



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
AMAZONAS
CAMPUS MANAUS DISTRITO INDUSTRIAL
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

PEDRO IVAN DAS GRAÇAS PALHETA NETO

SISTEMA AUTOMATIZADO PARA CULTIVO DE PLANTAS BASEADO EM
INTERNET DAS COISAS

MANAUS - AM
2023

PEDRO IVAN DAS GRAÇAS PALHETA NETO

SISTEMA AUTOMATIZADO PARA CULTIVO DE PLANTAS BASEADO EM
INTERNET DAS COISAS

Trabalho de Conclusão de Curso
apresentado como requisito parcial à
obtenção do título de Bacharel em
Engenharia de Controle e Automação, do
Instituto Federal de Educação, Ciência e
Tecnologia do Amazonas, Campus
Manaus Distrito Industrial - IFAM/CMDI.

Orientador: Me. Pedro Ivan Das Graças
Palheta

Coorientador: Dr. Cleonor Crescêncio das
Neves

MANAUS - AM
2023

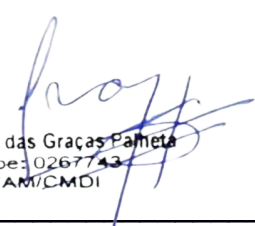
Dados Internacionais de Catalogação na Publicação (CIP)

P153s	Palheta Neto, Pedro Ivan das Graças. Sistema automatizado para cultivo de plantas baseado em internet das coisas / Pedro Ivan das Graças Palheta Neto. — Manaus, 2023. 93f.: il. color. Monografia (Graduação) — Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, <i>Campus</i> Manaus Distrito Industrial, Curso de Engenharia de Controle e Automação, 2023. Orientador: Prof.º Pedro Ivan das Graças Palheta, Me. Coorientador: Prof.º Cleonor Crescêncio das Neves, Dr. 1. Estufa. 2. Monitoramentos. 3. Sensores. 4. Thingspeak. I. Palheta, Pedro Ivan das Graças. II. Neves, Cleonor Crescêncio das. III. Instituto Federal de Educação, Ciência e Tecnologia do Amazonas. IV. Título. CDD 629.89
-------	--

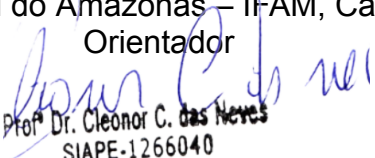
Elaborada por Oziane Romualdo de Souza (CRB11/ nº 734)

Folha de Aprovação

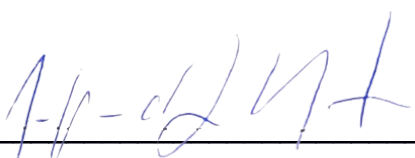
BANCA EXAMINADORA


Pedro Ivan das Graças Palheta
Siape: 0267743
IFAM/CMDI

Prof. Msc. Pedro Ivan das Graças Palheta
Instituto Federal do Amazonas – IFAM, Campus CMDI
Orientador


Prof. Dr. Cleonor C. das Neves
SIAPE-1266040
CMDI-IFAM

Prof. Dr. Cleonor Crescêncio das Neves
Instituto Federal do Amazonas – IFAM, Campus CMDI
Coorientador


Prof. DSc. José Pinheiro de Queiroz Neto
Instituto Federal do Amazonas – IFAM, Campus CMDI


Prof. Msc. Alexandre Lopes Martiniano
Instituto Federal do Amazonas – IFAM, Campus CMDI

A minha família e amigos.

AGRADECIMENTOS

Primeiramente agradecer a Deus, pois graças a Ele tenho a oportunidade de existir e realizar alguns feitos em minha vida, como esse TCC, às vezes fico me perguntando se isso é bom ou não.

Agradeço a minha mãe Iones Gomes Palheta e meu pai Pedro Ivan Das Graças Palheta Júnior, grande parte do que sou foram moldados por esses dois tanto geneticamente falando como por valores e educação passada. Sempre colocam as minhas necessidades acima deles mesmos, me apoiando e motivando nos momentos mais difíceis.

Agradeço aos meus amigos Rebecca Oliveira Lopes, Lucas Paiva Monteiro, Janderson Mike Do Nascimento Jati, Hylbert Bentes Rocha Rodrigues, entre outros, que foram de grande ajuda através de discursos, trocas de ideias, feedback e revisões. Suas companhias proporcionaram momentos essenciais para minha estabilidade emocional.

Gratidão às correções e ensinamentos transmitidos pelo meu orientador, professor e avô Pedro Ivan Das Graças Palheta e meu coorientador Cleonor Crescencio das Neves.

Por último, mas não menos importante, agradeço às pessoas que conheci ao longo desses anos de estudos no Instituto Federal do Amazonas. Foi uma experiência que contribuiu para a minha construção profissional, acadêmica e pessoal.

El lobo siempre cuida su loba.

RESUMO:

Este trabalho consiste no desenvolvimento de uma estufa indoor para o cultivo de plantas por meio de um sistema automatizado utilizando o princípio da internet das coisas. Com o emprego do controle das variáveis de temperatura, umidade do ar e do solo e luminosidade, utilizando o sensoriamento conectado à plataforma ESP32. Proporciona-se ao usuário o controle e acesso ao status da estufa, fornecendo um ambiente para cultivo de qualidade e de fácil construção.

Palavras-chave: Estufa. Monitoramento. Sensores. Thingspeak.

ABSTRACT:

This work consists of developing an indoor greenhouse for growing plants through an automated system using the principle of the internet of things. Using the control of temperature variables, air and soil humidity and luminosity, using sensing connected to the ESP32 platform. The user is given control and access to the status of the greenhouse, providing an environment for quality cultivation and easy construction.

Keywords:

Greenhouse. Monitoring. Sensors. Thingsspeak.

LISTA DE ILUSTRAÇÕES

Figura 1 - Sequência do método hipotético-dedutivo.....	21
Figura 2 - ESP32-D0WDQ6.....	32
Figura 3 - Diagrama de blocos funcionais.....	33
Figura 4 - Sensor de temperatura encapsulado a prova d' água DS18B20.....	35
Figura 5 - Sensor de umidade e temperatura DHT11.....	36
Figura 6 - Sensor higrômetro FC-28.....	38
Figura 7 - Sensor de luminosidade LDR 5mm.....	39
Figura 8 - Sensor de nível de água tipo boia.....	40
Figura 9 - Lâmpada led.....	41
Figura 10 - Lâmpada incandescente.....	42
Figura 11 - Bomba submersa para aquário.....	43
Figura 12 - Cooler fan ventoinha.....	44
Figura 13 - Módulo relé 4 canais 5V.....	45
Figura 14 - Módulo relé de estado sólido SSR de 1 canal.....	46
Figura 15 - Mini fonte chaveada estabilizada colmeia.....	47
Figura 16 - Regulador de tensão LM2596.....	48
Figura 17 - Diagrama do sistema IOT.....	49
Figura 18 - Representação funcionamento do projeto.....	51
Figura 19 - Site ThingSpeak.....	52
Figura 20 - Site de acesso ao ThingSpeak.....	53
Figura 21 - Site ThingSpeak.....	53
Figura 22 - Site de acesso ao ThingSpeak.....	54
Figura 23 - Site ThingSpeak.....	55
Figura 24 - Fluxograma de funcionamento do sistema.....	56
Figura 25 - Logo do Boxes py.....	57
Figura 26 - Logo do Boxes py.....	58
Figura 27 - Configuração para gerar case em MDF.....	59
Figura 28 - Abrindo um desenho no AutoCAD.....	60
Figura 29 - Desenho da caixa já modificada.....	60
Figura 30 - Ícone do RDWorksV8.....	61
Figura 31 - Opção de importação no RDWorks.....	61
Figura 32 - Desenho da caixa em formato na área de impressão do RDWors.....	62
Figura 33 - Área de configuração das camadas de cortes.....	62
Figura 34 - Configuração da camada de corte utilizada.....	63
Figura 35 - Display LCD mostrando a temperatura.....	64
Figura 36 - Display LCD mostrando o ajuste da temperatura.....	65
Figura 37 - Display LCD mostrando a umidade do solo.....	65
Figura 38 - Display LCD mostrando o ajuste da umidade do solo.....	65
Figura 39 - Display LCD mostrando umidade do ar.....	66

Figura 40 - Display LCD mostrando o ajuste da umidade do ar.....	66
Figura 41 - Caixa de isopor com duas ventoinhas acopladas em seus devidos lugares.....	67
Figura 42 - Sensores (DTH11, DS18B20, Higrômetro) acoplados na parte interna da caixa de isopor.....	68
Figura 43 - Case de MDF com o sistema de controle e acionamento em sua parte interna.....	68
Figura 44 - Bibliotecas utilizadas no código.....	69
Figura 45 - Representação das ligações elétricas do sistema.....	75
Figura 46 - Pacote de sementes utilizadas como amostras teste.....	76
Figura 47 - Bandejas de plástico com 15 células usadas na amostra da externa e da estufa.....	77
Figura 48 - Bandeja de plástico com a tubulação de irrigação já instalada.....	77
Figura 49 - Bandejas de plástico já semeadas.....	78
Figura 50 - Configuração da estufa via IHM.....	79
Figura 51 - Gráfico de temperatura e temperatura apresentada através do dashboard do ThingSpeak, referentes ao DS18B20.....	80
Figura 52 - Gráfico de temperatura e temperatura apresentada através do dashboard do ThingSpeak, referentes ao DHT11.....	80
Figura 53 - Gráfico de umidade e umidade apresentada através do dashboard do ThingSpeak.....	81
Figura 54 - Gráfico de umidade do solo e umidade do solo apresentada através do dashboard do ThingSpeak.....	82
Figura 55 - Estado da caixa d' água, reservatório vazio (luz vermelha acesa) e reservatório cheio (luz verde acesa).....	82
Figura 56 - Painel com a localização da estufa (a esquerda), índice de luminosidade dentro da estufa (a direita).....	83
Figura 57 - Tela inicial do canal de monitoramento da estufa através do ThingSpeak.....	84
Figura 58 - Acessando a área de importação de dados do canal.....	85
Figura 59 - Ícone do arquivo.....	85
Figura 60 - Formato em que os dados são apresentados.....	86
Figura 61 - Gráfico da variação da temperatura e umidade.....	87
Figura 62 - Registro de desenvolvimento das microverdes 1.....	89
Figura 63 - Registro de desenvolvimento das microverdes 2.....	90
Figura 64 - Último registro da microverde na estufa.....	91
Figura 65 - Último registro da microverde na área externa.....	91

LISTA DE TABELAS

Tabela 1: Materiais utilizados na montagem do projeto. (continua).....	30
Tabela 1: Materiais utilizados na montagem do projeto. (conclusão).....	31
Tabela 2: Especificações técnicas ESP32-D0WDQ6 (continua).....	34
Tabela 2: Especificações técnicas ESP32-D0WDQ6 (conclusão).....	34
Tabela 3: Especificações Técnicas do Sensor de temperatura encapsulado a prova de água DS18B20.....	36
Tabela 4: Especificações técnicas do Sensor de umidade e temperatura DHT11.....	38
Tabela 5: Especificações técnicas do sensor higrômetro FC-28.....	39
Tabela 6: Especificações técnicas do sensor de luminosidade LDR 5mm.....	40
Tabela 7: Especificações técnicas do sensor de nível de água tipo boia.....	41
Tabela 8: Especificações técnicas da lâmpada led.....	42
Tabela 9: Especificações técnicas da lâmpada incandescente.....	43
Tabela 10: Especificações técnicas da bomba submersa para aquário.....	44
Tabela 11: Especificações técnicas do cooler fan ventoinha.....	45
Tabela 12: Especificações técnicas do módulo relé 4 canais 5V.....	46
Tabela 13: Especificações técnicas do módulo relé de estado sólido SSR de 1 canal.....	47
Tabela 14: Especificações da mini fonte chaveada estabilizada colmeia.....	48
Tabela 15: Especificações técnicas do regulador de tensão LM2596.....	49
Tabela 16: Tabela da ligação dos componentes com a ESP32.....	75
Tabela 17 - Análise estatística dos valores de temperatura, umidade do ar, umidade do solo.	

LISTA DE ABREVIATURAS E SIGLAS

2D	Duas dimensões
A	Ampère
AC	Corrente Alternada
ADC	Conversor analógico-digital
API	Application Programming Interface (Interface de Programação de Aplicação)
CSV	Comma Separated Values (Valores Separados por Vírgulas)
CLPs	Controlador Lógico Programável
cm	Centímetro
DC	Corrente Contínua
EEPROM	(Electrically-Erasable Programmable Read-Only Memory) Memória Somente de Leitura Programável Apagável Eletronicamente
GPIO	General Purpose Input/Output
HMI	Human Machine Interface (Interface Homem-Máquina)
HTTP	HyperText Transfer Protocol (Protocolo de Transferência de Hipertexto)
HTTPS	Hypertext Transfer Protocol Secure (Protocolo de Transferência de Hipertexto Seguro)
Hz	Hertz
IDE	Integrated Development Environment (Ambiente de Desenvolvimento Integrado)
IHM	Interface Homem Máquina
IOT	Internet das Coisas
IP	Internet Protocol
IP	Ingress Protection (Grau de Proteção)
LCD	Liquid Crystal Display (Display de Cristal Líquido)
LDR	Led Dependente de luz
LED	Light-Emitting Diode (Diodo Emissor de Luz)
lm/W	Eficiência Luminosa - lúmens por Watts
MDF	Medium Density Fiberboard (Fibras de Média Densidade)
mm	Milímetro
MQTT	MQTT (Message Queuing Telemetry Transport)
ms	Milissegundo
Ω	Ohms
PCB	Placas de Circuito Impresso
RF	Radiofrequência
TCC	Trabalho de Conclusão de Curso
TIC	Tecnologia da Informação e Comunicação
UR	Umidade relativa
V	Volts
VAC	Tensão alternada

VDC	Tensão contínua
W	Watt
Wi-Fi	Wireless Fidelity

LISTA DE SÍMBOLOS

%	Porcentagem
C°	Graus Celsius
KΩ	Kilohm
mA	miliamperes
mW	miliwatts
MΩ	Megaohm
mm/s	Milímetros por segundo

SUMÁRIO

1. INTRODUÇÃO.....	19
1.1 OBJETIVOS GERAL.....	19
1.2 OBJETIVOS ESPECÍFICOS.....	19
1.3 METODOLOGIA.....	20
1.3.1 FINALIDADE DA PESQUISA.....	20
1.3.2 OBJETIVO DA PESQUISA.....	20
1.3.3 DEFINIÇÃO DA ABORDAGEM.....	20
1.3.4 MÉTODO.....	20
1.4 JUSTIFICATIVA.....	21
1.5 HIPÓTESES DO TRABALHO.....	21
2. REFERENCIAL TEÓRICO.....	22
2.1 AGRICULTURA.....	22
2.2 ESTUFA.....	23
2.3 FATORES CLIMÁTICOS.....	23
2.3.1 TEMPERATURA.....	24
2.3.2 UMIDADE DO AR.....	24
2.3.3 VENTILAÇÃO.....	24
2.3.4 LUMINOSIDADE.....	25
2.3.5 IRRIGAÇÃO.....	25
2.3.5.1.1 IRRIGAÇÃO LOCALIZADA.....	26
2.3.5.1.2 IRRIGAÇÃO POR ASPERSÃO.....	26
2.3.5.1.3 IRRIGAÇÃO DE SUPERFÍCIE.....	26
2.3.5.1.4 IRRIGAÇÃO DE SUBSUPERFÍCIE.....	27
2.4 AUTOMAÇÃO.....	27
2.5 IOT.....	27
2.6 PROTOCOLO DE COMUNICAÇÃO HTTP E API REST.....	28
3. MATERIAIS E MÉTODOS.....	30
3.1 MATERIAIS.....	30
3.1.1 ESP32.....	32
3.1.2 SENSORES.....	35
3.1.2.1 SENSOR DE TEMPERATURA DS18B20.....	35
3.1.2.2 SENSOR DE TEMPERATURA E UMIDADE DHT11.....	37
3.1.2.3 SENSOR HIGRÔMETRO FC-28.....	38
3.1.2.4 SENSOR DE LUMINOSIDADE LDR.....	39
3.1.2.4 SENSOR DE NÍVEL DE ÁGUA TIPO BOIA 90° HORIZONTAL.....	40
3.1.3 ATUADORES.....	41
3.1.3.1 LÂMPADA LED.....	42
3.1.3.2 LÂMPADA INCANDESCENTE.....	43
3.1.3.3 BOMBA DE ÁGUA.....	43
3.1.3.4 VENTOINHAS.....	44

3.1.3.5 MÓDULO RELÉ.....	45
3.1.3.6 MÓDULO RELÉ DE ESTADO SÓLIDO.....	46
3.1.4 ALIMENTAÇÃO.....	47
3.1.4.1 MINI FONTE CHAVEADA.....	47
3.1.4.2 REGULADOR DE TENSÃO LM2596.....	48
3.1.5 PLATAFORMA THINGSPEAK.....	49
3.2 MÉTODOS.....	50
3.2.1 DIAGRAMA DE FUNCIONAMENTO DO PROJETO.....	50
3.2.2 CONFIGURAÇÃO THINGSPEAK.....	53
3.2.3 FLUXOGRAMA DE FUNCIONAMENTO DO SISTEMA.....	57
3.3.1 DESENVOLVIMENTO DO CASE EM MDF.....	58
3.3.2.1 DESENVOLVIMENTO DA CAIXA ATRAVES DO BOXES.PY.....	58
3.3.3 AUTOCAD.....	60
3.3.3.1 MODIFICAÇÕES FEITAS NO AUTOCAD.....	60
3.3.4 RDWORKSV8.....	61
3.3.4.1 CONFIGURAÇÕES RDWorksV8.....	62
4. ANÁLISE DOS RESULTADOS.....	65
4.1 INTERFACE HOMEM-MÁQUINA.....	65
4.2.1 TELAS DE NAVEGAÇÃO.....	65
4.3 IMPLEMENTAÇÃO DO PROTÓTIPO.....	67
4.4 IMPLEMENTAÇÃO DO CÓDIGO.....	69
4.4.1 DECLARAÇÃO DAS BIBLIOTECAS.....	70
4.4.2 MAPEAMENTO DO HARDWARE.....	71
4.4.3 CONSTANTES E OBJETOS.....	71
4.4.4 DEFINIÇÃO DAS VARIÁVEIS E CONSTANTES GLOBAIS.....	72
4.4.5 FUNÇÃO SETUP.....	72
4.4.6 FUNÇÃO CONECTA WIFI.....	73
4.4.7 FUNÇÃO LOOP.....	73
4.4.7.1 FUNÇÃO LEITURA SENSORES.....	73
4.4.7.2 FUNÇÃO ENVIAR THINGSPEAK.....	73
4.4.7.3 FUNÇÃO CONDIÇÃO DE FUNCIONAMENTO.....	74
4.4.7.4 FUNÇÃO BOTÕES.....	74
4.5 MONTAGEM DO HARDWARE DO SISTEMA.....	75
4.6 PREPARO DAS AMOSTRAS.....	77
4.6.1 SELEÇÃO DAS PLANTAS.....	77
4.6.2 PREPARAÇÃO DAS BANDEJAS.....	77
4.7 SISTEMA DE IRRIGAÇÃO EM FUNCIONAMENTO.....	79
4.7.1 CONFIGURAÇÃO VIA IHM.....	79
4.7.2 CAPTAÇÃO DE DADOS PELO THINGSPEAK E VISUALIZAÇÃO DOS DADOS VIA WEB E MOBILE.....	80
4.7.3 EXPORTAÇÃO DOS DADOS.....	84
4.7.4 ANÁLISES ESTATÍSTICAS.....	87
4.8 ANÁLISE COMPARATIVA QUALITATIVA ENTRE O CULTIVO REALIZADO DENTRO E FORA DO PROTÓTIPO.....	89

5. CONSIDERAÇÕES FINAIS.....	93
REFERÊNCIAS.....	96

1. INTRODUÇÃO

A internet das coisas (IoT) junto com a automação tem sido tecnologias cada vez mais presentes em nossas vidas, permitindo a conexão de dispositivos entre si possibilitando a coleta e análise de dados em tempo real. Assim a aplicação desses conceitos no acionamento e monitoramento de estufas para cultivo indoor demonstra-se com potencial benefício teórico e prático para a agricultura moderna.

A capacidade de medir e alterar variáveis como iluminação, umidade do solo e temperatura ambiente pode melhorar significativamente o crescimento e desenvolvimento das plantas, aumentando a eficiência e produtividade da produção agrícola. Além disso, ao automatizar esse processo, pode-se reduzir custos e mão de obra humana, facilitando o cultivo de hortaliças tanto para pequenos produtores quanto para uso doméstico.

1.1 OBJETIVOS GERAL

Desenvolver um sistema de monitoramento e controle para uma estufa automatizada indoor que será utilizada para a produção de hortaliças.

1.2 OBJETIVOS ESPECÍFICOS

1. Identificar as tecnologias de sensoriamento/controle mais utilizadas e recomendadas para a construção de um sistema de automação para estufas indoor por meio de uma revisão bibliográfica.
2. Desenvolver um código baseado em linguagem C++ para o microcontrolador ESP32, implementando as funções de leitura dos sensores de luminosidade, umidade do ar e temperatura, bem como as funções de controle.
3. Implementar um protocolo de comunicação entre o software desenvolvido e a plataforma ThingSpeak, a fim de enviar e receber dados em tempo real.
4. Construção da estufa, levando em consideração as características físicas e ambientais necessárias para o correto funcionamento do sistema.
5. Avaliar o desempenho do sistema desenvolvido, tanto em relação ao software quanto ao hardware, a fim de garantir a qualidade e confiabilidade do sistema.

1.3 METODOLOGIA

Este estudo baseou-se em uma estratégia quantitativa de pesquisa, de caráter explicativo, por meio de uma pesquisa aplicada.

1.3.1 FINALIDADE DA PESQUISA

A pesquisa aplicada concentra-se em torno dos problemas presentes nas atividades das instituições, organizações, grupos ou atores sociais. Ela está empenhada na elaboração de diagnósticos, identificação de problemas e busca de soluções. Responde a uma demanda formulada por “clientes, atores sociais ou instituições” (Thiollent, 2009, p.36).

1.3.2 OBJETIVO DA PESQUISA

Sobre as pesquisas explicativas, são aquelas pesquisas que têm como preocupação central identificar os fatores que determinam ou que contribuem para a ocorrência dos fenômenos. Este é o tipo de pesquisa que mais aprofunda o conhecimento da realidade, porque explica a razão, o porquê das coisas (GIL, 2008).

1.3.3 DEFINIÇÃO DA ABORDAGEM

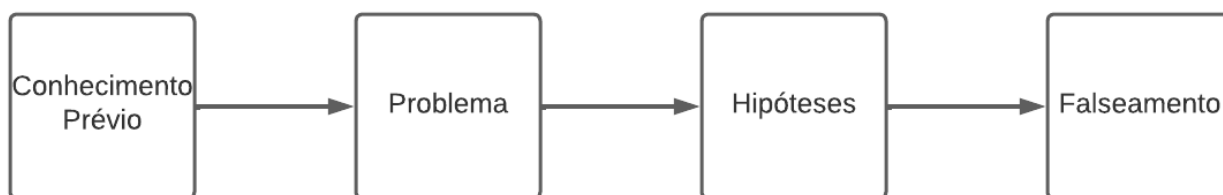
Diferentemente da pesquisa qualitativa, os resultados da pesquisa quantitativa podem ser quantificados. Como as amostras geralmente são grandes e consideradas representativas da população, os resultados são tomados como se constituíssem um retrato real de toda a população alvo da pesquisa. A pesquisa quantitativa se centra na objetividade. Influenciada pelo positivismo, considera que a realidade só pode ser compreendida com base na análise de dados brutos, recolhidos com o auxílio de instrumentos padronizados e neutros. A pesquisa quantitativa recorre à linguagem matemática para descrever as causas de um fenômeno, as relações entre variáveis, etc (FONSECA, 2002).

1.3.4 MÉTODO

O método utilizado foi o método hipotético-dedutivo. Conforme as etapas abaixo este método constrói hipóteses que são submetidas a testes de falseamento,

utiliza-se de discussão crítica, refutando ou aceitando as hipóteses, nesta diversidade de pensamentos sobre o método e o conhecimento, temos como certeza que o método hipotético dedutivo é um caminho que orienta em direção ao conhecimento das verdades e como fato se direciona ao objetivo que se busca ou mesmo se deseja, atingindo assim determinada meta dentro do conhecimento, (RAZUK,2015).

Figura 1 - Sequência do método hipotético-dedutivo.



Fonte: (RAZUK,2015, p.19), elaborado pelo autor.

1.4 JUSTIFICATIVA

Oferecer soluções tecnológicas acessíveis e sustentáveis para o cultivo de plantas em ambientes controlados para produtores e pequenos empreendedores baseado em soluções práticas e econômicas. O projeto tem a função de mimetizar um ambiente mais apropriado possível para o desenvolvimento de plantas saudáveis para consumo. O público-alvo deste trabalho são indivíduos que desejam cultivar suas próprias hortaliças ou plantas, porém enfrentam limitações de tempo e espaço adequado para cultivo. Além disso, também se destina a pequenos produtores que buscam maximizar sua produção em áreas reduzidas.

1.5 HIPÓTESES DO TRABALHO

A utilização do sistema de monitoramento e controle com IoT e microcontrolador ESP32 em ambientes controlados pode trazer benefícios significativos para o cultivo de hortaliças. Os parâmetros ambientais internos serão controlados e monitorados para maximizar a produtividade e qualidade das plantas cultivadas.

