feito a partir de um sinal PWM, que é um tipo de modulação que regula a proporção de tempo que um pulso digital é transmitido a partir do *duty cycle*, uma razão que efetivamente determinará a velocidade da ventoinha de até 1800RPM, conforme demonstrado na figura 17.

3.4.4.3 Umidade e temperatura do solo

A umidade e temperatura do solo podem ser influenciadas através da conexão de uma mini bomba d'água de 12V (figura 18) submersa em um reservatório conectada à tubos com saídas em aspersores que irrigam o local de maneira uniforme e controlada. O

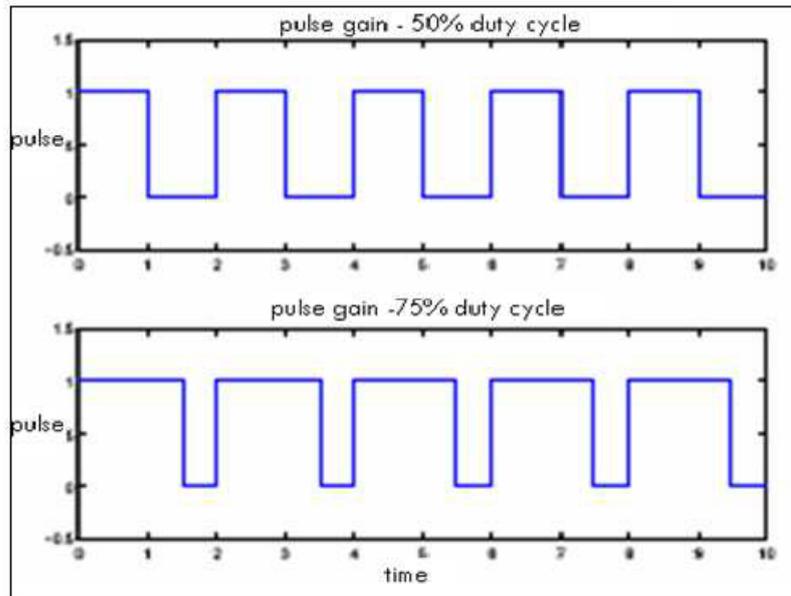


Figura 17 – Sinais PWM com 50% e 75% de *duty cycle*. Fonte: [Ullah, Gunawan e Muhida 2012]

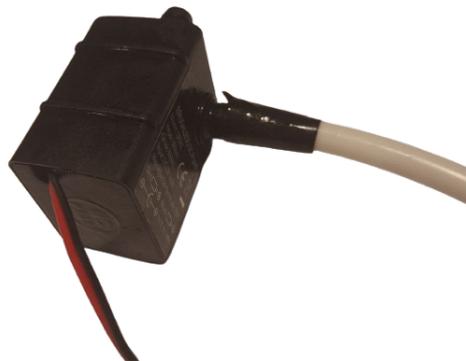


Figura 18 – Mini bomba d'água utilizada. Fonte: Autor, 2023

modelo utilizado tem uma potência de 4,2W e uma saída de 8mm, possibilitando uma vazão de 240L/h e uma coluna de elevação máxima de 3m.

3.4.4.4 Luminosidade

A luminosidade do local será diretamente controlado por uma lâmpada *smart* (figura 19), já que seu brilho e cor podem ser modificados através da conexão do dispositivo com a rede, abrindo a possibilidade de utilizar comandos na LAN para alterar os parâmetros desejados. O modelo utilizado utiliza uma potência de 8,5W e atua em uma faixa de temperatura entre 1700K e 6500K e um fluxo luminoso de 800lm.

Apesar da eficiência do componente disponível comercialmente escolhido, é recomendada a utilização de lâmpadas comumente utilizadas em estufas em casos críticos, como as lâmpadas que utilizam as tecnologias T-8 fluorescente, HID (High Intensity Discharge)



Figura 19 – Lâmpada *Yeelight E27* que será utilizada. Fonte: Autor, 2023

de iodetos metálicos com início de pulso e lâmpadas fluorescentes compactas, que são apropriadas para fornecer o brilho e as frequências e temperaturas de luz necessárias [Sanford 2004].

3.5 Opções de software e tecnologias

3.5.1 Protocolos de comunicação e mensagem

Para estabelecer a comunicação entre dispositivos e componentes, será necessário efetuar a escolha de um protocolo de comunicação eficiente e compatível com os objetivos do projeto.

3.5.1.1 Com fio

Apesar dos avanços tecnológicos das últimas décadas proporcionarem confiabilidade na transmissão de sinais a maiores distâncias em um canal com fio, a natureza das estufas agrícolas, possuindo umidade e luminosidade altas, o que acelera o processo de envelhecimento dos cabos, e o potencial de necessidade de um número alto de dispositivos, resultaria em um custo de instalação e manutenção elevado e uma expansibilidade limitada [Li et al. 2021]. Mesmo com algumas propostas de uso, como o emprego do protocolo I^2C em [Chhorn et al. 2022], os fatores negativos deste modo de comunicação se provaram como um impasse em grande parte das aplicações de projetos de sistemas de monitoramento de estufas agrícolas (figura 20).

3.5.1.2 Sem fio

A evolução rápida das tecnologias sem fio na última década transformou-as em uma opção viável, sendo uma escolha popular no âmbito de monitoramento ambiental e aplicações de IoT.

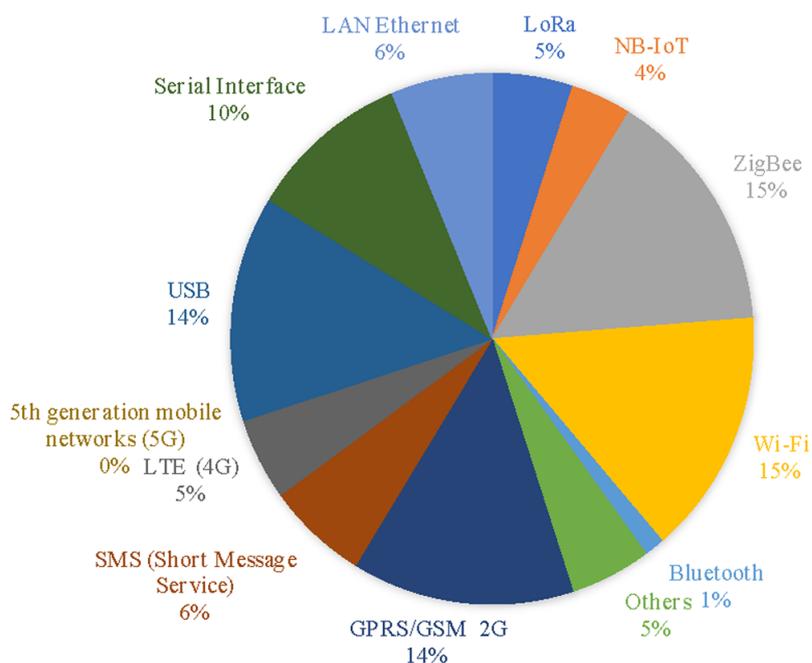


Figura 20 – Gráfico de distribuição de aplicações das tecnologias de comunicação em sistemas de monitoramento de estufas agrícolas. Fonte: [Li et al. 2021]

3.5.2 ZigBee

ZigBee é um protocolo que implementa uma *Low-Rate Wireless Personal Area Network* destinado à transmissão de sinais a baixas velocidades e distâncias curtas, com a proposta de obter um baixo consumo de energia e custo. Adotando o padrão IEEE 802.15.4, a tecnologia possui tempo de resposta rápido, uma taxa de dados de até 250kbps e suporta o uso de até 60 mil *nodes* com configuração automática e redundante, levando cerca 45ms para entrar em um estado acordado e conectar à rede [Li et al. 2021], que pode adotar as configurações de topologia estrela, a qual utiliza a comunicação de diversos dispositivos com um controlador central, árvore, que altera a topologia estrela através da adição de alguns roteadores, e malha, que tem estrutura similar à árvore, mas possibilita a comunicação direta entre todos os dispositivos (figura 21) [Zhang et al. 2007].

3.5.3 LoRa

LoRa é um protocolo que utiliza a modulação CSS e o padrão IEEE 802.15.4 g e é destinado ao uso em transmissões que necessitam de baixo consumo de energia e longas distâncias. Suas vantagens são similares às encontradas no ZigBee, adotando apenas a topologia estrela e possuindo suporte para até 60 mil *nodes*, manutenção automática, consumo de energia extremamente baixo e tempo de resposta rápido. No entanto, por ser uma tecnologia mais recente, a sua adoção ainda não é universal [Li et al. 2021].

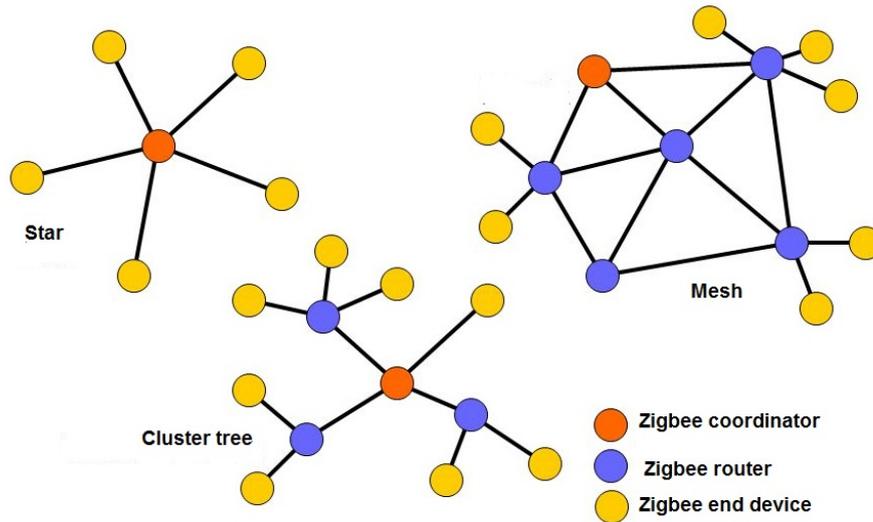


Figura 21 – Estruturas de configuração de rede ZigBee. Fonte: ECSTUFF4U, 2023

3.5.4 Bluetooth

Apesar de sua popularidade em aplicações relacionadas à estufas nos protótipos iniciais e sua taxa de dados de 1Mbps a 2 Mbps ser superior aos outros protocolos vistos até então, a tecnologia com base no padrão IEEE 802.15.1 caiu em desuso devido ao número de incompatibilidades criadas por suas diferentes versões, distância de transmissão curta, consumo de energia alto [Zhang et al. 2007] e processo de conexão complicado que resulta em uma inconveniência na aplicação de redes em larga escala [Li et al. 2021].

3.5.5 Wi-Fi

Um tipo específico de WLAN que utiliza o padrão IEEE 802.11, a rede *Wi-Fi* é caracterizada pela taxa de até 54 Mbps e baixa latência de transmissão, o que a torna capaz de transmitir vídeo em tempo real. No entanto, suas principais limitações, que incluem transmissão de distâncias curtas, alto consumo de energia [Zhang et al. 2007] e a capacidade de até 64 dispositivos conectados simultaneamente, tornam-na em uma opção viável apenas em aplicações de baixa escala [Li et al. 2021].

3.5.6 MQTT

O MQTT é um protocolo de mensageria leve projetado para utilizações em comunicações M2M e IoT. Seu funcionamento eficiente, confiável e escalável se dá através da implementação do modelo *publisher/subscriber* de dispositivos que se comunicam através de um intermediário chamado *broker*, um servidor central que recebe, gerencia e filtra todas as mensagens. O tamanho de cabeçalho e mensagens é reduzido, necessitando de uma banda menor para comunicação, e o protocolo possui 3 níveis de QoS (figura 22), sendo eles [Costa et al. 2021]:

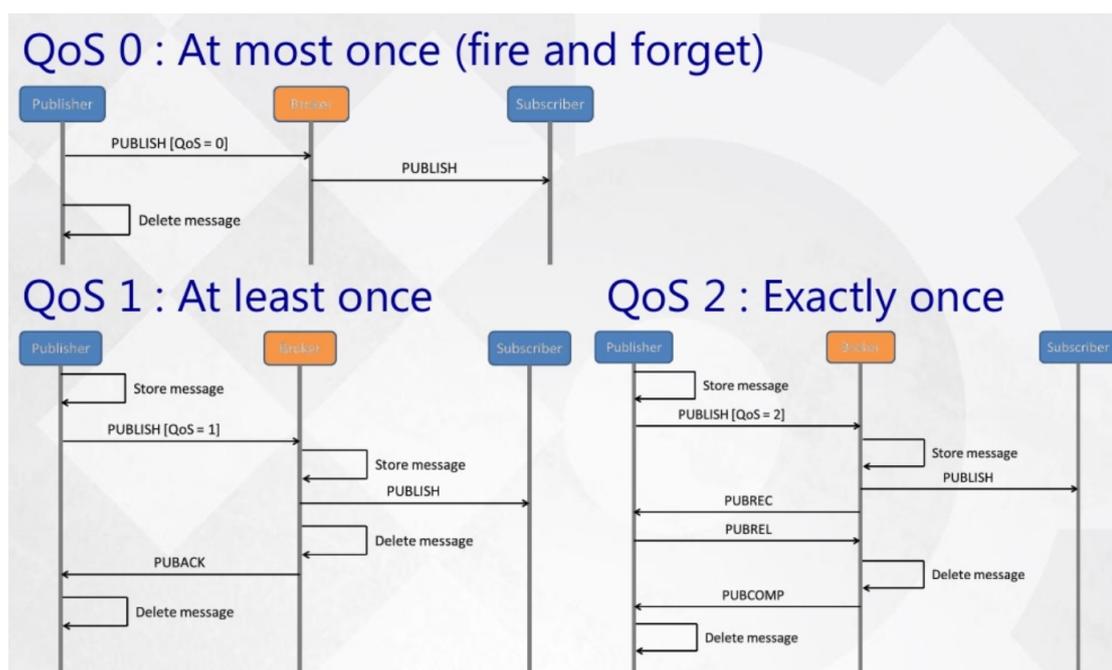


Figura 22 – Funcionamento do sistema de QoS do MQTT. Fonte: Devopedia, 2022

- Nível 0: A entrega de mensagem é feita no máximo uma vez, sem haver possibilidade de retransmissão ou armazenamento de dados e não é necessário o receptor reconhecer o recebimento da mensagem.
- Nível 1: A entrega de mensagem é feita pelo menos uma vez, em que, após a transmissão, o remetente manterá uma cópia até receber um pacote de confirmação enviado pelo receptor ou reenviará a mensagem caso não receba o pacote após determinado tempo.
- Nível 2: A entrega de mensagem é feita exatamente uma vez e, apesar de ser o nível que requer maior intensidade no uso de recursos, é o com maior confiabilidade. Seu funcionamento se dá através da publicação da mensagem até o recebimento do pacote de confirmação, momento em que o remetente poderá enviar um pacote indicando a possibilidade de liberar a mensagem para os assinantes. Após o recebimento do destinatário, será retornada a mensagem de confirmação do pacote anterior, resultando em uma situação em que os clientes e servidor tenham maior confiança que a entrega aos dispositivos foi bem sucedida.

