



# SISTEMA DE CONTROLE E MONITORIZAÇÃO REMOTO DE UMA ESTUFA

**ISABELA MELO DA SILVA**

julho de 2023

# SISTEMA DE CONTROLE E MONITORIZAÇÃO REMOTO DE UMA ESTUFA

Isabela Melo da Silva

Departamento de Engenharia Electrotécnica  
Mestrado em Engenharia Electrotécnica e de Computadores  
Área de Especialização em Automação e Sistemas



Relatório elaborado para satisfação parcial dos requisitos da Unidade Curricular de  
Tese/Dissertação do Mestrado em Engenharia Electrotécnica e de Computadores

Candidato: Isabela Melo da Silva, Nº 1210332, 1210332@isep.ipp.pt

Orientação científica: Ricardo Jorge Guedes da Silva Nunes da Costa, rjc@isep.ipp.pt

Coorientador: Victor Inácio de Oliveira, victor.inacio@mackenzie.br



Departamento de Engenharia Electrotécnica

Mestrado em Engenharia Electrotécnica e de Computadores

Área de Especialização em Automação e Sistemas

**2023**



“Quando o dedo aponta o céu, o idiota olha para o dedo.” - Amélie Poulain.

Esta citação serve como uma reflexão sobre a importância de olhar além das aparências e das convenções sociais, buscando compreender o verdadeiro significado por trás das coisas. Dedico esta dissertação a todos aqueles que têm a coragem de questionar, explorar e enxergar além do óbvio.



## *Agradecimentos*

Gostaria de agradecer a minha família, especialmente ao meu companheiro pela força e coragem que sempre me transmitiu, principalmente nos momentos mais sombrios. Não menos importante um agradecimento ao Professor Ricardo Jorge Guedes da Silva Nunes da Costa na qualidade de orientador e ao Professor Victor Inácio de Oliveira na qualidade de coorientador.



## *Resumo*

Este trabalho tem como objetivo o desenvolvimento de um sistema de controle e monitorização de dados de uma estufa, por isso foi desenvolvida uma estufa agrícola de modelo plano, onde é possível controlar a temperatura ambiente e a umidade do solo, também sendo possível verificar o nível de monóxido de carbono dentro da estufa e assim averiguar caso haja um incêndio e conferir se a planta cultivada está a receber a quantidade de horas de sol mínimas especificadas para o seu plantio.

Com o intuito de melhorar a segurança na transferência de dados entre o microcontrolador e o banco de dados, foi incrementada no sistema a funcionalidade de verificar o socket (endereço IP e porta de conexão) dos dispositivos que se conectarem ao sistema.

Para escrever o código de funcionamento dos sensores da estufa foi utilizada a plataforma Arduino que transfere os seus dados para um banco SQL localizado em um servidor e para a plataforma Cayenne, onde é possível visualizar os dados dos sensores e controlar os atuadores. O sistema de verificação do socket dos dispositivos foi realizado a partir do protocolo HTTP.

Com a finalização do projeto obteve-se um sistema que poderá ser implantado tanto em âmbito residencial, como em âmbito comercial, podendo ser aplicado em restaurantes que produzem os seus próprios alimentos, ou em larga escala, visando o agronegócio. Este sistema projetado automatizou parte do processo de cultivo de uma planta, trazendo conforto e segurança aos usuários. Para realizar a verificação do funcionamento do sistema foram realizados diversos testes com os sensores e atuadores separadamente e posteriormente em conjunto, para assegurar o correto funcionamento do circuito.

## *Palavras-Chave*

Estufas inteligentes, Controle remoto, automatização, cibersegurança, SQL.



## *Abstract*

This work aims at developing a control and monitoring system for greenhouse data. For this purpose, a flat model agricultural greenhouse was developed, which allows controlling the ambient temperature and soil humidity. It is also possible to monitor the level of carbon monoxide inside the greenhouse to detect potential fires and verify if the cultivated plant is receiving the specified minimum hours of sunlight for its growth.

In order to enhance data transfer security between the microcontroller and the database, the system was implemented with the functionality to verify the socket (IP address and connection port) of the devices connecting to it.

The Arduino platform was used to write the code for the greenhouse sensors, which transfer their data to a SQL database located on a server. The data can also be visualized and the actuators controlled through the Cayenne platform. The device socket verification system was implemented using the HTTP protocol.

Upon completion of the project, a system was obtained that can be deployed both in residential and commercial settings. It can be applied in restaurants that produce their own food or on a larger scale, targeting the agribusiness sector. This designed system automates part of the plant cultivation process, providing comfort and security to users. Various tests were conducted on the sensors and actuators individually and subsequently together to ensure the proper functioning of the circuit.

## *Keywords*

Smart greenhouses, Remote control, Automation, Cybersecurity, SQL.



## *Résumé*

Ce travail vise à développer un système de contrôle et de surveillance des données d'une serre. À cet effet, une serre agricole de modèle plat a été développée, permettant de contrôler la température ambiante et l'humidité du sol. Il est également possible de vérifier le niveau de monoxyde de carbone à l'intérieur de la serre pour détecter d'éventuels incendies et vérifier si la plante cultivée reçoit le nombre minimum d'heures d'ensoleillement spécifié pour sa croissance.

Afin d'améliorer la sécurité du transfert de données entre le microcontrôleur et la base de données, le système a été doté de la fonctionnalité de vérification du socket (adresse IP et port de connexion) des dispositifs se connectant au système.

La plateforme Arduino a été utilisée pour écrire le code des capteurs de la serre, qui transfèrent leurs données vers une base de données SQL située sur un serveur. Les données peuvent également être visualisées et les actionneurs contrôlés via la plateforme Cayenne. Le système de vérification du socket des dispositifs a été réalisé à l'aide du protocole HTTP.

À la fin du projet, un système a été obtenu qui peut être déployé à la fois dans des environnements résidentiels et commerciaux. Il peut être utilisé dans des restaurants produisant leur propre nourriture ou à plus grande échelle, en ciblant le secteur de l'agro-industrie. Ce système conçu automatise une partie du processus de culture des plantes, offrant confort et sécurité aux utilisateurs. Plusieurs tests ont été réalisés sur les capteurs et les actionneurs individuellement, puis en combinaison, pour garantir le bon fonctionnement du circuit..

## *Mots-clés*

Serres intelligentes, Contrôle à distance, Automatisation, Cybersécurité, SQL.



# Índice

AGRADECIMENTOS .....	I
RESUMO .....	III
PALAVRAS-CHAVE .....	III
ABSTRACT. ....	V
KEYWORDS .....	V
RESUME .....	VII
MOTS-CLES.....	VII
ÍNDICE.....	IX
ÍNDICE DE FIGURAS .....	XIII
ÍNDICE DE TABELAS .....	XVII
ACRÓNIMOS.....	XIX
<b>1. INTRODUÇÃO .....</b>	<b>23</b>
1.1.CONTEXTUALIZAÇÃO.....	25
1.2.OBJETIVOS.....	26
1.3.CALENDARIZAÇÃO .....	27
1.4.ORGANIZAÇÃO DO RELATÓRIO .....	27
<b>2. ESTUFAS AGRÍCOLAS – UM BREVE ESTADO DA ARTE.....</b>	<b>29</b>
2.1.IMPORTÂNCIA DA PRODUÇÃO EM ESTUFAS .....	30
2.2.TIPOS DE ESTUFAS EXISTENTES .....	31
2.2.1. <i>Modelo capela</i> .....	31
2.2.2. <i>Modelo arco</i> .....	32
2.2.3. <i>Modelo assimétrico</i> .....	33
2.2.4. <i>Modelo de encosto</i> .....	33
2.2.5. <i>Modelo plano</i> .....	34
2.3.CARACTERÍSTICAS DETERMINANTES PARA CULTIVO .....	35
<b>3. FUNCIONALIDADES DA ESTUFA A IMPLEMENTAR.....</b>	<b>37</b>
3.1.ESQUEMA FUNCIONAL.....	38
3.2.ENVIO DE DADOS .....	39
3.3.CONTROLE DA TEMPERATURA .....	39
3.4.MONITORIZAÇÃO DO NÍVEL DE MONÓXIDO DE CARBONO .....	39
3.5.CONTROLE DA UMIDADE DO SOLO .....	39
3.6.VERIFICAÇÃO DA LUMINOSIDADE.....	40

3.7. ATUADORES .....	40
3.8. BANCO DE DADOS .....	40
3.9. PÁGINA WEB.....	41
3.10. VERIFICAÇÃO DE NOVOS DISPOSITIVOS CONECTADOS AO MICROCONTROLADOR .....	41
<b>4. TECNOLOGIAS UTILIZADAS.....</b>	<b>42</b>
4.1. DIAGRAMA FUNCIONAL.....	43
4.2. HARDWARE .....	44
4.2.1. <i>Arduino</i> .....	44
4.2.2. <i>Plataforma Cayenne</i> .....	44
4.2.3. <i>Sensor de umidade e temperatura (DHT11)</i> .....	45
4.2.4. <i>Sensor de luminosidade (LDR)</i> .....	46
4.2.5. <i>Sensor de umidade do solo capacitivo (sku:sen0193)</i> .....	47
4.2.6. <i>Sensor de monóxido de carbono (MQ-9)</i> .....	48
4.2.7. <i>Microcontrolador</i> .....	48
4.2.8. <i>Cooler/Ventoinha</i> .....	50
4.2.9. <i>Bomba d'água</i> .....	51
4.3. SOFTWARE .....	51
CAMADA DE APLICAÇÃO: .....	51
4.3.1. <i>Protocolo HTTP</i> .....	52
4.3.2. <i>Protocolo MQTT</i> .....	52
4.3.3. <i>Segurança da informação</i> .....	52
4.3.4. <i>Criptografia</i> .....	53
4.3.5. <i>Tipos de ataques cibernéticos</i> .....	54
CAMADA DE TRANSPORTE: .....	55
4.3.6. <i>Protocolo TCP/IP</i> .....	55
CAMADA DE REDE: .....	56
4.3.7. <i>Protocolo IPv4</i> .....	56
4.3.8. <i>Protocolo 6LoWPAN</i> .....	56
CAMADA DE ENLACE: .....	56
4.3.9. <i>Tecnologias da comunicação</i> .....	56
<b>5. IMPLEMENTAÇÃO .....</b>	<b>60</b>
5.1. ARQUITETURA.....	62
5.1.1. <i>Configuração do software Arduino</i> .....	62
5.2. HARDWARE .....	66
5.3. SOFTWARE .....	68

<b>6. VERIFICAÇÃO EXPERIMENTAL.....</b>	<b>76</b>
6.1.METODOLOGIA.....	76
6.2.UMIDADE E TEMPERATURA DO AMBIENTE .....	78
6.3.MONÓXIDO DE CARBONO .....	79
6.4.UMIDADE DO SOLO .....	80
<b>7. CONCLUSÕES E TRABALHOS FUTUROS.....</b>	<b>83</b>
REFERÊNCIAS DOCUMENTAIS.....	85
ANEXO A. QUERY UTILIZADA NO SQL .....	89
ANEXO B. CÓDIGO EM C++ PARA O ARDUINO .....	91
ANEXO C. VÍDEO COM EXPLICAÇÕES SOBRE O PROJETO.....	107



## *Índice de Figuras*

<b>Figura 1</b>	Estufa no formato capela [Fonte]	32
<b>Figura 2</b>	Estufa com formato arqueado [11]	32
<b>Figura 3</b>	Estufa assimétrica [11]	33
<b>Figura 4</b>	Estufa do tipo de enconsto [13]	34
<b>Figura 5</b>	Estufa plana [Fonte]	35
<b>Figura 6</b>	Funcionamento da estufa [Autor]	38
<b>Figura 7</b>	Diagrama funcional [Autor]	43
<b>Figura 8</b>	Imagem ilustrativa da plataforma Cayenne [10]	45
<b>Figura 9</b>	Descrição da pinagem de um sensor DHT11 [Fonte]	46
<b>Figura 10</b>	Simbologia e aspecto de um LDR [37]	46
<b>Figura 11</b>	Conexão entre uma placa Arduino e um sensor sku:sen093 [Fonte]	47
<b>Figura 12</b>	Sensor MQ-9 e a descrição de seus componentes [Fonte]	48
<b>Figura 13</b>	ESP32 [Fonte]	50
<b>Figura 14</b>	Ventoinha [Fonte]	50
<b>Figura 15</b>	Mini bomba d'água [Fonte]	51
<b>Figura 16</b>	Datagrama com a definição do funcionamento dos sensores [Autor]	61
<b>Figura 17</b>	Menu arquivo [Autor]	63
<b>Figura 18</b>	Tela das preferências [Autor]	63

<b>Figura 19</b>	Menu sketch [Autor]	64
<b>Figura 20</b>	Biblioteca ESP32 [Autor]	64
<b>Figura 21</b>	Gerenciador de bibliotecas [Autor]	65
<b>Figura 22</b>	Definição de placas [Autor]	65
<b>Figura 23</b>	Desenho realizado no AutoCad Lite [Autor]	66
<b>Figura 24</b>	Estrutura da estufa [Autor]	67
<b>Figura 25</b>	Funcionamento das ventoinhas e da bomba em conjunto com os sensores DHT11 [Autor]	67
<b>Figura 26</b>	Na figura a) é apresentada a vista lateral da estufa, enquanto na figura b) é apresentada a vista superior [Autor]	68
<b>Figura 27</b>	Banco de dados no software phpMyAdmin [Autor]	69
<b>Figura 28</b>	Criação de tabela phpMyAdmin [Autor]	70
<b>Figura 29</b>	Dados no banco de dados [Autor]	72
<b>Figura 30</b>	Gráficos apresentados no Cayenne [Autor]	73
<b>Figura 31</b>	Na figura a) encontra-se a estrutura do banco de dados com as 4 tabelas presentes nele. Na figura b) tem-se as colunas presentes da tabela dht11 e na tabela ldr [Autor]	73
<b>Figura 32</b>	IP e Porta do cliente que está recebendo os dados da ESP32 [Autor]	74
<b>Figura 33</b>	Tela de monitorização e controle da estufa [Autor]	74
<b>Figura 34</b>	Característica da sensibilidade do sensor MQ9 [Fonte]	77
<b>Figura 35</b>	Na figura a) mostra-se a montagem do circuito com os sensores DHT11, enquanto na figura b), mostra-se a leitura realizada por eles [Autor]	78

- Figura 36** A figura a mostra a) leitura realizada pelo sensor MQ-9 e a figura b) mostra a montagem [Autor] 79
- Figura 37** A figura a) mostra os sensores de umidade do solo em um solo seco, a figura b) mostra os sensores num solo encharcado, enquanto a figura c) mostra os testes sendo realizados num solo umido [Autor] 81
- Figura 38** Botão para executar a query [Autor] 89
- Figura 39** Na imagem a) é apresentada a localização das ventoinhas, do sensor LDR e do sensor MQ-9, enquanto na imagem b) é apresentada a localização do sensor DHT11 [Autor] 107
- Figura 40** Na figura a) está sendo apresentada a localização da bomba d'água, enquanto na figura b) mostra-se a localização dos sensores de umidade de solo [Autor] 108
- Figura 41** Localização dos relés dos atuadores, do microcontrolador, dos sensores DHT11 internos e do módulo da fonte de 12 volts [Autor] 108



## *Índice de Tabelas*

Tabela 1	Calendarização do projeto [Autor]	28
Tabela 2	Característica Determinante Para o Cultivo [Fonte]	35
Tabela 3	Gastos com os materiais utilizados no projeto [Autor]	61



# Acrónimos

API	–	<i>Application Programming Interface</i>
BI	–	<i>Business Intelligence</i>
BLE	–	<i>Bluetooth Low Energy</i>
BR	–	<i>Basic Rate</i>
ATM	–	<i>Asynchronous Transfer Mode</i>
DDoS	–	<i>Distributed Denial of Service</i>
DoS	–	<i>Denial of Service</i>
EDR	–	<i>Enhanced Data Rate</i>
EXI	–	<i>Efficient XML Interchange</i>
GaAs	–	Arseniato de Gálio
HS	–	<i>Bluetooth High Speed</i>
IDE	–	<i>Integrated Development Environment</i>
IETF	–	<i>Internet Engineering Task Force</i>
IoT	–	<i>Internet of Things</i>
IPv4	–	<i>Internet Protocol version 4</i>
ITU	–	<i>International Telecommunication Union</i>
LGPD	–	Lei Geral de Proteção de Dados
LoRaWAN	–	<i>Long Range Wide Area Network</i>

- LTE – *Long Term Evolution*
- MITM – *Man in The Middle*
- MTU – *Maximum Transmit Unit*
- NFC – *Near Field Communication*
- ONU – *Organização das Nações Unidas*
- OWL – *Ontology Web Language*
- P&G – *Procter & Gamble*
- ppm – *Partículas por milhão*
- RDF – *Resource Description Framework*
- RFID – *Radio Frequency Identification*
- SDO – *Standards Development Organizations*
- SSID – *Service Set Identifier*
- UNB – *Ultra Narrow Band*
- WAF – *Web Application Firewall*





# 1. INTRODUÇÃO

No ano de 2022 a população mundial passou de 7 bilhões de pessoas para 8 bilhões, tendo decorrido 12 anos para que este fato ocorresse. A Organização das Nações Unidas (ONU) estima que a população levará cerca de 15 anos para alcançar os 9 bilhões [29].

A agricultura é uma das principais atividades praticadas pelo homem, sendo esta responsável por disponibilizar alimentos para a população, contudo, nações com menor desenvolvimento enfrentam problemas de insegurança alimentar, onde há a falta de alimentos para a população, sendo esta situação agravada por efeitos climáticos extremos, conflitos e choques econômicos, como é o caso de países como: Afeganistão, Haiti, Nigéria, Burquina Fasso, Mali, Somália, Sudão do Sul, Sudão e Iêmen que estão na lista de países com insegurança alimentar da ONU. Moçambique, apesar de não se encontrar na lista de países com insegurança alimentar, até março de 2023, tem registros pela ONU, de 3,1 milhões de pessoas que sofriam de insegurança alimentar aguda no território moçambicano [28].

Durante o processo de cultivo de um alimento, pode ocorrer uma alteração da qualidade e da quantidade dos alimentos colhidos nas lavouras, o que ocasiona uma alteração do valor das safras vendidas, afetando diretamente o preço cobrado dos consumidores. Quando a safra de um produto é ruim, ou quando ocorre algo que limite a quantidade deste produto nos mercados, o consumidor necessita gastar de acordo com a variação sofrida. Um exemplo destas variações é o preço de produtos lácteos que em junho de 2023 tiveram uma queda de 3,2% em seus preços em relação a abril do mesmo ano e o preço do açúcar que teve um aumento mensal consecutivo de 5,5% em seus preços em relação a abril, chegando a um valor 31% maior do que o mesmo período no ano de 2022 [30].

Uma tecnologia utilizada para aprimorar as técnicas de plantio e que auxilia no crescimento do setor, é a implementação do sistema de *Internet of Things (IoT)* para o monitoramento remoto, que pode ser utilizado para administrar o desempenho das máquinas, em tempo real e otimizar o consumo de combustível dos equipamentos, tendo assim um aumento na disponibilidade dos equipamentos e um ganho na eficiência [1]. Ao longo dos anos passou-se a implementar dentro da estrutura das estufas agrícolas sistemas de IoT responsáveis por verificar aspectos importantes para o desenvolvimento da planta sendo cultivada, como a umidade presente no solo, a temperatura do ambiente, ou o PH presente no solo.

Entre os oito eixos definidos no Plano Nacional de IoT, há um deles que define o uso eficiente de recursos naturais e insumos. Neste eixo, são apresentadas aplicações que, caso sejam implementadas, proporcionam um alto impacto à eficiência no plantio e na colheita. Algumas dessas aplicações são: as aplicações que geram alertas sobre potencial de chuvas e pragas, como as aplicações de monitoramento de clima por sensores ou mini estações; aplicações que apoiem a tomada de decisão na colheita, no plantio, no momento em que se deve retornar ao campo ou na quantidade de irrigação necessária. Outro tipo de solução importante é a aplicação para o monitoramento do solo para assim se aumentar a fertilidade do solo e a produção [1].

Para se incrementar uma rede de monitoramento remoto para realizar análises dos dados colhidos dos sensores dispostos numa estufa, exige-se um certo nível de infraestrutura,

conhecimento e dinheiro que pode viabilizar o projeto. Portanto, realizar a implementação de um sistema simples e de baixo custo seria ideal para suprir estas situações. Com isso, espera-se desenvolver um sistema de monitoramento via internet de dados extraídos de sensores e atuadores, presentes na estufa, que serão enviados para um servidor. Por isto, propõe-se integrar o microcontrolador ESP32 com três sensores de umidade e temperatura do ambiente (DHT11), um sensor de gás monóxido de carbono (mq9), um sensor de luminosidade (LDR), dois sensores de umidade do solo capacitivo (sku:sen0193), com dois coolers e uma bomba d'água, obtendo-se os dados de umidade e temperatura do ambiente, umidade do solo, nível de gases nocivos no ambiente e quantidade de luz absorvida pela planta e os enviando para o servidor online. Os dados coletados serão guardados numa base de dados com o objetivo de poder consultar estes dados posteriormente e se realizar um estudo mais aprimorado de determinadas características, como verificar se a planta cultivada recebeu a quantidade mínima de luminosidade ao longo do mês. Esta verificação seria realizada através de um gráfico realizado com os dados presentes no banco de dados.