2. REFERENCIAL TEÓRICO

2.1 AGRICULTURA

A agricultura compreende a atividade econômica responsável pela produção de alimentos que ao longo da história da humanidade ocupou as terras férteis de vales de rios e posteriormente, desenvolveu técnicas e procedimentos que tornam os solos mais produtivos, buscando sempre uma maior produtividade. Com o advento da indústria e o fortalecimento das cidades, a agricultura tornou-se um setor dependente das inovações técnicas industriais e estabeleceu-se uma interdependência entre os setores. É a agricultura que gera o alimento consumido por toda a população, seja essa população rural ou urbana.(Lima et al. 50-68)

De forma generalizada costuma-se definir a agricultura como o conjunto de técnicas utilizadas para cultivar plantas com o objetivo de obter alimentos, fibras, energia, matéria-prima para roupas, construções, medicamentos, ferramentas, etc. Sua origem remonta ao período neolítico há mais de 10 mil anos, em algumas regiões pouco numerosas e relativamente pouco extensas do planeta. Embora datem seu início, há tempos bem remotos, convém lembrar que o homem surgiu há mais de 100 mil anos. Foi nesse período histórico que ocorreu a evolução da pedra lascada para a pedra polida, bem como, o surgimento da cerâmica. (MAZOYER ; ROUDART, 2010).

Agricultura é uma atividade econômica fundamental para a produção de alimentos, fibras, energia e materiais diversos, tendo evoluído ao longo do tempo com o desenvolvimento de técnicas que visam aumentar sua produtividade. Com a crescente interdependência entre os setores, a agricultura tornou-se um setor dependente das inovações tecnológicas. Apesar disso, é essencial que essa atividade seja praticada de forma sustentável para evitar danos ao meio ambiente e garantir a segurança alimentar a longo prazo, enquanto ainda proporciona acesso a tecnologias e práticas agrícolas modernas para aumentar a produtividade e a qualidade dos produtos.

2.2 ESTUFA

As casas de vegetação são um instrumento de proteção ambiental para produção de plantas, como hortaliças e flores. Por definição, casas de vegetação são estruturas construídas com diversos materiais, como madeira, concreto, ferro, alumínio, etc, cobertas com materiais transparentes que permitem a passagem da luz solar para crescimento e desenvolvimento das plantas. O uso destas estruturas pode ser de caráter parcial ou pleno, dependendo das características exploradas. Um exemplo característico do uso parcial é a utilização de cobertura da estrutura para obter-se o efeito 'guarda-chuva', muito comum em regiões tropicais. Por outro lado, é possível explorar todo o potencial deste tipo de estrutura, construindo-se uma casa-de-vegetação completa, com todos os controles para a cobertura e para a proteção das plantas em relação a parâmetros meteorológicos adversos, como a precipitação pluviométrica, e com cortinas laterais para geração e aprisionamento do calor (REIS, 2005).

O monitoramento e controle do ambiente dentro de uma estufa desempenham um papel crucial na produção de hortaliças cultivadas, é essencial desenvolver um sistema automatizado que execute essas funções. Automatizar o ambiente de cultivo em estufas é de extrema importância para assegurar que as ações tomadas sejam baseadas em dados precisos e que erros causados por intervenção humana sejam praticamente eliminados. Assim permitindo que o cultivo em estufas seja altamente eficiente e confiável, resultando em melhores colheitas de hortaliças.

Uma estufa pode ser definida como uma estrutura fechada que serve para proteger as plantas de fatores externos, como condições climáticas e poluição. Oferece um desenvolvimento sustentável e eficiente das plantas ao longo do ano. Os fatores básicos que afetam o crescimento das plantas são luz solar, teor de água no solo, temperatura, umidade. (DANITA et al., 2018, tradução nossa).

2.3 FATORES CLIMÁTICOS

O clima é fundamental para o desenvolvimento de plantas, os fatores climáticos como temperatura e luminosidade podem interferir de forma benéfica ou maléfica no desenvolvimento da planta, sendo assim, controlar esses fatores é de

suma importância e o uso do ambiente protegido vem somar a essa busca por melhores resultados (SANTOS; SEABRA JUNIOR; NUNES, 2010).

2.3.1 TEMPERATURA

Existem muitos métodos descritos na literatura para regular a temperatura dentro da estufa, com ventilação, sistema de resfriamento evaporativo, filtro de radiação líquida, etc. No entanto, esses métodos podem consumir muita energia. A ventilação pode reduzir a temperatura trocando o ar entre a estufa e o ambiente. No entanto, em condições de verão extremamente quentes, o ar ambiente muito quente não pode fornecer um efeito de resfriamento adequado (Liu et al. #, tradução nossa).

Cada função vital do vegetal necessita de temperaturas críticas e acima ou abaixo delas não são realizadas ou são impedidas. Cada espécie vegetal, em cada momento crítico de seu ciclo biológico, necessita de uma temperatura ótima para seu desenvolvimento normal (CERMEÑO, 2005, tradução nossa).

2.3.2 UMIDADE DO AR

A umidade relativa do ar no interior de um ambiente protegido é determinada diretamente pela temperatura, numa relação inversa entre ambas. Ela pode variar num período de 24 horas de 30% a 100%, sendo que diminui durante o dia e aumenta durante a noite. Ela está vinculada ao equilíbrio hídrico das plantas, onde um déficit pode alterar a evapotranspiração, alterando a capacidade do sistema radicular de absorver a água que é o nutriente. Dessa forma, o manejo da umidade do ar, também vai depender da cultura visando atender sua fisiologia de crescimento e desenvolvimento. (PURQUERIO; TIVELLI, 2005)

A umidade relativa diurna de 25% a 70% costuma ser baixa o suficiente para inibir a infecção por patógenos. Porém, quando a umidade se eleva acima de 90%, principalmente por longos períodos, a incidência de doenças também pode aumentar (PFEUFER et al., 2016, tradução nossa).

2.3.3 VENTILAÇÃO

A ventilação pode ser considerada como a troca de ar entre o meio externo e interno da estufa. O principal objetivo da ventilação é retirar a sobrecarga de calor de dentro da estufa. A importância de se fazer circular e renovar o ar interno se deve ao

fato de evitar a presença e surgimento de pragas, controlando a umidade do ar na estufa. Para tanto, existem dois tipos de ventilações, a natural e a mecânica. A primeira forma de ventilação é caracterizada por aberturas laterais ou superiores existentes na estufa, o que permite a renovação gasosa interna (AZEVEDO JUNIOR, 2016).

Uma solução prática, rápida e eficaz é um mini-ventilador. Ele ajudará a diminuir a temperatura porque há a renovação do ar no interior do cultivo. Isso é muito importante porque as plantas trocam gases e vapor de água com o meio em que se encontram de forma contínua (PLANTEI, 2017).

2.3.4 LUMINOSIDADE

De acordo com HOPKINS(2006), a luz desempenha um papel crucial na sobrevivência das plantas, pois por meio do processo de fotossíntese, elas convertem moléculas de água e dióxido de carbono em energia química, ao mesmo tempo em que liberam oxigênio, contribuindo para a renovação e purificação do ar. Essas moléculas são posteriormente utilizadas pelas plantas em seu processo de respiração.

A iluminação artificial aplicada na agricultura tem o intuito de fornecer luz que não provém do sol, mas sim de lâmpadas. Estas, são lâmpadas LED, basicamente devido à capacidade destes em sintetizar luz em comprimentos de onda específicos aliada a uma elevada eficácia luminosa (lm/W) (Luz et al, 2022).

2.3.5 IRRIGAÇÃO

A água é um dos fatores limitantes da produção agrícola, considerando sua participação nos vários processos metabólicos da planta. Portanto, a água deve ser fornecida às mudas na quantidade necessária e no tempo certo. Excesso de água pode propiciar condições anaeróbicas em torno das raízes, reduzindo a respiração e limitando a fotossíntese e, ainda, favorecendo o aparecimento de doenças foliares e do solo. Por outro lado, o suprimento de água insuficiente provoca perdas excessivas de água por meio da transpiração, conduzindo a enrolamento, amarelecimento e queda de folhas. O ideal é manter um fornecimento de água necessário para evitar esses problemas (Scarpate Filho, 1995; Martins et al., 1999).

O manejo racional da irrigação consiste na aplicação da quantidade necessária de água às plantas no momento correto. Por não adotar um método de controle da irrigação, o produtor rural usualmente irriga em excesso, temendo que a cultura sofra um estresse hídrico, o que pode comprometer a produção. Esse excesso tem como consequência o desperdício de energia e de água, usados em um bombeamento desnecessário (LIMA, J. E. F. W.; FERREIRA, Raquel Scalia Alves; CHRISTOFIDIS, Demetrios, 1999).

2.3.5.1 TIPOS DE IRRIGAÇÃO

2.3.5.1.1 IRRIGAÇÃO LOCALIZADA

No método da irrigação localizada a água é, em geral, aplicada em apenas uma fração do sistema radicular das plantas, empregando-se emissores pontuais (gotejadores), lineares (tubo poroso ou “tripa”) ou superficiais (microaspersores). A proporção da área molhada varia de 20 a 80% da área total, o que pode resultar em economia de água (ANDRADE; BRITO, 2006).

2.3.5.1.2 IRRIGAÇÃO POR ASPERSÃO

Nestes sistemas, a água é distribuída na forma de gotas sobre a cultura e superfície do solo, imitando o efeito da chuva. A formação das gotas é obtida pela passagem da água pressurizada através de orifícios existentes em dispositivos mecânicos chamados aspersores ou sprays (TESTEZLAF, 2011).

2.3.5.1.3 IRRIGAÇÃO DE SUPERFÍCIE

Na irrigação localizada a água é aplicada sobre o solo em uma área restrita, preferencialmente debaixo da área sombreada pela copa da cultura ou perto do caule, buscando umedecer somente o volume de solo explorado pelo sistema radicular da planta. Esses sistemas utilizam pequenas vazões, quando comparados a outros sistemas de irrigação, devido ao emprego de emissores com diâmetros de saída reduzidos submetidos a baixas pressões (TESTEZLAF, 2011).

2.3.5.1.4 IRRIGAÇÃO DE SUBSUPERFÍCIE

Na irrigação de subsuperfície, a aplicação de água é realizada abaixo da superfície do solo, diretamente nas raízes das culturas, aproveitando a ocorrência do fenômeno de ascensão capilar, onde a água se eleva ao longo do perfil do solo por diferença de potencial total (TESTEZLAF, 2011).

2.4 AUTOMAÇÃO

A automação envolve a implantação de sistemas interligados e assistidos por redes de comunicação, compreendendo sistemas supervisórios e interfaces homem,máquina que possam auxiliar os operadores no exercício da supervisão e da análise dos problemas que porventura venham a ocorrer. A vantagem de utilizar sistemas que envolvam diretamente a informatização é a possibilidade da expansão utilizando recursos de fácil acesso; nesse contexto, são de extraordinária importância os centros lógicos programáveis (CLPs), que tornam a automação industrial uma realidade onipresente (CASTRUCCI; MORAES, 2007).

O uso de sistemas automatizados em estufas vem crescendo ano após ano, pois os produtores buscam cada vez mais tecnologias que possam auxiliar no dia a dia, para diminuir a possibilidade de erros comuns no cuidado das plantações, aumento da produção e também para baixar os custos com contratação de mão de obra. Processos de automação garantem um controle total do ambiente, proporcionando maior desenvolvimento da produtividade e qualidade dos produtos com maior eficiência e uso coerente dos recursos.(FERNANDES; PREUSS; SILVA, 2017).

Assim a automação é fundamental para o controle e monitoramento dos fatores internos da estufa, assim permitindo criar um ambiente ideal para um desenvolvimento saudável das hortaliças cultivadas. Assim tendo uma maior produtividade, qualidade e economia de recursos, além da possibilidade do acesso e monitoramento possibilitando futuros estudos.

2.5 IOT

A Internet das Coisas (IoT) é um termo abrangente para uma ampla gama de tecnologias e serviços subjacentes que, por sua vez, fazem parte de um ecossistema mais amplo. A interconexão de dispositivos físicos com possibilidades

de detecção e comunicação (por meio de sensores e atuadores), não é um conceito novo, no entanto, no que cerne à Internet das Coisas, os endpoints físicos estão conectados através de endereços de IP exclusivos pelos quais dados podem ser reunidos e comunicados. Com previamente elucidado, essa comunicação de dados acontece através de software, sendo, por fim, armazenada na nuvem (CARRION E QUARESMA, 2019, p. 55).

A IoT é capaz de interagir sem intervenção humana. Algumas aplicações preliminares de IoT já foram desenvolvidas em saúde, transporte e indústrias automotivas. As tecnologias de IoT estão em seus estágios iniciais; no entanto, muitos novos desenvolvimentos ocorreram na integração de objetos com sensores na Internet. O desenvolvimento da IoT envolve muitas questões, como infraestrutura, comunicações, interfaces, protocolos e padrões (GOKHALE; BHAT; BHAT, 2018, tradução nossa).

2.6 PROTOCOLO DE COMUNICAÇÃO HTTP E API REST

HTTP (Hypertext Transfer Protocol, 'protocolo de transferência de hipertexto') é um protocolo cliente/servidor sem conexão onipresente nas TIC e na web. Como existem inúmeras ferramentas de código aberto que usam HTTP e como cada linguagem de coding tem bibliotecas HTTP, é muito acessível (SEMLE e ARON, 2016, tradução nossa).

O foco do HTTP na IoT gira em torno do REST (Transferência Representativa do Estado, 'transferência de estado representacional'), que é um modelo sem estado onde os clientes podem acessar recursos no servidor através de encomendas. Na maioria dos casos, um recurso é um dispositivo e as informações que ele tal dispositivo contém (SEMLE e ARON, 2016, tradução nossa).

Use HTTP para enviar grandes quantidades de informações, como leituras de temperatura minuto a minuto, minuto a cada hora. Não use HTTP para obter informações vídeo de alta velocidade. HTTP pode operar sob o segundo, mas atualizações de cem milissegundos (100 ms) com HTTP são difíceis. Isso implica um pouco de sobrecarga por mensagem, então enviar mensagens pequenas é ineficiente. E sempre proteja a comunicação com HTTPS. A sobrecarga é mínima (SEMLE e ARON, 2016, tradução nossa).

O ThingSpeak possibilita o uso tanto das APIs REST quanto o MQTT, neste presente trabalho estaremos apenas utilizando a API REST, utilizando das chamadas da API REST como GET, POST, PUT e DELETE, podendo assim criar, atualizar, limpar e excluir um canal.

3. MATERIAIS E MÉTODOS

3.1 MATERIAIS

Tabela 1: Materiais utilizados na montagem do projeto. (continua)			
Especificações	Quantidade	Valor Unitário	Valor Total
Placa ESP32 WROOM-32 WiFi	1	R\$ 71,70	R\$ 71,70
Sensor de Umidade do Solo Higrômetro	1	R\$ 16,00	R\$ 16,00
Sensor de Temperatura Encapsulado a Prova D'água DS18B20	1	R\$ 24,00	R\$ 24,00
Módulo Sensor de Temperatura e Umidade DHT11 com LED	1	R\$ 24,00	R\$ 24,00
Módulo Serial I2C para Display LCD	1	R\$ 15,00	R\$ 15,00
Display LCD 16x2 Backlight Azul	1	R\$ 32,50	R\$ 32,50
Kit Cabo Jumper Macho-Fêmea 10 unidades 20cm	2	R\$ 10,00	R\$ 20,00
Kit Cabo Jumper Macho-Macho 10 Unidades 20cm	1	R\$ 5,00	R\$ 5,00
Fonte Ajustável Step Down Regulador de Tensão LM2596	1	R\$ 20,00	R\$ 20,00
Mini Fonte Chaveada Estabilizada 12V 5a Colmeia	1	R\$ 49,50	R\$ 49,50
Módulo Relé de Estado Sólido SSR de 2 Canais	1	R\$ 38,00	R\$ 38,00
Módulo Relé 2 Canais 5V	1	R\$ 20,00	R\$ 20,00
Chave Gangorra 2 Terminais Preto	1	R\$ 1,50	R\$ 1,50
Chave Botão 2 Terminais Com Trava	1	R\$ 4,50	R\$ 4,50

Tabela 1: Materiais utilizados na montagem do projeto. (conclusão)			
Especificações	Quantidade	Valor Unitário	Valor Total
Botão Push Button Chave Táctil Pequeno 6x6x7,3mm - Com Furo	4	R\$ 0,65	R\$ 2,60
Capas Coloridas Push Button Chave Táctil redonda de 6*6*7,3mm	4	R\$ 0,50	R\$ 2,00
Caixa de Isopor de 40 litros	1	R\$ 35,00	R\$ 35,00
Ventoinha Cooler 12v	2	R\$ 12,00	R\$ 24,00
Bandeja 15 células	2	R\$ 2,99	R\$ 5,98
Sementes Microverdes de Ervilha	1	R\$ 2,00	R\$ 2,00
Tubo de silicone 4mm 1 metro	2	R\$ 2,50	R\$ 5,00
Pote de Plastico de 3,6 Litros	1	R\$ 7,00	R\$ 7,00
Papel de alumínio	1	R\$ 11,00	R\$ 11,00
Bocal Pendente com Rabicho Plástico	2	R\$ 5,00	R\$ 10,00
Lâmpada 5w LED RGB	1	R\$ 20,00	R\$ 20,00
Lampada Incandescenza 100w 110v	1	R\$ 8,00	R\$ 8,00
Total		R\$ 474,28	
Fonte: Elaborada pelo autor.			

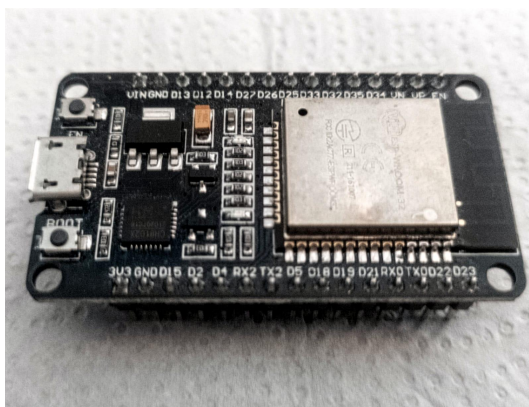
3.1.1 ESP32

O ESP32 foi projetado para dispositivos móveis, eletrônicos vestíveis e aplicativos de Internet das Coisas (IoT). Ele apresenta todas as características de última geração de chips de baixa potência, incluindo *clock gating* de granulação fina, vários modos de energia, e escala de potência dinâmica. Por exemplo, em um cenário de aplicativo de hub de sensor IoT de baixa potência, o ESP32 é ativado periodicamente apenas quando uma condição específica é detectada. O ciclo de trabalho baixo é usado para minimizar a quantidade de energia que o chip gasta. A

saída do amplificador de potência também é ajustável, contribuindo assim para uma compensação ideal entre alcance de comunicação, taxa de dados e consumo de energia. É uma solução altamente integrada para aplicações Wi-Fi, Bluetooth IoT, com cerca de 20 interfaces externas componentes. O ESP32 integra um interruptor de antena, RF, amplificador de potência, amplificador de recepção de baixo ruído, filtros, e módulos de gerenciamento de energia. Como tal, toda a solução ocupa uma placa de circuito impresso (PCB) mínima área. (ESPRESSIF SYSTEMS).

O ESP32 é de baixo custo, possui conectividade Wi-Fi e Bluetooth integrada, tem potência de processamento para lidar com tarefas complexas, é flexível em relação às linguagens de programação e possui diversos pinos digitais e analógicos, o que facilitou o sensoramento e controle de parâmetros dentro da estufa. Essas características me permitiram desenvolver uma solução eficiente, econômica e de fácil monitoramento e controle remoto da estufa. Na Figura 2 podemos ver o microcontrolador usado no projeto.

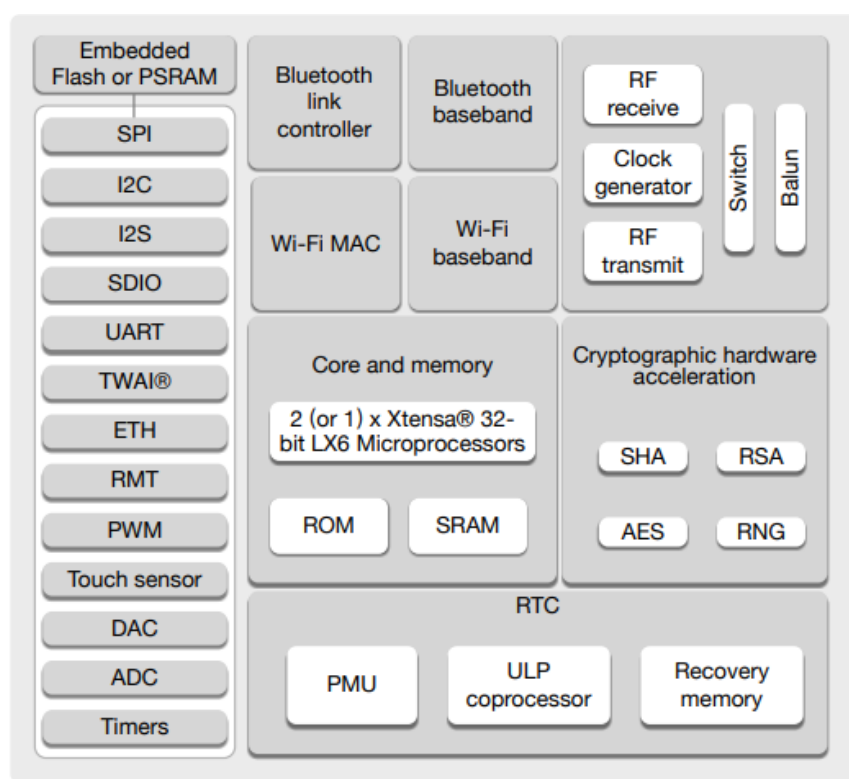
Figura 2 - ESP32-D0WDQ6



Fonte: Elaborada pelo autor.

A estrutura do ESP32 pode ser representada em um diagrama de blocos, que mostra os principais componentes do microcontrolador que permite sua operação. Isso pode ser útil para entender como o ESP32 funciona e como ele pode ser utilizado em projetos. A Figura 3 apresenta em forma de diagrama de blocos a estrutura do ESP32:

Figura 3 - Diagrama de blocos funcionais



Fonte: ESPRESSIF SYSTEMS,2023, p.24.

Na Tabela 2 podemos observar algumas especificações do microcontrolador ESP32.

Tabela 2: Especificações técnicas ESP32-D0WDQ6 (continua)	
Especificações	Detalhes
Chip Base	ESP32-D0WDQ6
Processador	Xtensa 32-Bit LX6 Dual Core
Clock	80 a 240 MHz (Ajustável)
Memória ROM	448 KB
Memória SRAM	520Kb
Memória Flash Externa	32-Bit de acesso e 4Mb
Tensão de Alimentação	4,5 à 12,0 VDC (Pino Vin)
Tensão de Nível Lógico	3,3VDC (não tolera 5V)
Corrente de Consumo	80mA (típica)
Corrente de Consumo	500mA (máxima)

Tabela 2: Especificações técnicas ESP32-D0WDQ6 (conclusão)	
Especificações	Detalhes
Interfaces	Cartão SD, UART(3 canais), SPI (3 canais), SDIO, I2C (2 canais), I2S (2 canais), IR, PWM LED (2 canais) e PWM motor (3 canais)
GPIO	Digital IO (36), ADC 12-Bits (16 canais), DAC 8-Bits (2 canais), Sensor Capacitivo (10 canais); LNA pré-amplificador
WIFI 802.11 b/g/n	2.4 a 2.5 GHz
Segurança WIFI	WPA / WPA2 / WPA2-Enterprise / WPS
Criptografia WiFi	AES / RSA / ECC / SHA
Bluetooth	4.2 BR / EDR e BLE (Bluetooth Low Energy)
Temperatura de Trabalho	-40° à +85° C
Dimensões	27,5 x 51,0 x 7,0 mm
Fonte: Elaborada pelo autor.	

3.1.2 SENSORES

O termo “sensor” é empregado para designar dispositivos sensíveis a alguma forma de energia, que pode ser luminosa, térmica ou cinética. O objetivo é relacionar informações sobre uma grandeza que precisa ser medida, como temperatura, pressão, velocidade, corrente, aceleração, posição, entre outras. (THOMAZINI ; ALBUQUERQUE, 2010)

Os sensores têm um papel fundamental na implementação de um sistema de monitoramento e controle em uma estufa indoor, pois fornecem informações precisas sobre o ambiente interno da estufa. Com o uso de sensores de temperatura, umidade do ar, umidade do solo e luminosidade, é possível garantir que as plantas estejam crescendo em condições ideais, aumentando a eficiência do sistema de cultivo.

Com o monitoramento contínuo, é possível identificar rapidamente variações nos parâmetros e ajustar o sistema para mantê-los dentro de uma faixa ideal. Dessa forma é possível maximizar a produtividade da estufa, reduzindo a perda de plantas e aumentando a qualidade dos produtos colhidos.