Todas as mensagens são publicadas em tópicos, uma *string* utilizada pelo *broker* para filtrar cada um dos clientes conectados, que podem assinar múltiplos tópicos para receber mensagens diversas, possuindo um modelo de conexão persistente que regularmente envia mensagens de *heartbeat* ao *broker*. O MQTT em si não possui recursos avançados de segurança, no entanto suas aplicações podem ser executadas em cima de TLS e SSL para criptografia e autenticação [OASIS 2014].

3.5.7 Editores e compiladores

3.5.7.1 ThingSpeak

Uma opção popular nas aplicações de monitoramento, a *ThingSpeak Cloud* é uma plataforma IoT que possibilita a coleção, análise, processamento e visualização de dados em tempo real remotamente de componentes conectados à rede utilizando uma SNS (*Social Networking Service*) via API [Danita et al. 2018].

3.5.7.2 Node-RED / JavaScript

O *Node-RED* é uma ferramenta de programação visual de fluxo intuitiva criada pela empresa IBM utilizada para a conexão de dispositivos de *hardware* e serviços online, sendo popular para aplicações no espaço de IoT e automação. Sua construção foi feita com base no *Node.js*, habilitando a capacidade de operações de I/O assíncronas e abrindo a possibilidade de injeção de códigos *JavaScript* durante o processo de programação [Node-RED Documentation 2023].

3.5.8 Software

Para garantir a finalização e conclusão do projeto, alguns outros sistemas devem ser considerados para utilização:

3.5.8.1 Arduino IDE / C++

O *Arduino IDE* é um ambiente de desenvolvimento de *software* de código aberto dedicado para a programação de placas *Arduino* e adjacentes com uma linguagem simplificada baseada em C/C++, possuindo diversas bibliotecas e ferramentas úteis integradas [Arduino Documentation 2023].

3.5.8.2 Raspberry Pi OS

Raspberry Pi OS é uma distribuição do sistema operacional *Linux* baseado em *Debian* otimizado para utilização nos dispositivos *Raspberry Pi*. A versão que será utilizada é o *Raspberry Pi OS Lite*, que é uma edição minimalista do OS, não possuindo interface gráfica e vindo com menos pacotes pré-instalados, consequentemente consumindo menos recursos. Em sua configuração, o sistema pode ser pré-configurado para disponibilizar o acesso remoto via SSH [Raspberry Pi Documentation 2023].

3.5.8.3 Docker / Portainer

Docker é uma plataforma de virtualização a nível de sistema operacional de *containers*, que são aplicações, criadas via a disponibilização de uma imagem, executadas em

ambientes e processos isolados acessíveis, em sua maioria, via uma interface *web* [Docker Docs 2023]. O *software Portainer*, uma ferramenta e *container* de gerenciamento *Docker*, será utilizado por disponibilizar uma interface gráfica que facilita o processo de implementação de *containers* e *templates* de diversas imagens [Portainer Documentation 2023].

3.5.8.4 Nginx

Nginx é um *software* de servidor *web* de código aberto que será utilizado pelas suas funções de *proxy* reverso, podendo direcionar o tráfego de entrada para servidores de *back-end*, o que será útil para tornar o *dashboard* da estufa automatizada público sem comprometer a segurança.

4 CONSTRUÇÃO DO PROJETO

4.1 Lista de componentes

Os componentes escolhidos, representados fisicamente na figura 25, são:

1. Fonte DC de 3,3 V
2. Fonte DC de 5 V
3. Fonte DC de 12 V
4. 2x relés de 3,3 V
5. Mini bomba d'água
6. DHT11
7. Abajur e lâmpada inteligente
8. Ventoinha PWM 12 V
9. Raspberry Pi 3 Model B+
10. ESP8266
11. Resistor de $10k\Omega$
12. LDR
13. 2x diodos 1N4007
14. YL-38
15. YL-69

Uma fonte ATX de 200W (figura 23) foi utilizada para a derivação de todas as tensões DC utilizadas, e foram representadas individualmente no circuito.



Figura 23 – Fonte ATX de 200W. Fonte: Autor, 2023

4.2 Modelo

4.2.1 Lógica de funcionamento

A divisão do projeto será dada através do funcionamento em 3 camadas ou subsistemas com objetivos específicos, centradas em um microprocessador e microcontroladores, havendo um fluxo de informação e dados constante entre cada dispositivo e seção (figura 24):

4.2.1.1 Camada de medições e atuações

Para as aplicações que serão exigidas na camada de medições, onde se realizará a leitura e interpretação dos sensores, será utilizado o ESP8266, em qual todos os atuadores e sensores serão conectados através do GPIO nativo. É a camada que diretamente interage com o ambiente e onde se encontram os sensores, fazendo-o responsável por interpretar os dados de leitura, e atuadores, onde toda decisão feita pela camada de processamento terá que ser traduzida em uma instrução para um dos atuadores conectados. Toda a comunicação com a camada acima será feita utilizando o protocolo MQTT, devido à sua natureza leve e de baixo consumo e potencial de escalabilidade alto.

4.2.1.2 Camada de processamento

Onde se encontra o cérebro do sistema, é responsável por enviar relatórios para a camada acima com todas as informações coletadas e instruções para de atuação com base em decisões lógicas programadas. Esta camada, juntamente à de aplicativo, será centralizadas no microcomputador *Raspberry Pi*, devido à sua capacidade de processar uma grande quantidade de dados e ter seu sistema operacional com base no *Linux*, o que garante alta compatibilidade com alguns dos outros programas e recursos utilizados. Ele será responsável pela tomada de decisões a partir das leituras e hospedará a página com a interface gráfica.

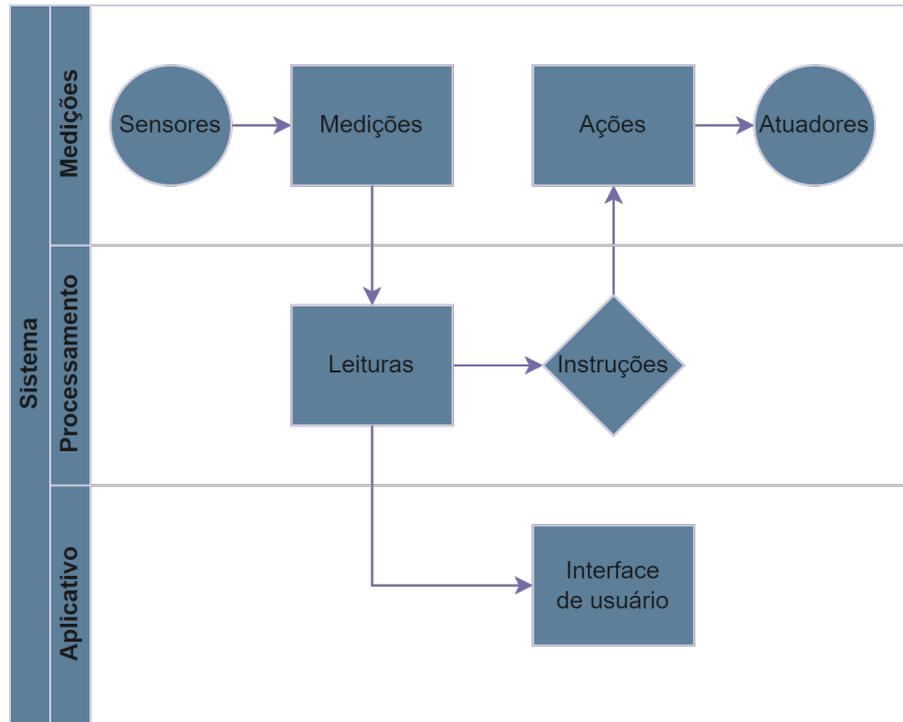


Figura 24 – Diagrama de blocos com o fluxo de informações entre os dispositivos e camadas.
Fonte: Autor, 2023

4.2.1.3 Camada de aplicativo

A camada mais elevada é a diretamente acessada pelo usuário via interface de usuário gráfica *web*, onde é possível ler as medições em tempo real, acessar relatórios e fazer o controle dos outros subsistemas. No *dashboard* também estarão disponíveis todas as possibilidades controle separados em 3 seções dedicadas à cada um dos parâmetros que serão controlados, onde o usuário irá poder decidir em automatizar os processos e influenciá-los a partir de valores ideais que ele decidir, ou realizar um controle manual de cada atuador.

4.2.2 Circuito

O circuito montado, representados pelas em simulação pela figura 25 e fisicamente pela figura 26, utiliza as seguintes conexões:

4.2.2.1 Sensores

4.2.2.1.1 Temperatura e umidade do ar

O sensor utilizado para temperatura e umidade do ar é o DHT11. Os 4 pinos serão conectados da seguinte maneira:

- O pino VCC alimentará o sensor e será conectado à fonte de 5V.

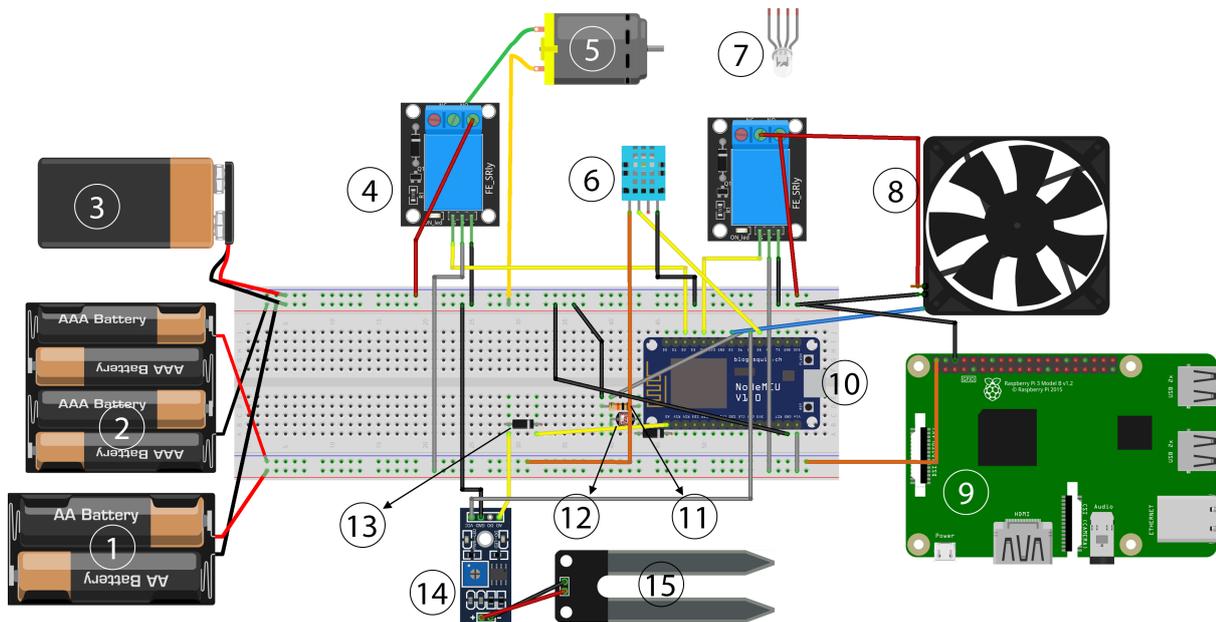


Figura 25 – Representação do circuito. Fonte: Autor, 2023 utilizando o *software* Fritzing

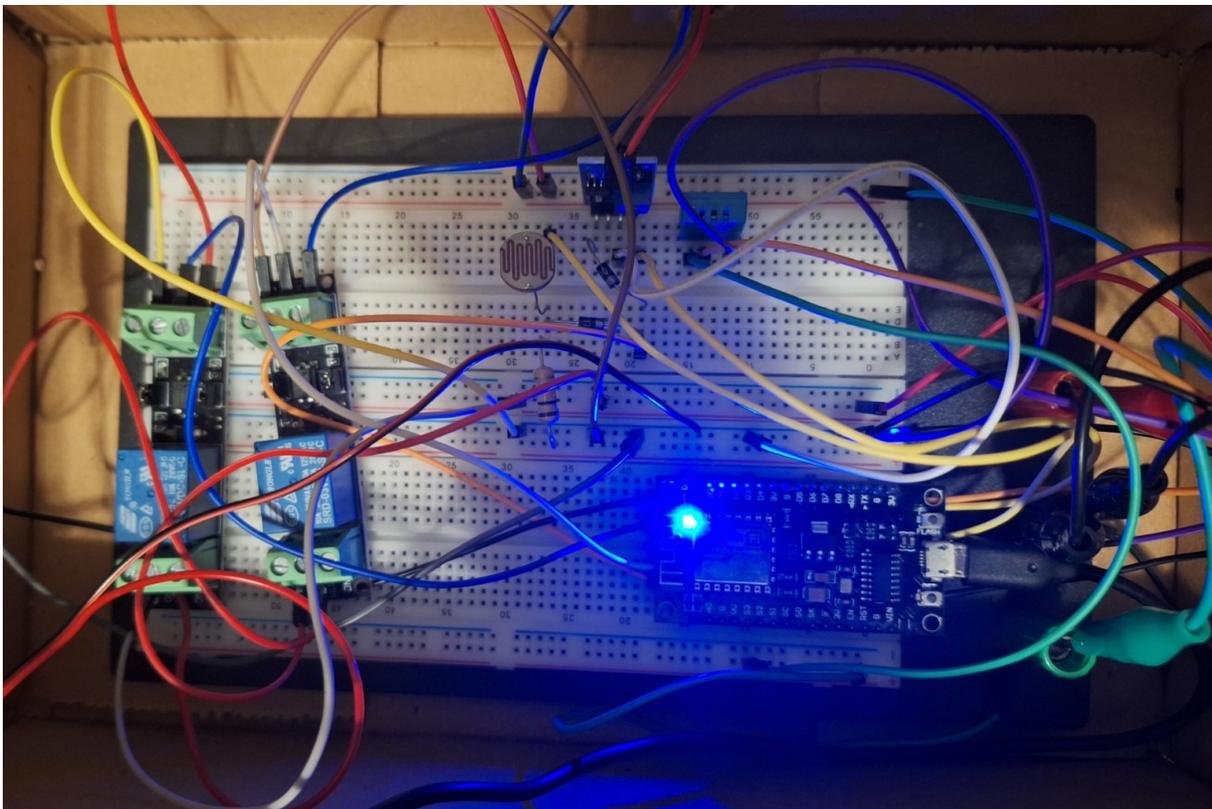


Figura 26 – Circuito do protótipo montado. Fonte: Autor, 2023

- O pino *signal* é responsável por enviar os dados digitais coletados pelo sensor, portanto será conectado ao pino GPIO digital D8 do ESP8266.
- O pino *ground* será conectado ao terra comum.