## 1.1. CONTEXTUALIZAÇÃO

O IoT no setor do agronegócio atua como um poderoso mecanismo na colaboração ao controle de todos os níveis de produção da cadeia alimentar. Ele tem o papel de detecção e monitorização da produção, avaliando o desenvolvimento de culturas, o processamento de alimentos e o desempenho zootécnico, além de poder prever e precaver-se das condições meteorológicas, e auxiliar na descoberta de tratamentos para pragas e infecções [12].

Em relação ao ciclo de vida da informação (coleta, armazenamento, tratamento, compartilhamento e descarte), a Lei Geral de Proteção de Dados (LGPD), Lei nº 13.709/2018 [18], regula regras explícitas para o seu funcionamento. Visando uma melhor proteção para todos a LGPD propõe que sejam realizados procedimentos de mapeamento, entendimento e análise de vulnerabilidades de processos nas organizações,

classificando-os e mapeando-os no ambiente de segurança da informação para estar em conformidade com as normas [14].

Este projeto surgiu do desejo de realizar um trabalho voltado para a área de agricultura e de segurança da informação, para criar uma estufa com valor acessível e para fortalecer a segurança dos sistemas IoT e assim diminuir e prevenir a quantidade de ataques cibernéticos que ocorrem atualmente.

## 1.2. OBJETIVOS

O objetivo principal deste projeto é a aquisição dos dados provenientes de uma estufa IoT, a sua integração com um banco de dados e a visualização destes dados através de uma página web. Dada a complexidade inerente a este objetivo, sentiu-se a necessidade de o subdividir em múltiplas tarefas de realização mais simples, tais como:

- Pesquisar o que é IoT;
- Pesquisar tipos de estufas existentes;
- Levantar problemas de vulnerabilidade que podem ocorrer num sistema de IoT;
- Testar o funcionamento dos sensores e atuadores;
- Desenvolver o banco de dados;
- Desenvolver um código no software Arduino que seja possível realizar a leitura dos sensores, ligando os atuadores sempre que possível, que armazene os dados em um banco de dados e que possibilite visualizar os dados e a porta e o endereço IP do dispositivo que está recebendo os dados;
- Realizar testes para a verificar se o software está funcionando corretamente;
- Criar a parte física do sistema e integrar com o software.

### 1.3. CALENDARIZAÇÃO

Sendo à aquisição dos dados provenientes de uma estufa IoT, o controle e monitoramento da estufa a motivação deste trabalho, a sua prossecução conduziu à calendarização apresentada na Tabela 1. Esta inclui um conjunto de tarefas, como por exemplo: o estudo do que é um sistema IoT; alguns tipos de estufas existentes; o desenvolvimento de uma estufa IoT; teste e validação da solução proposta.

### 1.4. ORGANIZAÇÃO DO RELATÓRIO

O relatório é composto por sete capítulos, onde o capítulo um apresenta a introdução e contextualização do tema abordado, o capítulo dois apresenta um breve estado da arte sobre as estufas agrícolas, o capítulo três mostra as funcionalidades da estufa a ser implementada. O capítulo quatro mostra os principais conceitos e tecnologias correlacionadas com o tema abordado. No capítulo cinco, são reunidos os desenvolvimentos realizados e os resultados obtidos, enquanto no capítulo seis são apresentados os testes realizados para a verificação experimental, e por fim, no capítulo sete se encontram as principais conclusões do projeto e os trabalhos futuros.

**Tabela 1** Calendarização do projeto [Autor]

Fase	Atividade	ago/22	set/22	out/22	nov/22	dez/22	jan/23	fev/23	mar/23	abr/23	mai/23	jun/23
Pesquisa	O que é IoT	X										
Pesquisa	Tecnologias de Comunicação	X										
Pesquisa	Dispositivos utilizados em IoT	X										
Pesquisa	Definição dos sensores DHT11		X									
Pesquisa	Definição de microprocessador e do microprocessador ESP32		X									
Pesquisa	IoT no Agronegócio		X									
Pesquisa	Criptografia			X								
Pesquisa	Tipos de ataques cibernéticos			X								
Pesquisa	Estufas agrícolas			X								
Pesquisa	Tipos de estufas existentes			X								
Desenvolvimento	Desenvolvimento da parte de hardware do sistema IoT			X								
Desenvolvimento	Desenvolvimento do programa em				X	X						
Desenvolvimento	Conexão do programa com o Cayenne e com o banco de dados				X	X						
Desenvolvimento	Verificação dos acessos ocorridos					X	X					
Desenvolvimento	Confecção da parte estrutural da							X				
Desenvolvimento	Junção da parte física e da parte virtual da estufa							X	X			
Análise	Discussão dos Resultados									X		
Redação	Redação das Seções									X	X	
Revisão	Conclusão e escrita do projeto										X	X
Entrega	Envio do Projeto Final											X

## 2. ESTUFAS AGRÍCOLAS – UM BREVE ESTADO DA ARTE

Após a segunda guerra mundial a agricultura à base de estufas para fins comerciais teve um grande desenvolvimento, surgido no norte da Europa nas primeiras décadas do século 20. As estufas foram desenvolvidas com o intuito de ultrapassar os problemas de cultivo de espécies de plantas sensíveis ao frio, tendo se iniciado o cultivo em estufas em regiões com um inverno mais ameno, como a Bacia Mediterrânea, após o aparecimento do plástico nos anos 60. Como o uso de estufas permite o uso da água de uma forma racional, a produção de alimentos de forma protegida expandiu-se para alguns países onde a água utilizada para a agricultura é escassa [25].

Em 1980 a Petroquímica Triunfo introduziu no Brasil as estufas plásticas para difundir o uso do filme de polietileno de baixa densidade aditivado com anti-UV, o anti-UV é adicionado ao filme de polietileno com a aplicação de um composto químico ao material, este composto químico auxilia na proteção do material contra efeitos climáticos e aumenta a vida útil do filme polietileno. Foi utilizada madeira no primeiro modelo de estufa proposto para utilização do plástico, que tinha como qualidade, a durabilidade, podendo ser utilizado por até quatro anos, caso não tivesse danos físicos. Este modelo de estufa foi bastante difundido na região sul do Brasil, por causa da sua construção simples e rústica [15]. Nos subtópicos seguintes será abordada a importância da produção em estufas, alguns tipos de estufas existentes e algumas das características determinantes para o cultivo de hortaliças.

## 2.1. IMPORTÂNCIA DA PRODUÇÃO EM ESTUFAS

A necessidade de se obter diversos produtos durante o ano todo, a busca da melhoria da produtividade e da qualidade do produto e os consumidores terem se tornado mais exigentes, fez com que os cultivadores de plantas buscassem estruturas cobertas para se cultivar diversas culturas.

Nas estufas podem ser cultivadas espécies hortícolas e ornamentais, sendo as estufas estruturas com diferentes formas e tamanhos, cobertas com materiais transparentes, como plástico ou vidro, onde o ambiente é controlado. As estufas criam um clima artificial dentro de si, protegendo as plantas do vento e do frio enquanto elevam a temperatura e umidade, situação esta que proporciona o aumento da produção e a melhoria na qualidade [25].

Para proporcionar microclimas favoráveis é necessário implementar o cultivo em estufas ou em outros modos de cultivo em ambientes controlados. A produção fora de época de plantas ornamentais e de alimentos de alto valor em áreas de clima frio está associada às estufas, pois estas possibilitam que as culturas se desenvolvam em locais onde ao ar livre não seria possível [25].

## 2.2. TIPOS DE ESTUFAS EXISTENTES

Os tipos e modelos de estufas existentes variam dependendo do tipo de cultivo desejado e dos tipos de recursos utilizados, resultando em diversas classificações.

- Modo de operação;
- Tamanho e forma;
- Tipo de construção;
- Tecnologia utilizada no crescimento de plantas;
- E características técnicas [13].

O tamanho das estufas é dividido em pequeno, médio e grande, sendo este fator determinante para a escala de produção. A seguir são apresentadas algumas das formas das estufas mais utilizadas no Brasil:

### 2.2.1. MODELO CAPELA

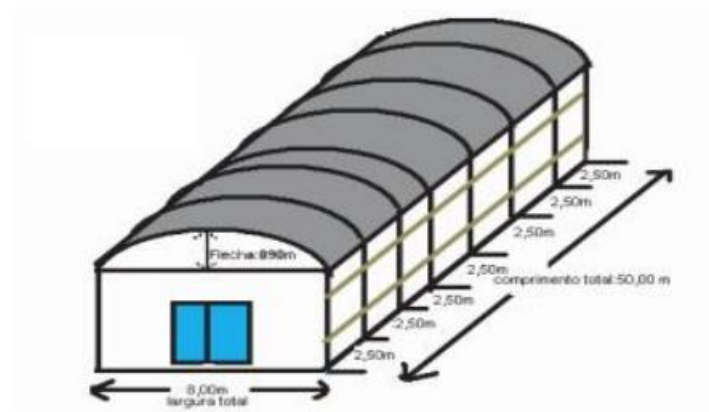
Apresenta o teto plano e simétrico, formado por duas águas. A inclinação do teto destas construções varia entre 20° e 30°, permitindo o escoamento da água da chuva. Os materiais mais utilizados em sua construção são a madeira, peças metálicas, ou o PVC, sendo utilizado filme plástico, vidro ou madeira para realizar a vedação [11]. Na Figura 1 é possível visualizar um exemplo do modelo de estufa.



**Figura 1** Estufa no formato capela [Fonte<sup>1</sup>]

### 2.2.2. MODELO ARCO

Este modelo de estufa suporta melhor o peso da precipitação, por causa de sua forma. Sendo normalmente coberto de folhas de polietileno, com o esqueleto composto por arcos de metal ou de PVC, Figura 2, tendo uma grande resistência a ventos fortes laterais [13].



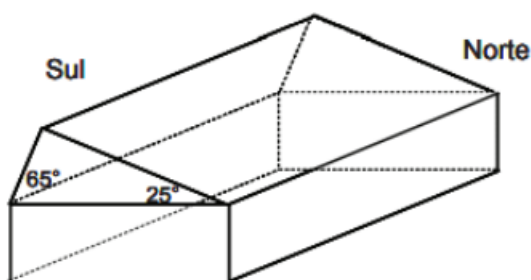
**Figura 2** Estufa com formato arqueado [11]

---

<sup>1</sup> <https://www.codevasf.gov.br/linhas-de-negocio/irrigacao/projetos-publicos-de-irrigacao/boletim-informativo-dos-projetos-da-codevasf/25a-edicao/estufa-em-formato-capela.jpg/view>

### 2.2.3. MODELO ASSIMÉTRICO

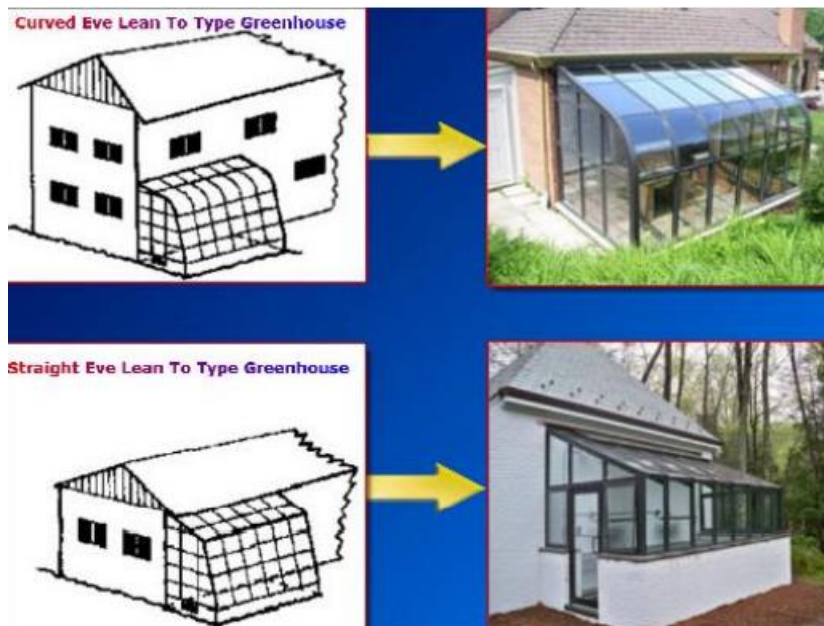
O teto deste modelo é plano e assimétrico, possuindo duas águas, como pode ser observado na Figura 3. Sua diferença em relação ao modelo capela é a inclinação de suas águas e o formato arquitetônico. Para que haja uma melhor condição de luminosidade no interior da estufa é necessário que o lado do telhado com menor inclinação esteja voltado para o norte. Este tipo de estufa é mais utilizado em regiões com altitudes elevadas e que apresentam uma grande variação de temperatura, como o sul do Brasil [11].



**Figura 3** Estufa assimétrica [11]

### 2.2.4. MODELO DE ENCOSTO

Este modelo de estufa é construído na lateral de um edifício, utilizando a estrutura do prédio como um dos seus lados, Figura 4. Esta é uma estrutura de baixo custo por necessitar de menos materiais para a sua construção, tendo como vantagem a proximidade a sistemas de água e de eletricidade. De desvantagens este modelo apresenta a limitação em seu tamanho, que é proporcional a altura da parede de suporte, recepção de luz limitada e dificuldades no controle da temperatura e ventilação do ambiente [13].



**Figura 4** Estufa do tipo de enconsto [13]

#### 2.2.5. MODELO PLANO

Este modelo é uma variação do modelo parral, que é utilizado na região de Almeria, na Espanha. Possui múltiplos vãos, uma inclinação menor da cobertura e sua estrutura de suporte para o filme plástico é constituída por uma malha de arame, Figura 5. Por causa de seu teto plano, a utilização deste modelo em regiões com alto índice de pluviosidade não é recomendada, pois o alto volume de chuvas pode acarretar em problemas quanto ao escoamento da água [11].



**Figura 5** Estufa plana [Fonte<sup>2</sup>]

### 2.3. CARACTERÍSTICAS DETERMINANTES PARA CULTIVO

Para determinar os limites de acionamento dos atuadores foram utilizados os dados obtidos pela Embrapa Hortaliças. Os dados são apresentados na Tabela 2, onde contém os dados de hortaliças que fazem parte do grupo das 50 espécies mais comercializadas no Brasil.

**Tabela 2** Característica Determinante Para o Cultivo [Fonte]

Hortaliça	Característica Determinante Para o Cultivo	Tempo de cultivo (dias)
Alho	No mínimo 13 horas de luz por dia.	150 - 180
Cebola	No mínimo 10 horas de luz por dia; Necessita de temperaturas altas.	120 - 180
Cebolinha	No máximo 25°C.	80 - 100
Coentro	Cultura de clima quente.	50 - 60

---

<sup>2</sup> <https://www.zanatta.com.br/cultivo-prottegido-em-estufas-agricolas/>

Tomate	Cultura de clima tropical; Não se dá bem com regiões muito úmidas e quentes.	100 - 120
Pimenta	Sensíveis a baixas temperaturas e intolerantes a geadas; São exigentes com o calor.	90 - 120
Salsa	Em torno de 20°C.	60 - 70
Rúcula	Melhor produção em temperaturas amenas (15° e 25° C).	40 - 60
Couve-chinesa	Melhor produção em temperaturas amenas (15° e 25° C).	60 - 70
Repolho	Melhor produção em temperaturas amenas (15° e 25° C).	90 - 110
Quiabo	Exige altas temperaturas; Não tolera o frio.	70 - 80
Pimentão	Melhor produção em temperaturas amenas (15° e 25° C).	100 - 120

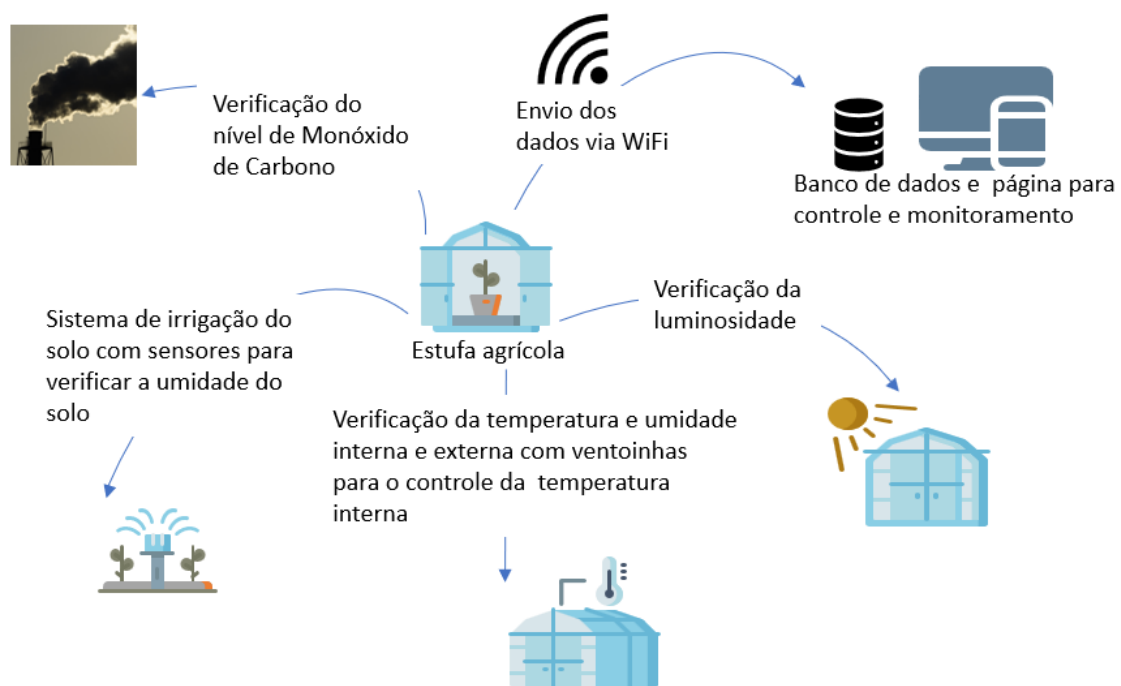
Neste capítulo pode-se ter uma breve noção sobre o que é uma estufa e saber alguns dos tipos de estufas existentes, sendo utilizado o modelo plano como exemplo para o modelo criado neste projeto. As especificações da estufa a ser desenvolvida serão descritas no capítulo a seguir.

# 3. FUNCIONALIDADES DA ESTUFA A IMPLEMENTAR

Neste capítulo será explicada as funcionalidades abrangidas pela estufa a ser implementada, como o esquema funcional do projeto, o envio de dados, o controle de temperatura e umidade do ambiente da estufa a ser empregado, a monitorização do nível de monóxido de carbono dentro da estufa, o controle da umidade do solo, os atuadores utilizados, o banco de dados, a página web e a verificação de novos dispositivos conectados ao microcontrolador.

### 3.1. ESQUEMA FUNCIONAL

O esquema funcional foi criado para exemplificar de forma sucinta o funcionamento da estufa. O projeto a ser desenvolvido possuirá um sistema de irrigação, onde sensores de umidade do solo coletarão o nível de umidade presente na terra em volta da planta e caso o valor lido esteja abaixo do esperado, a bomba d'água será acionada. Já o controle da temperatura do ambiente será realizado através de sensores de umidade e temperatura que coletarão os dados lidos e caso a temperatura estiver maior do que a temperatura especificada para a planta sendo cultivada, será acionada as ventoinhas. No projeto há também a verificação da quantidade de monóxido de carbono existente dentro da estufa e a monitorização da luminosidade que a estufa está recebendo, podendo ser implementado um sistema que controla o nível de luz que a estufa recebe, contudo, no projeto apresentado não consta este controle da luminosidade. Na Figura 6 pode ser visto o funcionamento da estufa a ser implementada.



**Figura 6** Funcionamento da estufa [Autor]

## 3.2. ENVIO DE DADOS

Todos os dados coletados pelo microcontrolador serão enviados para um banco de dados localizado em um servidor e para uma plataforma online, onde será possível monitorar o funcionamento dos sensores e controlar os atuadores. O envio dos dados será realizado através da rede Wi-Fi, pois este projeto tem como objetivo implementar uma estufa que possa ser utilizada em uma residência, em um comércio que produz as suas hortaliças, ou em estufas voltadas para o agronegócio.

## 3.3. CONTROLE DA TEMPERATURA

Neste projeto os sensores de umidade e temperatura do ambiente serão responsáveis por coletar os dados de temperatura e umidade do ambiente dentro da estufa, realizar uma leitura do valor médio lido entre os dois sensores internos, para verificar se algum destes sensores está com problema e verificar se os dados lidos no ambiente estão dentro do valor de temperatura esperado, caso o valor esteja maior, são acionadas as ventoinhas.

## 3.4. MONITORIZAÇÃO DO NÍVEL DE MONÓXIDO DE CARBONO

O sensor de gás verifica constantemente o nível de gases nocivos (monóxido de carbono) no ambiente para informar caso ocorra um incêndio e ativar um indicador no software Cayenne, sinalizando que o sistema de água foi acionado para apagar o fogo.