3.1.2.1 SENSOR DE TEMPERATURA DS18B20

O termômetro digital DS18B20 fornece 9 bits para medições de temperatura Celsius de 12 bits e tem um função de alarme superior não volátil programável pelo usuário e pontos de gatilho mais baixos. O DS18B20 se comunica em um barramento 1-Wire que, por definição, requer apenas uma linha de dados (e terra) para comunicação com uma central microprocessador. Além disso, o DS18B20 pode derivar energia diretamente da linha de dados (*"parasite power"*), eliminando a necessidade de uma fonte de alimentação externa. Cada DS18B20 possui um código serial exclusivo de 64 bits, que permite que vários DS18B20s funcionem no mesmo 1-Wire. Assim, é simples usar um microprocessador para controlar muitos DS18B20s distribuídos em uma grande área. Os aplicativos que podem se beneficiar desse recurso incluem Controles Ambientais, monitoramento de temperatura, sistemas dentro de edifícios, equipamentos ou máquinas, e sistemas de monitoramento e controle de processos. (Maxim Integrated Products)

O sensor DS18B20 foi escolhido para ser utilizado no projeto devido à sua precisão e facilidade de uso. Ele é capaz de medir a temperatura com grande acurácia em um intervalo amplo de temperaturas, tornando-o ideal para ser utilizado em sistemas de controle de temperatura. Na Figura 4 podemos mostrar o modelo de sensor utilizado no interior da estufa.

Figura 4 - Sensor de temperatura encapsulado a prova d' água DS18B20.



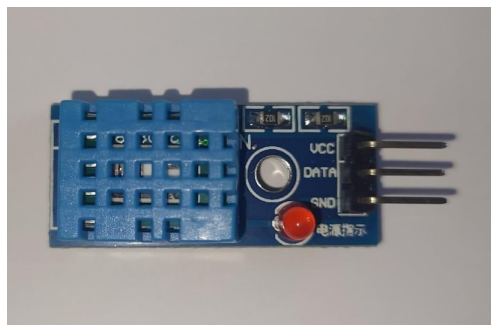
Fonte: Elaborada pelo autor.

Tabela 3: Especificações Técnicas do Sensor de temperatura encapsulado a prova de água DS18B20.	
Especificações	Detalhes
Tensão de alimentação	3.0 - 5.5VDC
Leitura de Temperatura	- 55 a + 125 C°
Tempo de atualização	< 750 ms
Precisão	+ ou - 0,5 C°
Resolução	9 ou 12 bits
Diâmetro	6mm
Comprimento do fio	90cm
Fonte: Elaborada pelo autor.	

3.1.2.2 SENSOR DE TEMPERATURA E UMIDADE DHT11

Conforme a ficha de dados do sensor, este sensor utiliza um termistor para medir a temperatura e um sensor capacitivo para medir a umidade do ambiente. O sensor funciona com uma tensão de alimentação entre 3 e 5,5VDC e possui uma única linha de comunicação serial de dados. O formato dos dados transmitidos consiste em 8 bits de dados de umidade integral, 8 bits de dados de umidade decimal, 8 bits de dados de temperatura integral, 8 bits de dados de temperatura decimal e 8 bits de soma de verificação. Na Figura 5 podemos ver o modelo do sensor DTH11 já soldado em um módulo para facilitar sua conexão.

Figura 5 - Sensor de umidade e temperatura DHT11.



Fonte: Elaborada pelo autor.

O sensor possui alta confiabilidade e estabilidade de longo prazo, dentro da sua faixa de operação, além de uma resposta rápida e baixo consumo de energia. No entanto, é importante observar as condições de operação do sensor, evitando exposição a materiais químicos, luz solar intensa e operação em condições não normais, o que pode acelerar o processo de envelhecimento do sensor. Além disso, a qualidade dos fios de conexão também pode afetar a qualidade e a distância da comunicação, sendo recomendado o uso de fios de blindagem de alta qualidade. Na Tabela 4 podemos observar algumas características técnicas do sensor DHT11.

Tabela 4: Especificações técnicas do Sensor de umidade e temperatura DHT11.	
Especificações	Detalhes
Tensão de alimentação	3.0 - 5.5VDC
Corrente	200uA a 500mA
Faixa de medição de umidade	20 a 90% UR
Faixa de medição de temperatura	0° a 50°C
Precisão de umidade de medição	± 5,0% UR
Precisão de medição de temperatura	± 2.0 °C
Tempo de resposta	< 5s
Dimensões	23mm x 12mm x 5mm
Fonte: Elaborada pelo autor.	

3.1.2.3 SENSOR HIGRÔMETRO FC-28

O sensor de umidade do solo Higrômetro resistivo é um dispositivo utilizado para medir o nível de umidade no solo. Ele é constituído por um conjunto de duas hastes metálicas, geralmente feitas de cobre ou aço inoxidável, que são inseridas no solo. Quando o solo está úmido, a resistência elétrica entre as hastes é menor, e quando está seco, a resistência é maior.

Essa diferença na resistência elétrica é medida pelo sensor e convertida em um sinal elétrico que pode ser interpretado por um microcontrolador ou outro dispositivo eletrônico. Dessa forma, é possível monitorar continuamente o nível de umidade do solo e tomar medidas adequadas para garantir o crescimento saudável das plantas. Na Figura 6 podemos ver o modelo do sensor sensor higrômetro.

Figura 6 - Sensor higrômetro FC-28.



Fonte: Elaborada pelo autor.

O sensor de umidade do solo Higrômetro resistivo é uma solução de baixo custo e fácil de usar, sendo muito utilizado em sistemas de irrigação automatizados e projetos de jardinagem inteligente. No entanto, é importante ressaltar que a precisão da medição pode variar de acordo com as características do solo, como o tipo de solo, a densidade e a temperatura, além de outros fatores ambientais.

Tabela 5: Especificações técnicas do sensor higrômetro FC-28	
Especificações	Detalhes
Tensão de alimentação	3.3V ou 5V
Corrente máxima	35 mA
Sinal da Tensão de saída	0V ~ 4.2V
Fonte: Elaborada pelo autor.	

3.1.2.4 SENSOR DE LUMINOSIDADE LDR

O LDR, também conhecido como resistor dependente de luz, é um componente eletrônico que tem a propriedade de ter sua resistência elétrica alterada conforme a quantidade de luz que incide sobre ele. Dessa forma, quando o LDR é exposto a uma maior intensidade de luz, sua resistência diminui, permitindo a passagem de uma maior corrente elétrica, enquanto que, na ausência de luz, sua resistência aumenta, dificultando a passagem de corrente elétrica. Na Figura 7 podemos ver o modelo do sensor LDR.

Figura 7 - Sensor de luminosidade LDR 5mm.



Fonte: Elaborada pelo autor.

Tabela 6: Especificações técnicas do sensor de luminosidade LDR 5mm.	
Especificações	Detalhes
Tensão Máxima	150VDC
Potência Máxima	100mW
Tensão de Operação	-30°C a 70°C
Resistência no Escuro	1 MΩ
Resistência na Luz	10-20 KΩ
Diâmetro	5mm
Fonte: Elaborada pelo autor.	

O sensor LDR foi muito útil para o monitoramento da luz recebida pelas plantas, o que é essencial para o sucesso do cultivo. Com o sensor LDR, é possível medir com certa precisão a quantidade de luz natural e artificial que as plantas estão recebendo, permitindo um controle mais preciso e eficiente do tempo de iluminação.

Isso é especialmente importante em estufas indoor, onde as plantas podem não receber luz natural, necessitando de luz artificial para um crescimento saudável.

Com o sensor LDR, é possível controlar o tempo de luz artificial de forma automatizada, ajustando o tempo de iluminação para cada tipo de planta, garantindo as condições ideais para o seu desenvolvimento.

3.1.2.4 SENSOR DE NÍVEL DE ÁGUA TIPO BOIA 90° HORIZONTAL

O sensor de nível de água por boia vertical é um dispositivo simples e eficiente para controlar o nível de líquidos em tanques e reservatórios. Composto por uma bóia que sobe ou desce com o nível do líquido, acionando um interruptor mecânico magnético, o sensor pode ser integrado a sistemas de automação e

permitir o acionamento de dispositivos elétricos, como bombas e alarmes. Sua simplicidade, baixo custo, ausência de fonte de energia externa e precisão ajustável tornam esse sensor uma solução versátil para o monitoramento e controle de líquidos em diversas aplicações.

Figura 8 - Sensor de nível de água tipo boia.



Fonte: Elaborada pelo autor.

Tabela 7: Especificações técnicas do sensor de nível de água tipo boia.	
Especificações	Detalhes
Tensão de chaveamento (máx.)	100VDC
Corrente de chaveamento (máx.)	0,5A
Tensão do contato aberto (máx.)	100VDC
Resistência Contato (máx.)	100M Ohms
Temperatura de Operação	-20 a +80°C
Comprimento do Cabo	36 cm
Fonte: Elaborada pelo autor.	

O sensor de nível de água é muito importante para controlar a quantidade de água no reservatório da estufa. Com ele, é possível garantir que as plantas tenham sempre água suficiente para crescer e se desenvolver, alertando o usuário da necessidade de encher o reservatório, evitando o risco de falta de água e, consequentemente, a morte das plantas.

3.1.3 ATUADORES

Os atuadores são dispositivos fundamentais para o controle e operação de sistemas em diversas áreas, tais como automação industrial, robótica e sistemas de controle. Eles convertem sinais elétricos em diferentes formas de energia física, como movimento mecânico, magnetismo, calor, luz, som, entre outras, para produzir uma ação física em um sistema. Os atuadores são projetados para atender a requisitos específicos de aplicação, como tamanho, força, velocidade, precisão e outros, e são usados para converter sinais elétricos em ações físicas precisas e eficientes.

3.1.3.1 LÂMPADA LED

O uso de lâmpadas LED para controle de iluminação em ambientes de cultivo é uma opção eficiente e promissora. As vantagens oferecidas incluem a eficiência energética, redução de calor irradiado, estabilidade da temperatura ambiente e melhor desempenho fotossintético das plantas. Tendo em vista esses aspectos foi definido o uso da lâmpada LED na estufa para cultivo, resultando em um ambiente de cultivo mais adequado para o desenvolvimento das plantas. Na Figura 9 podemos ver o modelo que será utilizado no protótipo.

Figura 9 - Lâmpada led.



Fonte: Elaborada pelo autor.

Tabela 8: Especificações técnicas da lâmpada led.	
Especificações	Detalhes
Tensão Operação	127V
Potência	15W
Fluxo Luminoso	90 lumens
Fonte: Elaborada pelo autor.	

3.1.3.2 LÂMPADA INCANDESCENTE

O uso de lâmpadas incandescentes tem como objetivo manter a temperatura da estufa caso ela caia para um valor abaixo do definido pelo usuário. O calor emitido pela lâmpada consegue estabilizar a temperatura rapidamente, sendo assim selecionada devido seu baixo custo e fácil acessibilidade. Na Figura 10 podemos ver o modelo que será utilizado no protótipo.

Figura 10 - Lâmpada incandescente.



Fonte: Elaborada pelo autor.

Tabela 9: Especificações técnicas da lâmpada incandescente.	
Especificações	Detalhes
Tensão Operação	127V
Potência	15W
Fluxo Luminoso	90 lumens
Fonte: Elaborada pelo autor.	

3.1.3.3 BOMBA DE ÁGUA

O controle da irrigação é fundamental para o desenvolvimento adequado das plantas. A utilização de uma bomba submersa de aquário (Figura 11) foi uma excelente opção para o bombeamento da água até os vasos das plantas. Além da facilidade de instalação e manutenção, a bomba submersa é compacta e silenciosa, e sua utilização permite otimizar o uso de água, garantindo a irrigação adequada das plantas. Assim, a escolha de uma bomba submersa de aquário é uma alternativa eficiente e econômica para o controle de irrigação em pequenas estufas indoor.

Figura 11 - Bomba submersa para aquário.



Fonte: Elaborada pelo autor.

Tabela 10: Especificações técnicas da bomba submersa para aquário.	
Especificações	Detalhes
Tensão de Entrada	110 - 120 V
Potência	2,5 W
Taxa de Fluxo	220 Litros por Hora
Grau de Proteção IP	X8
Dimensões	35x35x40mm
Fonte: Elaborada pelo autor.	

3.1.3.4 VENTOINHAS

As ventoinhas são um componente importante na estufa indoor, ajudando a controlar a temperatura e manter um ambiente adequado para o crescimento saudável das plantas. Através da circulação do ar, as ventoinhas permitem a dissipação de calor, impedindo o acúmulo excessivo em pontos específicos e evitando o superaquecimento do ambiente. Além disso, elas ajudam a manter uma temperatura uniforme em toda a estufa, o que é crucial para o desenvolvimento homogêneo das plantas.

Figura 12 - Cooler fan ventoinha.



Fonte: Elaborada pelo autor.

Tabela 11: Especificações técnicas do cooler fan ventoinha.	
Especificações	Detalhes
Alimentação	12VDC
Temperatura de operação	-20 à 75°C
Dimensões	60 x 60 x 10mm
Fonte: Elaborada pelo autor.	

3.1.3.5 MÓDULO RELÉ

O Módulo Relé é um componente fundamental no controle dos atuadores da estufa indoor, permitindo a automação do acionamento de diferentes dispositivos elétricos, tais como lâmpadas LED, bomba de água e ventoinhas.

Figura 13 - Módulo relé 4 canais 5V.



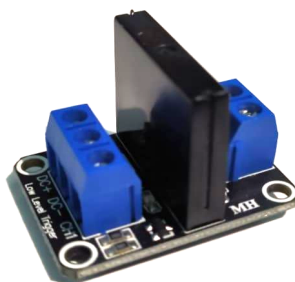
Fonte: Elaborada pelo autor.

Tabela 12: Especificações técnicas do módulo relé 4 canais 5V.	
Especificações	Detalhes
Tensão de Operação	5VDC
Corrente Típica de Operação	15~20mA
Tensão de Saída	(30 VDC a 10A) ou (250VAC a 10A)
Tempo de Resposta	5~10ms
Dimensões	8 x 6 x 2cm
Fonte: Elaborada pelo autor.	

3.1.3.6 MÓDULO RELÉ DE ESTADO SÓLIDO

O uso de um módulo relé de estado sólido oferece diversas vantagens no controle dos atuadores em uma estufa indoor. Ao contrário dos relés eletromecânicos tradicionais, os módulos relé de estado sólido não possuem componentes móveis, o que significa que não há desgaste mecânico e a vida útil é significativamente maior. Além disso, eles operam de forma mais silenciosa e têm um tempo de resposta mais rápido, garantindo uma ação imediata no controle dos atuadores.

Figura 14 - Módulo relé de estado sólido
SSR de 1 canal



Fonte: Elaborada pelo autor.

O módulo relé de estado sólido é uma opção viável para o acionamento da lâmpada incandescente, devido à sua alta confiabilidade e vida útil prolongada. Embora seu custo seja mais elevado em comparação ao módulo relé mecânico, seu desempenho é superior, principalmente em casos de comutação frequente. Dessa forma, considerando que o acionamento da lâmpada incandescente ocorre com mais frequência do que os outros atuadores na estufa, o uso do módulo relé de estado sólido se apresenta como uma vantagem para o controle preciso da temperatura do ambiente.

Tabela 13: Especificações técnicas do módulo relé de estado sólido SSR de 1 canal.	
Especificações	Detalhes
Tensão de Operação	5V
Tensão Máxima de Carga	240VAC
Corrente Máxima de Carga	2A
Dimensões	34 x 25 x 21,5mm
Fonte: Elaborada pelo autor.	

3.1.4 ALIMENTAÇÃO

3.1.4.1 MINI FONTE CHAVEADA

A fonte elétrica é um componente essencial para o funcionamento de dispositivos eletrônicos, convertendo a energia elétrica da rede em uma tensão e

corrente adequadas para o consumo dos equipamentos. No caso do desenvolvimento de uma estufa indoor, a escolha da fonte adequada é crucial para garantir o bom desempenho dos componentes eletrônicos utilizados, como sensores, microcontroladores e atuadores. Na Figura 15 podemos ver o modelo de fonte que será utilizado na alimentação de parte da estufa.

Figura 15 - Mini fonte chaveada estabilizada colmeia.



Fonte: Elaborada pelo autor.

Optou-se pelo uso da Mini Fonte Chaveada, que permite a transformação de 127V da rede elétrica em 12V AC/DC. Essa fonte será utilizada para alimentar a ESP32, responsável pelo controle dos dispositivos da estufa, além de ventoinhas e alguns sensores de monitoramento. A escolha da Mini Fonte Chaveada é vantajosa devido ao seu tamanho compacto, alta eficiência energética e estabilidade na saída de tensão. Segue na tabela algumas especificações técnicas da fonte.

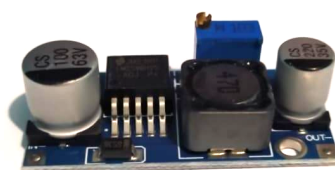
Tabela 14: Especificações da mini fonte chaveada estabilizada colmeia.	
Especificações	Detalhes
Potência	60W
Amperagem	5A
Tensão de Entrada	110-220v
Frequência	50/60Hz
Tensão de Saída	AC/DC 12v
Dimensões	8,5cm x 5,7cm x 3,2 cm
Fonte: Elaborada pelo autor.	

3.1.4.2 REGULADOR DE TENSÃO LM2596

O módulo regulador de tensão LM2596 trabalha como um conversor DC DC no modo Step Down, sendo capaz de reduzir uma carga de até 3A com ótima eficiência. A tensão de saída pode ser ajustada entre 1,5 a 35v, tendo como entrada 3,2 a 40v (MAKER HERO).

Na Figura 15 podemos observar o modelo do regulador de tensão utilizado, seguido da sua tabela (tabela 16) com algumas especificações técnicas.

Figura 16 - Regulador de tensão LM2596.



Fonte: Elaborada pelo autor.

Tabela 15: Especificações técnicas do regulador de tensão LM2596.	
Especificações	Detalhes
Tensão de Entrada	7V ~ 35V
Tensão de Saída	1.25V ~ 30V
Corrente de Saída	3A (máx.)
Potência de Saída	15W
Eficiência da Transferência	92% (máx.)
Temperatura de Trabalho	-40°C ~ +85°C
Dimensões	4.7 cm x 2.3 cm x 1.4 cm
Fonte: Elaborada pelo autor.	

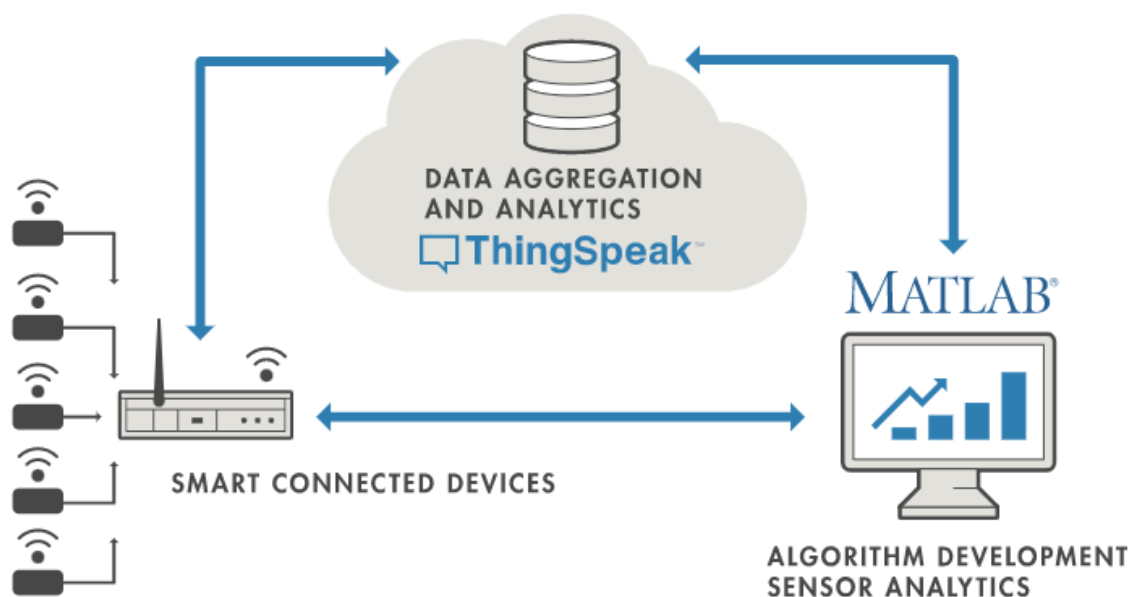
3.1.5 PLATAFORMA THINGSPEAK

O ThingSpeak é um serviço de plataforma de análise de IoT que permite agregar, visualizar e analisar fluxos de dados ao vivo na nuvem. Você pode enviar dados para ThingSpeak™ de seus dispositivos, criar visualizações instantâneas de dados ao vivo e enviar alertas usando serviços da web como Twitter® e Twilio®.

Com a análise MATLAB ® dentro do ThingSpeak, você pode escrever e executar código MATLAB para realizar pré-processamento, visualizações e análises. O ThingSpeak permite que engenheiros e cientistas criem protótipos e construam sistemas IoT sem configurar servidores ou desenvolver software web. (“ThingSpeak Documentation”)

A Figura 17 apresenta o diagrama de comunicação de dispositivos IoT com o ThingSpeak.

Figura 17 - Diagrama do sistema IOT.



Fonte: ThingSpeak (2023).

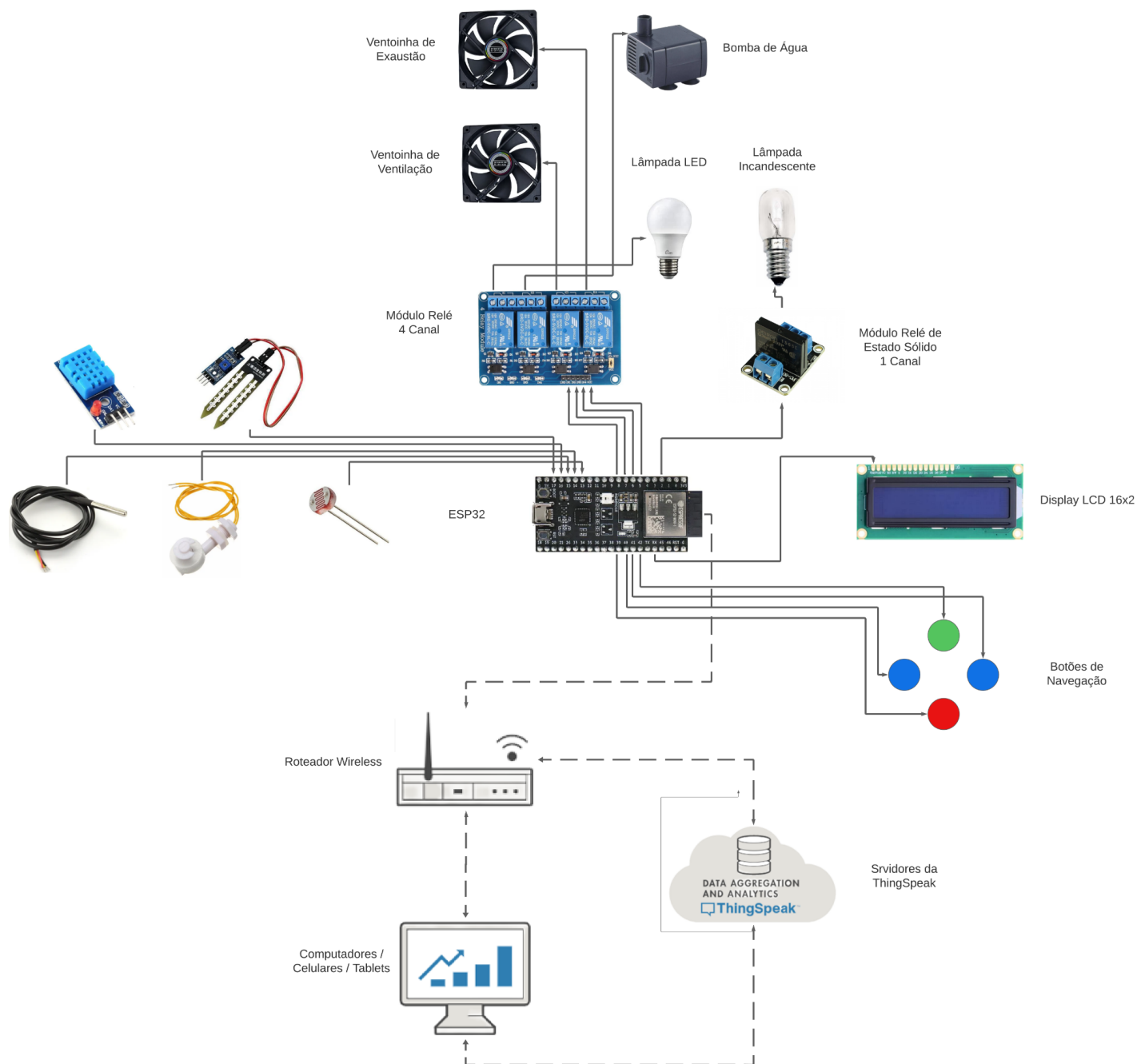
Foi utilizada a plataforma Thingspeak como meio de receber, armazenar e apresentar os dados coletados pelos sensores. Utilizarei a ESP32 como microcontrolador, que será responsável pela comunicação com os sensores e envio dos dados para a plataforma. Através do uso do protocolo HTTP E API REST, os dados serão enviados em tempo real para a plataforma, permitindo o monitoramento da umidade e temperatura do solo em tempo real, a partir de qualquer lugar com acesso à internet. A combinação da plataforma Thingspeak e a ESP32 permitirá criar um sistema de monitoramento eficiente e confiável, utilizando tecnologia IoT para melhorar a produção de plantas.