4.2.2.1.2 Umidade do solo

A umidade do solo será mensurada através dos sensores YL-38 e YL-69. As saídas serão conectadas na configuração a seguir:

- O pino VCC pode ser alimentado por uma fonte de tensão entre 3,3V e 5V, e será conectado ao pino GPIO D7 do ESP8266, para que seja possível controlar o sensor como um interruptor.
- Um diodo será conectado entre a saída analógica e a entrada A0 do ESP8266 para evitar problemas de corrente reversa durante o processo de multiplexação.
- O pino GND será conectado ao terra.

Como as saídas de ambos os sensores de umidade do solo e luminosidade são analógicas, e o ESP8266 tem apenas uma entrada que comporta esse tipo de informação, será necessário utilizar um processo de multiplexação dos dados através do ciclo a seguir:

1. Ambos os sensores começam desligados.
2. Utilizando uma das portas digitais, um dos sensores será ligado e o pino A0 irá digitalizar a informação.
3. O primeiro sensor será desligado e o segundo será ligado através da segunda porta digital, possibilitando a coleta de dados através do pino analógico.

4.2.2.1.3 Luminosidade

A medição de luminosidade será feita através de um LDR conectado à um diodo e a porta analógica do microcontrolador, com uma configuração de divisor de tensão utilizando um resistor de 10k Ω para evitar a saturação da entrada. Similarmente ao sensor de umidade do solo, para que seja possível aplicar o método de multiplexação, a alimentação do sensor será feita através do controle do pino digital D6 do ESP8266.

4.2.2.2 Atuadores

4.2.2.2.1 Temperatura e umidade do ar

A ventoinha utilizada possui o modelo de conexão de 4 pinos, que possibilita o controle de sinal PWM e a leitura de velocidade de rotação das pás. Como um *duty cycle* mínimo não resulta em uma desativação total da rotação, será necessário controlar seu estado através de um relé:

- O pino de 12V da ventoinha será conectado ao terminal Normally Open do relé.
- O sinal PWM da ventoinha será controlado pelo pino D5 do ESP8266.
- O Common Contact do relé será conectado à fonte de 12V para alimentar a ventoinha com a tensão devida.
- A fonte de 3,3V será utilizada para alimentar o relé através do pino VCC.
- O pino D4 do microcontrolador será conectado ao pino *input* do relé e será utilizado como um comando de interruptor.
- Os pinos *ground* da ventoinha e relé serão conectados ao terra comum.

4.2.2.2.2 Umidade do solo

A bomba d'água será controlada através da ativação de um segundo relé:

- O pino de 12V da bomba d'água será conectado ao Normally Open do relé.
- A alimentação da bomba d'água é de 12V DC, e será conectada ao pino Common Contact.
- O pino VCC do relé será alimentado pela fonte de 3,3V.
- O pino *input* do relé será controlado pelo pino D2 do ESP8266.
- Os pinos *ground* da bomba d'água e relé serão conectados ao terra.

4.3 Configuração dos dispositivos

4.3.1 Programação do microcontrolador

O ESP8266 será programado através do *software Arduino IDE* utilizando uma conexão via USB, e será necessário utilizar as bibliotecas do sensor DHT, *Wi-Fi* e MQTT. Será necessário estabelecer uma conexão com a rede local e ao servidor MQTT para enviar

as medições dos sensores a cada 2 segundos e receber as instruções dos atuadores. O código comentado está disponível na apêndice A.

4.3.2 Lâmpada

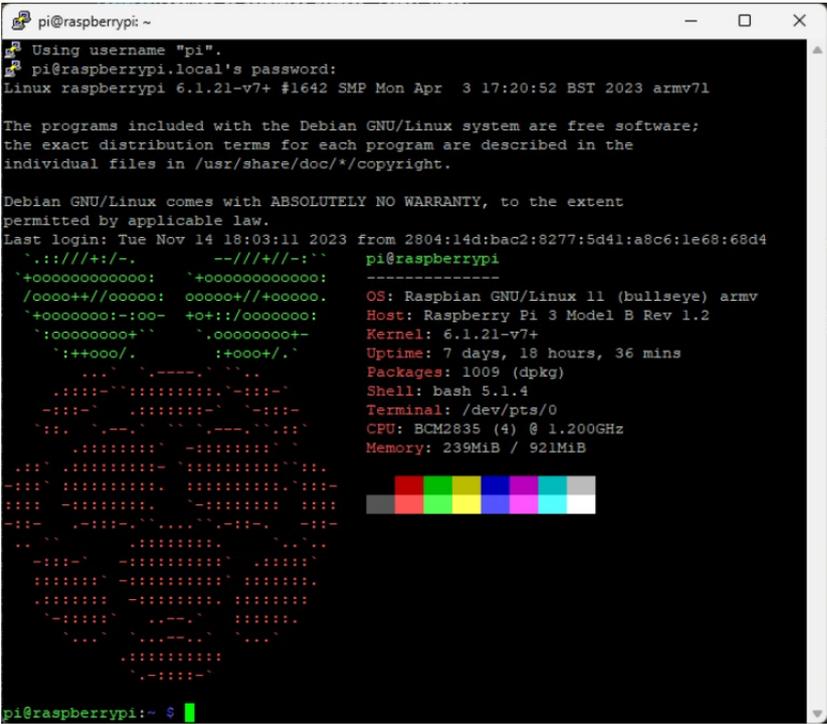
A lâmpada utilizada foi conectada à rede de 220V AC e configurada inicialmente através do aplicativo *Android* oficial *Yeelight*, onde foram estabelecidas as credenciais de conexão à *Wi-Fi* e a opção de controle LAN foi habilitada. Um IP estático também foi estabelecido para o endereço MAC da lâmpada através das configurações do roteador local.

4.3.3 Configuração do Raspberry Pi

O microcomputador que será utilizado foi conectado à uma fonte DC de 5V e configurado através da instalação do sistema operacional:

4.3.3.1 Raspberry Pi OS

Utilizando o *Imager* oficial, a versão lite do *Raspberry Pi OS* foi instalada em um cartão SD de 32GB e configurada para automaticamente habilitar o acesso via SSH e estabelecer uma conexão à rede *Wi-Fi* local com um IP estático para evitar futuros problemas relacionados à conexões e abertura de portas, possibilitando o controle remoto por *desktop* (figura 27) [Raspberry Pi Documentation 2023].



```
pi@raspberrypi ~
└─$ ssh pi@raspberrypi.local
Using username "pi".
pi@raspberrypi.local's password:
Linux raspberrypi 6.1.21-v7+ #1642 SMP Mon Apr  3 17:20:52 BST 2023 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Nov 14 18:03:11 2023 from 2804:14d:bac2:8277:5d41:a8c6:1e68:68d4
pi@raspberrypi
-----
OS: Raspbian GNU/Linux 11 (bullseye) armv7
Host: Raspberry Pi 3 Model B Rev 1.2
Kernel: 6.1.21-v7+
Uptime: 7 days, 18 hours, 36 mins
Packages: 1009 (dpkg)
Shell: bash 5.1.4
Terminal: /dev/pts/0
CPU: BCM2835 (4) @ 1.200GHz
Memory: 239MiB / 921MiB

pi@raspberrypi:~$
```

Figura 27 – Tela de terminal do Raspberry Pi acessado via chave SSH com o *software Putty*. Fonte: Autor, 2023

4.3.4 Docker / Portainer

Utilizando o terminal acessado, alguns comandos *bash* foram executados para a instalação dos programas de virtualização que serão essenciais para a execução do projeto e instalação dos próximos *containers*, sendo acessados via web a partir do IP local do dispositivo utilizando a porta 9000 (figura 28) [Docker Docs 2023] [Portainer Documentation 2023]. Este processo foi realizado para garantir o isolamento de execução dos *softwares*, evitar problemas de incompatibilidade com o sistema operacional e disponibilizar uma interface gráfica *web* para cada um dos programas nativamente:

```
# Adiciona as chaves GPG Docker
sudo apt-get install ca-certificates curl gnupg
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/raspbian/gpg |
sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg

# Configura o repositório API Docker:
echo \
  "deb [arch="$(dpkg --print-architecture)"
  signed-by=/etc/apt/keyrings/docker.gpg]
  https://download.docker.com/linux/raspbian \
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Atualiza os repositórios
sudo apt-get update

# Instala o Docker
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

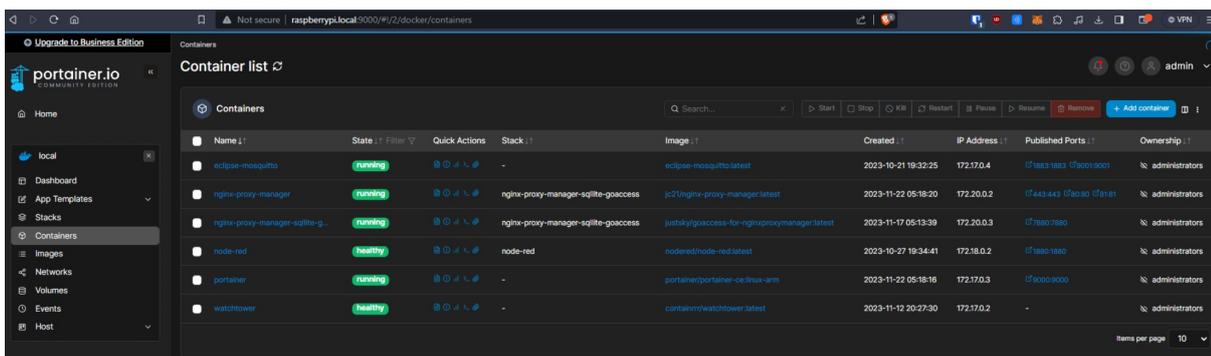


Figura 28 – Interface web da instalação Portainer. Fonte: Autor, 2023

```
docker-buildx-plugin docker-compose-plugin

# Instala e executa o container Portainer com a porta 9000
docker run -d -p 9000:9000 -p 9443:9443
--name portainer --restart=always
-v /var/run/docker.sock:/var/run/docker.sock
-v portainer_data:/data portainer/portainer-ee:latest
```

Também se demonstrou necessário realizar uma alteração no arquivo `/etc/docker/daemon.json` para que fosse definido um DNS primário local e um DNS secundário externo, já que sem essa configuração, os programas virtualizados falhavam em tentar estabelecer uma conexão com domínios da *internet*:

```
{
  "dns": ["10.0.0.2", "8.8.8.8"]
}
```

4.3.4.1 Eclipse Mosquitto

A imagem do *Eclipse Mosquitto*, uma implementação leve de servidor MQTT, será utilizada como *broker*, e um arquivo `config/pwfile` configurado no arquivo `mosquitto.conf` foi criado com um usuário e senha encriptado para que todas as conexões estabelecidas com o servidor sejam autorizadas, utilizando os seguintes comandos [Mosquitto Documentation 2023]:

```
# Login interativo no container
sudo docker exec -it <ID-CONTAINER> sh

# Adiciona usuário
mosquitto_passwd -c /mosquitto/config/pwfile #USUARIO

# Reinicia o container
sudo docker restart <ID-CONTAINER>
```

4.3.4.2 Node-RED

Além da instalação base do *Node-RED*, acessível pela porta 1880, alguns *nodes* extras necessários foram configurados:

The image shows the 'Edit Proxy Host' configuration window in Nginx. It has a title bar with a close button. Below the title bar are four tabs: 'Details' (selected), 'Custom locations', 'SSL', and 'Advanced'. The main content area is divided into sections: 'Domain Names' with a text input field containing 'estufaautomatizada.duckdns.org'; 'Scheme', 'Forward Hostname / IP', and 'Forward Port' each with a dropdown or text input field (values: 'http', '192.168.0.169', '1880'); three toggle switches for 'Cache Assets', 'Block Common Exploits', and 'Websockets Support', all currently turned off; and an 'Access List' section with a text input field containing 'Publicly Accessible'. At the bottom right, there are 'Cancel' and 'Save' buttons.

Figura 29 – Configuração de domínio e *host* no Nginx. Fonte: Autor, 2023

- *node-red-contrib-yeelight*: Utilizado para estabelecer uma conexão com a lâmpada utilizada, a configuração necessita do IP estático definido via roteador para o funcionamento.
- *node-red-dashboard*: Este *node* será utilizado para desenvolver a interface gráfica de controle e monitoramento que será utilizada pelos usuários da estufa. Um usuário e senha também foi estabelecido especificamente para autorizar o acesso à essa página.
- *node-red-node-email*: Utilizado para a criação de um sistema de notificação de condições críticas.
- *node-red-node-ui-webcam*: Um *node* utilizado para fazer uma transmissão ao vivo do local através de uma *webcam*.

4.3.4.3 Nginx

Para que seja possível acessar o *dashboard* em outras redes, um domínio DDNS foi criado gratuitamente utilizando o serviço *Duck DNS* e configurado no gerenciador de *proxy* reverso *Nginx* (figura 29). A porta devida foi então aberta nas configurações do roteador para o IP desejado, fazendo com que seja possível acessar de qualquer local a interface de maneira segura através do domínio selecionado [Nginx Documentation 2023].

4.3.4.4 Motion

Um módulo extra que será implementado no projeto será a utilização de um *feed* de *webcam* disponível para monitoramento na interface gráfica. A câmera *Logitech C922* foi

conectada diretamente ao microcomputador utilizando um cabo USB e o *software Motion* foi utilizado, que permite o monitoramento *web* do dispositivo. Algumas configurações do arquivo `/etc/motion/motion.conf` foram alteradas [Motion Documentation 2023]:

```
# Porta utilizada para o controle web
webcontrol_port 8080

# Remove restrições de conexões ao localhost
webcontrol_localhost off
stream_localhost off
```

4.4 Controle e monitoramento

O controle manual e automatizado do sistema será realizado através da programação via *Node-RED*, e a configuração será feita utilizando os seguintes *nodes*, que são blocos programáveis e podem ser interconectados e combinados para realizarem funções específicas [Node-RED Documentation 2023]:

- `mqtt in`: Este *node* é utilizado para receber as informações dos sensores publicadas pelos microcontroladores, e necessitam do IP do servidor MQTT e nome do tópico de entrada.
- `mqtt out`: Utilizado para enviar instruções aos atuadores conectados nos microcontroladores, este *node* necessita do IP do *broker* e o tópico de saída definido no ESP8266.
- `slider` e `switch`: Botões interativos no *dashboard* que alteram valores utilizados para controlar a ativação ou intensidade dos atuadores definidos.
- `chart`: Gráficos utilizados para monitoramento das medições feitas pelos sensores na interface.
- `yeelight` e `colour picker`: *Nodes* importantes para estabelecer a conexão e controle da lâmpada, sendo o segundo um seletor de cores disponível no *dashboard*.
- `webcam`: Utilizado para exibir a transmissão de uma imagem *web* na interface.
- `delay`, `trigger` e `inject`: Funções de manipulação de sinais relativas ao tempo. Respectivamente, controlam o atraso para transmissão do sinal, por quanto tempo o dado será transmitido antes de ser automaticamente alterado e o intervalo de tempo entre cada transmissão.
- `change`: Altera as propriedades da mensagem sendo transmitida.

- *function*: Habilita a utilização de código *JavaScript* para o processamento dos sinais.
- *email*: Utilizado para o envio automatizado de *e-mail*.

4.4.1 Sensores

Os 3 sensores são representados por 4 *nodes mqtt in*; umidade do solo e ar, temperatura e luminosidade. Eles serão diretamente conectados aos *charts* com intervalos de 10 minutos e 1 hora para que o usuário consiga acompanhar gráficos em tempo real das leituras dos parâmetros ambientais de maior importância.

4.4.2 Controle manual

As seguintes configurações foram programadas para efetuar o controle manual dos atuadores (figura 30), com o *dashboard* representado pela figura 31.

4.4.2.1 Ventoinha

A ventoinha pode ser controlada a partir do *node switch*, que alterna entre os estados *ON* e *OFF* e é conectado à um *node mqtt out* inscrito no tópico *home/FanControl* para ligar e desligar o relé da ventoinha, e *slider*, que envia um valor entre 0 e 1023 representando o *duty cycle* do sinal PWM, que é conectado diretamente ao *mqtt out* inscrito no tópico *home/fanSpeedControl* para alterar a velocidade de rotação.

4.4.2.2 Bomba d'água

Similarmente à ventoinha, a bomba d'água será controlada pelo relé comandado pelo tópico *home/pumpControl* e um *switch* que alterna entre os estados *ON* e *OFF*.

4.4.2.3 Lâmpada

Em primeiro momento, o ajuste da lâmpada será feito utilizando um *switch* que envia o valor de *payload* 0 ou 1, conectado ao *node yeelight*, sendo possível ligar e desligá-la.

4.4.3 Controle automatizado

O controle automatizado dos atuadores será determinado a partir de cálculos com base na medição de parâmetros dos sensores respectivos e os valores ideais selecionados pelo usuário. Para minimizar os efeitos de erros e variações instantâneas causadas pelos sensores durante as medições, um *array* de valores será extraído utilizando a seguinte função:

```
var valorArray = flow.get('valorArray') || [];
```

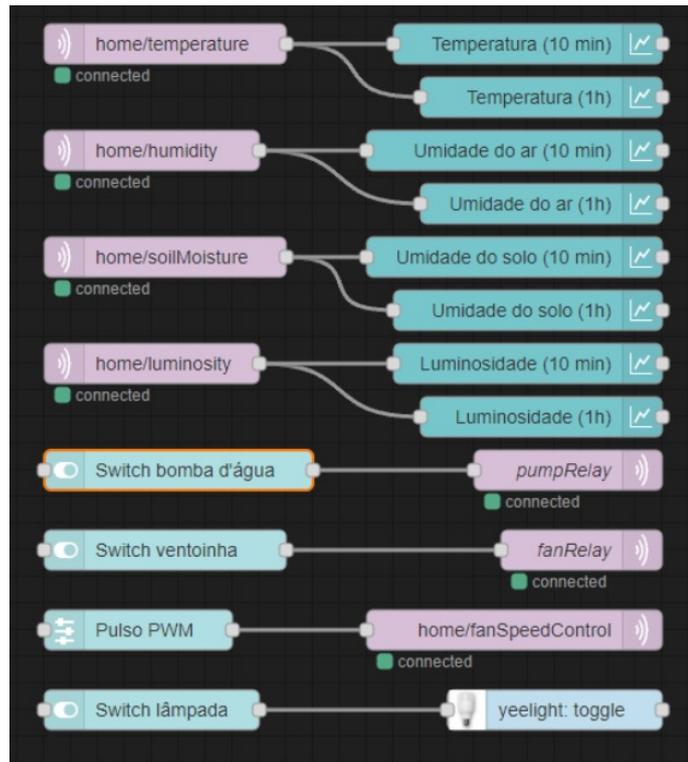


Figura 30 – Fluxo *Node-RED* com gráficos e controle manual. Fonte: Autor, 2023

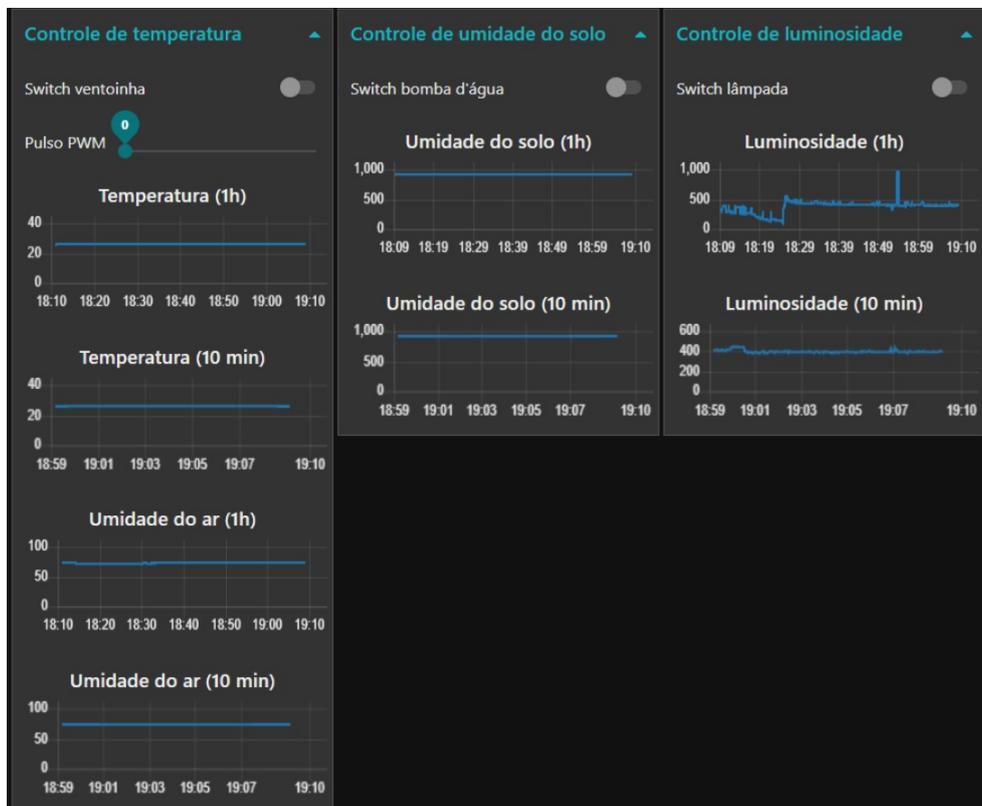


Figura 31 – *Dashboard* com gráficos e controle manual. Fonte: Autor, 2023

```
// Converte o payload para um número e adiciona ao array
var currentTemp = parseFloat(msg.payload);
if (!isNaN(valorInst)) {
    tempAir.push(valorInst);
    flow.set('valorArray', valorArray);
}

return null;
```

Em sequência, uma função de mediana, com um intervalo de tempo que deve ser um múltiplo do intervalo de medições determinado a partir da utilização de um *node inject*, será adquirida e alterada utilizando o *node change* para representar o valor de mensagem *payload* como um *flow* de nome específico:

```
var valorArray = flow.get('valorArray') || [];

if (valorArray.length > 0) {
    // Organiza o array
    valorArray.sort(function (a, b) {
        return a - b;
    });

    // Calcula a mediana
    var middle = Math.floor(valorArray.length / 2);
    var median = tempAir.length % 2 === 0 ? (tempAir[middle - 1]
+ valorArray[middle]) / 2 : valorArray[middle];

    // Limpa o array
    flow.set('valorArray', []);

    return { payload: median };
} else {
    return null;
}
```

O usuário terá a opção realizar de controle automatizado ou manual de cada um dos atuadores a partir de um interruptor que altera um parâmetro global definido pelo *node function* a seguir:

```
// Configura a função global que acompanha o estado do interruptor
global.set("autoMode", msg.payload);

return msg;
```

Uma função será então inserida no caminho do fluxo de cada atuador para os modos automatizado e manual, bloqueando ou ativando os dados a partir do *script*:

```
// Função que checa se está no modo automático
var autoModeon = global.get("autoMode") || false;