## 3.5. CONTROLE DA UMIDADE DO SOLO

Os sensores de umidade de solo são responsáveis por verificar se a planta necessita de mais água, ou se ela está com mais água do que o necessário. Sendo acionada a bomba

d'água caso o solo da planta fique mais seco do que o recomendado para a planta a ser cultivada.

### **3.6. VERIFICAÇÃO DA LUMINOSIDADE**

O sensor de luminosidade capta a quantidade de luz que a planta recebe e armazena no banco de dados se a planta está a receber luz ou não, esta informação é coletada para ser realizada a conferência se a planta está recebendo luz o suficiente, ou se será necessário implementar algum estímulo artificial.

### **3.7. ATUADORES**

As ventoinhas são responsáveis por resfriar a estufa, caso ela fique com uma temperatura acima da recomendada para a planta sendo cultivada, a bomba d'água umedecerá a planta quando ela necessitar. Os atuadores só serão acionados ou desligados pela página web por 200 milissegundos, pois a estufa foi programada para ajustar os seus parâmetros automaticamente, podendo ocasionar algum dano na planta cultivada caso os atuadores permaneçam ligados ou desligados por muito tempo sem seguir as especificações pré-determinadas.

### **3.8. BANCO DE DADOS**

Após ser realizada a coleta dos dados, eles são enviados para o banco de dados MySQL, que está presente em um servidor online. Neste banco de dados poder-se-á visualizar todo o histórico de dados da estufa; os dados serão armazenados para serem analisados aspectos específicos do cultivo. Para se analisar os dados coletados e armazenados no banco de dados é necessário que a pessoa, ao manuseá-los, possua um conhecimento prévio do significado dos dados sendo lidos pelos sensores, para assim extrair os dados do banco de dados e criar gráficos por meio da ferramenta Microsoft Excel. Para a criação do

banco de dados foram utilizadas as queries, que são a linguagem de programação utilizadas dentro de bancos de dados MySQL.

### **3.9. PÁGINA WEB**

Os dados lidos pelos sensores serão enviados para uma página web, sendo esta uma página localizada dentro de uma plataforma específica para o controle de sensores e atuadores, onde será possível acompanhar a evolução dos dados por interfaces gráficas e interagir com os atuadores, podendo ligar ou desligar a bomba e a ventoinha.

### **3.10. VERIFICAÇÃO DE NOVOS DISPOSITIVOS CONECTADOS AO MICROCONTROLADOR**

Para se obter uma maior segurança relacionada aos dados enviados pelo microcontrolador, acrescentou-se ao código um trecho responsável por realizar a verificação da porta e do endereço IP para o qual os dados estão sendo enviados, o “cliente” conectado ao microcontrolador, permitindo assim que seja visualizado caso haja uma invasão na rede e caso esteja ocorrendo algum tipo de ataque cibernético. Para adquirir a porta e o IP de um dispositivo novo que se conecte ao microcontrolador foi utilizado o protocolo HTTP, por meio de uma requisição HTTP, de onde são extraídas as informações desejadas.

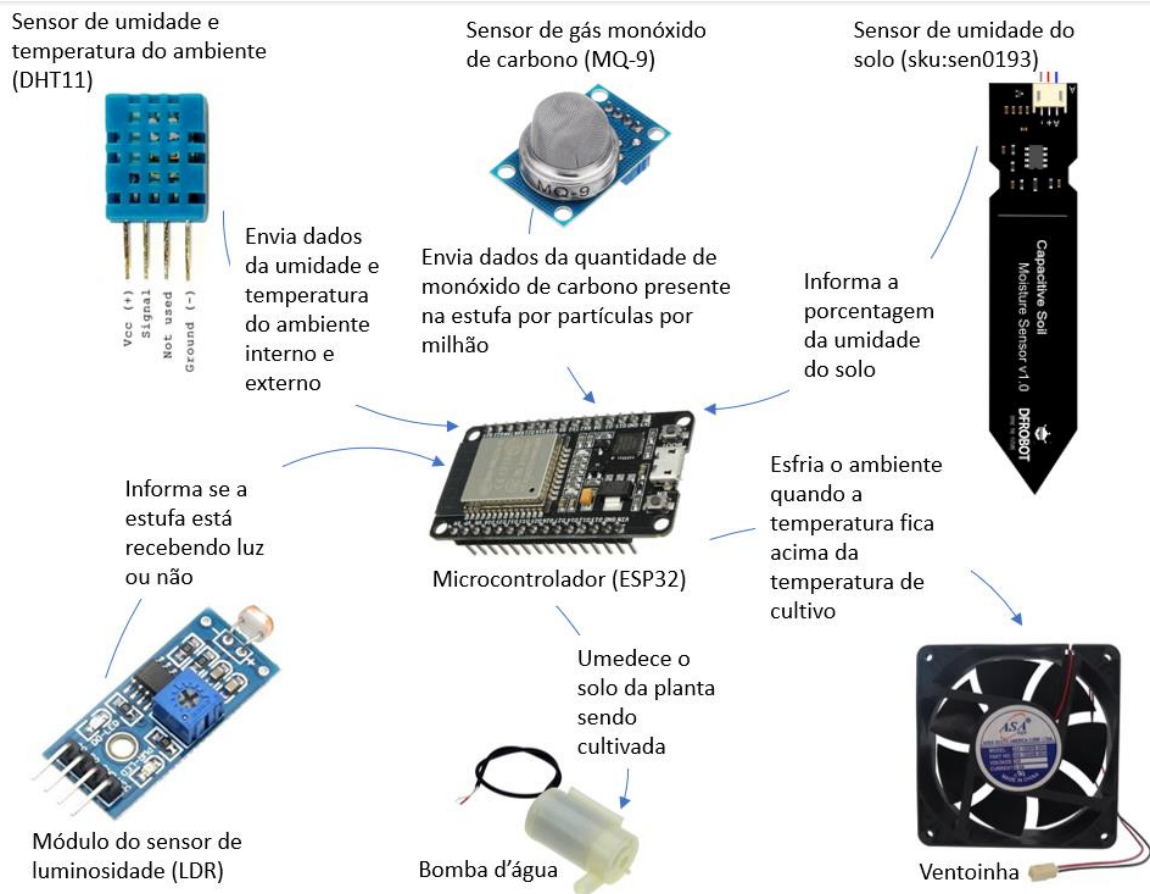
Para implementar as funcionalidades desejadas no projeto é necessário conhecer as tecnologias e conceitos utilizados, por isso o próximo capítulo abordará as tecnologias e conceitos utilizados na criação desta estufa.

## 4. TECNOLOGIAS UTILIZADAS

Neste tópico serão analisadas as tecnologias e conceitos utilizados no projeto para a parte de hardware e de software, como também será apresentado um diagrama funcional do projeto. Alguns dos temas abordados nos tópicos subjacentes são: microcontroladores, tecnologias da comunicação, os atuadores e sensores, os protocolos TCP/IP, HTTP e MQTT, a plataforma de desenvolvimento utilizada e os tipos de ataques cibernéticos, e serão esclarecidos alguns conceitos, como segurança da informação e criptografia.

## 4.1. DIAGRAMA FUNCIONAL

No diagrama funcional, mostrado na Figura 7, são apresentados todos os sensores e atuadores empregados neste projeto. Sendo descrita a função dos sensores de temperatura e umidade, que informam quando a temperatura lida ultrapassou o valor da temperatura estipulada para o cultivo da planta; a função do sensor de umidade do solo, que informa quando o solo está com o nível de água abaixo do recomendado para a planta sendo cultivada; o objetivo do sensor de luminosidade, que capta se a estufa está recebendo luz ou não e armazena essa informação no banco de dados e na página web; a funcionalidade do sensor de gás, que verifica o nível de monóxido de carbono no ambiente, para alertar as pessoas caso haja um incêndio; e os atuadores que são acionados conforme a necessidade.



**Figura 7** Diagrama funcional [Autor]

O conceito *Internet of Things* (IoT) é utilizado quando dispositivos se conectam à internet e trocam informações entre si para realizar um melhor controle do sistema que fazem parte. Por isto, é cabível de se dizer que este conceito foi utilizado dentro deste projeto,

pois no sistema a ser desenvolvido os atuadores são controlados por meio dos valores lidos pelos sensores, sendo armazenados os seus resultados em um banco de dados e numa página web através da utilização da internet.

## 4.2. HARDWARE

Na parte do hardware foi necessário estudar a plataforma Arduino, a plataforma Cayenne, os sensores de umidade e temperatura, o sensor de luminosidade, os sensores de umidade do solo capacitivo, o sensor do gás monóxido de carbono, os microcontroladores, as ventoinhas e a bomba d'água.

### 4.2.1. ARDUINO

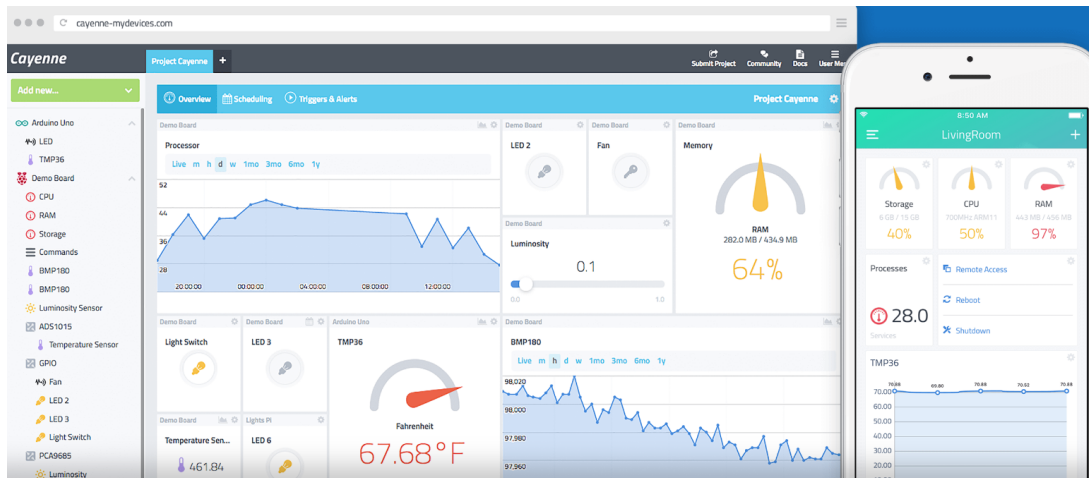
O Arduino é uma plataforma eletrônica de código aberto (não é necessário uma licença para utilizar a plataforma), onde é possível ao usuário desenvolver objetos interativos independentes, ou conectados a softwares do computador [2]. Para programar na plataforma Arduino é necessário a utilização de uma placa simples de entrada/saída, há diversos modelos de placas que podem ser utilizadas no Arduino, sendo cada uma específica para um tipo de utilização. Essas placas podem ser montadas manualmente, ou compradas pré-montadas no próprio site do Arduino ou em outras interfaces. Para programar na plataforma Arduino é utilizada a linguagem de programação padrão C++ [6].

### 4.2.2. PLATAFORMA CAYENNE

Na plataforma Cayenne é possível criar um painel personalizado com widgets de arrastar e soltar para visualizar, gerenciar e controlar os dispositivos conectados. Os widgets são utilizados para visualizar os dados, status e ações dos dispositivos. Cada dispositivo, sensor e atuador adicionado ao Cayenne possui um ou mais widgets associados, dependendo dos recursos de hardware.

Os projetos realizados na plataforma Cayenne podem ser acessados a partir de uma página web, ao acessar sua conta Cayenne, ou através de um aplicativo para celular, Figura 8. É possível conectar os dispositivos ao Cayenne para realizarem o envio dos

dados via MQTT, sendo possível também conectar dispositivos para realizarem a sua transmissão dos dados através da rede LoRaWan [10].

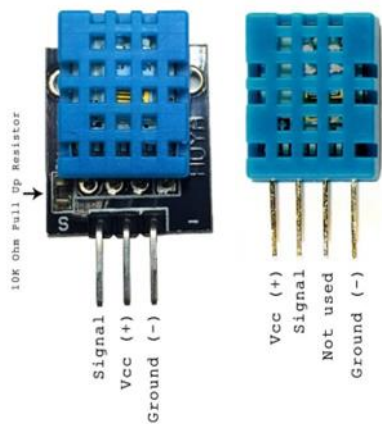


**Figura 8** Imagem ilustrativa da plataforma Cayenne [10]

O Cayenne suporta vários tipos de dados, incluindo dados de proximidade, temperatura, luminosidade e pressão. Os tipos de dados e a unidade de medida podem ser configurados no painel ou definidos diretamente ao enviar os dados [10].

#### 4.2.3. SENSOR DE UMIDADE E TEMPERATURA (DHT11)

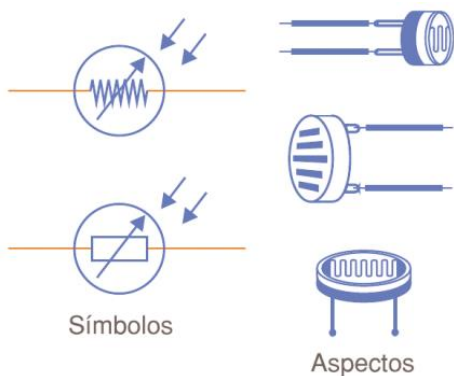
O sensor DHT11 é utilizado para se realizar medições de temperatura e umidade, tendo como resposta de saída um sinal digital. Possui um microcontrolador de 8 bits, um elemento resistivo do tipo NTC, este resistor é responsável por realizar a medição da temperatura. Esse aparelho mede, a cada 2 segundos, a temperatura na faixa de 0 a 50°C e a umidade é medida em porcentagem, sendo que a sua faixa de medição varia de 20 a 80%.



**Figura 9** Descrição da pinagem de um sensor DHT11 [Fonte<sup>3</sup>]

É um sensor de baixo custo em que a tensão de alimentação pode ser de 3 a 5 V, possui uma corrente durante as medições dos dados de 2,5 mA, com precisão de 5% para as medições de umidade e pode variar 2°C para as medições de temperatura. Dispõe de 15,5 mm de altura, 12 mm de largura e 5,5 mm de profundidade e quatro pinos onde o primeiro da esquerda para a direita é onde deve se conectar a alimentação (Vcc), o segundo é por onde será transmitido a informação (Signal), o terceiro pino não será utilizado e o último é ligado ao terra (Ground) [34], a disposição dos pinos pode ser visualizada na Figura 9.

#### 4.2.4. SENSOR DE LUMINOSIDADE (LDR)



**Figura 10** Simbologia e aspecto de um LDR [37]

O sensor LDR é um fotodetector mais simples existente, também chamado de célula ou dispositivo de fotocondução. A fotocondutividade é um fenômeno que ocorre quando a intensidade de luz sobre um material é alterada, ocasionando na variação da condutividade do material [35]. A Figura 10 mostra as simbologias para representar o

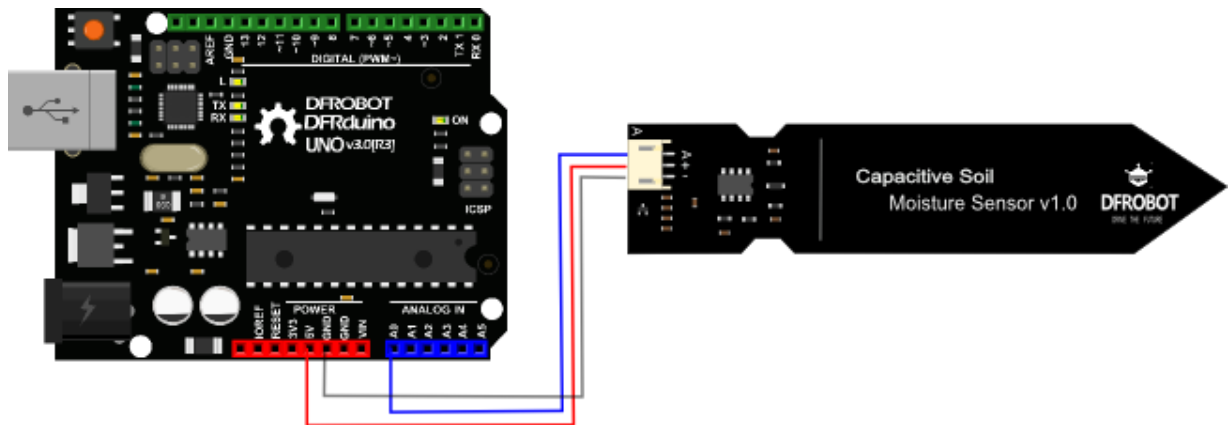
<sup>3</sup> <https://capsistema.com.br/index.php/2020/03/08/guia-de-como-instalar-o-sensor-de-umidade-dht11-em-um-arduino/>

LDR e alguns dos aspectos físicos do LDR.

O LDR é um resistor dependente da luz, agindo em função da quantidade de luz incidente na sua superfície. O Arsenieto de Gálio (GaAs) compõe a superfície sensível do LDR, este material altera a sua resistência de acordo com a incidência de luz, isto ocorre, pois, as partículas de luz liberam elétrons do material, causando a alteração da sua resistividade. Como estas são grandezas inversamente proporcionais, quanto menor a resistência existente no LDR, maior é a liberação de elétrons [37].

#### 4.2.5. SENSOR DE UMIDADE DO SOLO CAPACITIVO (SKU:SEN0193)

O sensor SKU:SEN0193 é um sensor de umidade do solo capacitivo de baixo custo facilmente adquirido no mercado, com dimensões de 9,8 x 2,3 cm. O sensor pode ser facilmente integrado a microcontroladores de código aberto, como Arduino, ESP32 e outros. O datasheet do fabricante informa que o sensor opera na faixa de tensão entre 3,3 e 5,0 V, com tensão de saída entre 0,0 e 3,0 V, com corrente de operação de 5 mA e peso de 15 g [32]. A Figura 11 demonstra como é realizada a conexão do sensor SKU:SEN093 com uma placa Arduino UNO.



**Figura 11** Conexão entre uma placa Arduino e um sensor sku:sen093 [Fonte<sup>4</sup>]

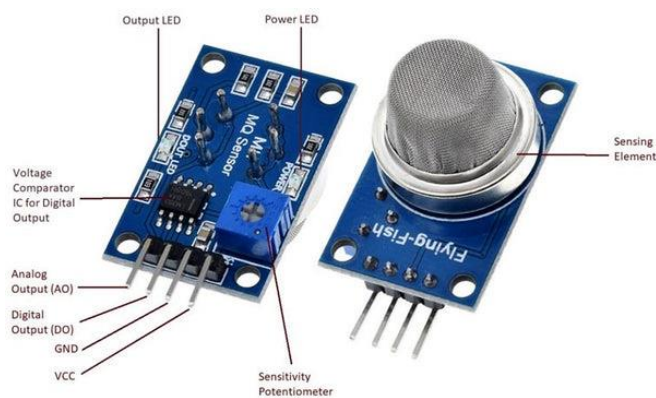
---

<sup>4</sup> [https://wiki.dfrobot.com/Capacitive\\_Soil\\_Moisture\\_Sensor\\_SKU\\_SEN0193](https://wiki.dfrobot.com/Capacitive_Soil_Moisture_Sensor_SKU_SEN0193)

Foi realizada a caracterização elétrica do sensor SKU:SEN0193 e observaram que a capacitância do sensor no ar era da ordem de 6,5 pF, enquanto a frequência do sensor variava de 260 Hz em um solo úmido a 520 Hz em um solo com baixa umidade [32].

#### 4.2.6. SENSOR DE MONÓXIDO DE CARBONO (MQ-9)

O MQ-9 é um sensor que possui a capacidade de verificar o nível de determinados gases existentes no ambiente, seu encapsulamento possui 6 pinos, dos quais 4 são utilizados para obter sinais e os outros 2 para fornecer corrente de aquecimento. O sensor funciona com tensões entre 5V e 12V CA ou CC e suas vantagens são: boa sensibilidade a CO/gás combustível; alta sensibilidade a metano, propano e CO; longa vida útil e baixo custo; e possui um circuito de acionamento simples.



**Figura 12** Sensor MQ-9 e a descrição de seus componentes [Fonte<sup>5</sup>]

Uma vez alimentado, a saída do sensor normalmente está em nível HIGH, mas fica em nível LOW quando o gás é detectado [17]. Na Figura 12 pode-se ver o sensor MQ-9 e a descrição dos seus componentes, como a descrição de cada pino e o potenciômetro de sensibilidade do sensor.