3.2 MÉTODOS

3.2.1 DIAGRAMA DE FUNCIONAMENTO DO PROJETO

A composição do sistema de monitoramento e controle da estufa proposto neste TCC está representada na Figura 18. Os sensores irão capturar os dados internos da estufa como sua temperatura, umidade do solo e ar, luminosidade e nível de água do reservatório. Esses dados serão transmitidos a ESP32 que vai processar esses dados e transformar em informação útil. Com essas informações podemos saber o status interno da estufa, assim dependendo das condições vamos controlar os atuadores utilizando módulos relés que serão comutados através da ESP32 assim acionando ou desligando as lâmpadas e motores.

Figura 18 - Representação funcionamento do projeto.



Fonte: Elaborada pelo autor.

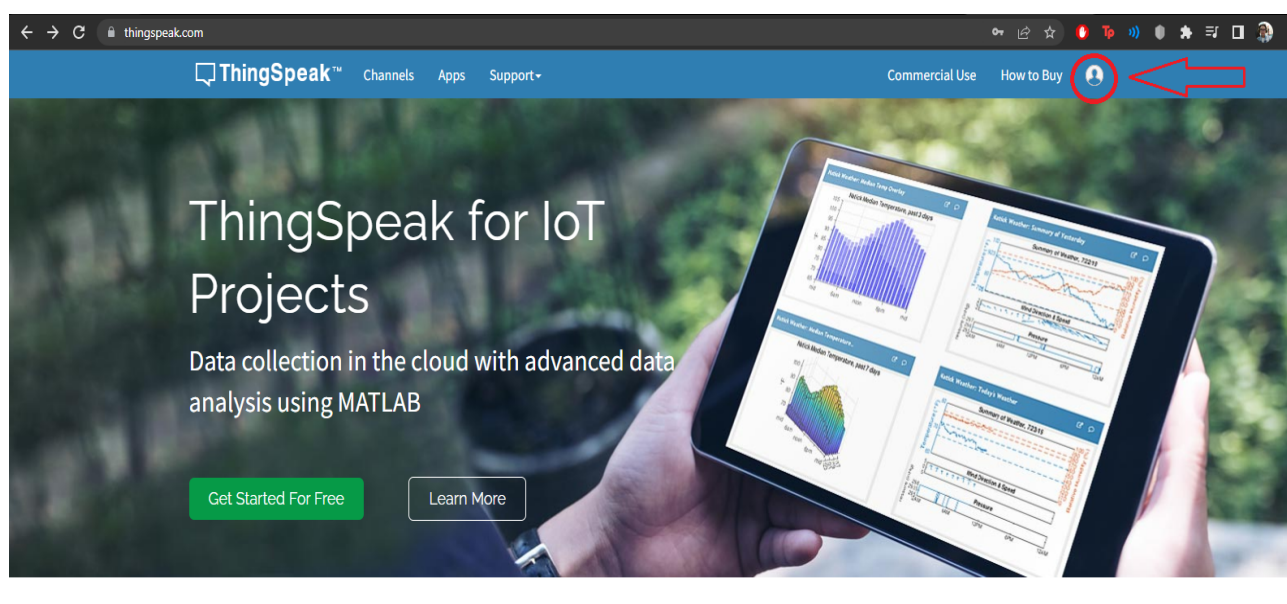
Através do display e botões é possível visualizar a temperatura, umidade do ar e umidade do solo pelo display LCD, também temos o controle de set da temperaturas desejadas na parte interna da estufa, navegando pelo menu do display sendo as configurações de ajustar temperatura, ajustar umidade do ar e ajustar umidade do solo.

Em paralelo a esse sistema que funciona offline temos o sistema online que depende da conexão WI-FI que vai transmitir as informações processadas para a plataforma ThingSpeak, através dessa plataforma vamos monitorar a estufa remotamente por um *dashboard* e poder baixar os dados que foram coletados durante um determinado período de tempo possibilitando futuros estudos do crescimento das plantas variando parâmetros da condição interna.

3.2.2 CONFIGURAÇÃO THINGSPEAK

Para configurar a plataforma ThingSpeak começa acessando o site da plataforma (<https://thingspeak.com/>), clicando no ícone *sign in* vamos a página de acesso e cadastro na plataforma (lado superior direito da tela), conforme mostra a Figura 19.

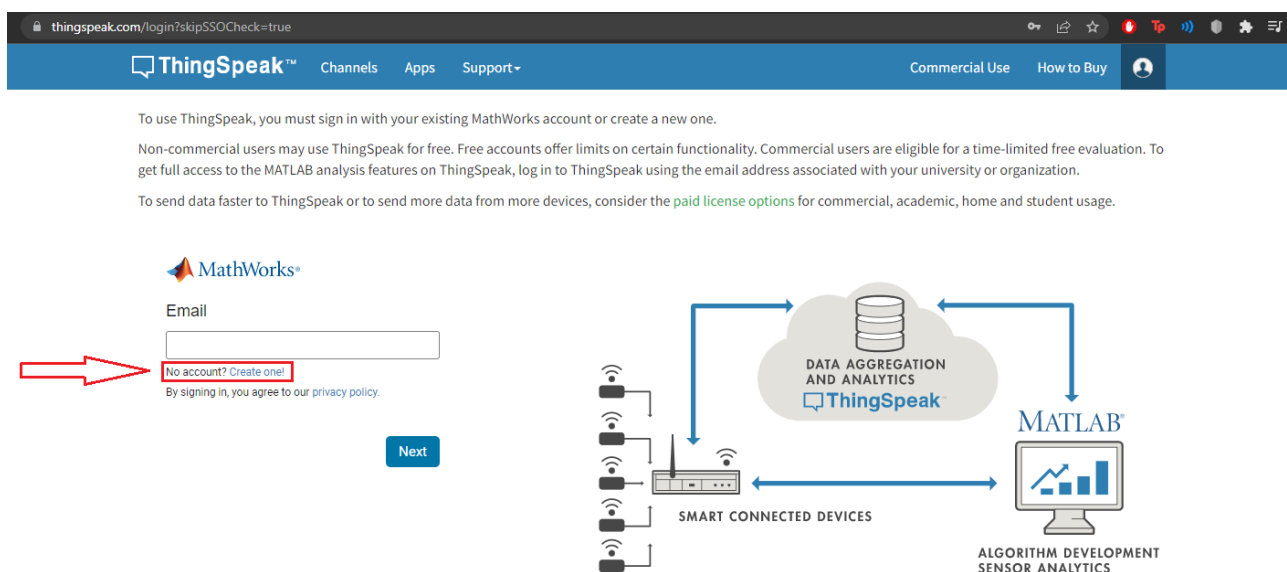
Figura 19 - Site ThingSpeak.



Fonte: ThingSpeak (2023).

Na página de acesso à sua conta é possível criar uma nova conta clicando em “*Create one!*”, que fica abaixo da caixa onde se deve digitar o email. Na Figura 20 podemos ver destacado a área para criar uma nova conta.

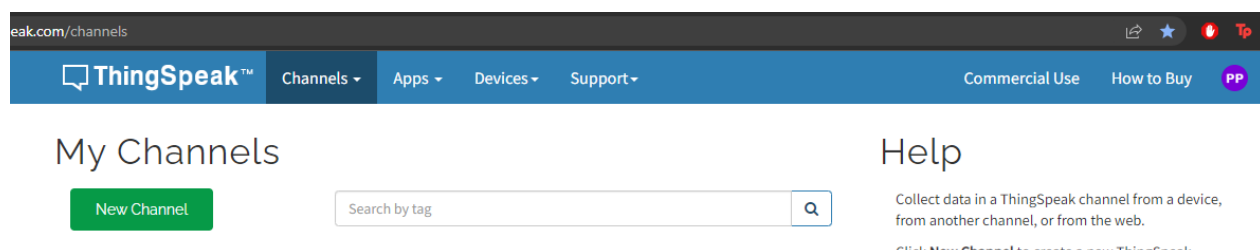
Figura 20 - Site de acesso ao ThingSpeak.



Fonte: ThingSpeak (2023).

Após criar e autenticar seu cadastro, criamos um canal no ThingSpeak para receber os dados.

Figura 21 - Site ThingSpeak.



Fonte: ThingSpeak (2023).

Podemos visualizar através da Figura 22 os canais que foram criados para cada variável que será monitorada.

Figura 22 - Site de acesso ao ThingSpeak.

Channel Settings

Percentage complete 50%

Channel ID 2024648

Name Estufa - Pedro Palheta Neto (DS18B20/DHT11/Higrôme

Description

Field 1 Temperatura DS18B20 ☒

Field 2 Temperature DTH11 ☒

Field 3 Umidade DTH11 ☒

Field 4 Higrômetro ☒

Field 5 Estado Nível Bomba ☒

Fonte: ThingSpeak (2023).

Depois de criar o canal, é possível controlar suas configurações, como definir se ele será público (permitindo que qualquer pessoa com o link acesse os dados) ou privado (restringindo o acesso apenas ao proprietário do canal). Na página, você também pode configurar novos campos, exportar e importar dados. No entanto, o aspecto mais importante é o acesso à chave de escrita do canal, que funciona como uma senha para a comunicação entre o Arduino e o ThingSpeak. Essa chave está presente no código do software e deve ser mantida em segredo para garantir a segurança das informações. Na Figura 23, é mostrado onde você pode encontrar essa chave.

Figura 23 - Site ThingSpeak.

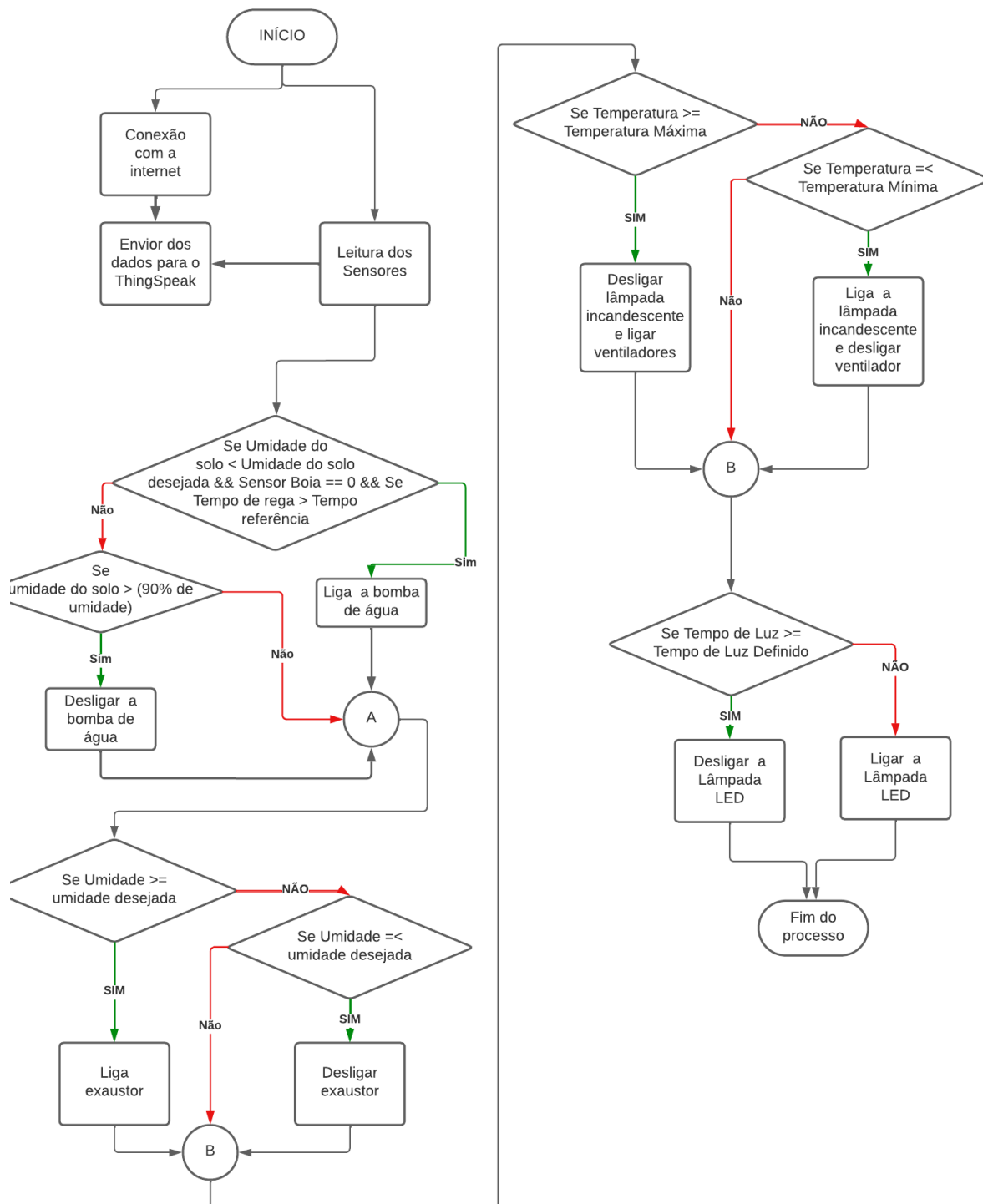
The screenshot shows the 'API Keys' tab in the ThingSpeak interface. At the top, there are navigation links: 'Private View', 'Public View', 'Channel Settings', 'Sharing', and 'API Keys'. Below these, the 'Write API Key' section is visible, featuring a red arrow pointing to a 'Key' field containing 'JQU8MVUCIOEQ3700' and a 'Generate New Write API Key' button. The 'Read API Keys' section below it also has a red arrow pointing to a 'Key' field containing 'JBTY5QWD09FQYSNI'. This section includes a 'Note' text area and two buttons: 'Save Note' and 'Delete API Key'.

Fonte: ThingSpeak (2023).

Após definir a chave de escrita no código de preparação do ThingSpeak, conforme mostrado na Figura 23, uma função chamada "*updateThingSpeak*" foi criada para enviar as informações usando o método POST. A função começa com uma verificação de conexão. Se o Arduino conseguir se conectar ao canal especificado e enviar as informações corretamente, o contador de falhas é reiniciado. No entanto, se a conexão falhar, o contador "*failedCounter*" é incrementado, e o *Shield Ethernet* é reiniciado após três tentativas sem sucesso. Isso garante que o Arduino continue tentando estabelecer a conexão corretamente.

3.2.3 FLUXOGRAMA DE FUNCIONAMENTO DO SISTEMA

Figura 24 - Fluxograma de funcionamento do sistema.



Fonte: Elaborada pelo autor.

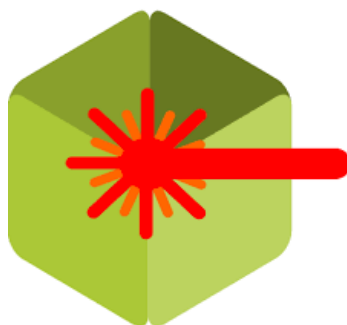
3.3.1 DESENVOLVIMENTO DO CASE EM MDF

No desenvolvimento do case foi cogitado adotar a abordagem de um desenvolvimento de uma caixa totalmente desenhada através do AutoCAD para depois fazer o corte através de uma cortadora a laser mas com uma breve pesquisa foi identificado um gerador de caixas para corte laser desenvolvido em Python, isso veio a facilitar e muito pois seria deveras trabalhoso o desenvolvimento de uma case do zero.

3.3.2 BOXES.PY

Boxes.py é um gerador de caixas para corte a laser, foi essencial para economizar tempo e trabalho já que levaria um tempo considerável para desenhar cada encaixe da caixa. Assim, ao invés de dedicar tempo a criar a estrutura da caixa, foi focado esse tempo em medir e analisar onde fica melhor disposto os furos para encaixe do botão, display e passagem de cabos de alimentação e sensoramento. O site para acesso ao Boxes.py é: “<https://festi.info/boxes.py/?language=en>”.

Figura 25 - Logo do Boxes py

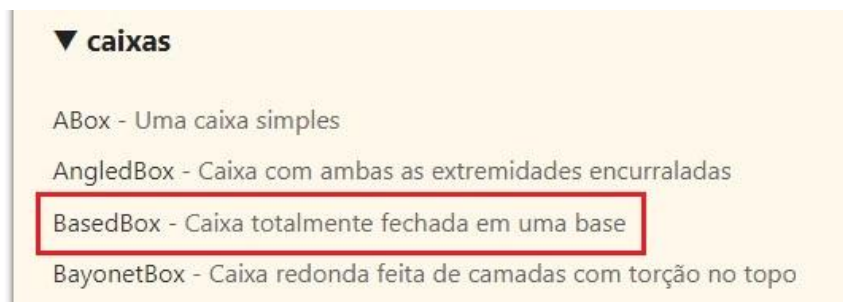


Fonte: boxes.py

3.3.2.1 DESENVOLVIMENTO DA CAIXA ATRAVES DO BOXES.PY

Ao acessar o site teremos uma variedade de modelos de caixa, a de nosso interesse encontrasse na opção “CAIXAS” e logo em seguida clicar na opção “BasedBox” que são caixas totalmente fechadas com um base, na Figura 26 podemos observar algumas opções e a selecionada para este projeto.

Figura 26 - Logo do Boxes py



Fonte: boxes.py

A página será atualizada e agora teremos que configurar algumas preferências sobre a caixa, maior parte foi deixada por padrão do site as configurações modificadas foram em relação à dimensão, que foi de 2000mm x 1400mm x 1200mm. Essas medidas são referentes a parte interna da caixa.

O MDF utilizado tem espessura de 3 milímetros, isso é de suma importância pois se essa configuração não for posta corretamente no site, os furos e abas não iriam se encaixar corretamente. O formato de saída foi em ".dxf", pois é um dos formatos de arquivo suportado pelo AutoCAD.

Após esse procedimento basta clicar em "Download" e um arquivo será gerado e baixado para seu aparelho. Na Figura 27 podemos observar destacado em vermelho as configurações descritas neste pequeno tutorial.

Figura 27 - Configuração para gerar case em MDF.

► **Configurações para articulações dos dedos**

▼ **Configurações da BasedBox**

x	2000.0	largura interna em mm (a menos que fora selecionado)
y	1400.0	profundidade interna em mm (a menos que fora selecionado)
h	1200.0	altura interna em mm (a menos que fora selecionado)

fora ☐ trate os tamanhos como medidas externas ⓘ

▼ **Configurações padrão**

grossura	3.0	espessura do material (em mm) ⓘ
formatar	dxf	formato do arquivo resultante ⓘ
guias	0.0	largura das abas que seguram as peças no lugar (em mm)(não suportadas em todos os lugares) ⓘ
Código QR	☐	Adicione um QR Code com link ou linha de comando à saída gerada
depurar	☐	imprimir caixas envoltentes para algumas estruturas ⓘ
rótulos	☒	rotule as peças (quando disponível)
referência	100	imprimir retângulo de referência com comprimento determinado (em mm) (zero para desativar) ⓘ
cantos_internos	laço	estilo para cantos internos ⓘ
queimar	0.1	correção de queimadura (em mm)(valores maiores para ajuste mais justo) ⓘ

Gerar Salvar no URL

Ativar o Window ⓘ

Fonte: boxes.py

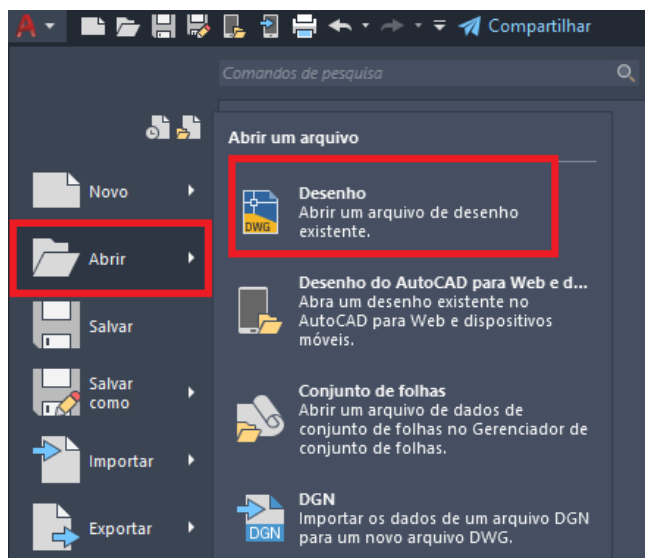
3.3.3 AUTOCAD

O AutoCAD é um software para desenhos técnicos e projetos, desenvolvido pela Autodesk, com ele é possível criar desenhos em duas ou três dimensões. O acesso a esse programa foi através da licença gratuita universitária.

3.3.3.1 MODIFICAÇÕES FEITAS NO AUTOCAD

O Boxes.py adiantou grande parte do desenvolvimento do case fazendo a estrutura da caixa mas ainda foi necessário utilizar uma ferramenta de desenho em 2D para fazer alguns furos para fixação de componentes e passagem de cabos. Na Figura 28 podemos observar o procedimento de como abrir um desenho feito fora do autocad, selecionar “Abrir” e clicar em “Desenho”.

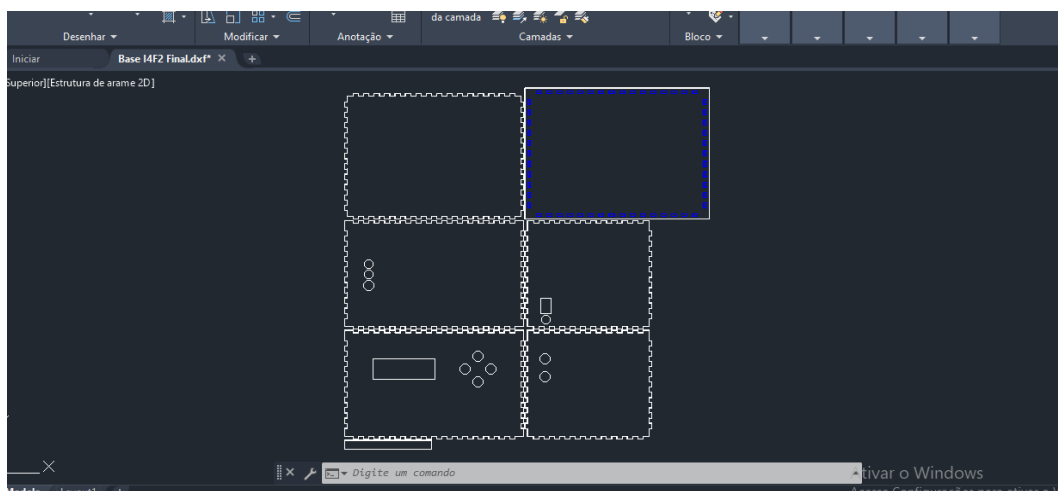
Figura 28 - Abrindo um desenho no AutoCAD.



Fonte: Elaborada pelo autor.

Após realizar esse procedimento o desenho vai ser importado para o autocad, estando disposto para realizar as modificações necessárias para encaixe de botões, Leds, LCD e passagem da fiação, na Figura 29 podemos observar as modificações feitas no desenho original.

Figura 29 - Desenho da caixa já modificada.



Fonte: Elaborada pelo autor.

3.3.4 RDWORKSV8

Nesta etapa, vamos utilizar o RDWorksV8, software que transforma desenhos vetoriais ou imagens bitmap em comandos para a cortadora laser. Assim gerando um trajeto preciso que a máquina seguirá, com base no arquivo enviado.

Figura 30 - Ícone do RDWorksV8

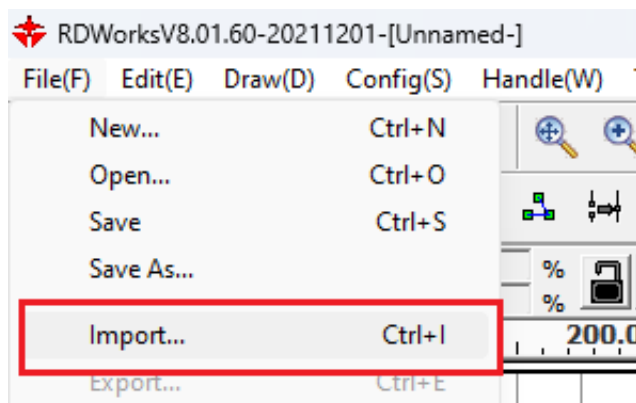


Fonte: Elaborada pelo autor.

3.3.4.1 CONFIGURAÇÕES RDWorksV8

Assim que inicializar o RDWorks iremos acessar a opção “File(F)” e logo em seguida clicar na opção “Import...” assim abrindo o painel de navegação de arquivos do Windows. Vamos na pasta onde se encontra o desenho no formato .dxf e vamos abrir. Na Figura 31 podemos ver onde fica a opção de importação de desenhos para o RDWorks.