if (autoModeon) {
    return msg; // return null se estiver no caminho
    // do controle manual
} else {
    return null; // return msg se for inserido
    // no caminho do controle manual
}
```

4.4.3.1 Sistemas de controle

O modelo projetado poderá usufruir de alguns métodos de controle dos atuadores. Para considerações de sistemas, é importante notar que não existem variações bruscas naturais da maioria dos parâmetros ambientais, e o tempo de resposta sempre estará na ordem de grandeza de minutos. Para a obtenção dos melhores resultados e maior precisão possível, os sensores devem ser posicionados próximos às plantas, o que os deixarão sob a influência dos atuadores, necessitando de um sistema de malha fechada para obter uma melhora na resposta e evitar um comportamento oscilatório de atuação devido à falta de realimentação. Em [Santos, Júnior e Neves 2017], um controlador PID (figura 32) foi implementado a partir da observação da temperatura para uma resposta ao degrau de um *cooler* (figura 33) e a dedução de uma curva característica de primeira ordem, representada pela equação 4.1:

$$G(s) = \frac{K}{\tau s + 1} \quad (4.1)$$

O controlador PID pode então ser projetado para atender as necessidades específicas do projeto como valores de erro, sobressinal e tempo de resposta, devendo seguir a estrutura de equação 4.2, fornecendo uma resposta e comportamento eficientes no controle de temperatura, como observado na figura 34:

$$C(s) = K \left(1 + \frac{1}{sT_I} \right) (1 + sT_D) \quad (4.2)$$

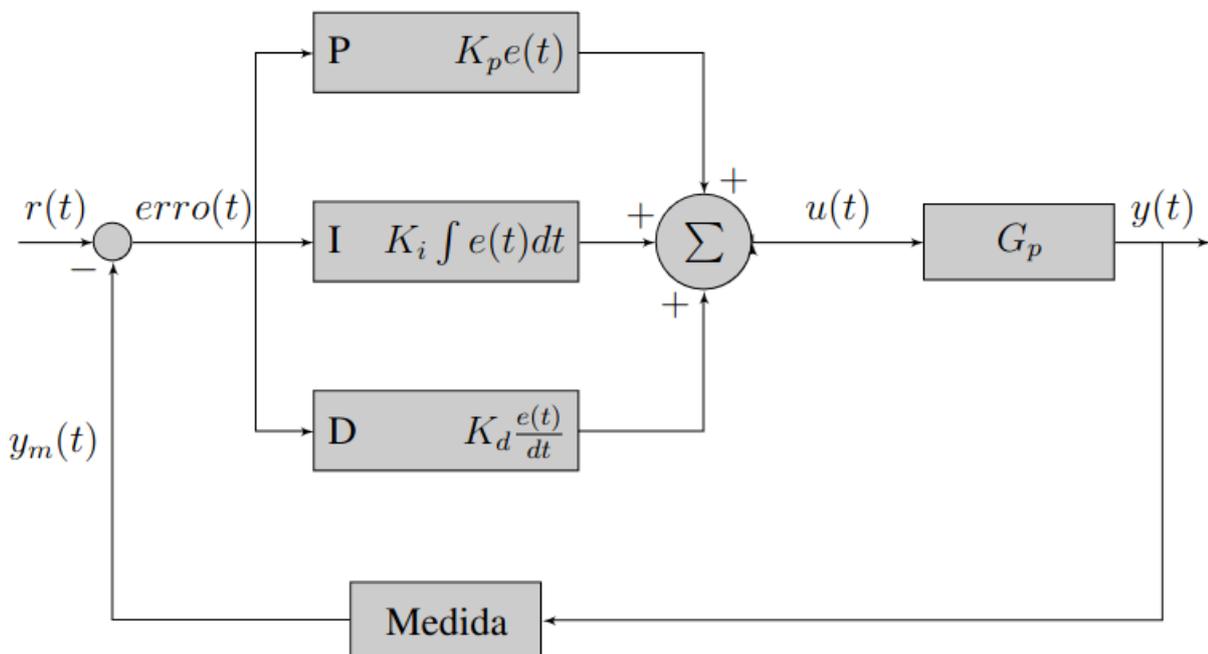


Figura 32 – Diagrama de blocos de um controlador PID. Fonte: [Santos, Júnior e Neves 2017]

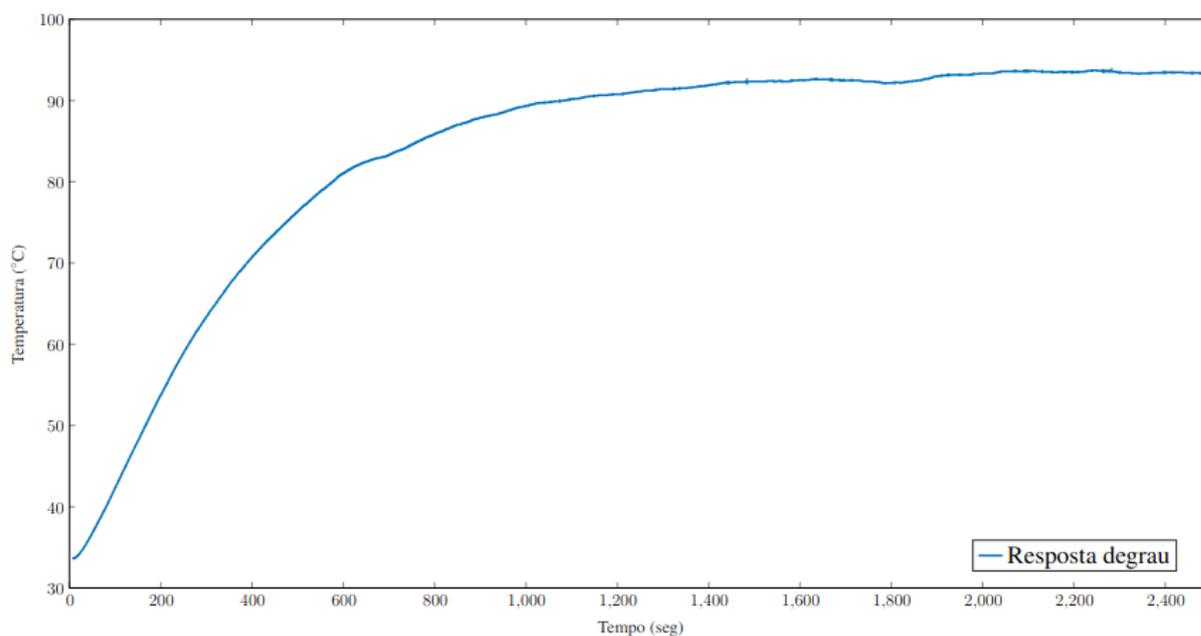


Figura 33 – Resposta ao degrau do sistema com o sinal PWM máximo δ . Fonte: [Santos, Júnior e Neves 2017]

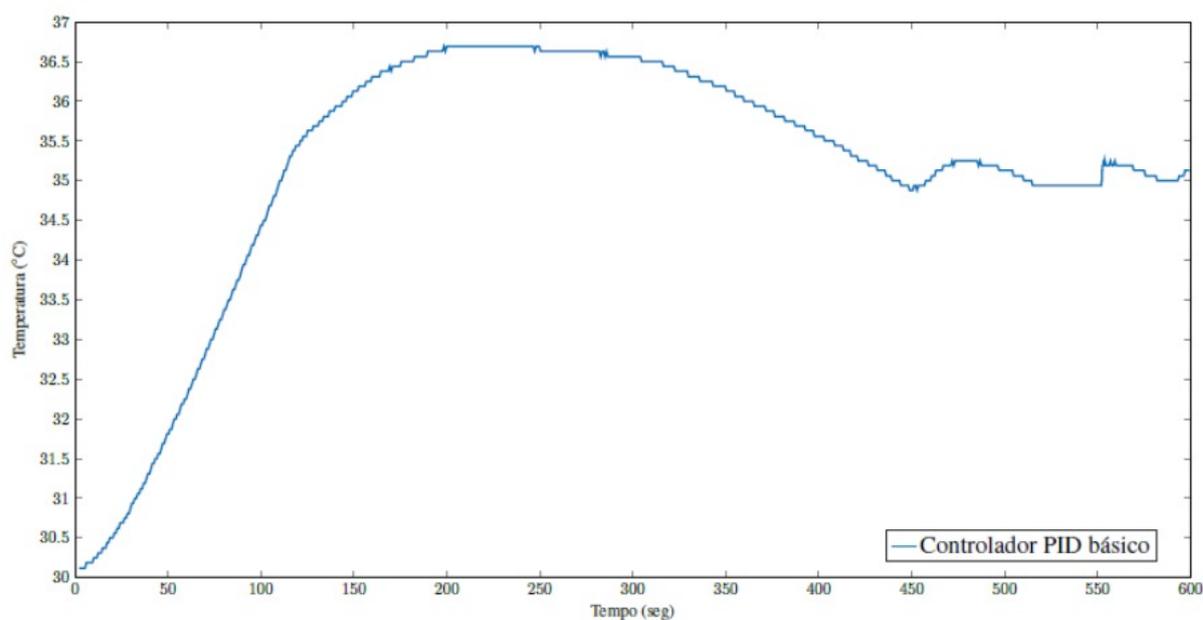


Figura 34 – Resposta do sistema com o controlador PID. Fonte: [Santos, Júnior e Neves 2017]

Apesar de soluções similares serem aplicadas, a resposta do sistema construído é, como já mencionado, não-linear, com parâmetros interdependentes influenciados pelo clima, local e configuração dos atuadores e sensores, de natureza imprecisa e com medição de base variável de acordo com a escolha do usuário, o que influencia diretamente na medição dos resultados. Por consequência, o princípio de superposição de efeitos não pode ser aplicado para que seja realizado o estudo dos efeitos de referência e perturbação mesmo com a redução dos efeitos de não-linearidade ao fechar a malha [Silva e Trofino], o que, quando combinado com a proposta de modularidade e condições variáveis de acordo com a aplicação do projeto, tornam um sistema de controle de do tipo *ON-OFF* que utiliza alguns sensores de referência que não se encontram dentro do campo de influência dos atuadores uma solução mais simples, com maior viabilidade e mais popular dentro das aplicações de estufas agrícolas, como no caso de estudo em [Danita et al. 2018]. Portanto, para a configuração do sistema de controle que está de acordo com o escopo do projeto, os sensores DHT11 e LDR serão posicionados em um local que representa o ambiente da estufa sem influência dos atuadores.

A regra base de ajuste dos atuadores foi definida em um esquema representado pelo fluxograma da figura 35, visando manter uma temperatura, umidade do solo e luminosidade mínima utilizando, respectivamente, a velocidade de rotação da ventoinha, ativação do relé da bomba d'água e alteração de brilho da lâmpada.

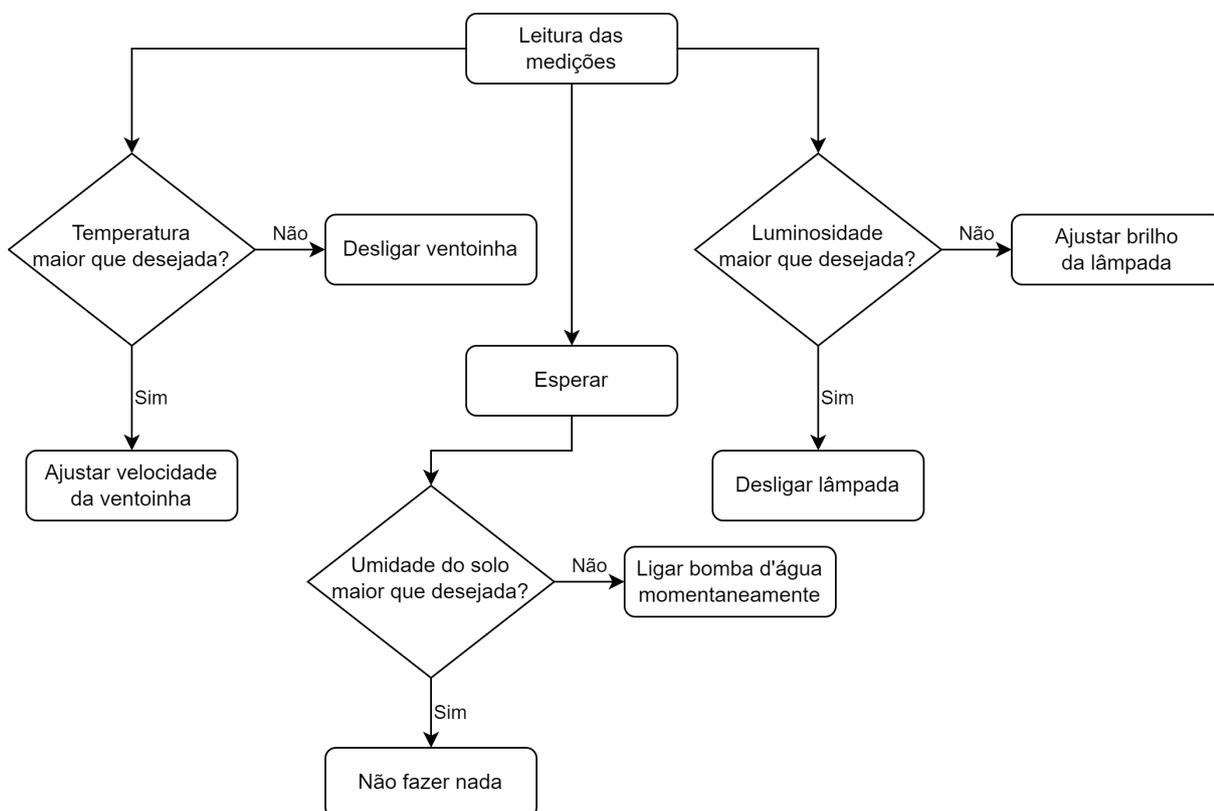


Figura 35 – Fluxograma da lógica de automatização dos atuadores. Fonte: Autor, 2023

4.4.3.1.1 Temperatura e circulação de ar

O sistema de ajuste de velocidade da ventoinha utilizará como inspiração uma versão modificada da curva de atuação comumente utilizada em ventoinhas de *desktop*, tendo como base um valor de temperatura ideal ajustável que seguirá um comportamento de equação quadrática. A velocidade de rotação atingirá o valor máximo quando o valor de temperatura medido foi igual ou superior ao valor ideal somado à um intervalo de temperatura, conforme observado na figura 36.

Os valores de sinal PWM máximo, temperatura desejada e intervalo de tolerância serão ajustados através de *sliders* e conectados junto à mediana de temperatura medida a cada 10s, com valores de *flow* diferentes em um *node function*. A função determinará a curva de atuação da ventoinha a partir do seguinte *script*:

```
var targetTemp = flow.get('targetTemp') || 35;
var toleranceTemp = flow.get('toleranceTemp') || 5;
var currentTemp = flow.get('currentTemp') || 25;
var pulseTemp = flow.get('pulseTemp') || 500;

let output;

if (currentTemp <= targetTemp) {
  // Se a temperatura alvo for maior que a medida, retornar 0
  output = 0;
```

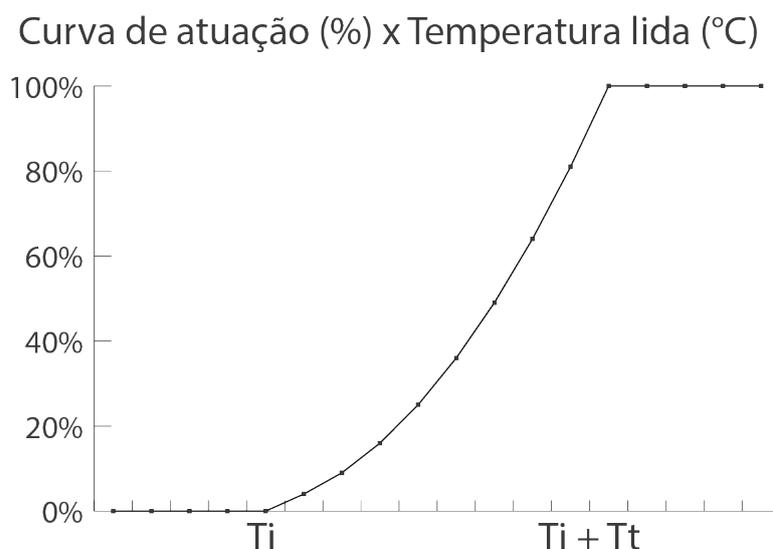


Figura 36 – Intensidade de atuação para valores entre T_i e a soma de T_i com T_t . Fonte: Autor, 2023

```
} else if (currentTemp >= targetTemp + toleranceTemp) {
  // Se a temperatura medida for maior que a soma do alvo e tolerância,
  // a saída será o valor máximo
  output = pulseTemp;
} else {
  // Se a medida estiver entre o alvo e a soma do alvo e tolerância,
  // a saída seguirá um comportamento linear
  output = Math.round(((currentTemp - targetTemp) / toleranceTemp)
    ** 2 * pulseTemp);
  output = Math.min(Math.max(output, 0), 1023);
}

msg.payload = output;

return msg;
```

Uma segunda função, apresentada a seguir, será utilizada para checar se o valor de velocidade é 0. Caso esta condição seja satisfeita, uma mensagem de estado *OFF* será enviada para o relé da ventoinha. Para evitar problemas com alterações de velocidade feitas simultaneamente à instrução de ativação do relé, um *delay* de 100ms foi adicionado ao *node* de ajuste de valor PWM.