#### 4.2.7. MICROCONTROLADOR

O microcontrolador possui as partes básicas de um microcomputador, como microprocessador, memórias não-voláteis e voláteis e portas de entrada e saída. É um

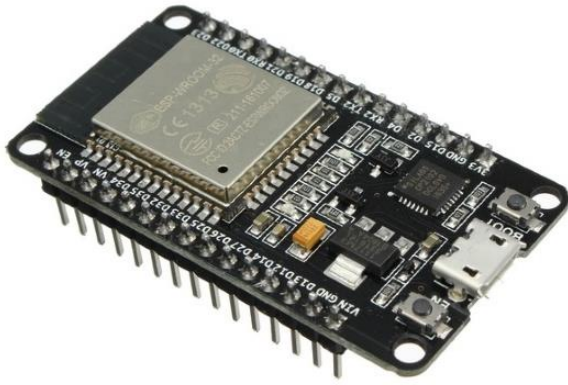
---

<sup>5</sup> <https://quartzcomponents.com/products/mq-9-gas-sensor-module-for-carbon-monoxide-methane-and-lpg>

circuito semiconductor em forma de circuito integrado que, normalmente, é limitado na quantidade de memória, sendo utilizado em aplicações que não necessitam armazenar grandes quantidades de dados, como automação residencial, automação predial, automação industrial e automação embarcada [35].

O ambiente Arduino UNO possui um baixo custo e fácil configuração, sendo muito utilizado em prototipagens. O Arduino é um microcontrolador customizável que possui um ambiente de desenvolvimento integrado *Integrated Development Environment (IDE)*, onde é possível escrever o seu próprio software, realizando a programação em uma linguagem baseada em C++, sem a necessidade da utilização de um firmware [39]. Por isto, Oliveira [26] afirma que “A maior vantagem desse processo é garantir maior poder de processamento, visto que o processador ficará totalmente dedicado ao programa.”

Para controlar outros microcontroladores pode-se utilizar o Arduino, como é o caso do módulo ESP32, que possibilita a comunicação via Bluetooth e Wi-Fi do sistema com um servidor, por exemplo, podendo ser enviado dados para a monitorização. Por isto o módulo ESP32 pode ser utilizado para aplicações de irrigação inteligente, devido à facilidade de configuração, manutenção e baixo custo.



**Figura 13** ESP32 [Fonte<sup>6</sup>]

O módulo ESP32 possui um microprocessador Xtensa Dual-Core 32-bit LX6, uma memória Flash programável de 4 MB, uma memória RAM de 520 KBytes, uma memória ROM de 448 KBytes, um clock que alcança uma frequência de até 240 MHz, 25 pinos digitais, um módulo Wireless 802.11 o qual utiliza uma frequência de 2,4 GHz, que é responsável pelo acesso da placa a internet, um módulo Bluetooth Low Energy padrão 4.2 e a tensão de alimentação pode ser de 4,5 a 9 V [33]. Na Figura 13 pode ser visto o modelo da ESP- WROOM-32, que é o modelo utilizado no projeto.

#### 4.2.8. COOLER/VENTOINHA



**Figura 14** Ventoinha [Fonte<sup>7</sup>]

Para este projeto foram utilizadas duas ventoinhas alimentadas por uma tensão 12 V DC, sendo uma ventoinha de 8cm x 8cm x 2,5cm e a outra de 6cm x 6cm x 2,5cm. Na Figura 14 é possível visualizar uma ventoinha.

---

<sup>6</sup> <https://www.makerhero.com/produto/modulo-wifi-esp32-bluetooth/>

#### 4.2.9. BOMBA D'ÁGUA



Neste projeto será utilizada uma mini bomba d'água que funciona com tensão de 3 a 6 V, com peso de 26g e 33mm de altura. Sendo esta uma bomba d'água que funciona submersa na água. Na Figura 15 pode-se ver uma mini bomba d'água.

**Figura 15** Mini bomba d'água  
[Fonte<sup>8</sup>]

### 4.3. SOFTWARE

Para um melhor entendimento das tecnologias e conceitos utilizados dividiu-se este tópico de acordo com as camadas presente no modelo OSI.

O modelo OSI é um modelo de rede de computador utilizado como padrão para os protocolos de comunicação, para assim ser possível a comunicação entre sistemas computacionais distintos. O modelo OSI é dividido em sete camadas: Aplicação; Apresentação; Sessão; Transporte; Rede; Enlace; e Física. Neste tópico serão abordados apenas conceitos presentes nas camadas de: Aplicação; Transporte; Rede; e Enlace.

Camada de Aplicação:

---

<sup>7</sup> [https://www.jelcoolers.com.br/MLB-1067774695-ventilador-cooler-ventoinha-adda-120x120x38-24v-rol-3000rpm-\\_JM](https://www.jelcoolers.com.br/MLB-1067774695-ventilador-cooler-ventoinha-adda-120x120x38-24v-rol-3000rpm-_JM)

<sup>8</sup> <https://www.electronicabyp.com.ar/producto/mini-bomba-de-agua-sumergible-2l-m-3v-5v-dc-arduino/>

A camada de aplicação é responsável por controlar o desenvolvimento de aplicações que operam em diversos sistemas finais e os seus protocolos de atuação.

#### 4.3.1. PROTOCOLO HTTP

O protocolo HTTP, Hyper Text Transfer Protocol, é um protocolo de comunicação da camada de aplicação que baseia-se na transmissão de documentos hipertexto utilizando o protocolo TCP/IP.

O HTTP utiliza a conexão cliente e servidor para realizar a requisição e transferências dos dados, na mensagem enviada para solicitar a informação pode conter alguns comandos, como o GET, que solicita um recurso, o POST, que envia as informações para o processamento e o HEAD, que solicita uma informação, que nem o comando HEAD, entretanto, não possui um corpo de mensagem [20].

#### 4.3.2. PROTOCOLO MQTT

O protocolo MQTT é um protocolo de mensagens que oferece suporte a mensagens entre dispositivos para a nuvem e da nuvem para o dispositivo, sendo um protocolo baseado em conjunto de regras ou padrões. O protocolo SSL é utilizado para proteger os dados sigilosos na comunicação MQTT.

O protocolo MQTT, para não ocorrer a conversa direta entre o destinatário e o remetente, utiliza um padrão de publicação/assinatura, onde um terceiro componente, denominado agente de mensagens, realiza a comunicação entre publicadores (remetente) e assinantes (destinatário). O agente é responsável por filtrar e distribuir todas as mensagens recebidas dos publicadores para os assinantes corretamente [3].

#### 4.3.3. SEGURANÇA DA INFORMAÇÃO

Para auxiliar na proteção das aplicações IoT algumas empresas criaram e disponibilizaram recursos para os indivíduos utilizarem. A Intel disponibilizou recursos que investem em quatro categorias essenciais da SI, a integridade da plataforma, que torna menor a ocorrência de adulterações por meio do aproveitamento da raiz do hardware e da proteção fundamentada na confiança dos dados críticos, código e de firmware da plataforma; a proteção aprimorada para dados, ID e chaves, que para diminuir as

modificações que ocorrem nos dados, chaves e IDs proporciona opções integradas e discretas para criação de credenciais; e também é oferecida a geração de chaves seguras e a aceleração da criptografia para melhorar a eficiência e o desempenho geral, garantindo assim que a execução ocorra de forma confiável e segura [9].

O Google Cloud Internet of Things Core foi projetado para conectar e gerenciar os dispositivos IoT de forma segura. Dentro deste serviço do Google está implementado o Cloud Armor, que é um serviço construído para proteger as aplicações IoT de ataques DDoS, ataques de rede distribuídos e filtrar as solicitações recebidas pela web por um host de parâmetros ou pela região geográfica. O Cloud Armor também é utilizado como um firewall de aplicativos da web, *Web Application Firewall (WAF)*, para evitar os ataques mais comuns na web e as tentativas de se explorar as vulnerabilidades do sistema; o Cloud Armor possui um conjunto de regras pré-configuradas que se baseiam nas principais regras do ModSecurity.

O Apigee possui políticas prontas para o uso, incorpora segurança as interfaces API, pode-se implementar as APIs com recursos para controlar o tráfego, melhorar o desempenho e reforçar a segurança.

A Samsung disponibiliza o Samsung Knox Manage cria um grupo completo de soluções empresariais que utilizam a plataforma Secured by Knox por meio da junção de bases de segurança integradas aos dispositivos Samsung [9].

#### 4.3.4. CRIPTOGRAFIA

É importante realizar a separação das informações importantes dos demais dados nos dispositivos que apresentam uma limitação em relação à capacidade de memória, processamento e ao consumo de energia devido ao seu tamanho. Essa separação é realizada de acordo com o grau de valor da informação ou o significado que o conjunto de dados possui, onde as informações têm de receber uma criptografia para assim os dados só serem consultados por quem é o seu destinatário [14].

Alguns dos ataques existentes são baseados exclusivamente na quebra da criptografia presente nos dados, como o ataque de canal lateral, a análise de criptografia e ataques MITM.

O ataque de canal lateral consiste no uso de técnicas de análise próprias da sincronização de informações, de vazamentos eletromagnéticos, do consumo de energia e de sons dos dispositivos de criptografia presentes em um sistema IoT. Com esse tipo de ataque o invasor é capaz de adquirir a chave criptografia utilizada para criptografar e descriptografar os dados compartilhados.

Na análise de criptografia o objetivo do agente mal-intencionado é encontrar a chave de criptografia através da quebra do método de criptografia do sistema, sendo que o agente já possui acesso ao texto cifrado [21].

#### 4.3.5. TIPOS DE ATAQUES CIBERNÉTICOS

A transmissão e armazenamento de dados de forma insegura, é composta por dispositivos que implementam protocolos da pilha TCP/IP e não utilizam um dispositivo gerenciador ou bridge (dispositivo que realiza a ponte entre redes), e precisam se conectar com a rede local para poder receber os comandos.

Um exemplo de vulnerabilidades causadas pelas transferências de dados de forma insegura seria o momento da conexão de uma lâmpada LED, onde o dispositivo, ao ser iniciado, disponibiliza um ponto de acesso livre e com *Service Set Identifier* (ou Identificador de Conjunto de Serviços, SSID) padrão. Ao ser disponibilizado o SSID, o agente malicioso conecta-se a esta rede facilmente e escaneia a rede para visualizar novos dispositivos conectados, esperando que ocorra a conexão do dispositivo de controle. Quando ocorre a conexão do dispositivo de controle, o agente malicioso inicia um ataque *Man in The Middle* (MITM), e ao obter sucesso no ataque, o agente passa a ter acesso ao dispositivo e às informações presentes nele [7].

Uma das aplicações cabíveis ao IoT é a telemetria, que consiste na coleta de dados em campo, medição, controle e comunicação destes dados para se realizar a gestão, controle, análises diversas e avaliações. Este tipo de aplicação pode sofrer diversos tipos de ataques cibernéticos, e um deles consiste na invasão dos dispositivos IoT, como impressoras, câmeras e geladeiras que estão conectados à internet para posteriormente causar um ataque *Distributed Denial of Service* (DDoS) [19].

Um ataque DDoS é um ataque de negação de serviço, *Denial of Service* (DoS) realizado por diversos dispositivos comprometidos, este tipo de ataque tenta indisponibilizar por sobrecarga um sistema [4]. Para se impedir que um ataque DDoS prossiga, geralmente, os servidores bloqueiam o tráfego de quem o está realizando; contudo, por serem vários dispositivos mandando requisições para o alvo, é impossível parar o ataque apenas bloqueando o tráfego de uma única fonte. Como não é possível monetizar este tipo de ataque, as causas possíveis para o motivo dele ocorrer é vingança ou ativismo [7].

Outros ataques que podem ocorrer a sistemas de IoT são o MITM e o aumento de privilégios. O MITM consiste no ato de um agente malicioso se posicionar entre a comunicação de dois dispositivos, como um computador recebendo os dados de um dispositivo IoT, podendo assim, receber todo o tráfego que está ocorrendo entre estes dispositivos. Para prevenir este tipo de ataque, é necessário a utilização de algum tipo de autenticação na troca de mensagens que identifique o remetente de forma única, ou criptografar as informações das mensagens por meio de algum protocolo específico para tal funcionalidade [7].

O ataque composto pelo aumento de privilégios de um usuário, consiste em elevar os acessos de um usuário, para assim este usuário poder realizar operações que normalmente não seriam autorizadas, e assim o agente malicioso poder coletar informações privadas, deletá-las, modificar arquivos ou inserir algum vírus [7].

Camada de Transporte:

A camada de transporte possui como principais protocolos o protocolo TCP e o protocolo UDP. Esta camada é responsável por transmitir os dados entre duas camadas do modelo OSI através dos dispositivos de destino e origem.

#### 4.3.6. PROTOCOLO TCP/IP

O protocolo TCP, Transmission Control Protocol, é um dos protocolos de comunicação, pertencente à camada de transporte da rede de computadores. Quando o dispositivo utiliza o protocolo TCP, antes de enviar as informações para o local de destino, primeiro o dispositivo garante a conexão com o servidor desejado, para só então dar início a troca de mensagens, sendo este um protocolo projetado para realizar o envio de todos os bytes

pertencentes a mensagem, possuindo também uma limitação da capacidade de transmissão de processos, para assim evitar a sobrecarga da rede [27].

Camada de Rede:

A camada de Rede é responsável pela movimentação de datagramas através de redes diferentes;

#### 4.3.7. PROTOCOLO IPV4

O protocolo *Internet Protocol version 4* (IPv4) foi o primeiro padrão de endereçamentos criado para endereçar os dispositivos em rede. Com este protocolo era possível acessar a internet, contudo, não foi previsto o grande crescimento de dispositivos conectados, por isso foi necessário confeccionar o IPv6, que é mais utilizado atualmente.

#### 4.3.8. PROTOCOLO 6LOWPAN

Em sistemas de IoT os dispositivos enviam pequenos pacotes de dados pela internet utilizando, unicamente o protocolo IPv6. No entanto os pacotes IPv6 possuem o tamanho maior que o tamanho do quadro de protocolos dos dispositivos IoT, por isso a *Internet Engineering Task Force* (IETF) criou o protocolo 6LoWPAN [31].

O 6LoWPAN foi desenvolvido inicialmente para ser utilizado pelo padrão IEEE 802.15.4, sendo uma camada de adaptação, sua maior função é compactar os pacotes IPv6, para assim os dispositivos que possuem baixo poder computacional, poderem utilizar o protocolo IPv6 [36].

Camada de Enlace:

A camada de enlace possui protocolos como o protocolo Ethernet, protocolo Wi-Fi e o protocolo Bluetooth. Esta camada é a camada responsável por transferir os datagramas criados na camada de rede entre roteadores e enviá-los ao dispositivo de destino pertencente à mesma rede.

#### 4.3.9. TECNOLOGIAS DA COMUNICAÇÃO

A tecnologia da comunicação pode ser definida como uma tecnologia utilizada para realizar a comunicação entre os dispositivos presentes em uma rede IoT com outros

dispositivos conectados à internet. Alguns exemplos de tecnologias de comunicação são: Rede Ethernet; Wi-Fi; ZigBee; Bluetooth Low Energy 3G/4G; LoRaWAN; e SigFox; para melhor entendimento de todos, será definido um pequeno resumo de cada tecnologia a seguir [36].

O protocolo Ethernet é um tipo de comunicação realizada com fio. Por ser um meio simples, de fácil adaptação e acesso e custos mais acessíveis, encontra-se presente na maioria das redes locais resistentes atualmente. Este padrão foi oficializado em 1983 pelo IEEE (IEEE 802.3), tendo atualmente dois tipos de cabos que podem ser implementados: par trançado, tendo como distância máxima 100 metros sem a utilização de repetidores e podendo atingir taxas de até 1 Gbps e fibra óptica, limitada a 2000 metros e com alcance de taxas de até 10 Gbps [40].

A tecnologia Wi-Fi foi lançada em 1997, se tornando um grande sucesso atualmente e estando presente em todos os lugares, desde lojas comerciais, casas residenciais e escritórios a até em espaços públicos das cidades. A regularização do Wi-Fi se deu por meio da norma IEEE 802.11, que define um agrupamento de padrões de codificação e transmissão, tendo como última atualização até o momento a versão IEEE 802.11bi, sendo que a versão IEEE 802.11ac prevê que as taxas de comunicação possam ser 600 Mbps ou 1300 Mbps [36].

Para a camada de enlace há o padrão IEEE 802.15.4 que a regulariza, sendo que o ZigBee se baseia nesse padrão. Possui um consumo energético e custo reduzido por causa da sua baixa vazão. Normalmente o ZigBee opera na faixa de frequência ISM (2,4 GHz), sendo capaz de operar em outras duas faixas de frequências, 868 MHz e 915 MHz. Tendo como taxa máxima 250 Kbps, mesmo que na prática as taxas alcançadas sejam inferiores [20].

A Ericsson propôs a utilização do protocolo Bluetooth Low Energy para substituir a comunicação serial RS-232. O protocolo Bluetooth se divide em duas categorias: o Bluetooth clássico e o *Bluetooth Low Energy* (BLE), sendo que o clássico possui mais duas subdivisões, *Basic Rate/Enhanced Data Rate* (BR/EDR), que é a subdivisão que engloba as versões 2.0 ou anteriores, enquanto o *Bluetooth High Speed* (HS) engloba a versão 3.0 e o BLE a versão 4.0 ou superior, posto que a versão BLE foi construída para dispositivos com baixo consumo de energia, enquanto as demais versões são focadas em aumentar a taxa

de comunicação. Atualmente o Bluetooth se encontra na versão 5.3 [8], lançada em 2021, que traz uma melhor capacidade de resposta, possui uma confiabilidade melhor e traz mais economia de energia para os dispositivos [20].

É possível realizar a aplicação dos padrões de telefonia celular 3G/4G às aplicações IoT. Possuindo um consumo energético muito alto, em relação a outras tecnologias, o padrão 3G/4G pode ser utilizado por projetos IoT quando estes abrangem grandes distâncias. As normas brasileiras especificam que as frequências que devem ser utilizadas para o protocolo 3G são 1900 MHz e 2100 MHz, estando na faixa UMTS, à medida que o 4G, *Long Term Evolution* (LTE) utiliza a frequência de 2500 MHz. A diferença entre a taxa de comunicação entre os dois padrões é visivelmente grande, dado que o padrão 3G alcança 1 Mbps e o padrão 4G alcança 10 Mbps [38].

O protocolo *Long Range Wide Area Network* (LoRaWAN) foi planejado para que dispositivos operados por bateria e possuindo a capacidade de comunicação sem fio, pudessem fazer parte de redes de longa distância, podendo ser uma rede global, nacional ou regional, o protocolo LoRaWAN é responsável por criar as redes de longo alcance. O padrão LoRaWAN fornece suporte ao protocolo IPv6, este suporte é feito a partir do protocolo denominado 6LoWPAN, funciona sobre a topologia estrela, possui uma comunicação segura e bi-direcional, tratamento e mobilidade de serviços de localização, trazendo assim, os requisitos básicos presentes na IoT. A taxa de comunicação do sistema pode variar entre 300 bps e 50Kbps. O protocolo LoRaWAN atua na faixa ISM e utiliza frequências em MHz, sendo as frequências mais usadas: 109 MHz, 433 Mhz, 866 MHz e 915 MHz. Por utilizar tais frequências, as ondas transmitidas penetram e transpassam grandes estruturas e superfícies, com distâncias de 2 km a 5km no meio urbano, ou até 45 km no meio rural. O *Maximum Transmit Unit* (MTU) utilizado pelo padrão LoRaWAN é de 256 bytes [20].

A tecnologia *Ultra Narrow Band* (UNB) é utilizada pelo protocolo SigFox. Este protocolo foi projetado para ser utilizado na transmissão de pequenas taxas de transferência de dados. Dispondo de um suporte a diversos dispositivos, a SigFox atua como uma operadora para IoT. Para que os usuários consigam implementar sistemas de IoT com mais facilidade, a SigFox possui como sua maior incumbência abstrair as dificuldades de

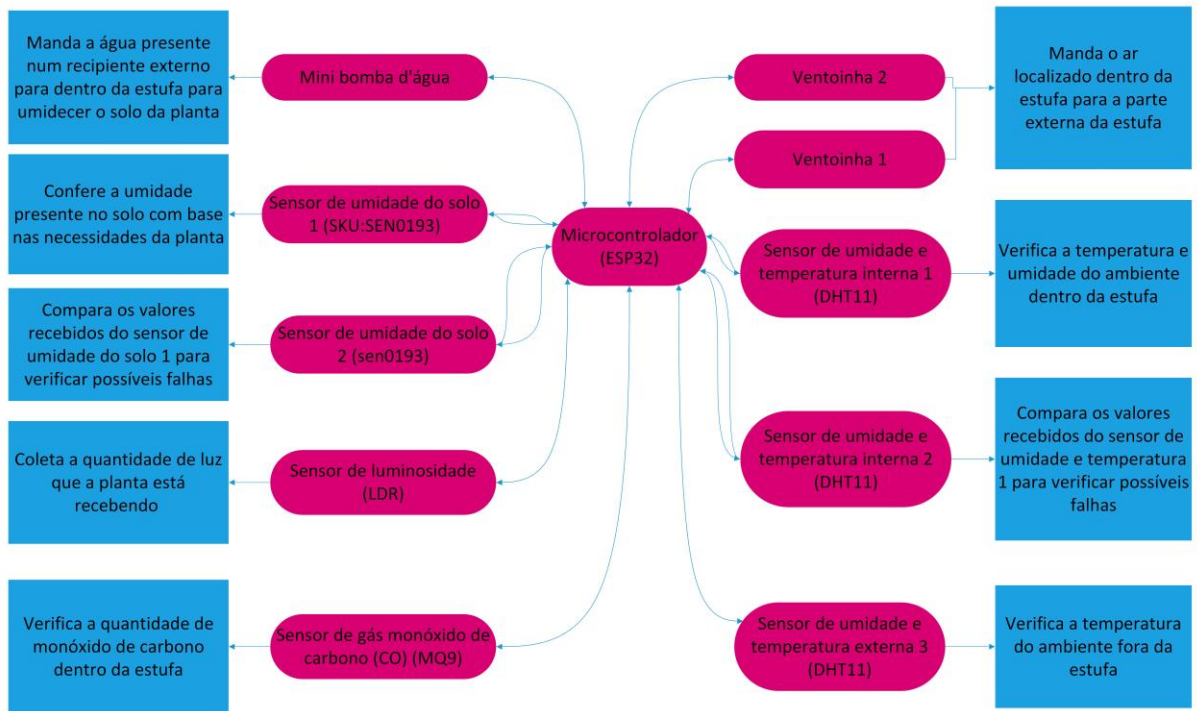
conexão e prover uma API. O SigFox pode abranger um raio de cobertura de 3 km a 10 km nas zonas urbanas e de 30 km a 50 km em zonas rurais. Dispondo de uma taxa de comunicação que pode variar entre 10 bps a 1000 bps e com o MTU de 96 bytes. A faixa de frequência em que o SigFox atua é de 900 MHz [36].