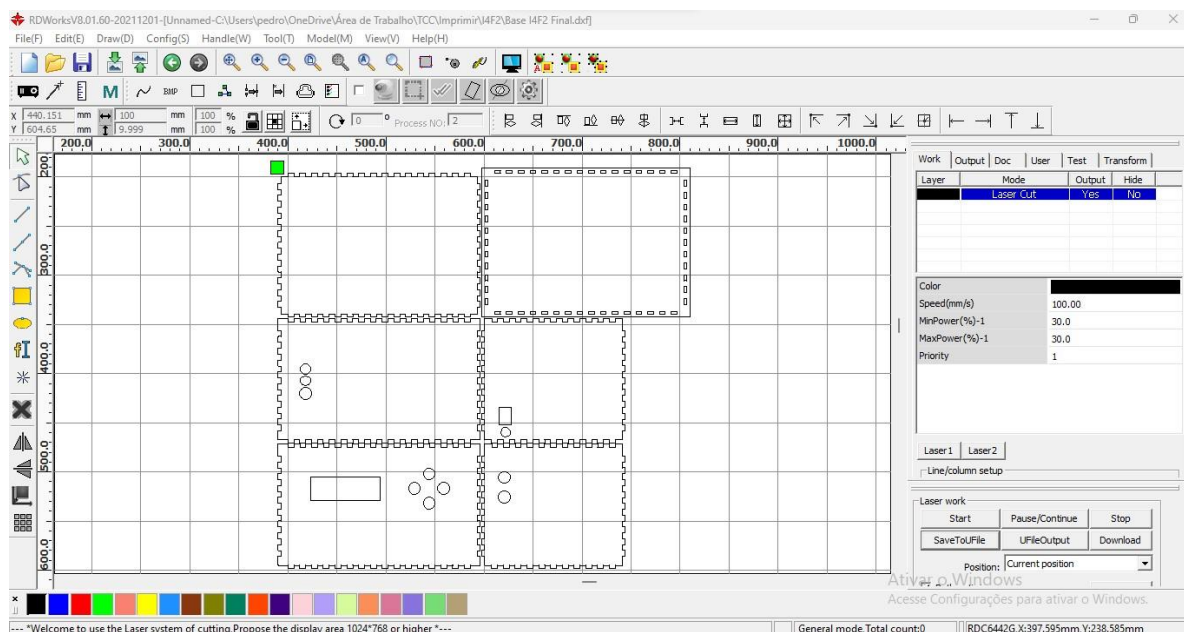
Figura 31 - Opção de importação no RDWorks.



Fonte: Elaborada pelo autor.

Logo que esse procedimento for executado, o desenho em .dxf vai aparecer na área de impressão do software, podemos definir de qual ponta o corte vai ser iniciado entre outras configurações, mas para facilitar a explicação utilizamos todas as configurações padrões do programa para esse corte. Na Figura 32 podemos ver como fica o desenho da caixa modificado no AutoCAD.

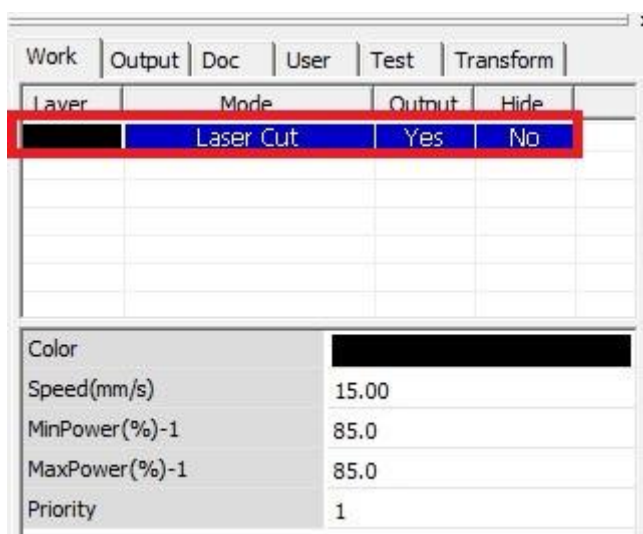
Figura 32 - Desenho da caixa em formato na área de impressão do RDWorks.



Fonte: Elaborada pelo autor.

Após abrir o desenho vamos configurar o laser, podemos ter vários tipos de corte, com diferentes “Layer” também conhecido como camadas de cortes, mas em nosso caso teremos apenas um que vai servir para fazer um furo passante no MDF. Para entrar nessas configurações de potência damos um duplo clique na camada que desejamos modificar, na Figura 33 está destacado de vermelho a área.

Figura 33 - Área de configuração das camadas de cortes.



Fonte: Elaborada pelo autor.

Para executar um corte no MDF de 3mm foram utilizadas as seguintes configurações, uma velocidade de 15 mm/s, 85% de potência do laser, e posto em

modo de corte. Tais configurações constam na Figura 34 nas áreas circuladas de vermelho.

Figura 34 - Configuração da camada de corte utilizada.

Layer Parameter

Parameter library

Layer: [Redacted]

Is Output: Yes

Speed(mm/s): 15

Repeat num: 1

Processing Mode: Cut

If Blowing: Yes

Min Power(%) Max Power(%)

1: 85 85

Fonte: Elaborada pelo autor.

4. ANÁLISE DOS RESULTADOS

Análise dos resultados obtidos do projeto aplicado em testes reais.

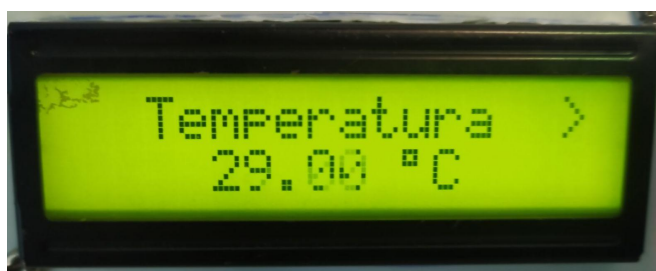
4.1 INTERFACE HOMEM-MÁQUINA

Uma IHM (Interface Homem Máquina) é a tradução do termo HMI (Human Machine Interface) e como o próprio nome diz é uma interface gráfica que permite o homem interagir com a máquina. As interfaces IHM de mercado variam bastante, podendo ser aplicadas desde no controle de plantas nucleares até na tela de um Iphone. (SILVEIRA, 2016).

4.2.1 TELAS DE NAVEGAÇÃO

Na Figura 35 podemos ver a tela do LCD com a informação da temperatura interna da estufa.

Figura 35 - Display LCD mostrando a temperatura.



Fonte: Elaborada pelo autor.

Caso o botão “Enter” seja pressionado, iremos acessar a tela de ajuste da temperatura, nesta parte podemos configurar a temperatura que desejamos que estufa fique, assim o sistema vai se ajustar para atingir esse parâmetros.

Na Figura 36 podemos ver a tela de configuração da temperatura. Para fazer o ajuste, utilizaremos os botões “Esquerda” e “Direita” para diminuir e aumentar a temperatura. Depois de configurar a temperatura desejada, acionamos o botão “Voltar” para a tela de temperatura.

Figura 36 - Display LCD mostrando o ajuste da temperatura.



Fonte: Elaborada pelo autor.

Podemos navegar por outros parâmetros utilizando os botões “Esquerda” e “Direita”. Na Figura 37 está mostrando a tela do LCD com a informação da umidade do solo.

Figura 37 - Display LCD mostrando a umidade do solo.



Fonte: Elaborada pelo autor.

Na Figura 38 podemos ver a tela de configuração da umidade. Para fazer os ajustes da umidade desejada é o mesmo procedimento para configurar a temperatura interna da estufa. Depois de configurar a umidade, acionamos o botão “Voltar” para a tela de Umidade Solo.

Figura 38 - Display LCD mostrando o ajuste da umidade do solo.



Fonte: Elaborada pelo autor.

O último parâmetro que podemos monitorar é a umidade do ar. Como podemos observar na Figura 39.

Figura 39 - Display LCD mostrando umidade do ar.



Fonte: Elaborada pelo autor.

Quando o botão “Enter” é acionado na tela de umidade do ar vamos acessar a tela de ajuste da umidade do ar. Para fazer o ajuste, utilizaremos os botões “Esquerda” e “Direita” para diminuir e aumentar a umidade do ar desejada.

Figura 40 - Display LCD mostrando o ajuste da umidade do ar.



Fonte: Elaborada pelo autor.

4.3 IMPLEMENTAÇÃO DO PROTÓTIPO

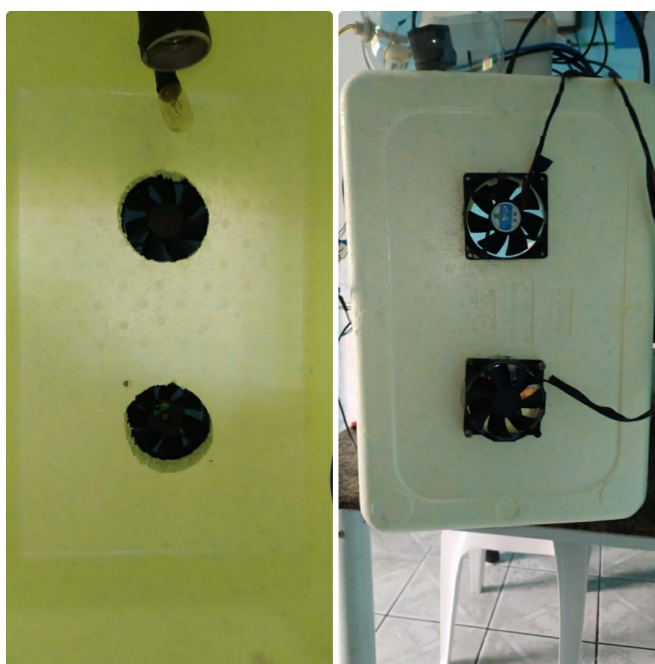
A implementação do protótipo foi feita utilizando em sua construção uma caixa de isopor como estrutura base para a estufa. A utilização de uma caixa de isopor apresentou vantagens significativas, já que o isopor é um material leve, viável economicamente e isolante térmico, o que ajuda na estabilização térmica dentro da estufa.

A caixa foi posta de lado e em pé de modo que tivemos um aproveitamento melhor da área para cultivo de hortaliças de tamanho pequeno e médio e também facilitando a implementação dos atuadores. As lâmpadas de led e incandescente estão acopladas na parte superior da caixa assim simulando uma iluminação natural, a lâmpada incandescente proporciona um meio eficaz de aquecimento, enquanto a lâmpada LED emite luz na faixa espectral necessária para a fotossíntese. Foram feitos dois furos, um para cada lâmpada, passamos os cabos pelos furos e depois

conectamos os bocais em cada par de fio, logo em seguida acoplamos as lâmpadas em seus bocais.

Para o encaixe das ventoinhas de exaustão e ventilação foi feito um furo na parte traseira da caixa de isopor utilizando uma faca de cozinha de serra. O diâmetro do furo circular foi feito utilizando como base o diâmetro das hélices, já que é a área de ventilação da ventoinhas, logo depois presas utilizando parafusos para fixação no isopor, já que cada ventoinha possui um suporte lateral com esse intuito.

Figura 41 - Caixa de isopor com duas ventoinhas acopladas em seus devidos lugares.



Fonte: Elaborada pelo autor.

Os sensores foram acoplados na parede da caixa de modo que ficassem na metade da altura da caixa de isopor, foram presos com abraçadeira de nylon, os cabos estão passando por pequenos furos feitos na lateral da caixa.

Figura 42 - Sensores (DTH11, DS18B20, Higrômetro) acoplados na parte interna da caixa de isopor.



Fonte: Elaborada pelo autor.

Foi projetado um case em MDF para a parte elétrica, assim ficando isolada da umidade ou partículas que poderiam causar danos ao circuito. Foram feitos alguns furos para a passagem dos cabos dos sensores e de alimentação dos atuadores. O display foi fixado através de parafusos e porcas para uma boa fixação, já os botões foram encaixados em seus furos e presos com *jet melt*.

Figura 43 - Case de MDF com o sistema de controle e acionamento em sua parte interna.



Fonte: Elaborada pelo autor.

4.4 IMPLEMENTAÇÃO DO CÓDIGO

Neste tópico é explorado o processo de desenvolvimento e implementação do código para operar a estufa. O código foi escrito e compilado utilizando a Arduino IDE e foi dividido em funções que vamos explorar nos próximos tópicos. A divisão em várias funções foi essencial para garantir a clareza, organização e modularidade durante a elaboração e futuras implementações.

4.4.1 DECLARAÇÃO DAS BIBLIOTECAS

No início do desenvolvimento do código é necessário incluir as bibliotecas necessárias. As bibliotecas foram adicionadas conforme o código e hardware foram sendo implementados. Na Figura 44 podemos visualizar as bibliotecas utilizadas.

Figura 44 - Bibliotecas utilizadas no código.

```
// --- Bibliotecas ---
#include <WiFi.h>           // Recursos de WiFi.
#include <OneWire.h>        // Protocolo de comunicação OneWire.
#include <DallasTemperature.h> // Comunicação com sensor DS18B20.
#include <ThingSpeak.h>     // Recursos e comunicação com nuvem ThingSpeak.
#include "DHT.h"           // Biblioteca Sensor DHT.
#include <Wire.h>           // Gerenciar a comunicação entre os dispositivos através do protocolo I2C.
#include <LiquidCrystal_I2C.h> // Permite controlar displays I2C.
#include <PubSubClient.h>   // Biblioteca para cliente para mensagens MQTT.
```

Fonte: Elaborada pelo autor.

<WiFi.h> - Essa biblioteca é utilizada para fornecer recursos de conectividade WI-FI ao código do projeto, permitindo que a ESP3 se comunique através de redes sem fio, assim possibilitando o envio e recebimento de dados. Além de recursos de segurança para proteger as comunicações.

<OneWire.h> - Biblioteca que facilita a comunicação com dispositivos que possuem o protocolo OneWire. Com um fio de comunicação é possível se conectar a uma diversidade de dispositivos, como sensores de temperatura, memórias EEPROM e outros periféricos.

<DallasTemperature.h> - A biblioteca Dallas Temperature proporciona a comunicação e o controle de sensores de temperatura baseados na família de dispositivos da Dallas Semiconductor. Podemos fazer calibração, ajuste de resolução e manipulação de múltiplos sensores permitindo um controle de temperatura avançado e preciso.

<ThingSpeak.h> - Biblioteca fundamental para a integração da ESP32 com a plataforma ThingSpeak. Permite o envio de dados coletados pela ESP32 para a nuvem através da rede WI-FI por meio da API do ThingSpeak. Através desta biblioteca foi possível monitorar e registrar informações em uma plataforma centralizada, além de possibilitar a criação de gráficos, análises e notificações com base nos dados coletados.

<DHT.h> - Através dessa biblioteca é possível capturar e monitorar os dados de temperatura e umidade dos sensores baseados na família DHT. Assim sendo possível ler os dados de temperatura e umidade de forma fácil e acessível.

<Wire.h> - Esta biblioteca desempenha a função de permitir comunicação serial I2C. Esse tipo de ferramenta é fundamental para estabelecer a comunicação com alguns sensores, módulos e outros microcontroladores por meio deste protocolo. A biblioteca oferece funções para enviar e receber dados em diferentes formatos, como bytes individuais ou sequências de bytes, além de suportar recursos como a configuração de endereços de dispositivos e a detecção automática de dispositivos conectados.

<LiquidCrystal_I2C.h> - Usando esta biblioteca é possível utilizar o display LCD de um modo mais intuitivo com linhas de comandos simplificadas que possibilitam definir as dimensões e o endereço do LCD, bem como controlar aspectos como brilho, contraste e posicionamento do texto. Podendo assim exibir textos, números e caracteres especiais de forma clara e legível.

<PubSubClient.h> - É através dessa biblioteca que é permitido a comunicação com o protocolo MQTT. Assim possibilitando que a ESP32 possa trocar dados com um servidor MQTT ou outros dispositivos conectados. Os dados são enviados e recebidos através de tópicos específicos.

4.4.2 MAPEAMENTO DO HARDWARE

Nesta parte do código foi definido as variáveis que serão associadas ao GPIO da ESP32, aqui estão incluídos os botões para navegação na IHM, os sensores para monitoramento e atuadores para o controle.

4.4.3 CONSTANTES E OBJETOS

Aqui definimos qual sensor DHT estamos utilizando, no caso desse projeto foi DTH11, e depois configuramos utilizando as funções da sua biblioteca. Também é definido a quantidade de menus de navegação que IHM vai possuir.

O display é configurado com o endereço I2C e também selecionado o tamanho do display.

Ambos os sensores de temperatura e umidade precisam ser configurados utilizando a biblioteca OneWire para informar seus pinos de conexão, assim permitindo a transmissão de dados corretamente.

4.4.4 DEFINIÇÃO DAS VARIÁVEIS E CONSTANTES GLOBAIS

Nesta parte foram definidas as variáveis e constantes globais utilizadas pelo programa. Temos a definição do número inicial para navegação do menu e submenu variáveis utilizadas para definir qual tela estará sendo exibida ao usuário.

Como não foi utilizado a função *delay()* em código, pois poderia acarretar e um acúmulo de pequenos atrasos interferindo na leitura e envio de dados e ações dos atuadores. Assim foi necessário criar variáveis temporais que define quanto tempo podemos ficar segurando um botão acionado sem que ele contabilize que foi acionado várias vezes. Para a navegação pelo menu e submenu foi definido o tempo de 1000 milissegundos e para os botões no modo ajuste o tempo foi de 500 milissegundos.

Temos a criação das variáveis que vão receber os dados dos sensores, variáveis de ajuste que vão servir para setar o valor que desejamos para o ambiente interno da estufa. Além disso, foram criadas as variáveis para auxiliar na programação do WIFI e das credenciais do ThingSpeak, definindo variáveis como nome e senha do WIFI utilizado e as credenciais de acesso à minha conta ThingSpeak.

4.4.5 FUNÇÃO SETUP

Em Setup ocorre a inicialização dos pinos da ESP32, podemos definir se serão utilizados como entrada ou saída, juntamente com a comunicação do microcontrolador com o servidor ThingSpeak. Também temos a inicialização e configuração do display, do monitor serial, sensores DHT11.

4.4.6 FUNÇÃO CONECTA WIFI

Essa função vai verificar o status da rede WIFI, fazendo uma verificação se o microcontrolador está conectado ou não à rede. Caso não esteja teremos um intervalo de um segundo entre uma tentativa e outra de conexão juntamente com uma mensagem de aviso de conectado ao WIFI no display.

4.4.7 FUNÇÃO LOOP

A função loop será chamada continuamente após a inicialização do programa, todo o código está inserido nessa função, de modo direto ou indireto. Para melhor modularidade, interpretação e desenvolvimento, o programa foi dividido em funções, cada um executando uma atividade específica.

4.4.7.1 FUNÇÃO LEITURA SENSORES

Função para ler e processar os dados dos sensores internos e externos. Armazenando essas informações em variáveis globais que serão utilizadas em outras funções.

4.4.7.2 FUNÇÃO ENVIAR THINGSPEAK

Essa função é responsável por enviar as informações captadas e processadas pela ESP32 para a plataforma ThingSpeak, aqui temos um condição que permite que o envio dos pacotes com um intervalo de 15 em 15 segundos, esse é o intervalo de tempo mínimo entre um envio e outro, limitação imposta pela própria plataforma para a versão gratuita.

O envio é feito através da função *ThingSpeak.setField* que vai enviar os valores dos sensores para canais definidos dentro da plataforma. É feita uma verificação para informar se ocorreu algum erro durante o update.

4.4.7.3 FUNÇÃO CONDIÇÃO DE FUNCIONAMENTO

Nesta função temos as condições de funcionamento dos atuadores baseado nos valores que estão sendo captados pelos sensores. Definindo as ações de rega, ventilação e iluminação. Podemos observar melhor como ocorre o funcionamento através do fluxograma contido na Figura 23 na página 57.

4.4.7.4 FUNÇÃO BOTÕES

Foi implementada uma função que vai permitir a navegação pelos menus do meu display LCD. Assim oferecendo uma navegação amigável para o usuário interagir com as configurações do sistema. Para determinar o tempo e quais menus vão aparecer é utilizado uma combinação de “if” e funções “milles”. Através dessas estruturas é possível monitorar por quanto tempo o botão é pressionado, antes de ir para a próxima tela, essa abordagem garante que o usuário tenha tempo suficiente para selecionar uma opção desejada antes de avançar.

4.4.7.5 FUNÇÃO TEMPERATURA, UMIDADE SOLO E UMIDADE

Funções que utilizam das informações coletadas e processadas pela ESP32 para mostrar temperatura ambiente, umidade do solo e umidade do ar em um display LCD, assim permitindo que o usuário tenha acesso a esses dados. Isso proporciona um maior controle e conhecimento sobre o ambiente em que o sistema está sendo operado.

4.4.7.6 FUNÇÃO AJUSTE TEMPERATURA, UMIDADE SOLO E UMIDADE

Função que exibe as informações da temperatura ambiente, umidade do solo e umidade do ar, com base nos dados obtidos pelos sensores internos. Essa função possui uma característica adicional, onde é possível configurar os valores desejados para essas variáveis, que representam as condições ideais de estabilização da estufa. Essa configuração é realizada através dos botões de navegação, que permitem ao usuário ajustar os parâmetros de temperatura e umidade desejados.

Dessa forma, é possível criar um ambiente controlado e adequado para o crescimento saudável das plantas na estufa.

4.5 MONTAGEM DO HARDWARE DO SISTEMA

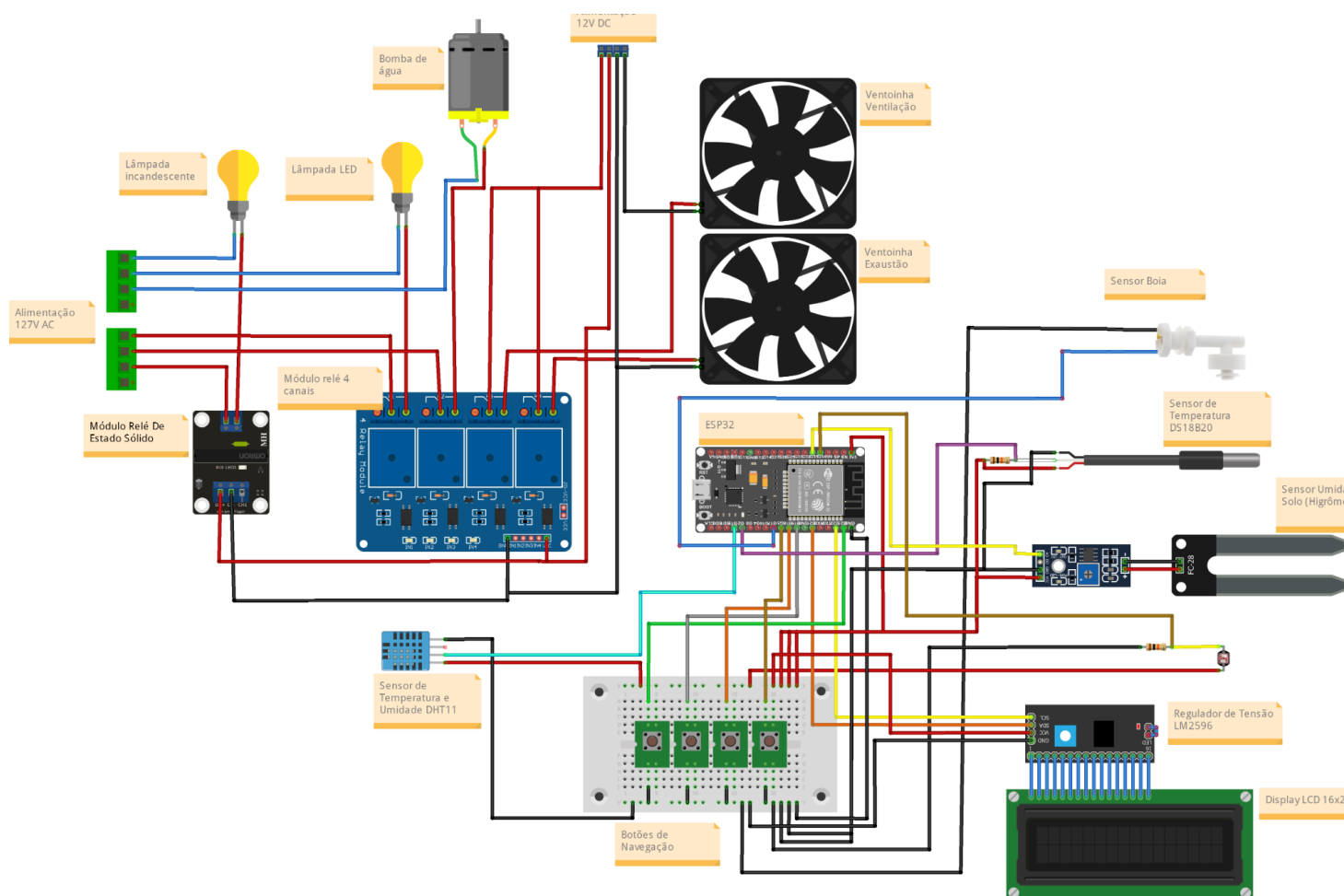
De acordo com JATI (2023), é necessário utilizar pinos do tipo ADC1 para os sensores. Esses pinos são capazes de converter um valor analógico, como a tensão, em uma forma digital. É importante evitar o uso dos conversores ADC2, pois eles são compartilhados com o driver do módulo de comunicação sem fio e não podem ser utilizados quando o Wi-Fi está em uso.

Tabela 16: Tabela da ligação dos componentes com a ESP32.			
ESP32	Tipo	Componentes	Tipo
GPIO_23	<i>INPUT</i>	Botão Direita	Digital
GPIO_19	<i>INPUT</i>	Botão Esquerda	Digital
GPIO_18	<i>INPUT</i>	Botão Enter	Digital
GPIO_5	<i>INPUT</i>	Botão Voltar	Digital
GPIO_15	<i>INPUT</i>	Pino Data do Sensor DHT11	Analógico
GPIO_2	<i>INPUT</i>	Sensor DS18B20	Analógico
GPIO_35	<i>INPUT</i>	Pino A0 do Sensor Higrometro Analogico	Analógico
GPIO_34	<i>INPUT</i>	Sensor LDR	Analógico
GPIO_17	<i>INPUT</i>	Sensor Nivel	Digital
GPIO_32	<i>OUTPUT</i>	Canal 1 Relé	Digital
GPIO_33	<i>OUTPUT</i>	Canal 2 Relé	Digital
GPIO_25	<i>OUTPUT</i>	Canal 3 Relé	Digital
GPIO_26	<i>OUTPUT</i>	Canal 4 Relé	Digital
GPIO_27	<i>OUTPUT</i>	Canal 1 Relé de Estado Sólido	Digital
GPIO_21	<i>INPUT</i>	Pino SDA do módulo I2C	I2C
GPIO_22	<i>INPUT</i>	Pino SCL do módulo I2C	I2C
Fonte: Elaborada pelo autor.			