```
if(msg.payload === 0) {
  // Se a entrada for 0, a saída será OFF
  msg.payload = "OFF";
} else if(msg.payload >= 1) {
  // Se a entrada for maior que 1, a saída será ON
  msg.payload = "ON";
}

return msg;
```

4.4.3.1.2 Umidade

Para este caso, o único parâmetro ajustável será o valor de umidade do solo desejada. O controle automatizado de umidade do solo é mais simples em sua composição, funcionando através de uma comparação da mediana do valor lido com o valor de umidade do solo desejada, sendo utilizado um *node inject*, que irá ativar a função a cada 1h. Caso o valor encontrado seja menor que o desejado, a bomba d'água será ativada por 2s e

automaticamente desativada através da utilização do *node trigger*. A função de comparação utilizada é:

```
var targetHum = flow.get('targetHum') || 50;
var currentHum = flow.get('currentHum') || 70;

if (currentHum < targetHum) {
  // Se a umidade alvo for maior que a medida, retornar 1
  msg.payload = "ON";
} else {
  // Se a umidade alvo for menor que a medida, retornar 0
  msg.payload = "OFF";
}

return msg;
```

Para transformar os valores medidos em um sistema similar à umidade relativa medida pelo sensor DHT11, um *node function* foi inserido em conexão direta com a entrada MQTT para inverter os valores, fazendo com que o mínimo represente uma umidade baixa, e que o intervalo aceito esteja entre 0 e 100:

```
// Transforma o valor absoluto da entrada em uma porcentagem
let absoluto = msg.payload;
let por cento = Math.round(100-((absoluto / 950) * 100));

msg.payload = por cento;

return msg;
```

4.4.3.1.3 Luminosidade

O controle de brilho utiliza os mesmos princípios do controle de temperatura. A automatização de nível de luminosidade terá 3 opções de ajustes para se adequar ao objetivo desejado.

- Luminosidade desejada: O valor de luminosidade ideal para manter.
- Intervalo de tolerância: Um *slider* que é utilizado para indicar o nível mínimo de luminosidade esperado.
- Brilho máximo: Controla o brilho máximo emitido pela lâmpada.

Similarmente ao comportamento de ajuste de temperatura, após a leitura da mediana de 5 medidas do LDR, o sistema irá comparar com o valor desejado. Por padrão, a lâmpada estará em um estado *OFF*, porém, caso a medida desejada for maior que o valor de mediana, o estado da lâmpada será modificado para *ON*. O brilho será então alterado com base no intervalo de tolerância decidido, seguindo o comportamento de uma equação quadrática, e atingindo o valor máximo de brilho quando o valor lido for menor que a subtração da luminosidade desejada pelo intervalo de tolerância, conforme demonstrado na figura 37. A função de ajuste de brilho foi empregada com o seguinte *script*:

```
var targetLum = flow.get('targetLum') || 50;
var toleranceLum = flow.get('toleranceLum') || 10;
var currentLum = flow.get('currentLum') || 70;
var maxLum = flow.get('maxLum') || 50;

let output;

if (currentLum >= targetLum) {
  // Se a luminosidade medida for maior que alvo, retornar 0
  output = 0;
} else if (currentLum <= targetLum - toleranceLum) {
  // Se a luminosidade medida for menor que a soma do alvo
  // e tolerância, retornar o valor máximo
  output = maxLum;
} else {
  // Se a medida estiver entre o alvo e a soma do alvo
  // e tolerância, a saída seguirá um comportamento linear
  output = Math.round(((targetLum - currentLum) / toleranceLum)
    ** 2 * maxLum);
  output = Math.min(Math.max(output, 0), 100);
}

msg.payload = output;

return msg;
```

Para alterar a cor da luz propagada, o *colour picker* com configuração de saída hexadecimal foi utilizado, sendo necessário conectar uma função que transforma a *string* em um valor de cor inteiro:

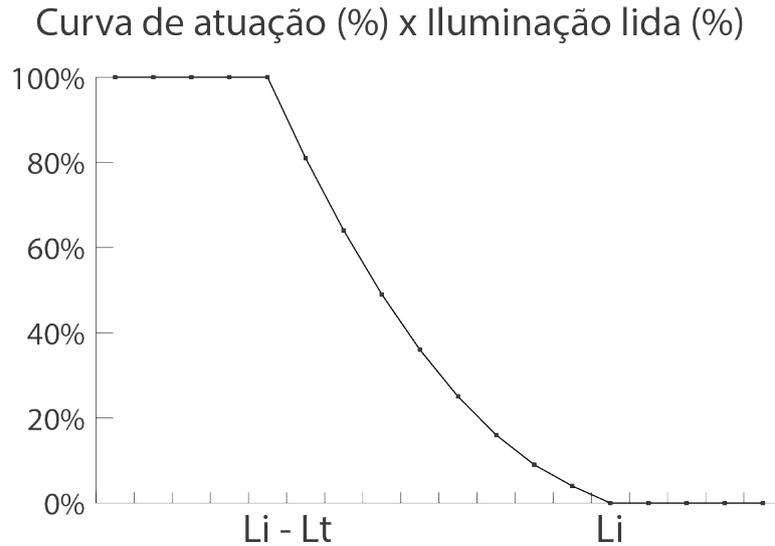


Figura 37 – Intensidade de atuação para valores entre Li e a soma de Li com Lt. Fonte: Autor, 2023

```
// Função que converte uma cor hexadecimal para uma inteira
msg.payload = parseInt(msg.payload.replace('#', ''), 16);

return msg;
```

4.5 Sistemas de contingência

Para adicionar uma camada extra de confiança ao usuário, outros 2 sistemas serão implementados.

1. O primeiro sistema utilizará o *feed* de câmera disponibilizado pelo *software Motion* para disponibilizar um método de acompanhamento visual em tempo real da estufa, que pode ser útil para identificação de problemas físicos dos atuadores ou a presença de elementos não desejados no local.

```
<iframe width="640" height="480"
src="http://#IPRPI:8081/0/stream" frameborder="0"></iframe>
```

2. A segunda implementação é um sistema de alertas via *node email*, que será configurado para cada um dos sensores com base nos valores ideais e de tolerância, quando aplicável, caso a opção de atuação automatizada estiver habilitada. Um *e-mail* será enviado quando o valor medido ultrapassar o limite de valor crítico, alertando o



Figura 38 – Exemplo de alerta enviado automaticamente via *e-mail*. Fonte: Autor, 2023

usuário com o valor de medição atual do parâmetro, conforme a figura 38. A seguir, será utilizada de exemplo a função de disparo do valor crítico de luminosidade:

```
var targetLum = flow.get('targetLum') || 50;
var toleranceLum = flow.get('toleranceLum') || 10;
var alertLum = flow.get('alertLum') || 70;