Após analisar todos os componentes e conceitos que compõem a estufa a ser projetada, no próximo capítulo será abordada a implementação do projeto e suas especificações.

# 5. IMPLEMENTAÇÃO

A implementação do projeto é composta pela metodologia utilizada, onde são descritos os procedimentos realizados ao longo do projeto, sendo esta metodologia subdividida em arquitetura, hardware e software.

Para a realização do estudo de caso de controle e monitoramento remoto de uma estufa IoT, inicialmente foi realizado um datagrama esquemático informando a função de cada dispositivo presente. Na Figura 16 é possível visualizar o datagrama com o funcionamento da estufa.



**Figura 16** Datagrama com a definição do funcionamento dos sensores [Autor]

A seguir são descritos os valores gastos com a criação deste projeto por meio da Tabela 3.

**Tabela 3** Gastos com os materiais utilizados no projeto [Autor]

Materiais utilizados	Preço (R\$)
Sensor MQ-9	22,90
2 Sensores capacitivos de umidade do solo	40,00
Fonte de alimentação 12 V	16,90
Módulo fonte 12 V	21,90
Protoboard 400 pontos	16,90
Mini bomba d'água	19,20
Módulo sensor LDR	17,90
2 Sensores DHT11	32,76
3 Sensores DHT11	37,90
Termo-higrômetro	25,97
Acrílico	0,00
Corte a laser	0,00
Cola de silicone	16,90
<b>Total</b>	<b>269,23</b>

Para realizar a implementação deste projeto foi necessário pensar na arquitetura a ser utilizada, por isso, o próximo tópico trata-se da arquitetura do projeto, contendo uma listagem de todos os componentes utilizados.

## 5.1. ARQUITETURA

A arquitetura deste projeto é composta pelos seguintes componentes:

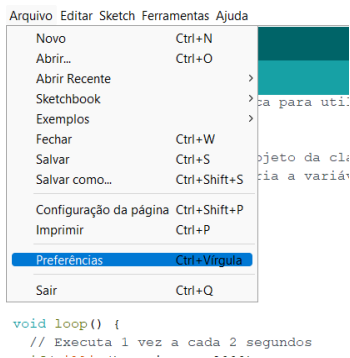
- 3 sensores DHT11 ;
- 1 microprocessador ESP32;
- 1 sensor de gás MQ-9;
- 2 sensores de umidade do solo capacitivo sku:sen0193;
- 1 módulo LDR;
- 2 coolers;
- 1 mini bomba d'água;
- 2 relés;
- 2 protoboards conectadas;
- 1 módulo de tensão 12 V;
- Jumpers para realizar a conexão;
- 1 estrutura de acrílico.

Para a utilização do Arduino foi necessário realizar a configuração do software que foi descrito no próximo subtópico.

### 5.1.1. CONFIGURAÇÃO DO SOFTWARE ARDUINO

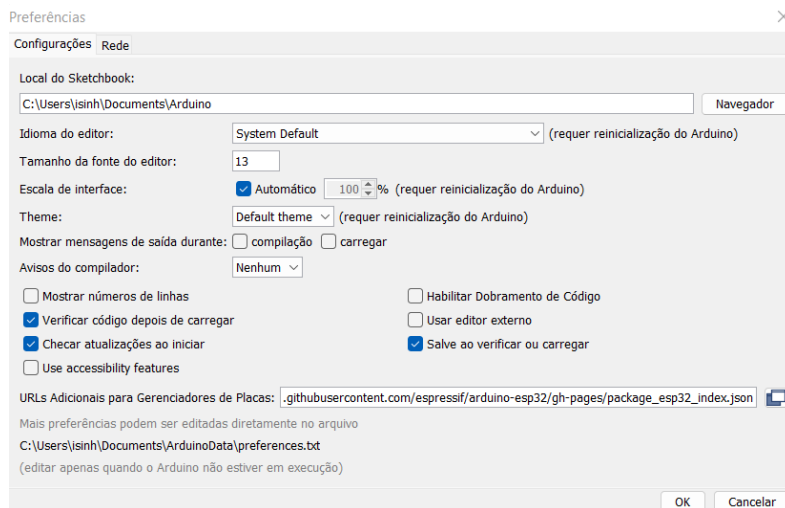
Para que o sistema Arduino consiga ler a porta USB em que o microcontrolador ESP está conectado, é necessário baixar o driver “CP210x Universal Windows Driver” disponibilizado gratuitamente pela empresa Silicon Labs.

Para o funcionamento da ESP32 no software Arduino foi necessário realizar o download da biblioteca ESP32, disponibilizado pela empresa Espressif.



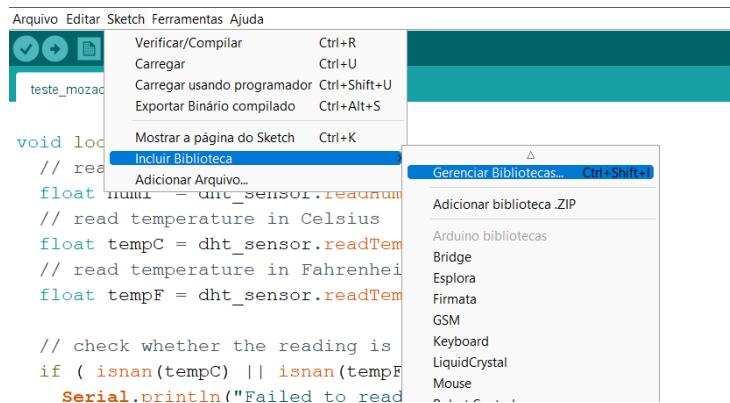
**Figura 17** Menu arquivo [Autor]

Para baixá-la é necessário copiar o link “[https://raw.githubusercontent.com/esp8266/Arduino/master/packages/package\\_index.json](https://raw.githubusercontent.com/esp8266/Arduino/master/packages/package_index.json)” disponibilizado na página da empresa, e colocar na aba “URLs Adicionais para Gerenciadores de Placas:” presente dentro do menu “Preferências” na aba “Arquivo”, como demonstrado nas Figura 17 e Figura 18.



**Figura 18** Tela das preferências [Autor]

Após acrescentar o link nas preferências será possível encontrar a biblioteca para a realização do download na parte do gerenciamento de bibliotecas, dentro do “Gerenciador de Bibliotecas” presente na aba “Sketch”, dentro da opção “Incluir Biblioteca”, ao se pesquisar “esp32”, conforme apresentado na Figura 19 e Figura 20.



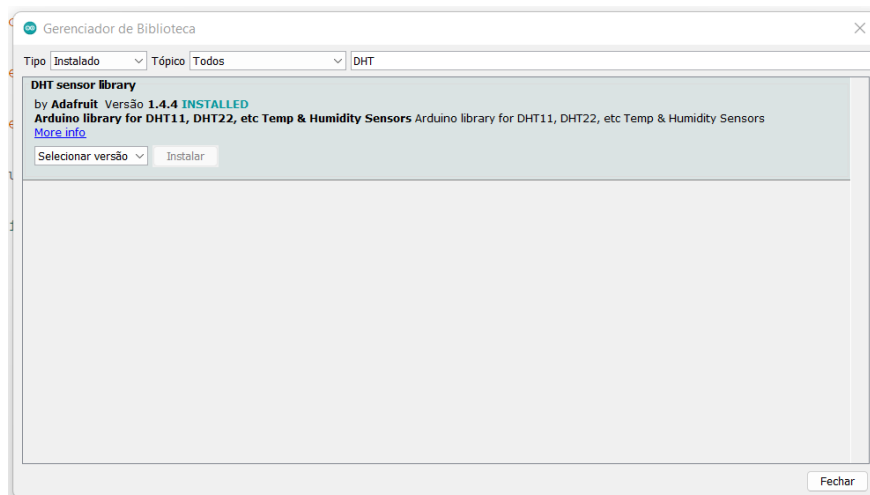
**Figura 19** Menu sketch [Autor]



**Figura 20** Biblioteca ESP32 [Autor]

Sendo necessário realizar o download da biblioteca do sensor DHT utilizável para dispositivos ESP32.

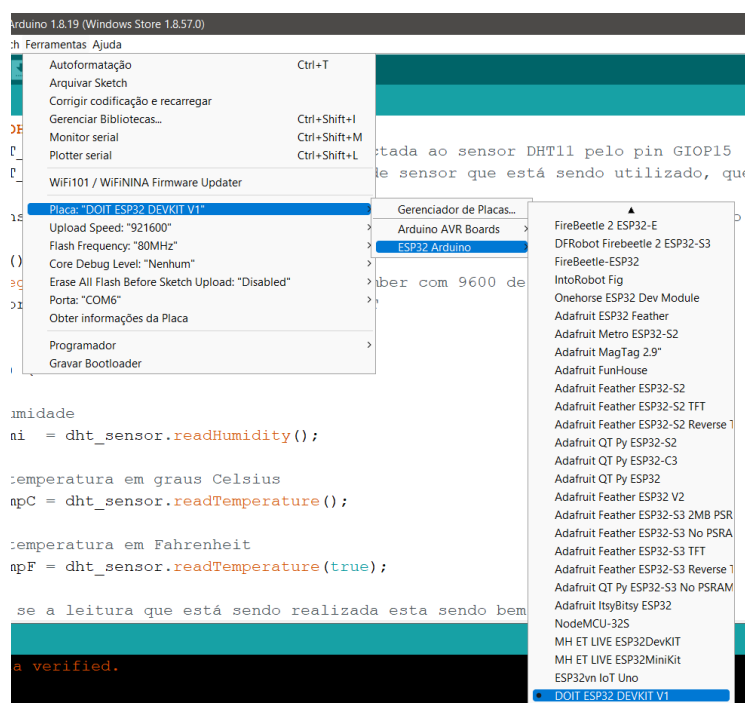
Para realizar o download da biblioteca DHT é necessário entrar na aba “Gerenciamento de Bibliotecas”, dentro da opção “Incluir Biblioteca” presente no menu “Sketch”, pesquisar por “DHT”, baixando a opção indicada, como demonstrado na Figura 21. A metodologia utilizada para baixar a biblioteca DHT foi utilizada para baixar as demais bibliotecas, como a biblioteca Adafruit Unified Sensor, AsyncTCP, MySQL Connector Arduino e OneWire.



**Figura 21** Gerenciador de bibliotecas [Autor]

Após a escrita do código de leitura do sensor DHT11, é necessário conectar o sistema Arduino a placa ESP32 e a porta USB utilizada, que no caso em questão a placa utilizada é a “DOIT ESP32 DEVKIT V1” e a porta em que o sistema está conectado é a “COM6”.

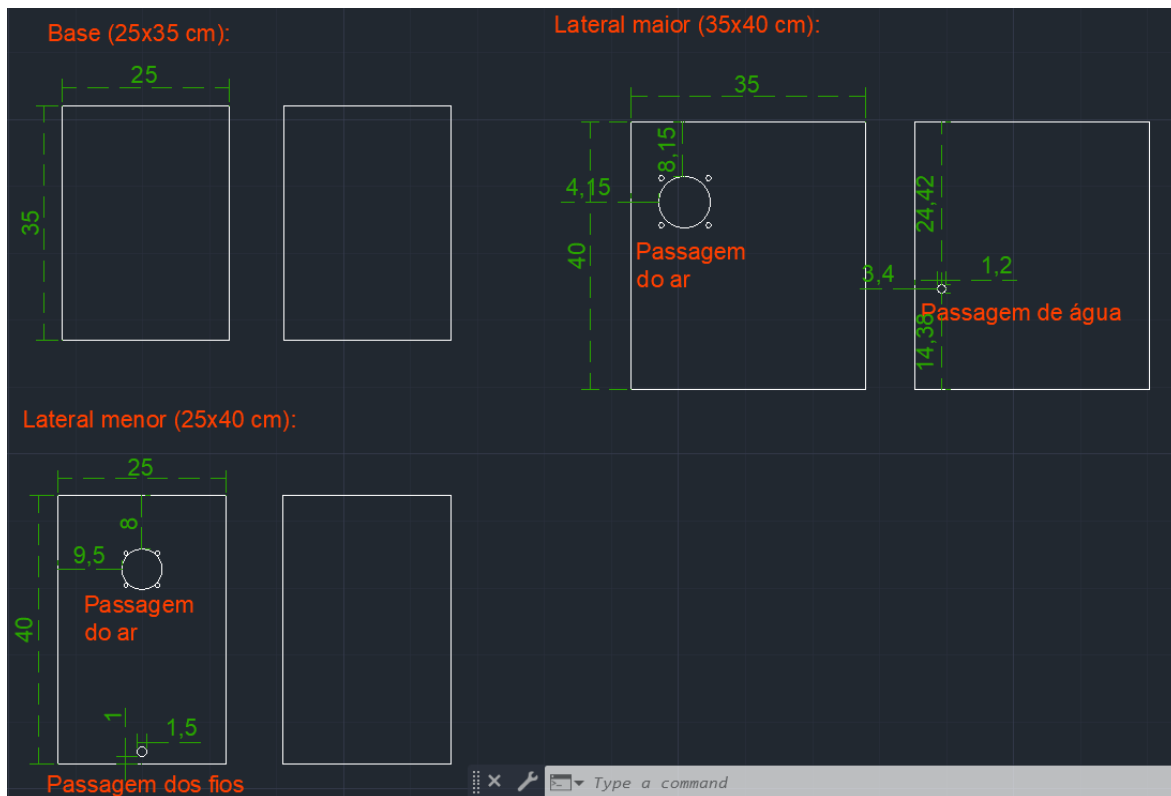
A opção da placa pode ser alterada no menu “Ferramentas”, dentro da aba “Placa”, onde é possível selecionar entre modelos de placas de Arduino e de ESPs, conforme pode se visualizar na Figura 22.



**Figura 22** Definição de placas [Autor]

## 5.2. HARDWARE

Para a criação da estrutura da estufa foi utilizado o desenho apresentado na Figura 23. A parte estrutural da estufa foi desenhada no AutoCad Lite, utilizando a unidade de medida em milímetros. O desenho foi passado para uma cortadora a laser, que cortou acrílicos de 30 mm, que foram colados com cola de silicone.



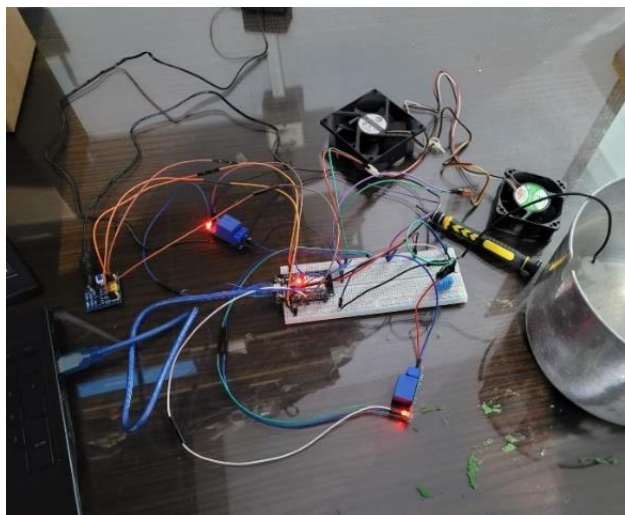
**Figura 23** Desenho realizado no AutoCad Lite [Autor]

Obtendo-se a estrutura mostrada na Figura 24, onde pode-se ver que na lateral de frente para a câmera, encontra-se o buraco para a passagem da mangueira d'água, à esquerda encontra-se o espaço para a ventoinha de 6 cm com um furo na parte de baixo para a conexão da ESP32. No lado seguinte estão localizados os furos para a conexão da ventoinha de 8 cm.



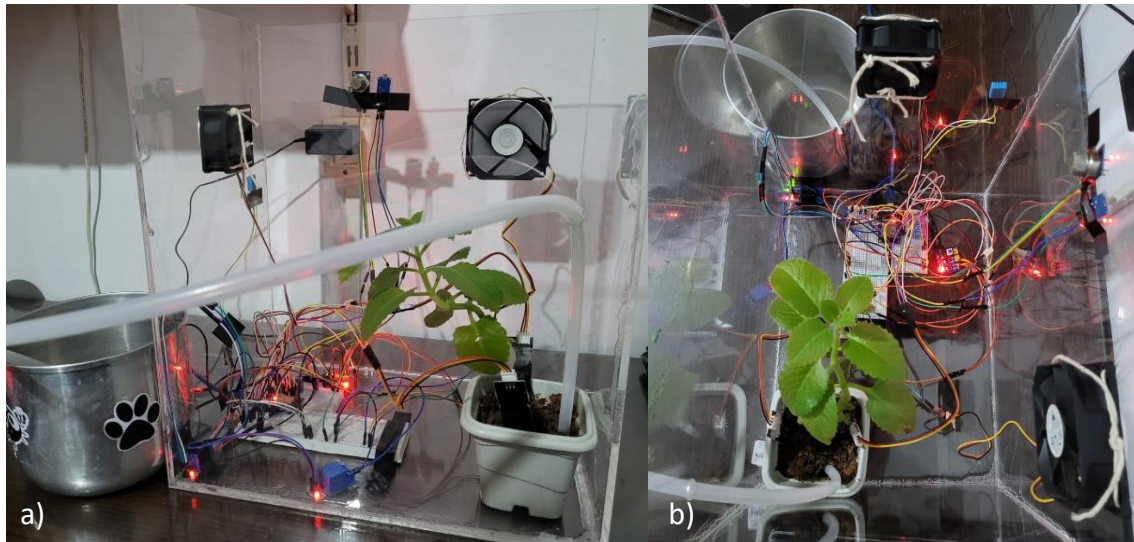
**Figura 24** Estrutura da estufa [Autor]

O circuito funcionou bem ao ser conectado às ventoinhas com a bomba para serem ativadas de acordo com a umidade e a temperatura sendo apresentada, conforme pode ser visto na Figura 25.



**Figura 25** Funcionamento das ventoinhas e da bomba em conjunto com os sensores DHT11 [Autor]

Após conectar todos os sensores à estrutura da estufa, a estufa ficou como apresentado na Figura 26.



**Figura 26** Na figura a) é apresentada a vista lateral da estufa, enquanto na figura b) é apresentada a vista superior [Autor]

### 5.3. SOFTWARE

Inicialmente foi realizado um software onde o programa era responsável por ler a umidade e temperatura do sensor DHT11, informando a temperatura em graus Celsius e em Fahrenheit e a umidade do ambiente no monitor serial do Arduino e em uma página HTTP que é acessada a partir do IP fornecido pelo sistema. Contudo, no decorrer do projeto decidiu-se realizar a visualização dos dados por meio do software Cayenne e pelo banco de dados MySQL.