Para o sensor de temperatura e umidade do ar, DHT11, sensor de temperatura, DS18B20, sensor de umidade do solo, Higrômetro e sensor de luminosidade, LDR, foram escolhidos os pinos analógicos. Foram utilizados cinco pinos para acionamento do relé que vai promover o controle da bomba de água, ventoinhas de exaustão e ventilação, as lâmpadas led e incandescente.

Teremos três pontos com diferentes níveis de tensão. Todo o sistema tem a alimentação principal vinda da rede elétrica de 127 volts AC, passando por um conversor AC/DC que vai fornecer uma alimentação de 12 volts DC e um regulador de tensão que vai regular os 12 volts para 5 volts. Foi necessário esses três pontos de alimentação para podermos fornecer a alimentação correta para diferentes tipos de componentes.

Figura 45 - Representação das ligações elétricas do sistema.



Fonte: Elaborada pelo autor.

A as lâmpadas de led e incandescentes estão sendo alimentadas pelo tensão de 127VAC, já as ventoinhas necessitam dos 12V DC do conversor AC/DC, a ESP32

e o display LCD está sendo alimentada através do seu Pino “Vin” com os 5V DC do regulador de tensão, já os sensores DTH11, DS18B20, Higrômetro e fotoresistor estão sendo alimentados pela tensão de 3,3V DC fornecida pela placa ESP32.

4.6 PREPARO DAS AMOSTRAS

Neste tópico é explorada a etapa da seleção das plantas que serão cultivadas nos dois ambientes de testes, assim como a preparação adequada da terra nas bandejas para plantio das microverdes.

4.6.1 SELEÇÃO DAS PLANTAS

A escolha das microverdes foi essencial para esses testes, tendo em vista o seu rápido tempo de germinação. Foram escolhidas a variedade e cenoura radesh, as cenouras radesh têm um tempo estimado de germinação entre 7 a 14 dias.

Figura 46 - Pacote de sementes utilizadas como amostras teste.



Fonte: Elaborada pelo autor.

4.6.2 PREPARAÇÃO DAS BANDEJAS

Como recipiente para plantio, foi selecionada uma bandeja de plástico com 15 células. Foi considerada a necessidade de oferecer espaço adequado para o crescimento de duas variedades de plantas de modo individualizado mas no mesmo

ambiente. Um recipiente também fornece o benefício de reter a umidade no solo e proteção contra danos externos.

Cada célula possui uma dimensão de x de largura, y de comprimento e z de altura.

Figura 47 - Bandejas de plástico com 15 células usadas na amostra da externa e da estufa.



Fonte: Elaborada pelo autor.

Na bandeja de teste que ficou dentro da estufa foi necessário adicionar a passagem de uma mangueira de silicone por cada célula para promover a irrigação adequada e autônoma das plantas cultivadas. Como podemos ver na Figura 48, a mangueira será acoplada ao motor que vai bombear a água do reservatório.

Figura 48 - Bandeja de plástico com a tubulação de irrigação já instalada.



Fonte: Elaborada pelo autor.

A terra utilizada foi proveniente do próprio quintal para o plantio, assim introduzindo e incorporando as condições reais de um solo caseiro em ambiente

imediatamente. A qualidade da terra não foi analisada tecnicamente mas sim através de uma análise qualitativa baseada em experiências empíricas, características como terra preta que foi coletada de um local com grande quantidade de matérias se decompondo. Na Figura 49 vemos os dois recipientes para plantios com terra coleta de uma mesma amostra.

Figura 49 - Bandejas de plástico já semeadas.



Fonte: Elaborada pelo autor.

4.7 SISTEMA DE IRRIGAÇÃO EM FUNCIONAMENTO

4.7.1 CONFIGURAÇÃO VIA IHM

Após ligar o sistema os valores de ajuste estão configurados de modo que os atuadores fiquem desenergizados, assim o sistema começa somente enviando os dados ao thingspeak. Fica a critério do usuário configurar os valores dos parâmetros da planta que será cultivada, isso torna a estufa flexível para uma maior diversidade de plantas.

Na Figura 50 podemos observar quais valores foram configurados para a germinação e crescimento das plantas.

Figura 50 - Configuração da estufa via IHM.



Configuração temperatura.



Configuração umidade do solo.



Configuração umidade do ar.

Fonte: Elaborada pelo autor.

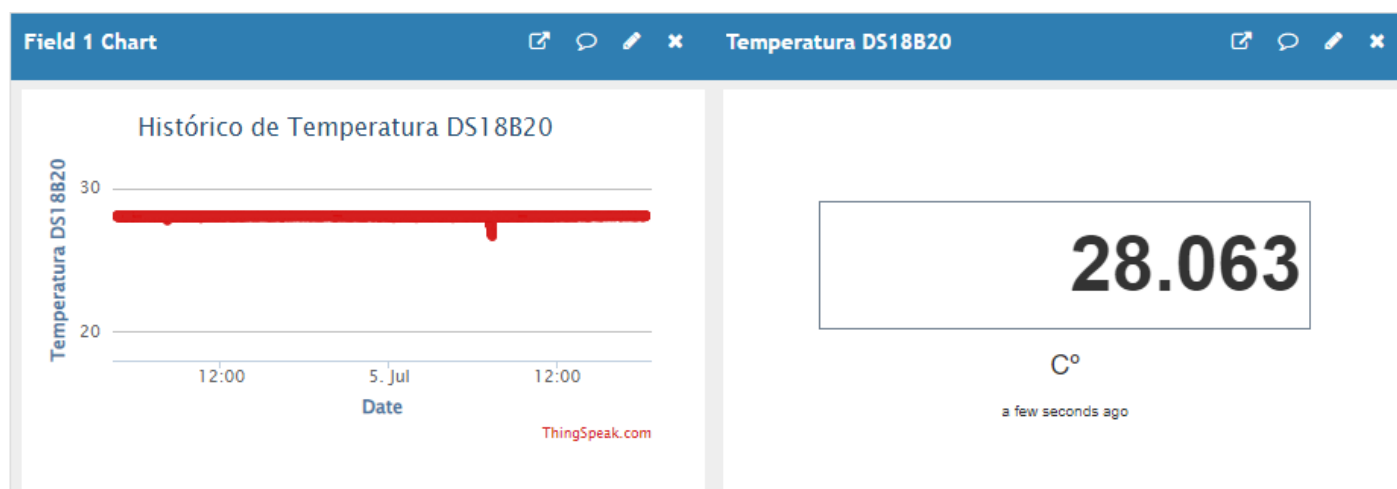
Uma observação que deve ser feita é que dentro do código foram aplicadas condições que impedem que o usuário configure valores fora de uma determinada faixa, pois em um exemplo hipotético, vamos imaginar que o usuário decida configurar a temperatura em 90 graus celsius, isso acarretaria na danificação da estufa e na provável morte do seu cultivo.

4.7.2 CAPTAÇÃO DE DADOS PELO THINGSPEAK E VISUALIZAÇÃO DOS DADOS VIA WEB E MOBILE

O próprio site proporciona algumas ferramentas de visualização como gráficos, display numérico e indicadores luminosos. Cada um pode ser configurado de tal forma que a informação seja entregue de modo eficiente, mostrando os principais indicadores do projeto.

Na Figura 51 temos um gráfico da temperatura pelo tempo do sensor DS18B20. O eixo y mínimo configurado para 15 e o eixo y máximo em 30, a temperatura está configurada para estabilizada entre os valores de 27 graus e 28 graus celsius, no gráfico podemos observar como a temperatura fica variando entre os valores de 27 e 28 graus. Ao lado podemos ver o valor da última temperatura.

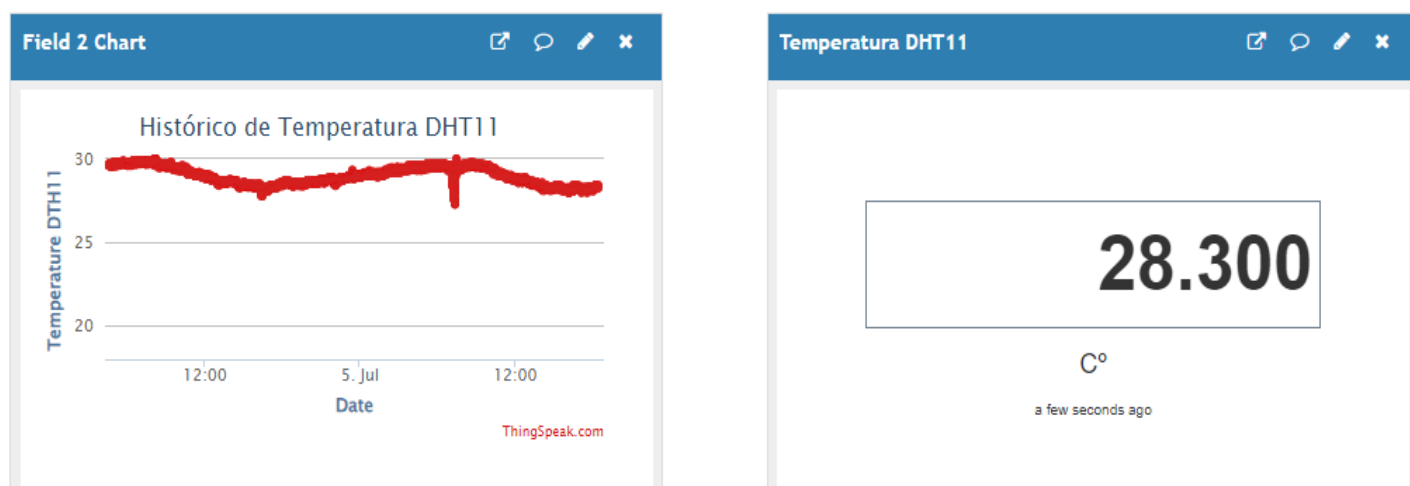
Figura 51 - Gráfico de temperatura e temperatura apresentada através do dashboard do ThingSpeak, referentes ao DS18B20.



Fonte: Elaborada pelo autor.

Na Figura 52 temos um gráfico da temperatura pelo tempo do sensor DTH11. O eixo y mínimo configurado para 15 e o eixo y máximo em 30, a temperatura está configurada para estabilizada em 28 graus celsius, não estamos utilizando os dados de temperatura do DTH11 como base para o controle autônomo da estufa devido a sua alta taxa de variação e baixa estabilidade, assim foi deixado apenas observação para futuras conclusões. Ao lado esquerdo que ocorre uma variação maior na temperatura comparado aos gráficos do sensor DS18B20, no lado direito podemos verificar a última temperatura capturada.

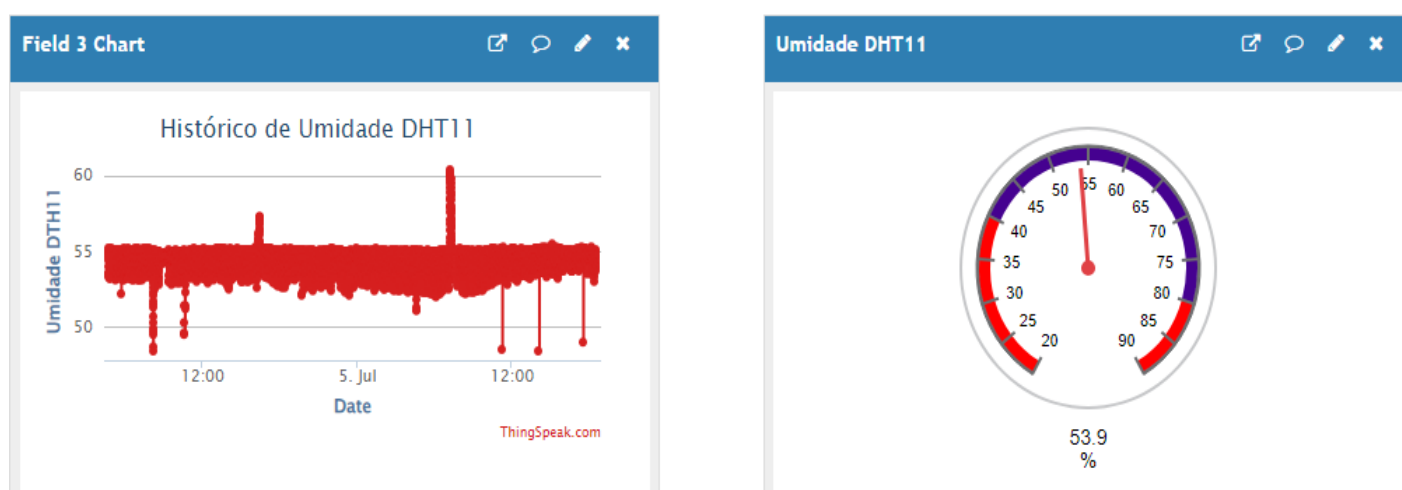
Figura 52 - Gráfico de temperatura e temperatura apresentada através do dashboard do ThingSpeak, referentes ao DHT11.



Fonte: Elaborada pelo autor.

Na Figura 53 temos um gráfico da umidade do ar captada pelo sensor DHT11. O eixo y mínimo configurado para 20 e o eixo y máximo em 90, a umidade está configurada para estabilizada entre os valores de 53% e 55%. Podemos observar que a umidade fica estabilizada nos valores definidos, com pequenos picos que podem ser decorrentes da abertura da estufa, variação da umidade externa ou da irrigação.

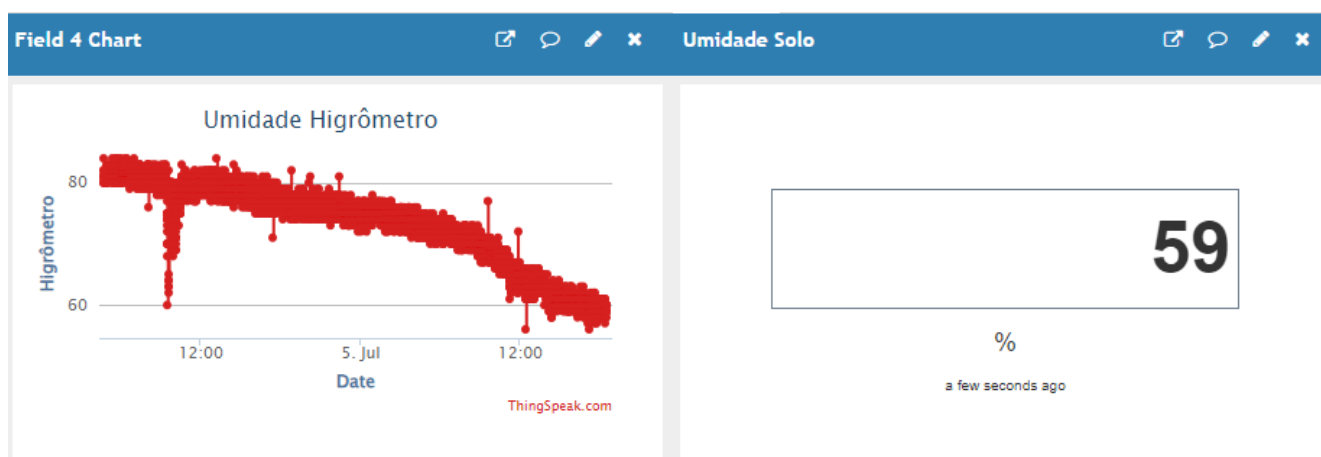
Figura 53 - Gráfico de umidade e umidade apresentada através do dashboard do ThingSpeak.



Fonte: Elaborada pelo autor.

Na Figura 54 temos um gráfico da umidade do solo captada pelo sensor higrômetro FC-28. O eixo y mínimo configurado para 20 e o eixo y máximo em 90, a variável da umidade do solo está configurada para estabilizada em 55%. É possível observar que a umidade do solo vai caindo com o tempo, devido a evaporação e absorção pelas sementes e raízes, até atingir a umidade mínima de 50% e a condição de tempo entre uma rega e outra. Ao lado podemos ver o valor da última medida da umidade do solo.

Figura 54 - Gráfico de umidade do solo e umidade do solo apresentada através do dashboard do ThingSpeak.



Fonte: Elaborada pelo autor.

O reservatório da estufa possui uma capacidade de 3,6 litros, isso promoveu uma autonomia que durou durante todo o experimento com as microverdes mas vendo a necessidade do usuário saber os principais status da estufa, foi posto duas lâmpadas indicadores no dashboard do thingspeak. Uma lâmpada verde que indica que os valores de água estão entre 3,6 litros e 0,5 litros, e a lâmpada vermelha que indica que o valor de água está abaixo de 0,5 litros, tendo a necessidade de colocar mais água dentro do reservatório.

Figura 55 - Estado da caixa d' água, reservatório vazio (luz vermelha acesa) e reservatório cheio (luz verde acesa).

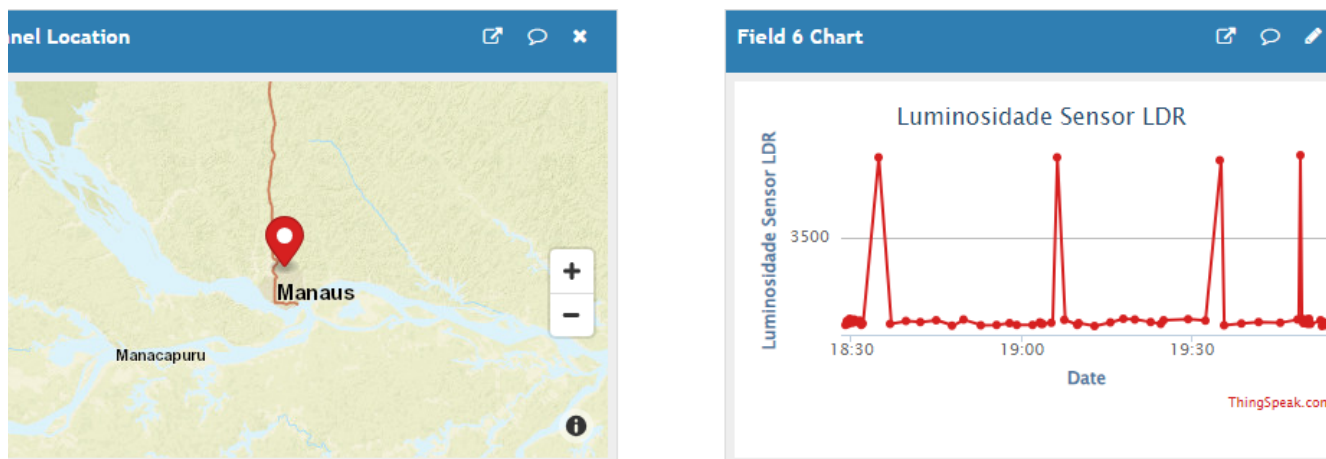


Fonte: Elaborada pelo autor.

Na Figura 56 no lado direito temos o gráfico com a intensidade luminosa pelo tempo, esses dados estão sendo capturados pelo sensor LDR fixado na parte

superior interna da estufa. E ao lado esquerdo temos a localização de onde se encontra a estufa no momento.

Figura 56 - Painel com a localização da estufa (a esquerda), índice de luminosidade dentro da estufa (a direita).



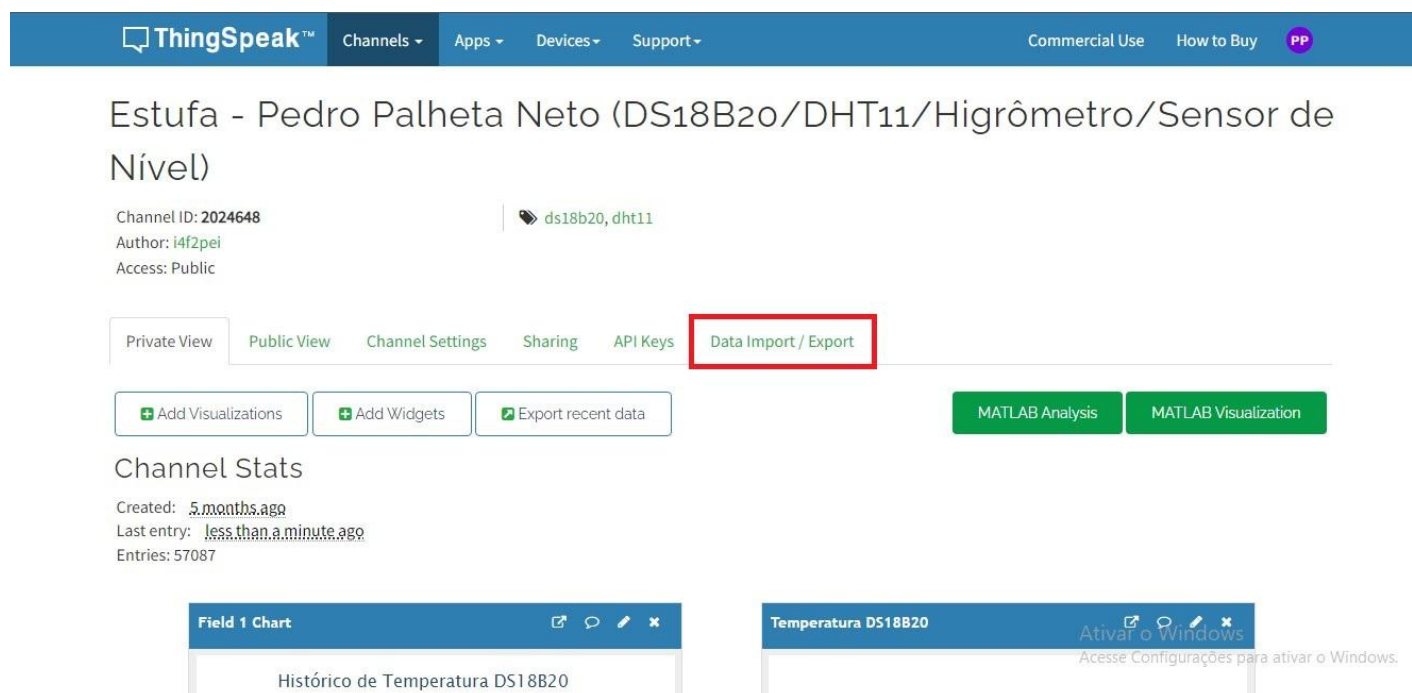
Fonte: Elaborada pelo autor.

4.7.3 EXPORTAÇÃO DOS DADOS

Caso o usuário queira o acesso a todos os dados que foram enviados ao Thingspeak deve seguir esse procedimento. Tais dados dentro do escopo do projeto tem uma grande importância para análise e visualização, pois podemos realizar análises mais detalhadas e obter conclusões a partir dos padrões e comportamentos registrados. Assim identificando correlações, ajudando a tomar decisões sobre novas configurações de setup para a estufa.

Depois de fazer login e acessar o canal da nossa estufa, vamos entrar no painel do canal, clicando na opção de “Data Import / Export” como mostrado na Figura 57.

Figura 57 - Tela inicial do canal de monitoramento da estufa através do ThingSpeak.



Fonte: Elaborada pelo autor.

Após acessar o Data Import / Export estaremos na página de exportação de dados, como podemos observar na Figura 58. Vamos em “Export” e clicamos “Download”. Dependendo da quantidade de dados que já foi coletada a exportação pode levar algum tempo.

Figura 58 - Acessando a área de importação de dados do canal.

ThingSpeak™ Channels Apps Devices Support Commercial Use How to Buy PP

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Import

Upload a CSV file to import data into this channel.

File Nenhum arquivo escolhido

Time Zone

Export

Download all of this Channel's feeds in CSV format.

Time Zone

Help

Import

The correct format for data import is provided in this [CSV Import Template File](#). Use the field names *field1*, *field2*, and so on, instead of custom field names.

CSV Import Format

```
created_at,field1,field3,field4,field8,elevation
2019-01-01T10:11:12-05:00,11,33,44,88,10
```

Other Import and Export Options

You can also use MATLAB, the REST API, or the MQTT API to import and export channel data.

[Read Data](#)

[Write Data](#)

Ativar o Windows

Fonte: Elaborada pelo autor.

Quando o download estiver concluído, um arquivo CSV será gerado, contendo todos os dados dos dados dos canais, na Figura 59 podemos ver um exemplo de arquivo.

Figura 59 - Ícone do arquivo.



Fonte: Elaborada pelo autor.