// Manda e-mail caso a medição for menor
// que a soma do alvo e tolerância
if (alertLum <= targetLum + toleranceLum) {
  msg.topic = "Alerta de luminosidade";
  msg.payload = "A luminosidade atingiu o valor crítico.
  Luminosidade atual: " + alertLum + "%";
  return [msg, null];
} else {
  return [null, msg];
}
```

O fluxo construído pode ser observado de maneira integral na figura 39.

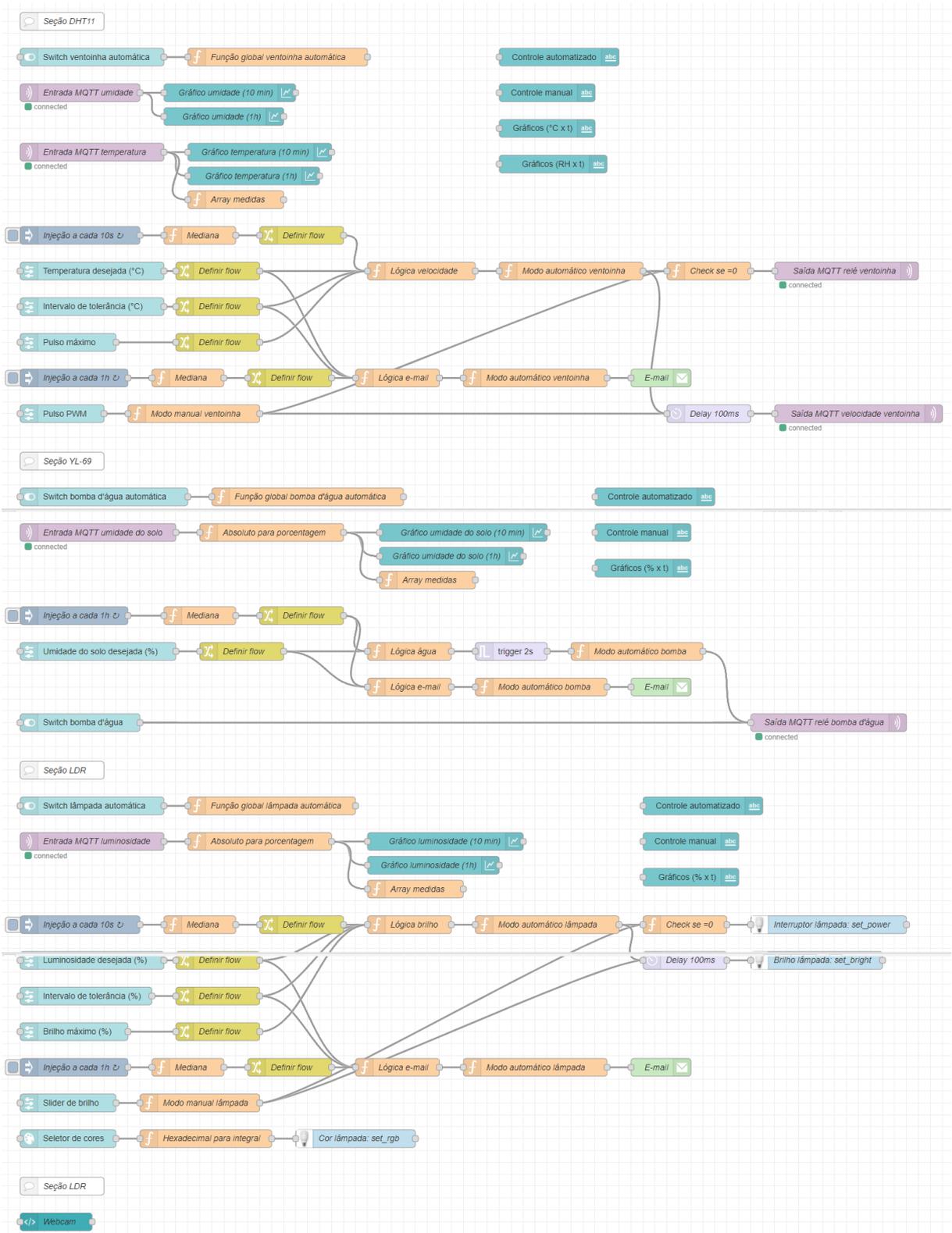


Figura 39 – Representação visual do sistema completo montado no Node-RED. Fonte: Autor, 2023

5 RESULTADOS

5.1 Testes e simulação

A partir das especificações e modelo projetado, é possível executar algumas simulações para observar o resultado de sistema de monitoramento e controle automatizado. A montagem do protótipo, representado pela figura 40, foi bem sucedida e feita de acordo com o circuito da figura 25. Utilizando o *dashboard* (figura 41), que foi dividido em 4 seções dedicadas à temperatura, umidade do solo, luminosidade e transmissão visual, é possível realizar o monitoramento direto dos dados coletados pelos sensores e ajustar todos os atuadores, conforme esperado.



Figura 40 – Visão geral do protótipo. Fonte: Autor, 2023

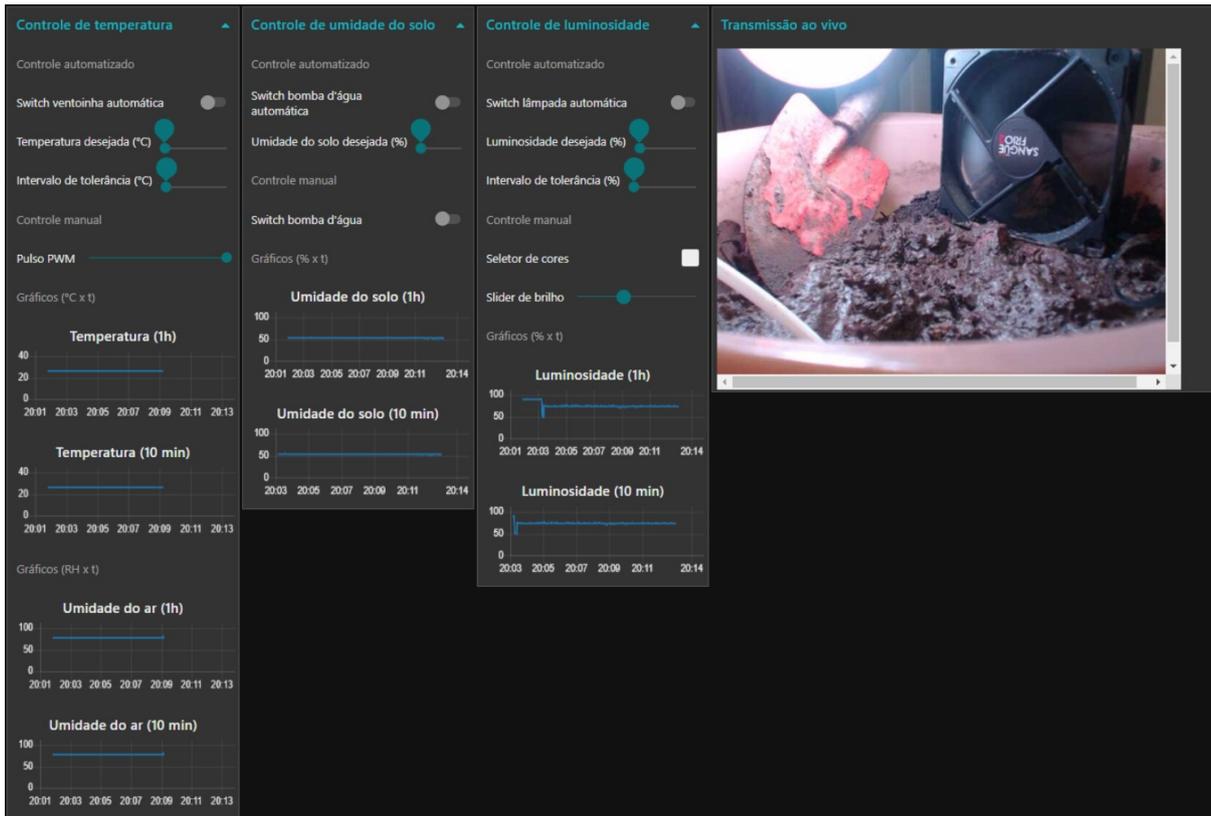


Figura 41 – *Dashboard* com opções de controle e dados. Fonte: Autor, 2023

Sendo o atuador com maior influência direta observável, o efeito de ajuste de brilho da lâmpada em um ambiente com pouca luz pode ser observado na figura 42. Para este caso, o sensor LDR foi movido para um local com linha de visada à 25cm de distância da lâmpada.

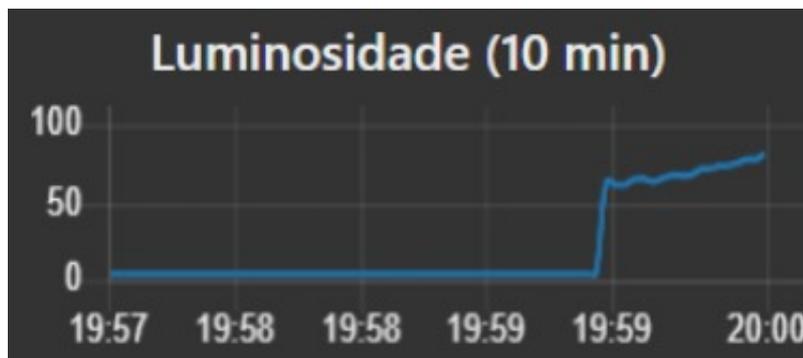


Figura 42 – Resultado do aumento gradativo de brilho de uma lâmpada a cerca de 25cm do LDR em um ambiente com pouca iluminação. Fonte: Autor, 2023

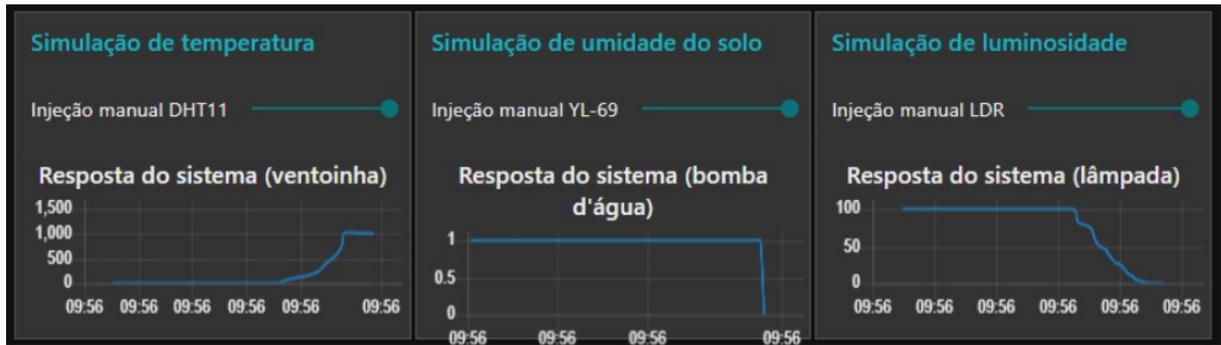


Figura 43 – Simulação de regulação dos atuadores. Fonte: Autor, 2023

Para executar um teste pragmático do ajuste automático dos atuadores, um segundo *dashboard* foi montado para observar os valores de resposta do sistema a partir da inserção de valores de leitura dos sensores de maneira manual. Observado na figura 43, a alteração gradativa das medições de entrada resultou em comportamentos de função parabólica dos atuadores de temperatura e luminosidade, e binário no caso de umidade do solo.

6 CONCLUSÃO

6.1 Síntese

A agricultura de precisão é uma tendência na indústria agrícola devido ao provimento de maior qualidade dos produtos resultantes, maior confiança no sistema e menor custo de trabalho [Sushmitha, Jyothi e Sain 2022], e a utilização de estufas automatizadas baseadas em conceitos de IoT podem reduzir o consumo de energia e água por aproximadamente 25% e 33% respectivamente, dobrar o crescimento médio de frutas como o tomate e construir um sistema de decisões lógicas mais consistentes e efetivas [Li et al. 2021]. Alguns outros benefícios que podem ser observados ao empregar o sistema incluem [Ardiansah et al. 2020]:

- Independência de temporada, solo e condições climáticas externas para o cultivo de plantas.
- Maior rendimento de espaço, como no caso de plantações verticais.
- Maior eficiência no consumo de energia.
- Redução no uso de fertilizantes.
- Melhor controle de pragas e doenças.
- Promoção do crescimento e amadurecimento das plantas e frutos.

O modelo e protótipo construído utilizou múltiplas referências para a escolha de componentes e métodos consistentes com o objetivo do projeto, proporcionando uma aplicação de monitoramento e controle dos parâmetros ambientais de maior importância utilizando dispositivos de baixo custo. No entanto, a natureza da proposta resulta em uma implementação customizável e versátil que deve ser ajustada conforme as necessidades da aplicação a partir de estudos de comportamento para garantir uma estabilidade dos parâmetros de base observados.

Este projeto tem um objetivo facilmente compreensível e de extrema importância, porém sua aplicação e otimização envolve diversos tópicos vistos com detalhe durante a graduação de engenharia elétrica, como processamento digital de sinais, controle, comunicações, eletrônica, programação e mecânica de fluidos, e uma solução totalmente satisfatória nunca foi implementada devido à sua complexidade elevada.

6.2 Próximos passos

6.2.1 Alternativas e melhorias

Existem inúmeras alternativas e melhorias que podem ser aplicadas ao projeto devido à sua capacidade de modularidade. Sensores de maior precisão estão disponíveis no mercado, e existem calibrações que podem ser feitas para melhor representar os parâmetros do ambiente e solo conforme a aplicação, como no caso apresentado em [Anire, Cruz e Agulto 2017], no qual foi aplicada uma técnica de utilização de 4 sensores DHT22 em um local isolado com valores ambientais de referência conhecidos para efetuar correções controladas das medidas.

Um dos atuadores com maiores limitações dentro do modelo empregado é a bomba d'água, que pode ser modificada para uma configuração em que seja possível customizar os tempos de ativação e intervalo entre cada verificação de leitura.

A aplicação em baixa escala possui algumas características que devem ser modificadas conforme as demandas de espaço e plantas aumentam. Especificamente, a tecnologia *Wi-Fi* deverá ser substituída por uma das alternativas citadas no capítulo 3, como *ZigBee* e *LoRa*, devido às limitações principais de alto consumo de energia e capacidade baixa de um volume de dispositivos conectados simultaneamente. Um sistema que utiliza múltiplos microcontroladores também deve ser desenvolvido, e existem 3 possibilidades de configuração, representados na figura 44:

1. Esta configuração utilizará uma média das leituras dos sensores para tomada de

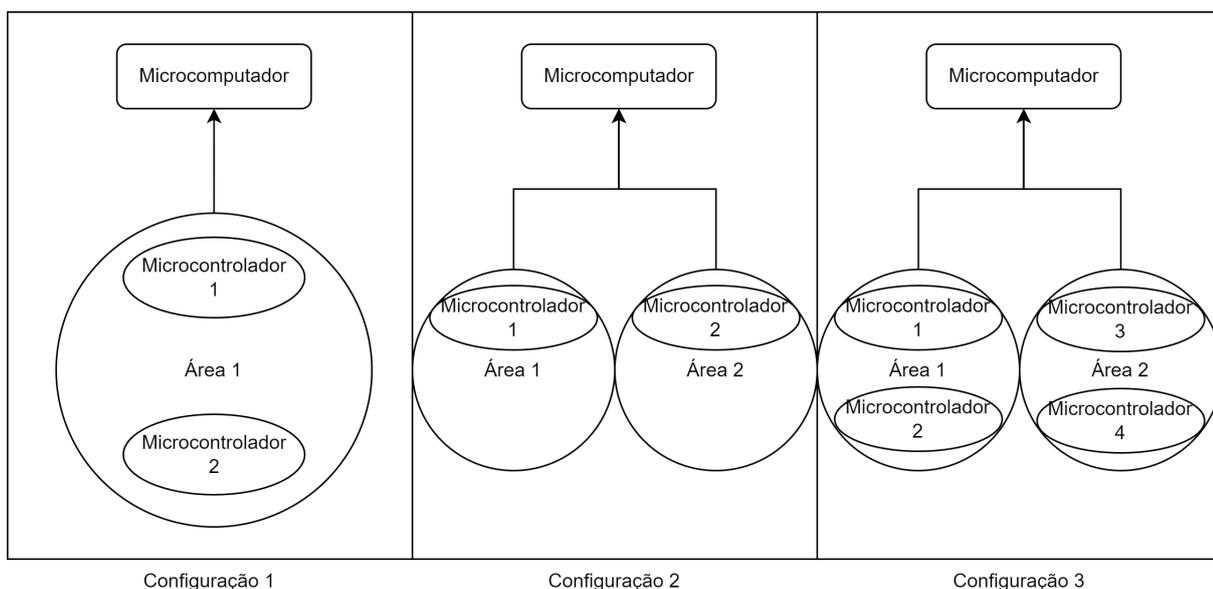


Figura 44 – Representação das configurações possíveis utilizando múltiplos microcontroladores. Fonte: Autor, 2023

decisões, tratando a estufa como um espaço unificado, e é útil em casos de pequena e média escala em que não existem grandes variações de parâmetros locais quando comparados aos globais.

2. A segunda configuração separa cada microcontrolador em uma área dedicada para decisões individuais. Um caso de aplicação seria em uma escala grande com objetivos ou plantas diferentes para cada um dos sensores e atuadores.
3. A última configuração utiliza os princípios das estruturas já mencionadas para possibilitar uma divisão em áreas que são calculadas a partir da média dos sensores presentes. Este modelo proporciona uma precisão alta de medição e uma possibilidade de atuação independente para cada uma das áreas.

6.2.2 Sugestões de possíveis modificações

Algumas modificações que podem ser relevantes dependendo da necessidade do projeto e plantas cultivadas são:

- A utilização de telas com dados medidos importantes para auxiliar qualquer pessoa que esteja ativamente interferindo e trabalhando no local.
- Um sensor de nível de água instalado no reservatório, capaz de enviar um alerta caso o volume de água esteja baixo.
- *Backups* de energia, sensores e atuadores, bem como um sistema de alarme no caso de falhas para evitar a interrupção de funcionamento do sistema [Arshad et al. 2020].
- A implementação de um sistema de resfriamento, como a utilização de condicionadores de ar e janelas deslizadoras, e aquecimento, para efetivamente influenciar a temperatura do local. O sistema de ventoinhas pode ser modificado para incluir um modelo de *intake* e *exhaust*, que auxilia na circulação de ar e resfriamento do local (figura 45).
- Um sistema de detecção visual utilizando a *webcam* para identificação da presença de animais e pragas ou doenças, bem como um acompanhamento de crescimento das plantas, utilizando princípios de processamento de sinais pode ser importante em alguns casos, como aplicações residenciais ou em céu aberto.
- Implementações, como o método de *buzzer* mencionado no capítulo 2, para controle de pragas.
- A obtenção de um modelo empírico de resposta dos atuadores para situações e épocas específicas em casos que se mostra necessário obter maior precisão no controle do sistema.

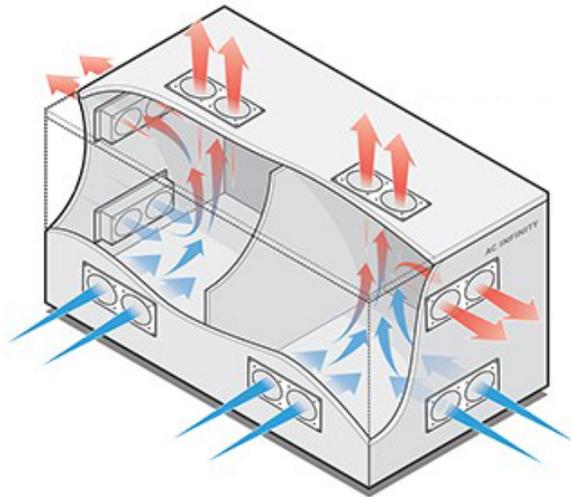


Figura 45 – Representação de um sistema de *intake* e *exhaust*. Fonte: Adaptado de AC Infinity

- A utilização de métodos de *machine learning* e AI para auxílio na tomada de decisões, análise de tendências e criação de alertas do sistema obtiveram bons resultados nos casos implementados recentemente e podem ser replicados [Li et al. 2021].
- Monitoramento e controle de outros parâmetros ambientais mencionados no capítulo 3, como a regulação de gás carbônico feita com base nas medições através do sensor TGS4161, uma solução custosa e de maior risco, mas eficiente [Ahonen, Virrankoski e Elmusrati 2008].
- Adição de um terceiro modo de controle que utiliza a possibilidade de programar eventos e ciclos de regulação dos atuadores [Chhorn et al. 2022].
- Implementação de técnicas de hidroponia, que são efetivas mas mais complexas e custosas, como mencionado no capítulo 2.

Referências

AHONEN, T.; VIRRANKOSKI, R.; ELMUSRATI, M. Greenhouse Monitoring with Wireless Sensor Network. IEEE, p. 403–406, 2008. Disponível em: <<https://ieeexplore.ieee.org/document/4735744>>. Citado 6 vezes nas páginas 29, 35, 38, 39, 40 e 88.

ANIRE, R. B.; CRUZ, F. R. G.; AGULTO, I. C. Environmental Wireless Sensor Network using Raspberry Pi 3 for Greenhouse Monitoring System. IEEE, p. 1–3, 2017. Disponível em: <<https://ieeexplore.ieee.org/document/8269426>>. Citado 2 vezes nas páginas 27 e 86.

ARDIANSAH, I. et al. Greenhouse Monitoring and Automation Using Arduino: a Review on Precision Farming and Internet of Things (IoT). International Journal on Advanced Science, Engineering and Information Technology, p. 703–706, 2020. Disponível em: <<http://insightsociety.org/ojaseit/index.php/ijaseit/article/view/10249>>. Citado 7 vezes nas páginas 13, 27, 28, 33, 34, 37 e 85.

ARDIANSAH, I. et al. Design of Micro-Climate Data Monitoring System for Tropical Greenhouse based on Arduino UNO and Raspberry Pi. IOP, p. 2–10, 2021. Disponível em: <<https://iopscience.iop.org/article/10.1088/1755-1315/757/1/012017>>. Citado 4 vezes nas páginas 13, 27, 40 e 41.

ARDUINO Documentation. 2023. Disponível em: <<https://docs.arduino.cc/>>. Citado na página 52.

ARSHAD, J. et al. Intelligent greenhouse monitoring and control scheme: An arrangement of Sensors, Raspberry Pi based Embedded System and IoT platform. Indian Journal of Science and Technology, p. 1–12, 2020. Disponível em: <<https://indjst.org/articles/intelligent-greenhouse-monitoring-and-control-scheme-an-arrangement-of-sensors-raspberry-pi-based->>. Citado 3 vezes nas páginas 27, 35 e 87.

BARBOSA, D. C. Sistema de Monitoramento e Controle de Cultivos em Estufas para Produção Hidropônica. IFSC, p. 18–25, 2019. Disponível em: <<https://repositorio.ifsc.edu.br/handle/123456789/1182>>. Citado 3 vezes nas páginas 27, 33 e 40.

CEPEA. Sumário executivo - PIB do agronegócio. CEPEA, p. 5–7, 2023. Disponível em: <https://www.cepea.esalq.usp.br/upload/kceditor/files/01sut.pib_mar_2023.jun2023%20-%20SUM%C3%81RIO%20EXECUTIVO.pdf>. Citado na página 27.

CHHORN, S. et al. Development of ESP32-Based Smart Greenhouse Controller. IEEE, p. 2–3, 2022. Disponível em: <<https://ieeexplore.ieee.org/document/10152112>>. Citado 2 vezes nas páginas 48 e 88.

COSTA, V. M. d. et al. Monitoramento e Suporte a Controle de Estufas Climatizadas Utilizando Nuvem e MQTT. SBC, p. 2–3, 2021. Disponível em: <<https://sol.sbc.org.br/index.php/ersi-rj/article/view/16982>>. Citado 3 vezes nas páginas 27, 29 e 50.

- CRUZ, R. S. d. et al. Calibração do Sensor YL-69 Para Fins de Monitoramento da Umidade de Composteiras. Editora Realize, p. 3–5, 2019. Disponível em: <https://www.editorarealize.com.br/editora/anais/join/2019/TRABALHO_EV124_MD1_SA45_ID1867_23082019232707.pdf>. Citado na página 44.
- DANITA, M. et al. IoT based Automated Greenhouse Monitoring System. IEEE, p. 1993–1936, 2018. Disponível em: <<https://ieeexplore.ieee.org/document/8662911>>. Citado 6 vezes nas páginas 29, 31, 43, 44, 52 e 72.
- DEUS, A. M. d. Aplicativo para Monitoramento de Estufa Agrícola com Comunicação Lorawan. UFSC, p. 19–20, 2022. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/237568>>. Citado na página 27.
- DHT11 Humidity and Temperature Sensor. p. 3–3, 2023. Disponível em: <<https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>>. Citado 2 vezes nas páginas 13 e 44.
- DOCKER Docs. 2023. Disponível em: <<https://docs.docker.com/>>. Citado 2 vezes nas páginas 53 e 62.
- FERNANDEZ, J. E.; BAILEY, B. J. The Influence of Fans on Environmental Conditions in Greenhouses. Silsoe Research Institute, p. 201–202, 1994. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0021863484710493>>. Citado na página 46.
- GANESAN, K. et al. Raspberry-Pi Based Automated Greenhouse. IRJET, p. 1–5, 2018. Disponível em: <<https://www.irjet.net/archives/V8/i1/IRJET-V8I1115.pdf>>. Citado na página 35.
- GROKHOTKOV, I. Esp8266 arduino cores documentation. 2017. Disponível em: <<https://arduino-esp8266.readthedocs.io/en/latest/>>. Citado na página 42.
- KUMAR, A. et al. Role of Greenhouse Technology in Agricultural Engineering. Int. J. Agric. Res, p. 364–372, 2006. Citado 2 vezes nas páginas 13 e 38.
- LI, H. et al. Towards automated greenhouse: A state of the art review on greenhouse monitoring methods and technologies based on internet of things. Elsevier, p. 1–14, 2021. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0168169921005755>>. Citado 12 vezes nas páginas 13, 28, 29, 32, 38, 39, 40, 48, 49, 50, 85 e 88.
- MOSQUITTO Documentation. 2023. Disponível em: <<https://mosquitto.org/documentation/>>. Citado na página 63.
- MOTION Documentation. 2023. Disponível em: <https://motion-project.github.io/motion_guide.html>. Citado na página 65.
- NGINX Documentation. 2023. Disponível em: <<https://nginx.org/en/docs/>>. Citado na página 64.
- NODA, T.; KOBAYASHI, T.; SUDA, I. Effect of soil temperature on starch properties of sweet potatoes. Carbohydrate Polymers, p. 239–246, 2001. Citado na página 39.

NODE-RED Documentation. 2023. Disponível em: <<https://nodered.org/docs/>>. Citado 2 vezes nas páginas 52 e 65.

OASIS. Mqtt version 3.1.1. 2014. Disponível em: <<https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf>>. Citado na página 51.

PORTAINER Documentation. 2023. Disponível em: <<https://docs.portainer.io/>>. Citado 2 vezes nas páginas 53 e 62.

RASPBERRY Pi Documentation. 2023. Disponível em: <<https://www.raspberrypi.com/documentation/>>. Citado 4 vezes nas páginas 13, 43, 52 e 61.

SANFORD, S. Energy-Efficient Agricultural Lighting. University of Wisconsin Extension, p. 1–2, 2004. Citado na página 48.

SANTOS, O. L. D.; JÚNIOR, J. D. S. M.; NEVES, M. R. M. Sistema de Controle de Temperatura para uma Estufa com Monitoramento via Aplicativo. Revista de Ciência e Tecnologia, p. 5–14, 2017. Disponível em: <<https://revista.ufr.br/rct/article/view/4004>>. Citado 4 vezes nas páginas 14, 69, 70 e 71.

SHAH, N. P.; BHATT, P. P. Greenhouse Automation and Monitoring System Design and Implementation. International Journal of Advanced Computer Research, p. 469–470, 2017. Disponível em: <https://www.researchgate.net/publication/322020853_GREENHOUSE_AUTOMATION_AND_MONITORING_SYSTEM_DESIGN_AND_IMPLEMENTATION>. Citado 3 vezes nas páginas 13, 43 e 44.

SILVA, A. S. e.; TROFINO, A. Princípios de Sistemas de Controle. Universidade Federal de Santa Catarina, p. 3–10. Citado na página 72.

SUSHMITHA, M.; JYOTHI, V.; SAIN, A. V. Implementation of Greenhouse Control and Monitoring System Using ESP32. IEEE, p. 1–2, 2022. Disponível em: <<https://ieeexplore.ieee.org/document/9792852>>. Citado 4 vezes nas páginas 27, 29, 45 e 85.

TANG, Y. et al. The effect of temperature and light on strawberry production in a solar greenhouse. Sol. Energy, p. 318–328, 2020. Citado na página 40.

TIMMERMAN, G. J.; KAMP, P. G. H. Computerised Environmental Control in Greenhouses. PTC+, p. 15–124, 2003. Citado 2 vezes nas páginas 13 e 41.

ULLAH, M. H.; GUNAWAN, T. S.; MUHIDA, R. Development of a Electrically Inspired Low Emission Microcontroller Based Hybrid Vehicle. American Journal of Applied Sciences, p. 1732–1732, 2012. Disponível em: <https://www.researchgate.net/publication/232031887_Development_of_a_Electrically_Inspired_Low_Emission_Microcontroller_Based_Hybrid_Vehicle>. Citado 2 vezes nas páginas 13 e 47.

UMAROV, A. et al. Microclimate Monitoring System for a Home Greenhouse as Part of ESP32. TEST Engineering Management, p. 4569–4570, 2020. Disponível em: <<https://www.testmagazine.biz/index.php/testmagazine/article/view/1558>>. Citado 2 vezes nas páginas 15 e 42.

ZHANG, Q. et al. A wireless solution for greenhouse monitoring and control system based on ZigBee technology. Zhejiang University, p. 1584–1585, 2007. Disponível em: <<https://link.springer.com/article/10.1631/jzus.2007.A1584>>. Citado 6 vezes nas páginas 13, 29, 32, 33, 49 e 50.

Apêndices

APÊNDICE A – Código em C++ final utilizado no ESP8266