Inicialmente decidiu-se utilizar o banco de dados MySQL Workbench, contudo, ao executar o código o banco de dados não se conectava com a ESP32, e continuava tentando realizar a conexão e falhando, por isso, decidiu-se adotar o banco de dados db4free, que é um banco projetado para a realização de testes e que pode ser administrado pela ferramenta phpMyAdmin. No trecho de código a seguir há um exemplo da estruturação da tabela criada no SQL. O código completo da query utilizada pode ser encontrado no Anexo A.

```
# Criação do banco, das tabelas e das  
variáveis
```

```

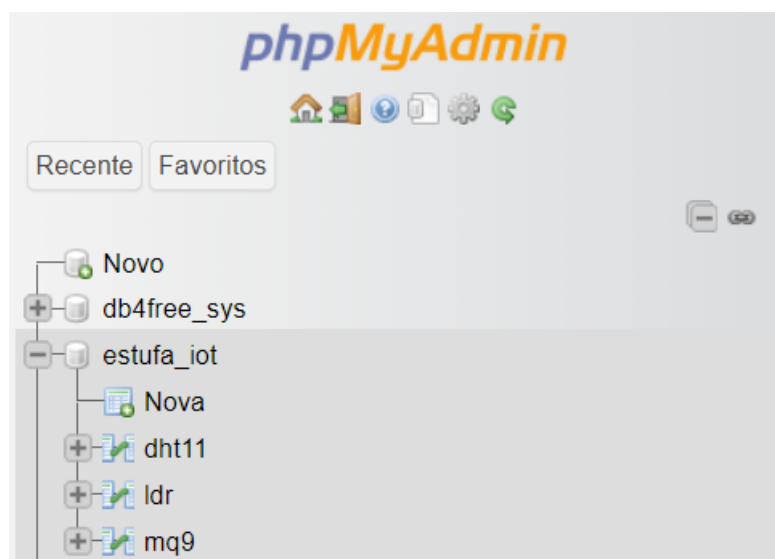
create schema estufa_iot; # Criação do banco
de dados utilizado

# Criação da tabela dht11 (uma das tabelas
presente no banco):
CREATE TABLE `estufa_iot`.`dht11` (
  `id_dht11` INT NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id_dht11`));

# Inclusão de colunas na tabela DHT11:
ALTER TABLE `estufa_iot`.`dht11`
ADD COLUMN `umidadeInterno1` FLOAT NULL AFTER
`id_dht11`,
ADD COLUMN `temperaturaInterno1` FLOAT NULL
AFTER `umidadeInterno1`,
ADD COLUMN `umidadeInterno2` FLOAT NULL AFTER
`temperaturaInterno1`,
ADD COLUMN `temperaturaInterno2` FLOAT NULL
AFTER `umidadeInterno2`,
ADD COLUMN `umidadeExterno3` FLOAT NULL AFTER
`temperaturaInterno2`,
ADD COLUMN `temperaturaExterno3` FLOAT NULL
AFTER `umidadeExterno3`,
ADD COLUMN `coolers` VARCHAR(45) NULL AFTER
`temperaturaExterno3`;

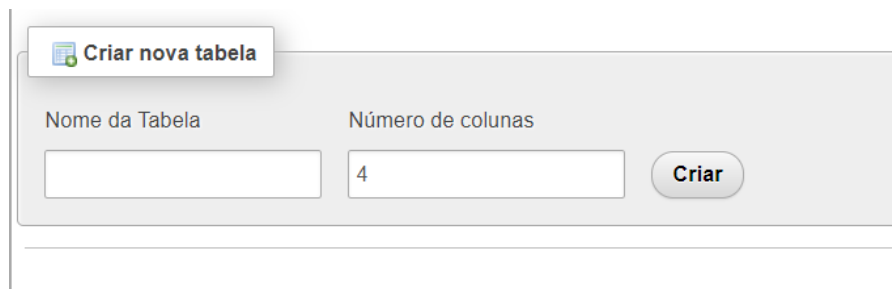
```

Após a criação do banco de dados estufa\_iot, com o usuário estufando, foi criada a estrutura do banco pela ferramenta phpMyAdmin, como pode se ver na Figura 27.



**Figura 27** Banco de dados no software phpMyAdmin [Autor]

Para adicionar novas tabelas dentro do banco de dados no phpMyAdmin basta ir no campo “Criar nova tabela” e inserir as especificações desejadas, Figura 28.

A imagem mostra a interface de usuário do phpMyAdmin para criar uma nova tabela. No topo, há um botão com um ícone de documento e o texto "Criar nova tabela". Abaixo, há dois campos de entrada: "Nome da Tabela" (vazio) e "Número de colunas" (contendo o número "4"). À direita dos campos, há um botão "Criar".

**Figura 28** Criação de tabela phpMyAdmin [Autor]

Para se aprofundar no sistema elaborado, a seguir apresenta-se uma breve descrição das principais bibliotecas utilizadas na confecção do código em C++ na plataforma Arduino. O código completo utilizado pode ser encontrado no Anexo B.

- AsyncTCP.h: É uma biblioteca que fornece uma implementação assíncrona do protocolo TCP para o ESP32. Isso permite que o ESP32 lide com várias conexões TCP de forma assíncrona e eficiente.
- ESPAsyncWebServer.h: É uma biblioteca que permite criar um servidor da Web assíncrono no ESP32. Ele fornece funcionalidades para lidar com solicitações HTTP e enviar respostas para os clientes.
- MySQL\_Connection.h: É uma biblioteca que permite a conexão do ESP32 a um servidor MySQL para realizar operações de banco de dados relacionados.
- MySQL\_Cursor.h: É uma biblioteca que fornece um cursor para executar consultas e manipular resultados de consultas em um banco de dados MySQL usando o ESP32.
- DHT.h: Esta biblioteca é usada para trabalhar com sensores DHT (umidade e temperatura) no ESP32. Ela fornece funções para ler os valores de umidade e temperatura do sensor DHT conectado ao ESP32.
- Wire.h: É uma biblioteca usada para comunicação I2C (Inter-Integrated Circuit). Ela permite que o ESP32 se comunique com outros dispositivos usando o protocolo I2C.

- CayenneMQTTESP32.h: É uma biblioteca utilizada para a integração do ESP32 com a plataforma de IoT Cayenne MQTT. Ela facilita o envio de dados do ESP32 para a plataforma Cayenne e permite o controle remoto do dispositivo.

O trecho de código mostrado abaixo pertence ao código escrito na plataforma Arduino e é responsável por realizar a inclusão das bibliotecas.

```
// Inclusão das bibliotecas
#include <DHT.h>
#include <Wire.h>
#include <CayenneMQTTESP32.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>
```

O trecho a seguir é responsável por enviar os dados lidos pela ESP32 para o banco de dados MySQL.

```
//MQ-9
sprintf(query, INSERT_SQL_MQ9, ppmCO);

Serial.print("sprintf: ");
Serial.println(query);

MySQL_Cursor *cur_mem = new
MySQL_Cursor(&conn);
cur_mem->execute(query);
delete cur_mem;
```

Ao enviar os dados coletados pelos sensores para o banco de dados, ocorreu o erro “Guru Meditation Error: Core 1 panic'ed (LoadProhibited). Exception was unhandled.” Que indica um erro na memória da ESP32 por tentar ler e enviar demasiados dados ao mesmo tempo, por isso optou-se por enviar os dados separadamente para o banco de dados. A Figura 29 apresenta o formato que os dados se encontram no banco.

SELECT \* FROM `umidade\_solo`

Perfil [ Editar em linha ] [ Editar ] [ Demonstrar SQL ] [ Criar código PHP ] [ Atualizar ]

Mostrar tudo | Número de linhas: 25 | Filtrar linhas: Procurar nesta tabela | Ordenar pela chave: Nenhum

Opções extras

	id_umidade_solo	sensor1Value	percent_sensor1	sensor2Value	percent_sensor2	BOMBA_PIN	data
<input type="checkbox"/> Editar Copiar Remover	1	NULL	NULL	NULL	NULL	0	2023-06-21 20:57:38
<input type="checkbox"/> Editar Copiar Remover	2	2515	NULL	NULL	NULL	NULL	2023-06-21 20:57:39
<input type="checkbox"/> Editar Copiar Remover	3	NULL	38	NULL	NULL	NULL	2023-06-21 20:57:39
<input type="checkbox"/> Editar Copiar Remover	4	NULL	NULL	2571	NULL	NULL	2023-06-21 20:57:40
<input type="checkbox"/> Editar Copiar Remover	5	NULL	NULL	NULL	36	NULL	2023-06-21 20:57:41
<input type="checkbox"/> Editar Copiar Remover	6	NULL	NULL	NULL	NULL	0	2023-06-21 20:57:53
<input type="checkbox"/> Editar Copiar Remover	7	2507	NULL	NULL	NULL	NULL	2023-06-21 20:57:55
<input type="checkbox"/> Editar Copiar Remover	8	NULL	38	NULL	NULL	NULL	2023-06-21 20:57:56
<input type="checkbox"/> Editar Copiar Remover	9	NULL	NULL	2556	NULL	NULL	2023-06-21 20:57:57
<input type="checkbox"/> Editar Copiar Remover	10	NULL	NULL	NULL	37	NULL	2023-06-21 20:57:58

**Figura 29** Dados no banco de dados [Autor]

Enquanto este outro pedaço do código mostrado a seguir, é uma parte da função responsável por enviar os dados lidos pelos sensores para a plataforma Cayenne. O código completo escrito na plataforma Arduino encontra-se no Anexo B.

```
// Função para publicar os dados no Cayenne
MQTT
void publishData() {
    // Calcula a média da temperatura dos
    sensores externos
    float temperaturaMediaInterna =
    (temperatureInterno1 + temperatureInterno2) /
    2.0;
    // Calcula a média das umidades dos
    sensores internos
    float umidadeMediaInterna =
    (humidityInterno1 + humidityInterno2) / 2.0;

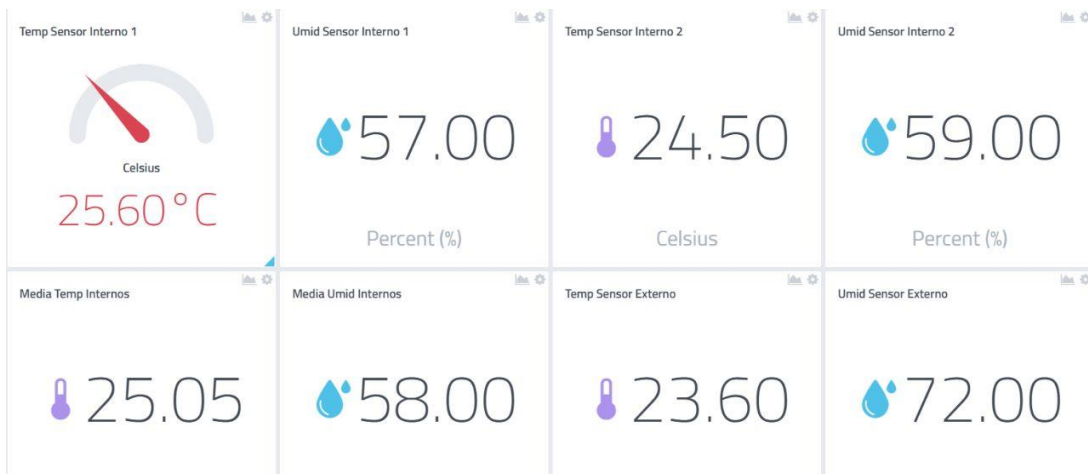
    // Publica as leituras dos sensores no
    Cayenne
    Cayenne.virtualWrite(0,
    temperatureInterno1, "temp", "c");
    Cayenne.virtualWrite(1, humidityInterno1,
    "rel_hum", "p");
    Cayenne.virtualWrite(2,
    temperatureInterno2, "temp", "c");
    Cayenne.virtualWrite(3, humidityInterno2,
    "rel_hum", "p");
    Cayenne.virtualWrite(4,
    temperaturaMediaInterna, "temp", "c"); //
```

```

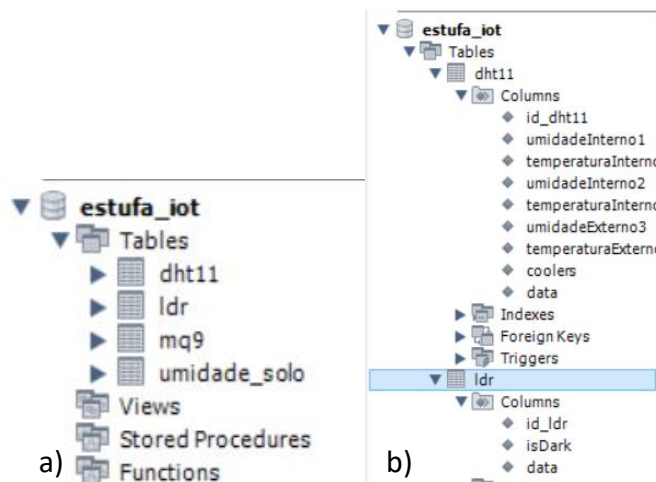
Envia a média da temperatura dos sensores
externos no canal 4
    Cayenne.virtualWrite(5,
umidadeMediaInterna, "rel_hum", "p"); //
Envia a média das umidades dos sensores
internos no canal 5
    Cayenne.virtualWrite(6,temperatureExterno3,
"temp", "c");

```

Os dados enviados ao servidor Cayenne foram possíveis de ser observados graficamente em tempo real, vide Figura 30 e também foram exportados para um banco de dados MySQL local, Figura 31.



**Figura 30** Gráficos apresentados no Cayenne [Autor]



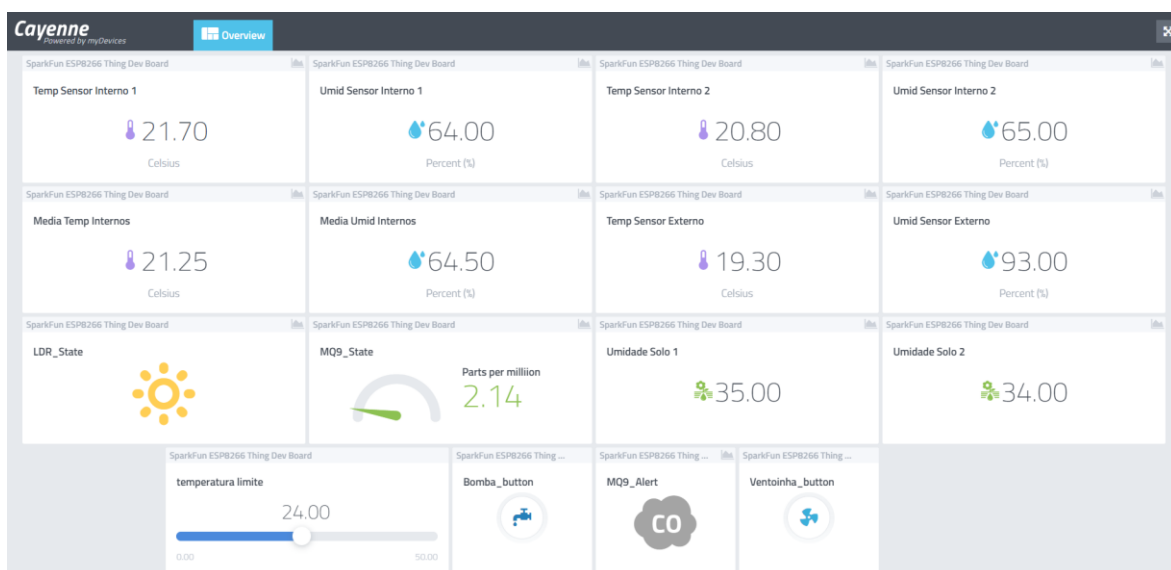
**Figura 31** Na figura a) encontra-se a estrutura do banco de dados com as 4 tabelas presentes nele. Na figura b) tem-se as colunas presentes da tabela dht11 e na tabela ldr [Autor]

Após inserir no código os comandos para verificar o IP e a porta do dispositivo que está recebendo os dados enviados pela ESP, foi possível visualizar o trecho da requisição HTTP que fornece estas informações, como pode ser verificado na Figura 32.

```
Nova requisição:  
Método: GET  
URL: /  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36  
Corpo:  
Endereço IP do cliente: 192.168.1.10  
Porta do cliente: 59520
```

**Figura 32** IP e Porta do cliente que está recebendo os dados da ESP32 [Autor]

Após a implementação de todos os sensores na plataforma Cayenne, obteve-se a seguinte tela de controle, onde é possível ligar e desligar os atuadores da bomba e da ventoinha e ajustar a temperatura limite que deve ser considerada para a estufa ligar ou desligar as ventoinhas. Nesta página também é mostrado o nível de umidade do ar e a temperatura dos três sensores DHT11, além da média dos valores adquiridos pelos sensores DHT11 internos, a porcentagem de água que cada sensor de umidade de solo está captando, o nível de partículas de gás CO captados no ambiente e o alerta caso este nível ultrapasse o recomendado, que no caso está sendo utilizado 3 partículas por milhão para demonstração do funcionamento e se a planta está ou não recebendo luz solar, como pode ser visualizado na Figura 33. Caso o usuário queira compartilhar a aplicação criada no Cayenne para outras pessoas poderem interagir com o projeto, é necessário compartilhar o link do projeto com as pessoas desejadas. Este link pode ser encontrado dentro da plataforma Cayenne.



**Figura 33** Tela de monitorização e controle da estufa [Autor]

Para verificar o funcionamento dos sensores e atuadores que compõem o sistema, foram realizados testes separados para cada sensor, sendo testados em conjunto os sensores que são iguais. Os testes realizados encontram-se no próximo capítulo deste trabalho, denominado verificação experimental.

## 6. VERIFICAÇÃO EXPERIMENTAL

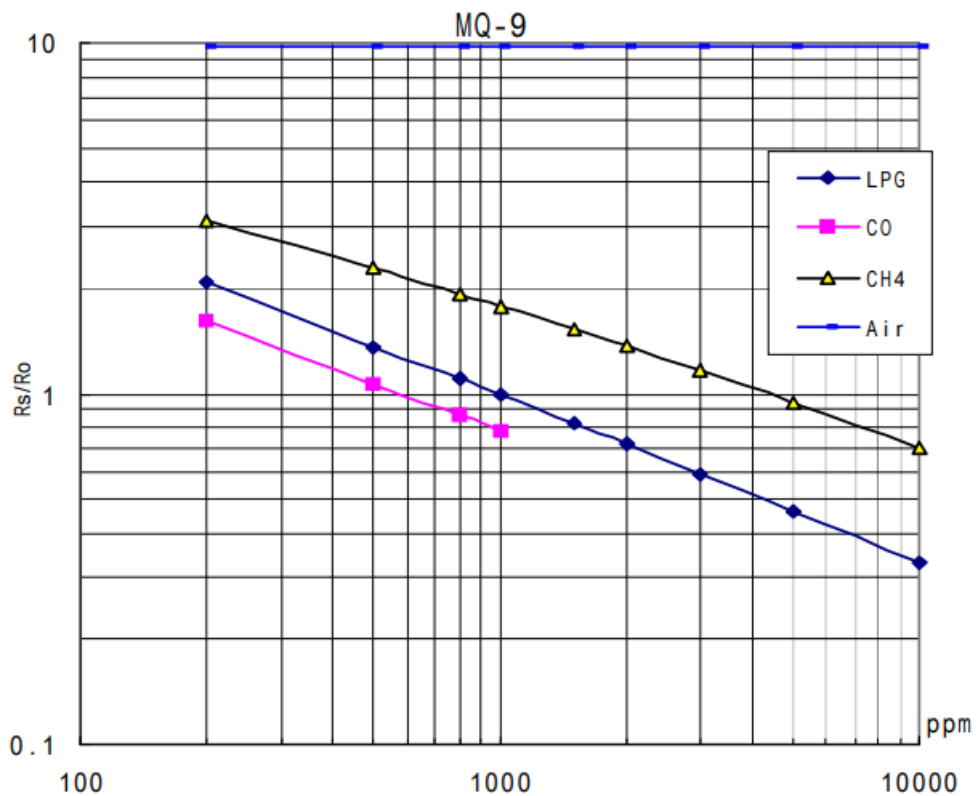
Após a confecção do diagrama conceitual e do datagrama foram realizados os testes individuais de cada sensor, sendo os sensores idênticos testados em conjunto. A verificação do funcionamento dos sensores, atuadores e do projeto em si é uma etapa importante para a confecção do projeto, pois assim se têm a certeza de que o sistema está funcionando por inteiro.

### 6.1. METODOLOGIA

Para a realização dos testes de verificação do funcionamento dos sensores de umidade e temperatura do ambiente foi utilizado um termo-higrômetro, que possui precisão na

temperatura de  $\pm 1^{\circ}\text{C}$  e com precisão da umidade de  $\pm 5\%$ . Caso fosse encontrado uma divergência extrema nos valores lidos, seria acrescentado ao código uma linha onde se insere mais ou menos a porcentagem de erro obtido, para assim as leituras se tornarem equivalentes.

Para configurar o sensor de monóxido de carbono foi utilizada a curva de calibração do sensor mq9 para o gás monóxido de carbono. Esta curva está presente no datasheet do sensor, Figura 34, e está na escala logarítmica.



**Figura 34** Característica da sensibilidade do sensor MQ9 [Fonte<sup>9</sup>]

A variável “curva” que foi criada dentro da função calcularPPMCO() foi criada a partir da seleção de dois pontos da curva original e da sua inclinação, o que fornece uma curva com o formato aproximado da curva original que foi fornecida no datasheet do sensor.