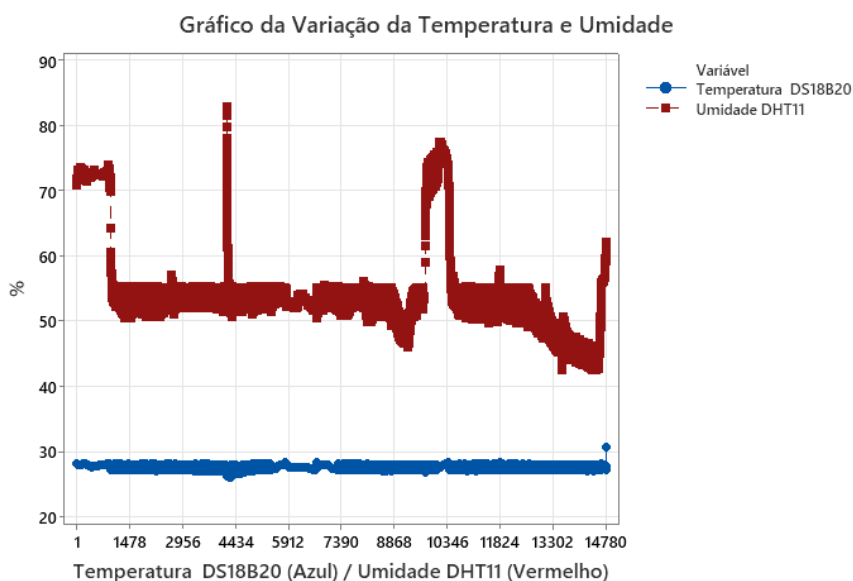
Na Figura 60 podemos ver como os dados são apresentados, abrimos o arquivo CSV através do bloco de notas. A primeira linha, "created_at,field1,field2,field3, field4, field6, latitude, longitude, elevation, status", contém os cabeçalhos das colunas. O "created_at" indica a data e hora em que os dados foram registrados. Os campos "field..." são referentes aos canais definidos no ThingSpeak. "latitude, longitude, elevation" se referem a localização de onde estão vindo os dados, esses campos estão em branco pois não está dando o envio de tais informações assim como do status.

A temperatura do DS18B20 apresentou uma média de 27,5 com erro padrão da média de 0,0027 o que é um valor baixo, demonstrando uma alta confiabilidade da média amostral calculada. Já o desvio padrão é de 0,334 indicando um baixo desvio dos valores individuais da média, o que é muito bom pois é um dos principais fatores que nos propomos a controlar.

Já em relação a umidade do ar tivemos uma média de 54,59 e apresentando um erro padrão da média de 0,058, claramente um erro um pouco maior que o da temperatura isso se deve ao fato que a umidade poderia variar entre os valores de 55% e 53%. O desvio padrão apresenta o valor de 7,053 demonstrando que os dados estão um pouco mais distantes da média, isso pode ter decorrido o ambiente com alta umidade em que a estufa se encontrava.

Os valores da umidade estavam previstos para variar de 90% a 50% mas devido a irrigação ocorrer próximo às raízes logo próximo ao sensor higrômetro atingimos valores de 100% em alguns momentos da rega e em um momento de troca de posição do sensor atingimos a umidade de 9%. A média se manteve em 91,97 mas tivemos valores de erro padrão da média e desvio padrão deveras altos, indicando que nem todos os valores da amostra estão próximos da média, sendo assim a média é um parâmetro ruim para avaliar a estabilidade de um valor da umidade do solo.

Figura 61 - Gráfico da variação da temperatura e umidade.



Fonte: Elaborada pelo autor.

4.8 ANÁLISE COMPARATIVA QUALITATIVA ENTRE O CULTIVO REALIZADO DENTRO E FORA DO PROTÓTIPO

A análise comparativa entre o cultivo no interior da estufa e no ambiente externo deixou claro algumas diferenças no desenvolvimento das plantas. Enquanto dentro da estufa houve um crescimento uniforme e rápido, com a germinação de uma quantidade maior de sementes comparada às semeadas e cultivadas em ambiente externo.

Já em ambiente externo houve um subdesenvolvimento das microverdes, com o insucesso da germinação de muitas sementes, uma menor quantidade de folhas por ramo desenvolvido e o ressecamento das folhas, assim tendo um desenvolvimento lento e irregular.

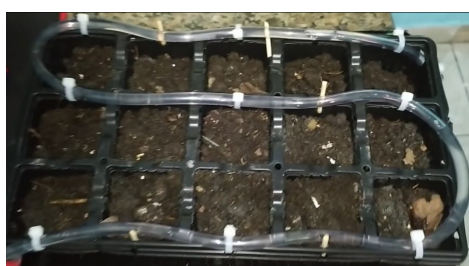
Podemos observar na Figura 62 e Figura 63, o desenvolvimento registrado por fotografia da planta, em diferentes dias para salientar a diferença em ambos os cultivos.

Figura 62 - Registro de desenvolvimento das microverdes 1.

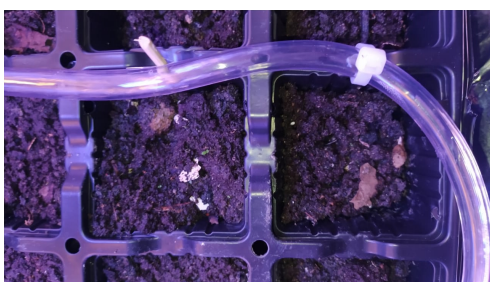
Amostra A



Dia 0



Dia 3



Dia 5



Dia 6

Amostra B



Dia 0



Dia 3



Dia 5



Dia 6

Fonte: Elaborada pelo autor.

Figura 63 - Registro de desenvolvimento das microverdes 2.

Amostra A



Dia 10



Dia 15



Dia 18

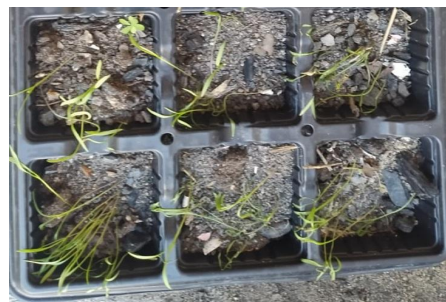


Dia 21

Amostra B



Dia 10



Dia 15



Dia 18



Dia 21

Fonte: Elaborada pelo autor.

Na Figura 64 podemos observar o último registro feito das microverdes na estufa, apresentando uma boa coloração e com novos crescimento de novas folhas em todos os ramos analisados.

Figura 64 - Último registro da microverde na estufa.



Fonte: Elaborada pelo autor.

Já na Figura 65 podemos observar o último registro feito das microverdes na área externa, apresentam uma boa coloração das folhas mas uma baixa densidade e com pouco desenvolvimento de novas folhas em seus ramos.

Figura 65 - Último registro da microverde na área externa.



Fonte: Elaborada pelo autor.

5. CONSIDERAÇÕES FINAIS

Os objetivos propostos no início deste projeto foram executados e testados com sucesso, dando origem a um sistema de monitoramento e controle para uma estufa indoor. As tecnologias de sensoriamento e acionamento estão disponíveis em diversas lojas de eletrônicos e com preços relativamente acessíveis, atendendo as exigências de captura de dados do projeto.

Todo o código foi desenvolvido em linguagem C++ dentro da plataforma IDE Arduino, todas as linhas de código estão com seus principais pontos comentados e serão postas no apêndice, as bibliotecas e funções utilizadas são de fácil utilização. A implementação com o ThingSpeak foi graças às bibliotecas que viabilizam o uso do protocolo de comunicação para envio de dados à plataforma.

Em relação a estrutura física desenvolvida para a estufa foi utilizado o advento de criatividade e de artifícios técnicos caseiros, cabos, caixa de isopor, canaletas, papel alumínio são materiais encontrados na maioria das lojas de materiais de construção e supermercados, tornando viável para qualquer um que queira replicar a estrutura modificando seu tamanho para atender melhor a sua réplica.

Resultados obtidos foram satisfatórios, o envio, armazenamento de informações e apresentação da mesma através do dashboard do ThingSpeak foram um sucesso sendo tal questão validada nas primeiras horas de testes, pequenos detalhes envolvendo a lógica de programação foram aperfeiçoados ao longo dos testes, um exemplo foi o tempo de luminosidade pela lâmpada LED, onde tivemos problemas nas condições que deixavam ela ligada ou desligada.

Os gráficos e análise das informações coletadas dos canais de sensoriamento apresentaram resultados dentro das expectativas, o controle de umidade do solo e umidade do ar ficaram dentro de suas taxas de variação definidas. Na questão da temperatura tivemos dificuldade em manter abaixo da temperatura máxima em certos períodos do dia devido a alta temperatura externa, mas isso não veio a interferir de forma prejudicial ao crescimento da amostra utilizada, mas deve se atentar caso a amostra precise de baixas temperaturas em futuros experimentos e validações. Os tempos de iluminação a luz proveniente da lâmpada LED foram um sucesso depois dos ajustes nas condições da programação.

Através da utilização de duas amostras controle foi constatado benefícios significativos do cultivo de hortaliças em uma estufa indoor automatizada, a coloração das folhas, tamanho dos ramos, número de folhas e quantidade de sementes germinadas apresentam características superiores a amostra cultivada em ambiente externo.

Em relação aos os instrumentos de coleta de dados tivemos duas ocorrências que devem ser citadas, no dia 7 do experimento o sensor DS18B20 apresentou um problema intermitente, onde a temperatura registrada às vezes estava em -127 graus celsius, isso implicou na lâmpada ligando algumas vezes para aquecer a estufa, essa questão foi observada rapidamente através dos gráficos de monitoramento. Foi efetuado a troca do sensor e não houve mais ocorrências com tal sensor.

Agora em relação ao sensor higrômetro resistido, ao passar do tempo as lâminas utilizadas pelo sensor apresentaram oxidação, assim os valores configurados inicialmente tiveram que ser alterados para que a porcentagem de umidade do solo estivesse de acordo com a realidade. Sugestão de nos próximos projetos ocorra a troca do sensor higrômetro por outro modelo.

Para pesquisas futuras que desejam utilizar este trabalho como apoio, aqui estão algumas sugestões de melhoria:

- A troca do sensor DHT11 pelo DHT22, pois ele possui um maior precisão nas medições e abrange uma faixa de temperatura e umidade.
- Substituição da lâmpada LED por uma lâmpada Grow que emite uma maior diversidade de espectro luminoso.
- Construção de uma estufa de maneira mais profissional, substituindo o papel alumínio em seu interior por mylar, que é um filme de poliéster que apresenta alta resistência ao calor e propriedades de isolamento.
- Acoplamento do reservatório de água a lateral da caixa da estufa e a fixação da case a parte superior da estufa.
- Estudo da utilização de inteligência artificial e implementação de câmeras para ajustes automáticos dos parâmetros utilizados para cada fase de desenvolvimento da planta cultivada.

REFERÊNCIAS

ANDRADE, C.; BRITO, R. A. L. **Métodos de irrigação e quimigação**. 2006.

AZEVEDO JUNIOR, Amber Leite de. **Sistema de monitoramento e climatização de estufa de pequeno porte em um contexto doméstico**. 2016.

BOXES.PY. **Logo do Boxes PY**. 2017. Disponível em:
<https://florianfesti.github.io/boxes/static/boxes-logo.svg> Acesso em: 24 de julho de 2023

CASTRUCCI, P.; MORAES C. C. de. **Engenharia de Automação Industrial**. 2 ed. Editora LTC, 2007.

CARRION, P.; QUARESMA, M. **Internet da Coisas (IoT): Definições e aplicabilidade aos usuários finais**. Human Factors in Design, Florianópolis, v. 8, n. 15, p. 049-066, 2019. DOI: 10.5965/2316796308152019049. Disponível em:
<https://www.revistas.udesc.br/index.php/hfd/article/view/2316796308152019049>. Acesso em: 26 abr. 2023.

CERMEÑO, Z. S. **Construcción de invernaderos**. Madrid: Mundi-Prensa, 2005.

DANITA, M. et al. **IoT based automated greenhouse monitoring system**. In: 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE, 2018. p. 1933-1937.

FERNANDES, G. D.; PREUSS, E.; SILVA, T. L. **Sistema Automatizado de Controle de Estufas para Cultivo de Hortaliças**. Frederico Westphalen, 2017. 32 p. Disponível em:
https://repositorio.ufsm.br/bitstream/handle/1/12958/TCCG_SIFW_2017_FERNANDES_DOUGLAS.pdf?sequence=1&isAllowed=y. Acesso em: 20 jun. 2023.

FONSECA, J. J. S. **Metodologia da pesquisa científica**. Fortaleza: UEC, 2002. Apostila.

GIL, Antônio Carlos. **Métodos e técnicas de pesquisa social**. 6 ed. São Paulo: Atlas, 2008.

GOKHALE, Pradyumna; BHAT, Omkar; BHAT, Sagar. **Introduction to IOT**. **International Advanced Research Journal in Science, Engineering and Technology**, v. 5, n. 1, p. 41-44, 2018.

HOPKINS, W. G. **Photosynthesis and Respiration**. Facts On File, Incorporated, 2006.

JATI, J. M. d. N. **Sistema para cultivo de plantas utilizando princípios de Internet das coisas**. Faculdade de Engenharia Automação e Controle. Manaus, pg. 68. 2023.

LIMA, J. E. F. W.; FERREIRA, Raquel Scalia Alves; CHRISTOFIDIS, Demetrios. **O uso da irrigação no Brasil. O estado das águas no Brasil. Agência Nacional de Energia Elétrica. CD-ROM**, 1999.

LUZ, Paulo César Vargas et al. **Suplementação luminosa aplicada ao cultivo de hortaliças**. SEPOC 2022, 2022.

MAKER HERO. **Regulador de Tensão LM2596 Conversor DC-DC Step Down**.

Disponível em:

<https://www.makehero.com/produto/regulador-de-tensao-lm2596-conversor-dc-dc-step-down/>. Acesso em: 06 jul. 2023.

PFEUFER, Emily; GAUTHIER, Nicole W.; KONOPKA, Philip. **Managing Greenhouse & High Tunnel Environments to Reduce Plant Diseases**. University of Kentucky. 2016. Disponível em

<https://www.controlledenvironments.org/wpcontent/uploads/sites/6/2017/06/Managing-Greenhouse-High-Tunnel-Environmentsto-Reduce-Plant-Diseases.pdf>. Acesso em: 20 jun. 2023.

PLANTEI. **Tudo o que Você Precisa Saber Sobre Cultivo Indoor**. 2017. Disponível em: <https://blog.plantei.com.br/cultivo-indoor/>. Acesso em: 20 jun. 2023.

PURQUERIO, L. F. V.; TIVELLI, S. W. **Manejo do Ambiente em Cultivo protegido-Instituto Agrônômico (IAC)**. 2005.

RAZUK, Paulo Cesar. **O Método Científico**. Disponível em:

<https://pt.scribd.com/document/125908704/o-metodo-cientifico-pdf#>. 2015. Acesso em: 21 jun. 2023.

REIS, Neville V. B. dos. **Construção de estufas para produção de hortaliças nas Regiões Norte, Nordeste e Centro-Oeste**. 2005. Disponível em:

<https://core.ac.uk/download/pdf/15428573.pdf>. Acesso em: 20 jun. 2023.

SANTOS, Leão Lucas; SEABRA JUNIOR, Santino; NUNES, Moitinho Cândida Maria. **Luminosidade, temperatura do ar e do solo em ambientes de cultivo protegido**. Revista de Ciências Agro-Ambientais, v. 8, n. 1, p. 83-93, 2010.

SCARPARE FILHO, J.A. Viveiros para formação de mudas. In: MINAMI, K. (Ed.) **Produção de mudas de alta qualidade em horticultura**. São Paulo: T.A. Queiroz, 1995. p. 47-51.

SEMLE, Aron. **Protocolos IIoT para considerar**. Aadeca Revista, v. 34, 2016.

SILVEIRA, C. B. **IHM: Saiba quais os Tipos e como Selecionar**. 2016. Disponível em: <https://www.citisystems.com.br/ihm/>. Acesso em: 29 jun. 2023.

TESTEZLAF, Roberto. **Irrigação: métodos, sistemas e aplicações**. 2011.

THIOLLENT, M. (2009). **Metodologia de Pesquisa-ação**. São Paulo: Saraiva.

APÊNDICE 1 - CÓDIGO DO PROJETO

```
//
=====

// --- Bibliotecas ---
#include <WiFi.h> //
Recursos de WiFi.
#include <OneWire.h> //
Protocolo de comunicação OneWire.
#include <DallasTemperature.h> //
Comunicação com sensor DS18B20.
#include <ThingSpeak.h> //
Recursos e comunicação com nuvem ThingSpeak.
#include "DHT.h" //
Biblioteca Sensor DHT.
#include <Wire.h> //
Gerenciar a comunicação entre os dispositivos através do protocolo I2C.
#include <LiquidCrystal_I2C.h> //
Permite controlar displays I2C.
#include <PubSubClient.h> //
Biblioteca para cliente para mensagens MQTT.
//
=====

//
=====

// --- Mapeamento de Hardware ---
#define botao_direita 23 //
Botão direita.
#define botao_esquerda 19 //
Botão esquerda.
#define botao_enter 5 //
Botão enter.
#define botao_voltar 18 //
Botão voltar.
```

```

#define      Sensor_DHT11                      15          //
Pino digital 2 conectado ao DHT11.

#define      Sensor_DS18B20                    2           //
Pino de conexão com o sensor DS18B20.


const  int  Sensor_Higrometro_Analogica        = 35;        //
Pino de conexão com o sensor DS18B20.


const  int  Sensor_LDR                        = 34;        //
Pino de conexão com o sensor LDR.


const  int  sensor_nivel                      = 17;        //
Pino de conexão com o sensor de nível.


const  int  rele1_lampada_led                  = 32;        //
Pino ao qual o Módulo 1 do Relé está conectado, controle da lâmpada
LED.

const  int  rele2_bomba_agua                  = 33;        //
Pino ao qual o Módulo 2 do Relé está conectado, controle da bomba de
água.

const  int  rele3_ventoinha_exaustao          = 25;        //
Pino ao qual o Módulo 3 do Relé está conectado, controle da ventoinha
exaustão.

const  int  rele4_ventoinha_ventilacao        = 26;        //
Pino ao qual o Módulo 4 do Relé está conectado, controle da ventoinha
ventilação.


const  int  rele_solido_lampada_incandescente = 27;        //
Pino ao qual o Módulo 4 do Relé está conectado, controle da lâmpada
incandescente.


//
=====
=====

// --- Constantes e Objetos ---

```

```

#define DHTTYPE DHT11 //
Sensor DHT11.
DHT dht(Sensor_DHT11, DHTTYPE); //
Configurando o Sensor DHT.

#define menu_max 2 //
Número máximo de menus existentes.

LiquidCrystal_I2C lcd(0x27, 16, 2); //
Endereço I2C e definição do tamanho do display.

OneWire oneWire(Sensor_DS18B20); //
É informado o pino de conexão com o DS18B20.
DallasTemperature sensor(&oneWire); //
Passando referência do objeto oneWire.
WiFiClient client; //
Objeto para operações via WiFi.

//
=====
=====
// --- Funções ---
void botoes(); // Função para receber
identificar quais botões estão acionados e encaminhar para suas
funções.
void temperatura(); // Função que vai mostrar a
temperatura ambiente.
void umidade_solo(); // Função que vai mostrar a
umidade do solo.
void umidade(); // Função que vai mostrar a
umidade do ar.

void ajustar_temperatura(); // Função que permite ajustar
a temperatura ambiente.
void umidade_solo(); // Função que permite ajustar
a umidade do solo.
void ajustar_umidade(); // Função que permite ajustar
a umidade do ar.

```



```

void condicao_funcionamento();           // Função com as condições de
funcionamento dos atuadores.

//
=====
=====
// --- Definindo variáveis e constantes globais ---
int          menu_num                    = 1;
// Número inicial do menu.
int          sub_menu                    = 1;
// Número inicial do submenu.

unsigned long tempo_botoes_menu          = 1000;
// Delay do botão durante a navegação pelo menu.
unsigned long tempo_botoes_ajustes       = 500;
// Delay do botão durante o ajuste das variáveis umidade/temperatura.

unsigned long tempo_rega                  = 3600000;
// Variável temporal para verificar se o tempo mínimo entre uma rega e
outra.
unsigned long tempo_referencia_rega       = 0;
//

unsigned long tempo_luz_ligada             = 72000000;
// Variável temporal para verificar o tempo que planta vai receber luz.
unsigned long tempo_luz_desligada         = 14400000;
// Variável temporal para verificar o tempo que planta vai receber luz.

unsigned long tempo_luz_atual              = 0;
//
unsigned long tempo_luz_anterior           = 0;
//

bool lampada_ligada                       = false;
//57600000

```

```

unsigned long tempo_botao_direito                = 0;
// Variável temporal para verificar se o tempo mínimo para clicar de
novo passou.
unsigned long tempo_botao_esquerdo              = 0;
// Variável temporal para verificar se o tempo mínimo para clicar de
novo passou.
unsigned long tempo_botao_enter                 = 0;
// Variável temporal para verificar se o tempo mínimo para clicar de
novo passou.
unsigned long tempo_botao_voltar                = 0;
// Variável temporal para verificar se o tempo mínimo para clicar de
novo passou.

int          comparador_direito                 = 0;
// Variável auxiliar para criar a logica do botão.
int          comparador_esquerdo               = 0;
// Variável auxiliar para criar a logica do botão.
int          comparador_enter                  = 0;
// Variável auxiliar para criar a logica do botão.
int          comparador_voltar                 = 0;
// Variável auxiliar para criar a logica do botão.

int          temperatura_valor                  = 28;
// Temperatura do ar inicialmente configurada por padrão.
int          umidade_solo_valor                 = 55;
// Umidade do solo inicialmente configurada por padrão.
int          umidade_ar_valor                  = 55;
// Umidade do ar inicialmente configurada por padrão.

float        Temperaura_Sensor_DS18B20         = 0;
// Variavel para guardar o valor da Temperatura do Sensor DS18B20.
float        Temperaura_Sensor_DTH11           = 0;
// Variavel para guardar o valor da Temperatura do Sensor DTH11.
float        Umidade_Sensor_DTH11              = 0;
// Variavel para guardar o valor da Umidade do Sensor DTH11.
float        Umidade_Sensor_Higrometro_Analogica = 0;
// Variavel para guardar o valor analogico do Sensor Higrômetro.
float        Umidade_Sensor_Higrometro         = 0;
// Variavel para guardar o valor mapeado do Sensor Higrômetro.

```

```

int          Luminosidade_Sensor_LDR                = 0;
// Variavel para guardar o valor da captado pelo LDR.

int          estado_do_nivel                        = 0;
// Variavel para guardar o estado lgico do sensor nível.

const char   *ssid                                =
"Wifi_Palheta";                                // Nome e senha para conexão.
const char   *password                            =
"Palheta281915";                                // com rede WiFi.

const long   CHANNEL                             = 2024648;
// Parâmetros de acesso ao canal.
const char   *WRITE_API                           =
"JQU8MVUCIOEQ3700";                            // específico na nuvem ThingSpeak:
const char   *READ_API                             =
"JBTY5QWD09FQYSNI";                            // Canal e chaves escrita/leitura.
int          intervalSensor                        = 2000;
// Parâmetros para definição de intervalo.
long         prevMillisThingSpeak                  = 0;
// mínimo entre duas escritas no ThingSpeak,
int          intervalThingSpeak                    = 15000;
// que deve ser de 15s para contas free
int          erro;
// Guarda valor de erro de escrita no ThingSpeak.

//
=====
=====
// --- Definindo função setup() ---
void setup()
{
    pinMode(botao_direita, INPUT_PULLUP);                //
    Pino conectado ao botao é um pino de entrada de sinal.
    pinMode(botao_esquerda, INPUT_PULLUP);                //
    Pino conectado ao botao é um pino de entrada de sinal.
    pinMode(botao_enter, INPUT_PULLUP);                    //
    Pino conectado ao botao é um pino de entrada de sinal.

```

```

    pinMode(botao_voltar, INPUT_PULLUP); //
Pino conectado ao botao é um pino de entrada de sinal.

    lcd.begin(); //
Inicializa a interface para a tela LCD.
    lcd.backlight(); //
Liga a luz de fundo do Display.

    sensor.begin(); //
Inicializa sensor.

    Serial.begin(115200); //
Inicializa monitor serial.

    ThingSpeak.begin(client); //
Inicializa o ThingSpeak.

    dht.begin(); //
Inicializa o sensor DHT11.

    pinMode(Umididade_Sensor_Higrometro_Analogica, INPUT); //
Define Umididade_Sensor_Higrometro_Analogica (pino analógico 35) como
entrada.

    pinMode(Luminosidade_Sensor_LDR, INPUT); //
Define Umididade_Sensor_Higrometro_Analogica (pino analógico 35) como
entrada.

    pinMode(sensor_nivel, INPUT_PULLUP); //
Define sensor_nivel (pino analógico 34) como entrada.

    pinMode(rele1_lampada_led, OUTPUT);
// Seta o pino como saída.
    pinMode(rele2_bomba_agua, OUTPUT);
// Seta o pino como saída.
    pinMode(rele3_ventoinha_exaustao, OUTPUT);
// Seta o pino como saída.
    pinMode(rele4_ventoinha_ventilacao, OUTPUT);
// Seta o pino como saída.

```

```

    pinMode(rele_solido_lampada_incandescente, OUTPUT);
// Seta o pino como saída.

    digitalWrite(rele1_lampada_led, HIGH);
// Seta o pino com nível logico baixo.
    digitalWrite(rele2_bomba_agua, HIGH);
// Seta o pino com nível logico baixo.
    digitalWrite(rele3_ventoinha_exaustao, HIGH);
// Seta o pino com nível logico baixo.
    digitalWrite(rele4_ventoinha_ventilacao, HIGH);
// Seta o pino com nível logico baixo.
    digitalWrite(rele_solido_lampada_incandescente, HIGH);
// Seta o pino com nível logico baixo.
}

//
=====
// --- Definindo função ConectaWiFi() ---
void ConectaWiFi()
{
    if (WiFi.status() != WL_CONNECTED)
    {
        WiFi.begin(ssid, password); //
        Essa função é necessária para que a placa se conecte a rede WiFi.
        Usando dois parâmetros para a função, o nome e senha da rede.

        while (WiFi.status() != WL_CONNECTED) //
        {
            WiFi.status(), que retorna uma constate informando se houve problema de
            conexão ou não.
            {
                lcd.setCursor(0,0); //
                Seta a posição em que o cursor vai inicializar (Linha 1).
                lcd.print("Conectando Wi-Fi"); //
                Escrever "Conectando Wi-Fi" na linha 1.
                lcd.setCursor(0,1); //
                Seta a posição em que o cursor vai inicializar (Linha 2).