```

#include <DHT.h>           // Biblioteca de sensores DHT
#include <ESP8266WiFi.h>   // Biblioteca WiFi
#include <PubSubClient.h> // Biblioteca MQTT

// Conexão ao servidor MQTT
const char* mqtt_server = "#IPRPI";
const char* mqtt_user = "#USUÁRIO"; // Usuário configurado
// no broker
const char* mqtt_password = "#SENHAMQTT"; // Senha configurada
// no broker
const char* clientId = "ESP8266Client";

// Credenciais do WiFi
const char* ssid = "#SSID";
const char* password = "#SENHAWIFI";

#define DHTPIN 15      // D8
#define DHTTYPE DHT11 // Tipo do sensor

#define SOIL_PIN 13    // D7
#define LDR_PIN 12     // D6

#define FAN_PIN 2      // D4
#define PUMP_PIN 4     // D2
#define PWM_FAN_PIN 14 // D5

DHT dht(DHTPIN, DHTTYPE);
WiFiClient espClient;
PubSubClient client(espClient);

void setup() {
  Serial.begin(9600);

```

```
pinMode(SOIL_PIN, OUTPUT);
pinMode(LDR_PIN, OUTPUT);
digitalWrite(SOIL_PIN, LOW);
digitalWrite(LDR_PIN, LOW); // Desligando ambos os sensores
// analógicos inicialmente

dht.begin();

setup_wifi();
client.setServer(mqtt_server, 1883);

pinMode(FAN_PIN, OUTPUT);
pinMode(PUMP_PIN, OUTPUT);

digitalWrite(FAN_PIN, LOW);
digitalWrite(PUMP_PIN, LOW); // Desligando ventoinha
// e bomba inicialmente

// Callback MQTT
client.setCallback(mqttCallback);

pinMode(PWM_FAN_PIN, OUTPUT);
analogWrite(PWM_FAN_PIN, 0); // Ajustando pulso PWM para 0
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  // Leitura dos parâmetros do DHT11
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  char temp_buff[8];
  char hum_buff[8];

  dtostrf(t, 1, 2, temp_buff);
```

```
    dtostrf(h, 1, 2, hum_buff); // Transformando medidas
    // float em string

    // Leitura YL-38 & YL-69
    digitalWrite(SOIL_PIN, HIGH); // Liga o sensor de
    // umidade do solo
    delay(100); // Delay para leitura
    // e estabilização
    int soilValue = analogRead(A0);
    digitalWrite(SOIL_PIN, LOW); // Desliga o sensor

    // Leitura LDR
    digitalWrite(LDR_PIN, HIGH);
    delay(100);
    int ldrValue = analogRead(A0);
    digitalWrite(LDR_PIN, LOW);

    String soilMsg = String(soilValue);
    String ldrMsg = String(ldrValue);

    client.publish("home/temperature", temp_buff);
    client.publish("home/humidity", hum_buff);
    client.publish("home/soilMoisture", soilMsg.c_str());
    client.publish("home/luminosity", ldrMsg.c_str()); // Nome
    // dos tópicos publicados

    delay(2000); // Medidas enviadas a cada 2 segundos
}

void mqttCallback(char* topic, byte* payload,
unsigned int length) {
    String topicStr = topic;

    // Converte o payload em string
    payload[length] = '\0';
    String payloadStr = String((char*)payload);

    if (topicStr == "home/fanControl") {
        if (payloadStr == "ON") {
```

```
        digitalWrite(FAN_PIN, HIGH);
    } else {
        digitalWrite(FAN_PIN, LOW);
    }
}

if (topicStr == "home/pumpControl") {
    if (payloadStr == "ON") {
        digitalWrite(PUMP_PIN, HIGH);
    } else {
        digitalWrite(PUMP_PIN, LOW);
    }
}

if (topicStr == "home/fanSpeedControl") {
    int pwmValue = payloadStr.toInt();
    if (pwmValue >= 0 && pwmValue <= 1023) {
        analogWrite(PWM_FAN_PIN, pwmValue);
    }
}
}

void setup_wifi() {
    delay(10);
    Serial.println();
    Serial.print("Conectando a ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi conectada");
    Serial.println("Endereço de IP: ");
    Serial.println(WiFi.localIP());
}

void reconnect() {
    while (!client.connected()) {
```

```
Serial.print("Estabelecendo conexão MQTT...");
if (client.connect(clientId, mqtt_user, mqtt_password)) {
  Serial.println("conectado");
} else {
  Serial.print("falha, rc=");
  Serial.print(client.state());
  Serial.println(" tente novamente em 5 segundos");
  delay(5000);
}
}
client.subscribe("home/fanControl");
client.subscribe("home/pumpControl");
client.subscribe("home/fanSpeedControl");
}
```