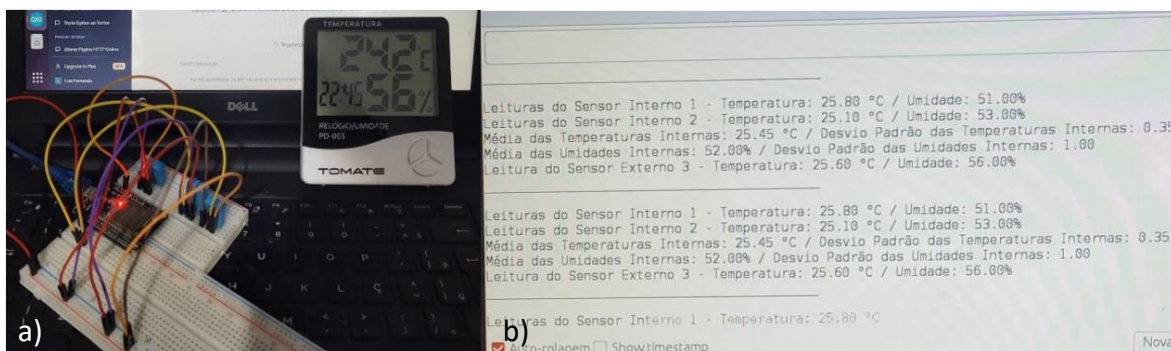
<sup>9</sup> [https://www.electronicoscaldas.com/datasheet/MQ-9\\_Hanwei.pdf](https://www.electronicoscaldas.com/datasheet/MQ-9_Hanwei.pdf)

Para realizar a medição dos sensores de umidade de solo foi realizado um teste de calibração, onde foi medido 100 vezes o valor analógico lido pelos sensores em um solo seco, sem a presença de água; em cum solo umido, com pouca água; e em um solo encharcado, com muita água. Com isso obteve-se um valor médio das medições para ser utilizado como referência para um solo seco e um solo umido e assim configurar o código.

Um fato importante de ser mencionado é que os dados lidos pelo sensor de luminosidade e pelos sensores de umidade do solo tiveram que ser invertidos para serem enviados para a página do Cayenne e para o banco de dados.

## 6.2. UMIDADE E TEMPERATURA DO AMBIENTE

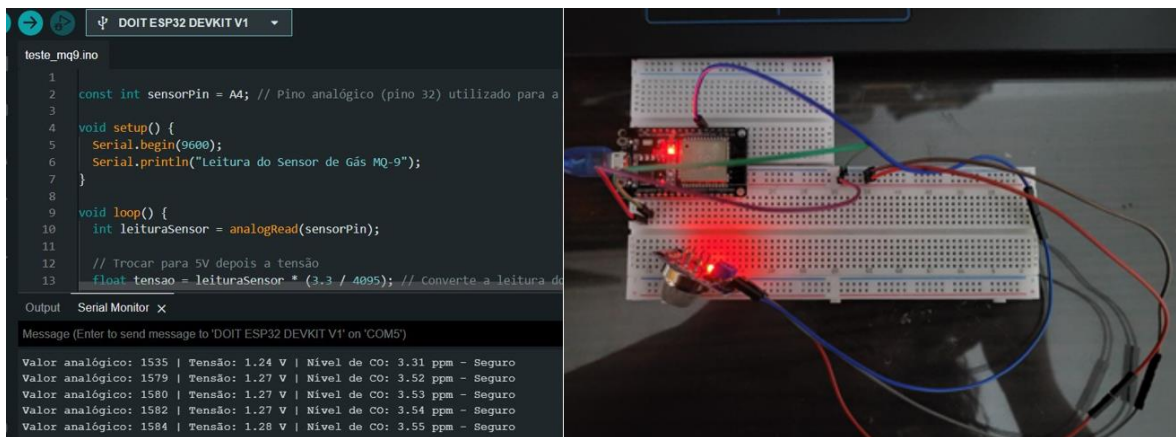
Os sensores DHT11 foram programados para que fosse realizada a verificação do funcionamento de cada sensor e se caso um deles, dois deles, ou todos os sensores não estivessem funcionando ser mostrada uma mensagem de erro, que informa qual é o sensor com problema e ser exibido os dados coletados pelos outros sensores. Os DHT11 também foram programados para, após ser coletado os dados, ser comparado o nível da temperatura interna da estufa com a temperatura máxima requerida e caso o nível da temperatura estiver maior do que o desejado, que seja realizada a ligação das ventoinhas para ventilarem a estufa. Para a calibração dos sensores foi utilizado um termo-higrômetro. Na Figura 35 vê-se o funcionamento dos sensores DHT11 com o termo-higrômetro ao lado.



**Figura 35** Na figura a) mostra-se a montagem do circuito com os sensores DHT11, enquanto na figura b), mostra-se a leitura realizada por eles [Autor]

## 6.3. MONÓXIDO DE CARBONO

Para o sensor MQ-9 foi realizada a sua medição, onde o código dá a tensão que está saindo do sensor, a sua conversão em partes por milhão (ppm) e se o ambiente está seguro ou não com esta concentração de partículas nocivas. A Figura 36 mostra o funcionamento do teste do sensor MQ-9.



**Figura 36** A figura a mostra a) leitura realizada pelo sensor MQ-9 e a figura b) mostra a montagem [Autor]

A função `calcularPPMCO()` é responsável por converter a tensão medida pelo sensor MQ-9 em ppm de monóxido de carbono (CO). Essa função recebe um parâmetro “tensao”, que representa a tensão medida pelo sensor MQ-9, e retorna o valor correspondente em ppm de CO. A função `calcularPPMCO()` é mostrada no trecho de código a seguir.

```
float calcularPPMCO(float tensao) {
    // Curva de calibração do sensor MQ-9 para
    // o nível de CO em ppm
    float curva[3] = {2.3, 0.72, -0.34};

    // Cálculo do nível de CO em ppm
    float ppmCO = curva[0] * pow(tensao, 2) +
    curva[1] * tensao + curva[2];

    return ppmCO;
}
```

A conversão é realizada utilizando uma curva de calibração específica para o sensor MQ-9, que relaciona a tensão de saída do sensor com a concentração de CO. A curva de

calibração é representada por um array curva contendo três valores: curva[0], curva[1] e curva[2].

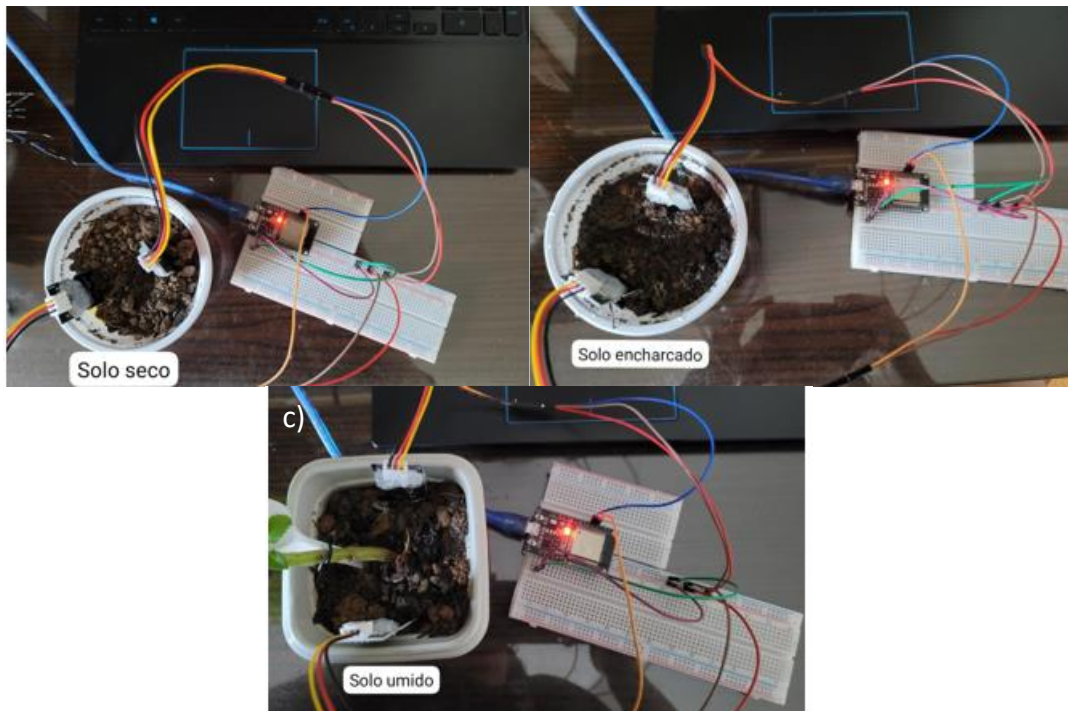
O cálculo do nível de CO em ppm é feito pela seguinte fórmula:

$$\text{ppmCO} = \text{curva}[0] * \text{pow}(\text{tensao}, 2) + \text{curva}[1] * \text{tensao} + \text{curva}[2]$$

Nessa fórmula, pow(tensao, 2) representa a elevação da tensão ao quadrado, e curva[0], curva[1] e curva[2] são os coeficientes da curva de calibração que são obtidos através da curva apresentada no datasheet. Portanto, essa função utiliza uma equação quadrática para estimar o valor em ppm de CO com base na tensão medida pelo sensor MQ-9.

## 6.4. UMIDADE DO SOLO

Para a calibração do sensor de umidade de solo foi realizado teste para verificar a tensão fornecida pelos sensores com o solo seco, úmido e encharcado de água. Com os sensores de umidade de solo não é possível realizar o cálculo do valor médio lido entre os dois sensores, pois mesmo sendo o mesmo tipo de sensor, em todas as leituras os sensores fornecem valores diferentes, tendo uma alta variação no valor lido. Na Figura 37 é mostrado os testes realizados com o sensor de umidade do solo.



**Figura 37** A figura a) mostra os sensores de umidade do solo em um solo seco, a figura b) mostra os sensores num solo encharcado, enquanto a figura c) mostra os testes sendo realizados num solo umido [Autor]

É importante destacar que os testes realizados no solo são necessários ser refeitos toda vez que troca o tipo de solo em que se está trabalhando, pois os sensores de solo capacitivo variam o seu valor analógico lido para cada tipo de solo.

Após realizar todos os testes nos sensores e atuadores a implementação e a fase de verificação experimental foram concluídas, dando seguimento para a fase de conclusões e trabalhos futuros, que se encontram no capítulo a seguir.



## 7. CONCLUSÕES E TRABALHOS FUTUROS

É possível realizar um sistema visando o controle e monitorização dos dados de uma estufa tanto em âmbito residencial como em âmbito de larga escala. Podendo este trabalho ser aplicado e vendido para restaurantes que produzem os seus próprios alimentos.

O sistema construído além de permitir a visualização em tempo real, com até 20 indicadores permitidos pelo aplicativo Cayenne gratuitamente, permite o controle manual da estufa, garantindo não só uma forma automatizada, como também uma forma prática de se realizar o processo confortavelmente. Pode-se realizar o procedimento uma vez montado, de qualquer lugar do mundo.

Como forma de ministrar os dados de maneira com que além de visuais, os dados também fossem armazenados de forma segura, o banco de dados feito em SQL trouxe uma grande facilidade, visto que sua arquitetura permite trabalhar com grande volume de dados e de forma gratuita.

É possível dizer que é um sistema de baixo custo já que utiliza elementos de baixo custo, e plataformas gratuitas.

É imprescindível dizer que o sistema de monitorização do protocolo TCP-IP entre a ESP e o Cliente traz uma segurança maior, principalmente para sistema do agronegócio, visto que vazamento de dados é algo preocupante entre quaisquer usuários, principalmente aqueles com grande volume de dados.

Por fim, os trabalhos futuros que podem ser realizados para dar continuidade a este projeto e aprimorá-lo, são: a automatização de uma página web para conter opções de hortaliças a serem cultivadas pelo responsável pela estufa, onde, ao clicar na hortaliça desejada, os sensores se ajustam automaticamente para as melhores condições para o cultivo da planta; adicionar atuadores para esquentar a estufa; colocar câmeras dentro da estufa para assim ser possível acompanhar o desenvolvimento da planta remotamente; e realizar a validação do projeto através de pessoas que possuem estufas.

## Referências Documentais

- [1] ALMEIDA, Rafael Chiaradia – Internet das Coisas e as Agtechs no Brasil. 2019, <http://ri.ufs.br/jspui/handle/riufs/14165>.
- [2] ARDUINO – Arduino - Home. 2023, <https://www.arduino.cc/>.
- [3] AWS – O que é MQTT? – Explicação sobre o protocolo MQTT – AWS, <https://aws.amazon.com/pt/what-is/mqtt/>.
- [4] AWS – O que é um ataque DDoS e como proteger seu site contra um deles, Disponível em: <https://aws.amazon.com/pt/shield/ddos-attack-protection/>.
- [5] AWS – O que é um Data Warehouse? | Conceitos principais | Amazon Web Services. 2022, <https://aws.amazon.com/pt/data-warehouse/>.
- [6] BANZI, Massimo; SHILOH, Michael – Primeiros passos com o Arduino. São Paulo: Novatec, p. p1, 2011, <https://s3.novatec.com.br/capitulos/capitulo-9788575222904.pdf>.
- [7] BARROS, Emerson de – Estudo de vulnerabilidades em dispositivos IoT TCP/IP. 2021, <https://www.lume.ufrgs.br/handle/10183/222479>.
- [8] BLUETOOTH – The 3 Updates Application Developers Need To Know About Bluetooth® Core Specification Version 5.3 | Bluetooth® Technology Website. 2022, <https://www.bluetooth.com/bluetooth-resources/the-3-updates-application-developers-need-to-know-about-bluetooth%ef%b8%8e-core-specification-version-5-3/>.
- [9] CARVALHO, André Ferreira Almeida de; SANTOS, Christyan Matteus Lima; GONÇALVES, Lucas Vaz – Segurança em IoT. 2022, <https://dspace.uniceplac.edu.br/handle/123456789/1610>.
- [10] CAYENNE – Developer | myDevices.com. 2023, <https://developers.mydevices.com/cayenne/features/>.
- [11] CONCEIÇÃO, Carlos Alexandre da – Carga de vento atuante em uma estufa agrícola no município de Alegrete-RS. 2022, <https://repositorio.unipampa.edu.br/jspui/handle/riu/6859>.
- [12] COSTA, Cainã L.; OLIVEIRA, Letícia; MÓTA, LS Michele – Internet das coisas (IOT): um estudo exploratório em agronegócios Internet of Things (IOT): na exploratory study in agribusiness. VI Simpósio da Cienc. do Agronegócio, 2018, <https://www.ufrgs.br/cienagro/wp-content/uploads/2018/10/Internet-das-coisas-IOT-um-estudo-explorat%C3%B3rio-em-agroneg%C3%B3cios.docx-Cain%C3%A3-Lima-Costa.pdf>.

- [13] DA SILVA, Rodrigo Sencadas – Sistema de Controlo Eletrónico de uma Estufa. 2019. Dissertação de Mestrado. Instituto Politécnico do Porto (Portugal), <http://hdl.handle.net/10400.22/15677>.
- [14] DE OLIVEIRA, N.; GOMES, M.; LOPES, R.; NOBRE, J. – Segurança da Informação para Internet das Coisas (IoT): uma Abordagem sobre a Lei Geral de Proteção de Dados (LGPD). Revista Eletrônica de Iniciação Científica em Computação, [S. l.], v. 17, n. 4, 2019. DOI: 10.5753/reic.2019.1704, <https://sol.sbc.org.br/journals/index.php/reic/article/view/1704>.
- [15] EMBRAPA – Condições favoráveis: Estufas plásticas para cultivos diversos. - Portal Embrapa. 2020, <https://www.embrapa.br/busca-de-publicacoes/-/publicacao/1119857/condicoes-favoraveis-estufas-plasticas-para-cultivos-diversos>.
- [16] FACCIONI FILHO, Mauro – Internet das coisas. Unisul Virtual, 2016, [https://www.researchgate.net/profile/Mauro-Fazion-Filho/publication/319881659\\_Internet\\_das\\_Coisas\\_Internet\\_of\\_Things/links/59c038d5458515e9cfd54ff9/Internet-das-Coisas-Internet-of-Things.pdf](https://www.researchgate.net/profile/Mauro-Fazion-Filho/publication/319881659_Internet_das_Coisas_Internet_of_Things/links/59c038d5458515e9cfd54ff9/Internet-das-Coisas-Internet-of-Things.pdf).
- [17] FALOHUN, A. S. et al – Dangerous gas detection using an integrated circuit and MQ-9. International Journal of Computer Applications, v. 135, n. 7, p. 30-34, 2016, [https://www.researchgate.net/profile/Falohun-As/publication/295278328\\_Dangerous\\_Gas\\_Detection\\_using\\_an\\_Integrated\\_Circuit\\_and\\_MQ-9/links/56cae7e708aee3cee541204d/Dangerous-Gas-Detection-using-an-Integrated-Circuit-and-MQ-9.pdf](https://www.researchgate.net/profile/Falohun-As/publication/295278328_Dangerous_Gas_Detection_using_an_Integrated_Circuit_and_MQ-9/links/56cae7e708aee3cee541204d/Dangerous-Gas-Detection-using-an-Integrated-Circuit-and-MQ-9.pdf).
- [18] GOVERNO FEDERAL BRASILEIRO. Lei Geral de Proteção de Dados Pessoais (LGPD), 2021, <https://www.gov.br/cidadania/pt-br/aceso-a-informacao/lgpd>.
- [19] HAYASHI, Victor T.; ALMEIDA, Felipe V. de; KOMO, Andrea E. – Segurança em Internet das Coisas: Uma Revisão da Literatura. In: ESCOLA REGIONAL DE REDES DE COMPUTADORES (ERRC), 19., 2021, Charqueadas/RS. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2021. p. 103-108, <https://doi.org/10.5753/errc.2021.18550>.
- [20] JESUS, Klebio de; SILVA, Ronaldo Ferreira da; VIEIRA, Cecília Cândida Frazão – APLICAÇÃO DE INTERNET DAS COISAS (IoT) NA AGRICULTURA DE PRECISÃO. 2021, <http://aprender.posse.ueg.br:8081/jspui/handle/123456789/274>.
- [21] LEITE, Leandro Rogério Corrêa – Internet das Coisas (IoT): vulnerabilidades de segurança e desafios. 2019, <http://ric.cps.sp.gov.br/handle/123456789/3978>.
- [22] MORAIS, Izabelly Soares D.; GONÇALVES, Priscila de F.; LEDUR, Cleverson L.; et al – Introdução a Big Data e Internet das Coisas (IoT). Grupo A, 2018. E-book. ISBN 9788595027640, <https://app.minhabiblioteca.com.br/#/books/9788595027640/>.
- [23] MOTRONI, Andrea; BUFFI, Alice; NEPA, Paulo – A Survey on Indoor Vehicle Localization Through RFID Technology. Acesso em IEEE, v. 9, p. 17921-17942, 2021. doi: 10.1109/ACCESS.2021.3052316, <https://ieeexplore.ieee.org/abstract/document/9328120>.

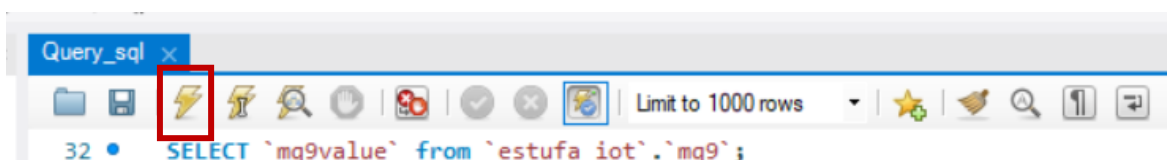
- [24] NASCIMENTO, Hugo Gomes do – Avaliação e prospecção do uso da IoT no setor agrícola no município de Petrolina/PE. TCC (Bacharelado em Agronomia) - Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano, Campus Petrolina Zona Rural, Petrolina, PE, 23 f., 2022, <http://hdl.handle.net/123456789/843>.
- [25] NASCIMENTO, Kiakisiki Quaresma do – Implementação das estufas agrícolas em São Tomé e Príncipe (STP): impacto social e económico nas comunidades rurais. 2019. Dissertação de Mestrado. Universidade de Évora, <http://hdl.handle.net/10174/26209>.
- [26] OLIVEIRA, Sérgio de – Internet das Coisas com ESP8266, Arduino e Raspberry Pi. [S.L]: Novatec, 2017, <https://books.google.com.br/books?id=vBjfJOnE1foC&pg=PA12&lpg=PA12&dq=practical+arduino#v=onepage&q=practical%20arduino&f=false>.
- [27] ORVALHO, João P. iCatch – Sistema de intelligence para caracterização de ciberataques multi-etapa. Ipbeja.pt, 2019, <http://hdl.handle.net/20.500.12207/5297>.
- [28] ONU – Haiti está entre os 18 focos de fome e situação deverá piorar. 31 mai. 2023, <https://news.un.org/pt/story/2023/05/1815122>.
- [29] ONU – População mundial atinge 8 bilhões de pessoas. 15 nov. 2022, <https://news.un.org/pt/story/2022/11/1805342#:~:text=ONU%20ressalta%20que%20n%C3%BAmero%20de,a%209%20bilh%C3%B5es%20em%202037>.
- [30] ONU – Um mês após leve alta, preço dos alimentos volta a cair em maio. 2 jun. 2023, <https://news.un.org/pt/story/2023/06/1815307>.
- [31] PAUFERRO, Gabriel Brogno Alcantara; DE PAIVA, Seila Vasti Faria; LESSA, Nayari Marie – IoT: conceitos de segurança de dados e criptografia. Cogitare, v. 3, n. 2, pág. 40-52, 2020, <https://ojs.ifsp.edu.br/index.php/cogitare/article/view/1483>.
- [32] PEREIRA, R. M.; SANDRI, D.; SILVA JÚNIOR, J. J. da – Evaluation of low-cost capacitive moisture sensors in three types of soils in the Cerrado, Brazil. Revista Engenharia na Agricultura - REVENG, [S. l.], v. 30, n. Contínua, p. 262–272, 2022. DOI: 10.13083/reveng.v30i1.14017, <https://periodicos.ufv.br/reveng/article/view/14017>.
- [33] ROBOCORE – ESP32 - WiFi + Bluetooth – RoboCore. 2022, <https://www.robocore.net/wifi/esp32-wifi-bluetooth>.
- [34] ROBOCORE – Sensor DHT11 de Temperatura e Umidade - RoboCore. 2022, <https://www.robocore.net/sensor-ambiente/sensor-de-temperatura-dht11>.
- [35] RUANO, T.; SERRÃO, L. DA S.; SANTANA, K.; NASCIMENTO, D. A. DO; TRUQUETE, M. K – PID Algorithm Applied on Led Lamps Luminosity Control. International Journal of Advanced Engineering Research and Science, v. 6, n. 12, 7 dez. 2019, <http://journal-repository.theshillonga.com/index.php/ijaers/article/view/1384>.