```

```

        lcd.print("      ....      ");
Escrever "      ....      " na linha 2.

    }
}
}

//
=====
=====
// --- Definindo função loop() ---
void loop()
{
    botoes();

    // Função para receber identificar quais botões estão acionados e
    identificar qual função sera selecionada.
    switch(menu_num)
    // Switch com 3 casos que são referentes aos menus de temperatura,
    umidade do solo e umidade do ar.
    {
        case 1: temperatura();    break;
    // Caso "menu_num" seja "1" entrar na função temperatura.
        case 2: umidade_solo();    break;
    // Caso "menu_num" seja "2" entrar na função umidade_solo.
        case 3: umidade();         break;
    // Caso "menu_num" seja "3" entrar na função umidade.
    } //fim switch

    // (Re)Conectando à rede WiFi //
    ConectaWiFi();
    // Função para se conectar a rede Wi-Fi.

    leitura_sensores();

    condicao_funcionamento();
    // Função para determinar em qual condição de funcionamento a estufa
    vai está.

```

```

    enviar_thingspeak();

    monitor_serial();
}

//
=====
//Função para leitura dos sensores.
void leitura_sensores()
{
    // Efetuando a leitura de temperatura no sensor DS18B20. //
    sensor.requestTemperatures();
    // Solicita que a função informe a temperatura do sensor.
    Temperatura_Sensor_DS18B20 = sensor.getTempCByIndex(0);
    // Passar o valor da temperatura medida para a variavel
    Temperatura_Sensor_DS18B20.

    // Efetuando a leitura de temperatura no sensor Higrômetro. //
    Umidade_Sensor_Higrometro_Analogica =
    analogRead(Sensor_Higrometro_Analogica); //
    Lê o valor analogico do Sensor Higrometro.
    Umidade_Sensor_Higrometro =
    map(Umidade_Sensor_Higrometro_Analogica, 4095, 2020, 0, 100); //
    Mapeia o valor analogico do Sensor Higrometro.

    if(Umidade_Sensor_Higrometro >= 100)
    {
        Umidade_Sensor_Higrometro = 100;
    }

    // Efetuando a leitura de temperatura no sensor DTH11. //
    Umidade_Sensor_DTH11 = dht.readHumidity();
    // Leitura da Umidade do Sensor DTH11.
    Temperatura_Sensor_DTH11 = dht.readTemperature();
    // Leitura da Temperatura do Sensor DTH11.

```

```

    // Efetuando leitura do estado do sensor de nível.
    estado_do_nivel                = digitalRead(sensor_nivel);
// Leitura do estado do sensor de nível.

    // Efetuando a leitura da luminosidade. //
    Luminosidade_Sensor_LDR = analogRead(Sensor_LDR);
// Leitura da Luminosidade com LDR.
}

//
=====
=====
//Função para enviar dados.
void enviar_thingspeak()
{
    // Se decorrido o intervalo mínimo entre uma e outra escrita no
    ThingSpeak
    if (millis() - prevMillisThingSpeak > intervalThingSpeak)
// Permite o envio somente se o interalo mínimo de 15 segundos tiver
passado.
    {
        ThingSpeak.setField(1, Temperatura_Sensor_DS18B20);
// Configurando campo 1 com valor lido.
        ThingSpeak.setField(2, Temperatura_Sensor_DTH11);
// Configurando campo 2 com valor lido.
        ThingSpeak.setField(3, Umidade_Sensor_DTH11);
// Configurando campo 3 com valor lido.
        ThingSpeak.setField(4, Umidade_Sensor_Higrometro);
// Configurando campo 4 com valor lido.
        ThingSpeak.setField(5, estado_do_nivel);
// Configurando campo 5 com valor lido.
        ThingSpeak.setField(6, Luminosidade_Sensor_LDR);
// Configurando campo 5 com valor lido.

        erro = ThingSpeak.writeFields(CHANNEL, WRITE_API);
// Escrevendo no canal.

```



```

// Informando se ocorreu escrita com sucesso ou algum erro
if (erro == 200)
{
    Serial.println(" >> Update realizado com sucesso!");
}
else
{
    Serial.println(" >> Problema no canal - erro HTTP " +
String(erro));
}
prevMillisThingSpeak = millis();
}
}

//
=====
=====
// --- Função com a condição de funcionamento dos atuadores ---
void condicao_funcionamento()
{

////////////////////////////////////
////////////////////////////////////
// Definindo ações caso o reservatório de água esteja cheio ou vazio.
if(estado_do_nivel == 0)
{
    Serial.println("Reservatorio Cheio");
}
else
{
    Serial.println("Reservatorio Vazio");
}

////////////////////////////////////
////////////////////////////////////

```

```

    // Definindo ações caso a estufa esteja com umidade do solo
    diferente da faixa que foi definida.

    if((Umidade_Sensor_Higrometro < umidade_solo_valor &&
    estado_do_nivel == 0) )                // Caso a umidade do solo esteja
    menor que a definida e tenha agua no resevatorio.

    //if((Umidade_Sensor_Higrometro < umidade_solo_valor) && ((millis()
    - tempo_referencia_rega) > tempo_rega) )                // Caso a
    umidade do solo esteja menor que a definida.

    {
        digitalWrite(rele2_bomba_agua, LOW);                //
    Ligar a bomba para regar a planta.
    }

    if else (Umidade_Sensor_Higrometro >= 90)
    // senão
    {
        digitalWrite(rele2_bomba_agua, HIGH);
    // Desligar a bomba.
        tempo_referencia_rega = millis();
    }

    //////////////////////////////////////
    //////////////////////////////////////

    // Definindo ações caso a estufa esteja com umidade do ar diferente
    da faixa que foi definida.

    if(Umidade_Sensor_DTH11 >= umidade_ar_valor)
    // Caso a umidade do ar esteja maior que a definida.

    {
        digitalWrite(rele3_ventoinha_exaustao, LOW);                //
    Ligar as exaustão para diminuir a umidade.
    }

    else if (Umidade_Sensor_DTH11 <= (umidade_ar_valor - 2) )
    {
        digitalWrite(rele3_ventoinha_exaustao,HIGH);
    }

```

```

////////////////////////////////////
////////////////////////////////////

    // Definindo ações caso a estufa esteja com temperatura diferente
da faixa que foi definida.
    if(Temperaura_Sensor_DS18B20 >= temperatura_valor)
// Caso a temperatura esteja menor que a definida.
    {
        digitalWrite(rele_solido_lampada_incandescente, HIGH);
// Desligar a lâmpada incandescente.
        if (Temperaura_Sensor_DS18B20 > (temperatura_valor + 0.2))
        {
            digitalWrite(rele4_ventoinha_ventilacao, LOW);
// Ligar as ventoinha para esfriar a estufa.
        }
    }
    else if (Temperaura_Sensor_DS18B20 <= (temperatura_valor - 1) )
    {
        digitalWrite(rele_solido_lampada_incandescente,LOW);
        digitalWrite(rele4_ventoinha_ventilacao, HIGH);
// Desligar as ventoinha para esquentar a estufa.
    }

////////////////////////////////////
////////////////////////////////////

    // Definindo ações para iluminação das plantas coma a luz de LED.
    tempo_luz_atual = millis();

    if (lampada_ligada)
    {
        if (tempo_luz_atual - tempo_luz_anterior >= tempo_luz_ligada)
        {
            // Desliga a lâmpada
            digitalWrite(rele1_lampada_led, HIGH);
            lampada_ligada = false;
            tempo_luz_anterior = tempo_luz_atual;
        }
    }

```

```

    }    else
    {
        if (tempo_luz_atual - tempo_luz_anterior >=
tempo_luz_desligada)
        {
            // Liga a lâmpada
            digitalWrite(rele1_lampada_led, LOW);
            lampada_ligada = true;
            tempo_luz_anterior = tempo_luz_atual;
        }
    }

```

```

        digitalWrite(rele1_lampada_led, LOW);
    }

```

```

//
=====
// --- Desenvolvimento das Funções ---
void botoes()
{
    if(!digitalRead(botao_direita) && sub_menu == 1 &&
comparador_direito == 0)    /// Se botao_direita estiver em nível
logico baixo e sub_menu igual a 1 e
    {
        /// menu_num estiver a baixo do seu valor máximo "3", menu_num vai
adicionar mais "1"
        if(menu_num <= menu_max) menu_num += 1;
        /// assim trocando para outro menu_num.

        tempo_botao_direito = millis();
        /// "tempo_botao_direito" é atualizado com o valor do millis().
        comparador_direito = 1;
        /// "comparador_direito" troca seu valor logico.
    } //fim botao_direita

```

```

    if((millis() - tempo_botao_direito) > tempo_botoes_menu)
    /// Se o tempo mínimo de 1000 milissegundos tiver passado, o
    "comparador_direito"
    {
    /// vai trocar o valor logico. Permitindo que o botão seja pressionado
    de novo.
        comparador_direito = 0;
    }

    if(!digitalRead(botao_esquerda) && sub_menu == 1 &&
    comparador_esquerdo == 0)    /// Se botao_esquerda estiver em nível
    logico baixo e sub_menu igual a 1 e
    {
    /// menu_num estiver a acima do seu valor mínimo "0", menu_num vai
    subtrair "1"
        if(menu_num > 0) menu_num -= 1;

        tempo_botao_esquerdo = millis();
    // "tempo_botao_esquerdo" é atualizado com o valor do millis().
        comparador_esquerdo = 1;
    // "comparador_esquerdo" troca seu valor logico.
    } //fim botao_esquerda

    if((millis() - tempo_botao_esquerdo) > tempo_botoes_menu)
    /// Se o tempo mínimo de 1000 milissegundos tiver passado, o
    "comparador_esquerdo"
    {
    /// vai trocar o valor logico. Permitindo que o botão seja pressionado
    de novo.
        comparador_esquerdo = 0;
    }

    if(!digitalRead(botao_enter) && comparador_enter == 0)
    /// Se botao_enter estiver em nível logico baixo e sub_menu estiver a
    baixo do seu

```

```

{
  /// valor máximo "3", sub_menu vai adicionar mais "1" assim trocando
  para outro sub_menu.
    if(sub_menu <= 3) sub_menu += 1;

    tempo_botao_enter = millis();
  /// "tempo_botao_enter" é atualizado com o valor do millis().
    comparador_enter = 1;
  /// "comparador_enter" troca seu valor logico.
} //fim botao_enter

if((millis() - tempo_botao_enter) > tempo_botoes_menu)
  /// Se o tempo mínimo de 1000 milissegundos tiver passado, o
  "comparador_enter"
  {
    /// vai trocar o valor logico. Permitindo que o botão seja pressionado
    de novo.
      comparador_enter = 0;
  }

  if(!digitalRead(botao_voltar) && comparador_voltar == 0)
  /// Se botao_voltar estiver em nível logico baixo e sub_menu estiver a
  acima do seu
  {
    /// valor mínimo "0", sub_menu vai subtrair "1" assim trocando para
    outro sub_menu.
      if(sub_menu > 0) sub_menu -= 1;

      tempo_botao_voltar = millis();
    /// "tempo_botao_voltar" é atualizado com o valor do millis().
      comparador_voltar = 1;
    /// "comparador_voltar" troca seu valor logico.
  } //fim botao_voltar

  if((millis() - tempo_botao_voltar) > tempo_botoes_menu)
  /// Se o tempo mínimo de 1000 milissegundos tiver passado, o
  "comparador_voltar"

```

```

    {
    /// vai trocar o valor logico. Permitindo que o botão seja pressionado
    de novo.
        comparador_voltar = 0;
    }
} //fim botoes

//
=====
=====
// --- Ajuste da temperatura ---
void temperatura()
{
    switch(sub_menu)
    {
        case 1: //
Caso 1 na função temperatura, permite acesso a visualização e
configuração da temperatura.
            lcd.setCursor(0,0); //
Seta a posição em que o cursor vai inicializar (Linha 1).
            lcd.print("  Temperatura  >"); //
Escrever "Temperatura >" na linha 1.
            lcd.setCursor(0,1); //
Seta a posição em que o cursor vai inicializar (Linha 2).
            lcd.print("    "); //
Deixa em branco um espaço de 4 caracteres.
            lcd.print(Temperatura_Sensor_DS18B20); //
Escreve o valor da temperatura interna da estufa.
            lcd.print(" ");
            lcd.write(B11011111);
            lcd.print("C");
            lcd.print("    "); //
Deixa em branco um espaço de 4 caracteres.
            break;

        case 2: //
Caso 3 leva a função "ajustar_temperatura".
            ajustar_temperatura();

```

```

        break;
    } //fim switch
} //fim temperatura

void ajustar_temperatura()
{
    lcd.setCursor(0,0); //
    // Seta a posição em que o cursor vai inicializar (Linha 1).
    lcd.print("  Ajustar Temp: ");
    // Escrever "Ajustar Temp:" na linha 1
    lcd.setCursor(0,1); //
    // Seta a posição em que o cursor vai inicializar (Linha 2).
    lcd.print("    "); //
    // Deixa em branco um espaço de 4 caracteres.
    lcd.print(temperatura_valor); // Escreve
    // o valor da temperatura interna da estufa.
    lcd.print(" ");
    lcd.write(B11011111);
    lcd.print("C");
    lcd.print("    "); //
    // Deixa em branco um espaço de 4 caracteres.

    if(!digitalRead(botao_direita))
    temperatura_valor=(temperatura_valor+1); // Se "botao_direita"
    for precioso dentro da função "ajustar_temperatura", vamos
    incrementar "1" no valor da temperatura.

    if(!digitalRead(botao_esquerda))
    temperatura_valor=(temperatura_valor-1); // Se "botao_direita"
    for precioso dentro da função "ajustar_temperatura", vamos
    decrementar "1" no valor da temperatura.
} //fim ajustar_temperatura

//
=====
=====

// --- Ajuste da Umidade do Solo ---
void umidade_solo()

```



```

{
    switch(sub_menu)
    {
        case 1: //
            Caso 1 na função temperatura, permite acesso a visualização e
            configuração da temperatura.
            lcd.setCursor(0,0); //
            Seta a posição em que o cursor vai inicializar (Linha 1).
            lcd.print("< Umidade Solo >"); //
            Escrever "< Umidade Solo >" na linha 1.
            lcd.setCursor(0,1); //
            Seta a posição em que o cursor vai inicializar (Linha 2).
            lcd.print("    "); //
            Deixa em branco um espaço de 5 caracteres.
            lcd.print(Umidade_Sensor_Higrometro); //
            Escrever nada na linha 2.
            lcd.print(" %");
            lcd.print("    "); //
            Deixa em branco um espaço de 5 caracteres.
            break;

        case 2:
            ajustar_umidade_solo();
            break;
    } //fim switch sub_menu
} //fim umidade_solo

void ajustar_umidade_solo()
{
    lcd.setCursor(0,0); //
    Seta a posição em que o cursor vai inicializar (Linha 1).
    lcd.print("Ajust. Umidade: "); //
    Escrever "Ajust. Umidade:" na linha 1.
    lcd.setCursor(0,1); //
    Seta a posição em que o cursor vai inicializar (Linha 2).
    lcd.print("    "); //
    Deixa em branco um espaço de 6 caracteres.
    lcd.print(umidade_solo_valor); //
    Escreve o valor da Umidade Solo configurada/desejada na estufa.

```

```

        lcd.print(" %"); //
Escreve o simbolo da porcentagem.
        lcd.print(" "); //
Deixa em branco um espaço de 6 caracteres.

        if(!digitalRead(botao_direita))
umidade_solo_valor=(umidade_solo_valor+1); // Se
"botao_direita" for preciosnado dentro da função "umidade_solo_valor",
vamos incrementar "1" no valor da Umidade Solo.

        if(!digitalRead(botao_esquerda))
umidade_solo_valor=(umidade_solo_valor-1); // Se
"botao_direita" for preciosnado dentro da função "umidade_solo_valor",
vamos incrementar "1" no valor da Umidade Solo.
    } //fim ajustar_umidade_solo

//
=====
=====
// --- Ajuste da Umidade do Ar ---
void umidade()
{
    switch(sub_menu)
    {
        case 1:
            lcd.setCursor(0,0); //
Seta a posição em que o cursor vai inicializar (Linha 1).
            lcd.print("< Umidade do Ar"); //
Escrever "< Umidade do Ar" na linha 1.
            lcd.setCursor(0,1); //
Seta a posição em que o cursor vai inicializar (Linha 2).
            lcd.print(" "); //
Deixa em branco um espaço de 5 caracteres.
            lcd.print(Temperaura_Sensor_DS18B20); //
Escreve o valor da Umidade do Ar na estufa.
            lcd.print(" %");
            lcd.print(" ");
            break;

```

```

        case 2:
            ajustar_umidade();
            break;
    } //fim switch sub_menu
} //fim umidade

void ajustar_umidade()
{
    lcd.setCursor(0,0); //
    Seta a posição em que o cursor vai inicializar (Linha 1).
    lcd.print("Ajust. Umid. Ar:"); //
    Escrever "Ajust. Umid. Ar:" na linha 1.
    lcd.setCursor(0,1); //
    Seta a posição em que o cursor vai inicializar (Linha 2).
    lcd.print("      "); //
    Deixa em branco um espaço de 6 caracteres.
    lcd.print(umidade_ar_valor); //
    Escreve o valor da Umidade do Ar configurada/desejada na estufa.
    lcd.print(" %"); //
    Escreve o simbolo da porcentagem.
    lcd.print("      "); //
    Deixa em branco um espaço de 6 caracteres.

    if(!digitalRead(botao_direita))
    umidade_ar_valor=(umidade_ar_valor+1); // Se "botao_direita"
    for precioso dentro da função "umidade_ar_valor", vamos incrementar
    "1" no valor da Umidade do Ar.

    if(!digitalRead(botao_esquerda))
    umidade_ar_valor=(umidade_ar_valor-1); // Se "botao_direita"
    for precioso dentro da função "umidade_ar_valor", vamos incrementar
    "1" no valor da Umidade do Ar.
} //fim ajustar_umidade

void monitor_serial()
{
    Serial.println("----- Dados Sensores -----");

```

```

Serial.print("Temperatura Sensor DS18B20: ");
Serial.println(Temperatura_Sensor_DS18B20);
Serial.print("Umidade Sensor DTH11: ");
Serial.println(Umidade_Sensor_DTH11);
Serial.print("Temperatura Sensor DTH11: ");
Serial.println(Temperatura_Sensor_DTH11);
Serial.print("Umidade Sensor Higrometro: ");
Serial.println(Umidade_Sensor_Higrometro);
Serial.print("Umidade Sensor Higrometro Analogico: ");
Serial.println(Umidade_Sensor_Higrometro_Analogica);
Serial.print("Luminosidade LDR: ");
Serial.println(Luminosidade_Sensor_LDR);

Serial.print("Estado do Relé 1 (LAP. LED): ");
Serial.println(digitalRead(rele1_lampada_led));
Serial.print("Estado do Relé 2 (BOMBA ÁGUA): ");
Serial.println(digitalRead(rele2_bomba_agua));
Serial.print("Estado do Relé 3 (VENT. EXAU.): ");
Serial.println(digitalRead(rele3_ventoinha_exaustao));
Serial.print("Estado do Relé 4 (VENT. VENT.) : ");
Serial.println(digitalRead(rele4_ventoinha_ventilacao));
Serial.print("Estado do Relé 5 (VENT. INCAN.) : ");
Serial.println(digitalRead(rele_solido_lampada_incandescente));

Serial.print("Estado do Botão 1 (DIREITA): ");
Serial.println(digitalRead(botao_direita));
Serial.print("Estado do Botão 2 (ESQUERDA):");
Serial.println(digitalRead(botao_esquerda));
Serial.print("Estado do Botão 3 (ENTER): ");
Serial.println(digitalRead(botao_enter));
Serial.print("Estado do Botão 4 (VOLTAR) :");
Serial.println(digitalRead(botao_voltar));
}

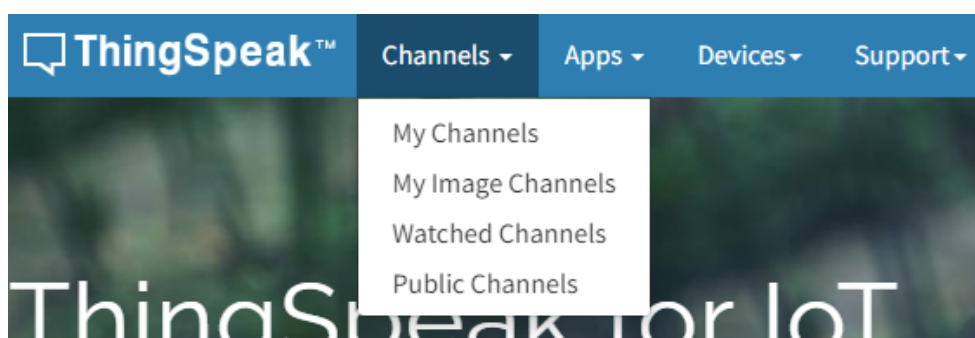
```

APÊNDICE 2 - COLETE DADOS EM UM NOVO CANAL

Este exemplo mostra como criar um novo canal para coletar dados analisados.

Crie um canal:

1. Faça login no ThingSpeak TM usando as credenciais da sua conta MathWorks [®] ou crie uma nova conta.
2. Clique em Canais > Meus Canais .



3. Na página Canais, clique em Novo canal .
4. Marque as caixas ao lado dos Campos 1–6. Insira estes valores de configuração de canal:

Nome :Dew Point Measurement
Campo 1 :Temperatura DS18B20
Campo 2 :Temperature DTH11
Campo 3 :Umidade DTH11
Campo 4 :Higrômetro
Campo 5 :Estado Nível Bomba
Campo 6 :Luminosidade Sensor LDR

New Channel

Name	Estufa - Pedro Palheta Neto (DS18B20/DHT11/Higrôme	
Description		
Field 1	Temperatura DS18B20	☒
Field 2	Temperature DTH11	☒
Field 3	Umidade DTH11	☒
Field 4	Higrômetro	☒
Field 5	Estado Nível Bomba	☒
Field 6	Luminosidade Sensor	☒
Field 7		☐

5. Clique em **Salvar canais** na parte inferior das configurações.

Agora você vê estas guias:

- Visualização privada** : esta guia exibe informações sobre seu canal que somente você pode ver.
- Visualização pública** : se você optar por disponibilizar seu canal publicamente, use esta guia para exibir os campos selecionados e as visualizações do canal.
- Configurações do canal** : esta guia mostra todas as opções de canal que você definiu na criação. Você pode editar, limpar ou excluir o canal nesta guia.
- Compartilhamento** : esta guia mostra as opções de compartilhamento do canal. Você pode definir um canal como privado, compartilhado com todos (público) ou compartilhado com usuários específicos.
- Chaves de API** : esta guia exibe as chaves de API do seu canal. Use as teclas para ler e escrever em seu canal.
- Importação/Exportação de dados** : esta guia permite importar e exportar dados do canal.

ANEXO 1 - DOCUMENTAÇÃO THINGSPEAK

OPÇÕES DE LICENÇA

ThingSpeak está disponível como um serviço gratuito para pequenos projetos não comerciais (<3 milhões de mensagens/ano ou ~8.200 mensagens/dia). Para projetos maiores ou aplicações comerciais, são oferecidos quatro tipos diferentes de licença anual: Standard, Academic, Student e Home. O ThingSpeak é comprado em unidades, onde uma unidade permite que 33 milhões de mensagens sejam processadas e armazenadas no período de um ano (~90.000 mensagens/dia). Uma unidade também oferece a capacidade de criar um número fixo de canais no ThingSpeak.

REQUISITOS DE SISTEMA

Dispositivos:

Os dados do sensor podem ser enviados para o ThingSpeak™ do Arduino®, Espressif ESP32 e ESP8266, Raspberry Pi™, BeagleBone Black e outros hardwares. Para que os dispositivos se comuniquem com o ThingSpeak, eles devem suportar os protocolos HTTP ou MQTT. Conexões seguras são recomendadas e exigem que seu dispositivo seja compatível com TLS 1.2. Seu firewall deve permitir a conexão com as portas padrão para esses protocolos. Explore informações e exemplos adicionais .

Local na rede Internet:

Para acessar o ThingSpeak por meio de um navegador da web, certifique-se de que seu sistema atenda aos seguintes requisitos.

Navegadores Suportados:

ThingSpeak é compatível com a maioria dos navegadores modernos executados em sistemas operacionais Windows®, Mac e Linux®, bem como Chromebooks.

Para navegadores (como Chrome e Firefox) que são atualizados automaticamente, a versão estável atual é suportada. Para outros navegadores, as versões suportadas são mostradas abaixo. Se você estiver usando uma versão mais antiga, deverá atualizar para a versão listada abaixo.

Para a melhor experiência geral, recomendamos o uso do Google Chrome.

Navegador	Versões suportadas
Google Chrome	Versão Atual

ThingSpeak é compatível com outros navegadores populares.

Navegador	Versões suportadas
Microsoft	Versão Atual
Mozilla Firefox	Versão Atual
Safári	Versão Atual

Configurações obrigatórias do navegador:

- Cookies ativados
- Pop-ups ativados
- JavaScript ativado

Conexão de internet:

Conexão de banda larga recomendada para uso interativo do site.