- [36] SANTOS, B. P., Silva, L. A., Celes, C. S., Borges Neto, J. B., Peres, B. S., Vieira, M. A. M., ... & Loureiro, A. A. – Internet das coisas: da teoria à prática. 2016, <https://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf>.
- [37] SANTOS, Diego Zeuner Fagundes – Controle de luminosidade de alta eficiência, adaptável ao ambiente, utilizando lâmpada de led. 2012, <https://repositorio.uniceub.br/jspui/handle/123456789/3100>.
- [38] SEUNGHWAN, Won; CHOI, Sang Won – Three Decades of 3GPP Target Cell Search through 3G, 4G, and 5G. Acesso em IEEE, vol. 8, pp. 116914-116960, 2020, doi: 10.1109/ACCESS.2020.3003012, <https://ieeexplore.ieee.org/abstract/document/9119377>.
- [39] SLIDESGO – Free estufa Google Slides themes and PowerPoint templates for presentations, <https://slidesgo.com/search?q=estufa#rs=search>.
- [40] TIMMIS, Harold – Practical Arduino Engineering. [S.L]: Technology In Action, 2011. 307 p, <https://books.google.com.br/books?id=vBjfJOnE1foC&pg=PA12&lpg=PA12&dq=practical+arduino#v=onepage&q=practical%20arduino&f=false>.
- [41] VIEIRA, Rony Anderson Uliana – Aplicação de processamento de imagens no controle de planta modelo: interoperabilidade com controles industriais utilizando comunicação Ethernet. 2022, <https://repositorio.ifes.edu.br/handle/123456789/1976>.

## Anexo A. Query utilizada no SQL

Neste anexo são descritos alguns dos pormenores do desenvolvimento do banco de dados MySQL local.

Para executar trechos isolados de código presente numa mesma query, é necessário apenas seleccionar o trecho desejado e clicar no botão com formato de raio, como pode ser visto na Figura 38.



**Figura 38** Botão para executar a query [Autor]

A seguir tem-se o código completo utilizado para criar o banco de dados SQL.

```
# Criação do banco, das tabelas e das
variáveis

create schema estufa_iot;

# Criação da tabela dht11:
CREATE TABLE `estufa_iot`.`dht11` (
  `id_dht11` INT NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id_dht11`));

ALTER TABLE `estufa_iot`.`dht11`
ADD COLUMN `umidadeInterno1` FLOAT NULL AFTER
`id_dht11`,
ADD COLUMN `temperaturaInterno1` FLOAT NULL
AFTER `umidadeInterno1`,
ADD COLUMN `umidadeInterno2` FLOAT NULL AFTER
`temperaturaInterno1`,
ADD COLUMN `temperaturaInterno2` FLOAT NULL
AFTER `umidadeInterno2`,
ADD COLUMN `umidadeExterno3` FLOAT NULL AFTER
`temperaturaInterno2`,
```

```

ADD COLUMN `temperaturaExterno3` FLOAT NULL
AFTER `umidadeExterno3`,
ADD COLUMN `coolers` VARCHAR(45) NULL AFTER
`temperaturaExterno3`;

# Criação da tabela mq9:
CREATE TABLE `estufa_iot`.`mq9` (
  `id_mq9` INT NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id_mq9`));

ALTER TABLE `estufa_iot`.`mq9`
ADD COLUMN `mq9value` FLOAT NULL AFTER
`id_mq9`,
ADD COLUMN `data` VARCHAR(255) NULL AFTER
`mq9value`;

INSERT INTO `estufa_iot`.`mq9` (data)
VALUES (CURRENT_TIMESTAMP());

# Para realizar visualizações na tabela:
SELECT * from `estufa_iot`.`mq9`;
SELECT `mq9value` from `estufa_iot`.`mq9`;

# Para apagar os dados da tabela:
TRUNCATE table `estufa_iot`.`mq9`;

# Criação da tabela ldr:
CREATE TABLE `estufa_iot`.`ldr` (
  `id_ldr` INT NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id_ldr`));

ALTER TABLE `estufa_iot`.`ldr`
ADD COLUMN `isDark` BOOLEAN NULL AFTER
`id_ldr`,
ADD COLUMN `data` BOOLEAN NULL AFTER
`isDark`;

INSERT INTO `estufa_iot`.`LDR` (data)
VALUES (CURRENT_TIMESTAMP());

# Criação da tabela umidade_solo:
CREATE TABLE `estufa_iot`.`umidade_solo` (
  `id_umidade_solo` INT NOT NULL
AUTO_INCREMENT,
  PRIMARY KEY (`id_umidade_solo`));

```

```
ALTER TABLE `estufa_iot`.`umidade_solo`
ADD COLUMN `leitura_sensor1` FLOAT NULL AFTER
`id_umidade_solo`,
ADD COLUMN `umidade_sensor1` FLOAT NULL AFTER
`leitura_sensor1`,
ADD COLUMN `leitura_sensor2` FLOAT NULL AFTER
`umidade_sensor1`,
ADD COLUMN `umidade_sensor2` FLOAT NULL AFTER
`leitura_sensor2`,
ADD COLUMN `bomba` VARCHAR(45) NULL AFTER
`umidade_sensor2`;
```

O trecho “INSERT INTO `estufa\_iot`.`LDR`(data) VALUES(CURRENT\_TIMESTAMP);” é responsável por incluir uma data e hora no campo data presente na tabela LDR do banco de dados estufa\_iot, toda vez que for selecionado e executado.

O trecho “SELECT \* from `estufa\_iot`.`mq9`;” seleciona todas as colunas e todos os dados presente na tabela mq9, que se encontra no banco de dados estufa\_iot e mostra o seu conteúdo.

Já o trecho “SELECT `mq9value` from `estufa\_iot`.`mq9`;” mostra o conteúdo presente apenas na coluna mq9value da tabela mq9 do banco de dados estufa\_iot. Enquanto o trecho “TRUNCATE table `estufa\_iot`.`mq9`;” apaga toda a tabela mq9.

## Anexo B. Código em C++ para o Arduino

Neste anexo é apresentado o código principal do projeto, escrito na plataforma Arduino em C++.

```
// Inclusão das bibliotecas
#include <DHT.h>
#include <Wire.h>
#include <CayenneMQTTESP32.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
```

```

#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>

// Definição dos pinos
#define DHT_PIN_INTERNO_1 2 // Pino do
sensor DHT11 interno 1
#define DHT_PIN_INTERNO_2 4 // Pino do
sensor DHT11 interno 2
#define DHT_PIN_EXTERNO_3 12 // Pino do
sensor DHT11 externo
const int LDR_PIN = 18; // Pino D5
(GPIO 5) para conexão do sensor LDR
const int MQ9_PIN = A5; // Pino 33
para conexão do sensor MQ-9
const int SENSOR1_PIN = A6; // Pino 34
para conexão do sensor de umidade do solo 1
const int SENSOR2_PIN = A7; // Pino 35
para conexão do sensor de umidade do solo 2
const int BOMBA_PIN = 5; // Pino
conectado à bomba d'água
const int VENTOINHA_PIN = 22; // Pino de
controle da ventoinha conectada ao pino 22

#define DHT_TYPE DHT11 // Define o
tipo de sensor DHT como DHT11

// Informações da rede WiFi
char ssid[] = "Luis"; //"Isa M.";
char wifiPassword[] = "*Paodequeijo";
//"nkkp3363";

// Informações de autenticação do Cayenne,
obtidas no Painel de Controle do Cayenne
char username[] = "4f818d40-2b3a-11eb-8779-
7d56e82df461";
char password[] =
"364ff1b84ef8a63532a79a7e80c8f3a6990ae93f";
char clientID[] = "dc83d9e0-0d74-11ee-8485-
5b7d3ef089d0";

// DECLARAÇÃO DE VARIÁVEIS PARA MySQL
IPAddress server_addr(85, 10, 205, 173); //
IP of the MySQL *server* here
char INSERT_SQL_MQ9[] = "INSERT INTO
estufa_iot.mq9 (mq9value) VALUES ('%f')";

```

```

char INSERT_SQL_LDR[] = "INSERT INTO
estufa_iot.ldr (ldrValue) VALUES ('%i')";
char INSERT_SQL_ventoinha[] = "INSERT INTO
estufa_iot.dht11 (VENTOINHA_PIN) VALUES
('%d')";
char INSERT_SQL_temp_externo[] = "INSERT INTO
estufa_iot.dht11 (temperatureExterno3) VALUES
('%f')";
char INSERT_SQL_temp_interno1[] = "INSERT
INTO estufa_iot.dht11 (temperatureInterno1)
VALUES ('%f')";
char INSERT_SQL_temp_interno2[] = "INSERT
INTO estufa_iot.dht11 (temperatureInterno2)
VALUES ('%f')";
char INSERT_SQL_umid_externo[] = "INSERT INTO
estufa_iot.dht11 (humidityExterno3) VALUES
('%f')";
char INSERT_SQL_umid_interno1[] = "INSERT
INTO estufa_iot.dht11 (humidityInterno1)
VALUES ('%f')";
char INSERT_SQL_umid_interno2[] = "INSERT
INTO estufa_iot.dht11 (humidityInterno2)
VALUES ('%f')";
char INSERT_SQL_bomba[] = "INSERT INTO
estufa_iot.umidade_solo (BOMBA_PIN) VALUES
('%d')";
char INSERT_SQL_sensor_solo1[] = "INSERT INTO
estufa_iot.umidade_solo (sensor1Value) VALUES
('%f')";
char INSERT_SQL_porcentagem_sensor1[] =
"INSERT INTO estufa_iot.umidade_solo
(percent_sensor1) VALUES ('%f')";
char INSERT_SQL_sensor_solo2[] = "INSERT INTO
estufa_iot.umidade_solo (sensor2Value) VALUES
('%f')";
char INSERT_SQL_porcentagem_sensor2[] =
"INSERT INTO estufa_iot.umidade_solo
(percent_sensor2) VALUES ('%f')";
char query[128];
char login_banco_sql[] = "estufando";
char senha_banco_sql[] = "12C4ractere$";

// INSTANCIANDO OBJETOS
WiFiClient client;
MySQL_Connection conn((Client *)&client);

```

```

// Inicializa os sensores DHT11
DHT dhtInterno1(DHT_PIN_INTERNO_1, DHT_TYPE);
DHT dhtInterno2(DHT_PIN_INTERNO_2, DHT_TYPE);
DHT dhtExterno3(DHT_PIN_EXTERNO_3, DHT_TYPE);

// Temperatura limite para ligar a ventoinha
//const float TEMPERATURA_LIMITE = 21.5;
int TEMPERATURA_LIMITE;

// Valores de calibração do sensor de umidade
do solo
const int VALOR_MINIMO1 = 871; // Solo
encharcado (apenas como parâmetro de análise
não é considerado em contas)
const int VALOR_MINIMO2 = 923; // Solo
encharcado (apenas como parâmetro de análise
não é considerado em contas)
const int VALOR_UMIDO1 = 1180; // Solo
úmido
const int VALOR_UMIDO2 = 1303; // Solo
úmido
const int VALOR_MAXIMO1 = 2115; // Solo seco
const int VALOR_MAXIMO2 = 2049; // Solo seco

// Concentração máxima desejada de água no
solo
const int CONCENTRACAO_MAXIMA1 = 1400;
const int CONCENTRACAO_MAXIMA2 = 1500;

// Nível máximo de partículas no ar
const int NIVEL_PARTICULAS = 39;

// Variáveis globais para armazenar as
leituras dos sensores
float humidityInterno1, temperatureInterno1;
float humidityInterno2, temperatureInterno2;
float humidityExterno3, temperatureExterno3;
int ldrValue;
float mq9Value;
float sensor1Value; // sensor de umidade do
solo
float sensor2Value; // sensor de umidade do
solo
float percent_sensor1;
float percent_sensor2;

```

```

float ppmCO;
float tensao;

// Função para leitura dos sensores
void sensorReadings() {

    Serial.println("Estados dos sensores:");
    Serial.println();

    humidityInterno1 =
dhtInterno1.readHumidity();
    temperatureInterno1 =
dhtInterno1.readTemperature();
    Serial.print("Humidity Interno 1: ");
    Serial.print(humidityInterno1);
    Serial.println("%");
    Serial.print("Temperature Interno 1: ");
    Serial.print(temperatureInterno1);
    Serial.println("°C");

    humidityInterno2 =
dhtInterno2.readHumidity();
    temperatureInterno2 =
dhtInterno2.readTemperature();
    Serial.print("Humidity Interno 2: ");
    Serial.print(humidityInterno2);
    Serial.println("%");
    Serial.print("Temperature Interno 2: ");
    Serial.print(temperatureInterno2);
    Serial.println("°C");

    humidityExterno3 =
dhtExterno3.readHumidity();
    temperatureExterno3 =
dhtExterno3.readTemperature();
    Serial.print("Humidity Externo 3: ");
    Serial.print(humidityExterno3);
    Serial.println("%");
    Serial.print("Temperature Externo 3: ");
    Serial.print(temperatureExterno3);
    Serial.println("°C");

    ldrValue = !digitalRead(LDR_PIN);
    Serial.print("LDR Value inverted: ");

```

```

Serial.println(ldrValue);

tensao = mq9Value * (3.3 / 4095);
ppmCO = calcularPPMCO(tensao); // Função
para calcular o nível de CO em ppm

mq9Value = analogRead(MQ9_PIN);
Serial.print("MQ9 Value: ");
Serial.println(mq9Value);

Serial.print("MQ9 ppm: ");
Serial.println(ppmCO);

sensor1Value = analogRead(SENSOR1_PIN);
Serial.print("Sensor 1 Value: ");
Serial.println(sensor1Value);

sensor2Value = analogRead(SENSOR2_PIN);
Serial.print("Sensor 2 Value: ");
Serial.println(sensor2Value);

// Realiza a leitura do Sensor 1 em %
percent_sensor1 = 100-map(sensor1Value,
4095, 0, 100, 0); // a função map apenas
contabiliza quão seco está o local sensoriado
Serial.print("Percent Sensor 1: ");
Serial.print(percent_sensor1);
Serial.println("%");

// Realiza a leitura do Sensor 2 em %
percent_sensor2 = 100-map(sensor2Value,
4095, 0, 100, 0); // a função map apenas
contabiliza quão seco está o local sensoriado
Serial.print("Percent Sensor 2: ");
Serial.print(percent_sensor2);
Serial.println("%");

// Linha adicional para separar as leituras
Serial.println("_____
_____");
Serial.println();

```

```

}

// Função para controlar a ventoinha e a
bomba d'água
void Control() {
    // Controla a ventoinha com base na
    temperatura
    Serial.println("Estados dos relés:");
    Serial.println();

    if (temperatureInterno1 >
TEMPERATURA_LIMITE || temperatureInterno2 >
TEMPERATURA_LIMITE) {
        digitalWrite(VENTOINHA_PIN, LOW); //
Liga a ventoinha
        Serial.println("Ventoinha Ligada");
    } else {
        digitalWrite(VENTOINHA_PIN, HIGH); //
Desliga a ventoinha
        Serial.println("Ventoinha Desligada");
    }

    // Controla a bomba d'água com base na
    umidade do solo
    if (sensor1Value > VALOR_UMIDO1 &&
sensor2Value > VALOR_UMIDO2) {
        digitalWrite(BOMBA_PIN, LOW); // Liga a
bomba d'água
        Serial.println("Bomba Ligada");
    } else {
        digitalWrite(BOMBA_PIN, HIGH); //
Desliga a bomba d'água
        Serial.println("Bomba Desligada");
    }

    Serial.println("
_____
_____");
}

// Função para publicar os dados no Cayenne
MQTT
void publishData() {

```

```

    // Calcula a média da temperatura dos
    sensores externos
    float temperaturaMediaInterna =
    (temperatureInterno1 + temperatureInterno2) /
    2.0;
    // Calcula a média das umidades dos
    sensores internos
    float umidadeMediaInterna =
    (humidityInterno1 + humidityInterno2) / 2.0;

    // Publica as leituras dos sensores no
    Cayenne
    Cayenne.virtualWrite(0,
    temperatureInterno1, "temp", "c");
    Cayenne.virtualWrite(1, humidityInterno1,
    "rel_hum", "p");
    Cayenne.virtualWrite(2,
    temperatureInterno2, "temp", "c");
    Cayenne.virtualWrite(3, humidityInterno2,
    "rel_hum", "p");
    Cayenne.virtualWrite(4,
    temperaturaMediaInterna, "temp", "c"); //
    Envia a média da temperatura dos sensores
    externos no canal 4
    Cayenne.virtualWrite(5,
    umidadeMediaInterna, "rel_hum", "p"); //
    Envia a média das umidades dos sensores
    internos no canal 5
    Cayenne.virtualWrite(6, temperatureExterno3,
    "temp", "c");
    Cayenne.virtualWrite(7, humidityExterno3,
    "rel_hum", "p");
    Cayenne.virtualWrite(8, ldrValue, "lum",
    "d");
    Cayenne.virtualWrite(9, ppmCO, "co",
    "ppm");
    Cayenne.virtualWrite(10, percent_sensor1,
    "soil_moist", "p");
    Cayenne.virtualWrite(11, percent_sensor2,
    "soil_moist", "p");

    //teste apenas para saber se o envio está
    sendo feito corretamente, o número a ser
    utilizado é o NIVEL_PARTICULAS
    if (ppmCO >= 3) {
        Serial.println(" - Perigoso (acima do
        limite recomendado para humanos)");
    }

```

```

        Serial.println("Alerta de incendio!");
        Cayenne.virtualWrite(14, HIGH, "co",
"d");
    } else {
        Serial.println(" - Seguro");
        Cayenne.virtualWrite(14, LOW, "co", "d");
    }
}

AsyncWebServer server(80);

// Para obtenção do endereço IP e porta do
cliente
void handleRequest(AsyncWebServerRequest
*request) {
    // Obtém informações da requisição
    String method = request->methodToString();
    String url = request->url();
    String userAgent = request->header("User-
Agent");
    String content = request->hasParam("plain")
? request->getParam("plain")->value() : "";

    // Obtém o endereço IP e a porta do cliente
    String clientIP = request->client()-
>remoteIP().toString();
    uint16_t clientPort = request->client()-
>remotePort();

    // Imprime as informações da requisição
    Serial.println("Nova requisição:");
    Serial.println("Método: " + method);
    Serial.println("URL: " + url);
    Serial.println("User-Agent: " + userAgent);
    Serial.println("Corpo: " + content);
    Serial.println("Endereço IP do cliente: " +
clientIP);
    Serial.println("Porta do cliente: " +
String(clientPort));
    Serial.println();
}

```

```

Serial.println("_____
_____");

    // Responde com uma mensagem de sucesso
    request->send(200, "text/plain",
"Requisição recebida com sucesso!");
}

void enviaDados() { // envia os dados para o
banco de dados
    Serial.print("MQ9 ppm p/ sql: ");
    Serial.println(ppmCO);

    //MQ-9
    sprintf(query, INSERT_SQL_MQ9, ppmCO);
    MySQL_Cursor *cur_mem = new
MySQL_Cursor(&conn);
    cur_mem->execute(query);

    // LDR:
    sprintf(query, INSERT_SQL_LDR, ldrValue);
    cur_mem->execute(query);

    delay(200);

// Umidade do solo:
    // Bomba
    sprintf(query, INSERT_SQL_bomba,
digitalRead(BOMBA_PIN));
    //Serial.print("sprintf: ");
    //Serial.println(query);
    cur_mem->execute(query);

    // Tensão lida no sensor de umidade do solo
1:
    sprintf(query, INSERT_SQL_sensor_solo1,
sensor1Value);
    cur_mem->execute(query);

    // Porcentagem de água lida no sensor de
umidade do solo 1:

```

```

    sprintf(query,
INSERT_SQL_porcentagem_sensor1,
percent_sensor1);
    cur_mem->execute(query);

    // Tensão lida no sensor de umidade do solo
2:
    sprintf(query, INSERT_SQL_sensor_solo2,
sensor2Value);
    cur_mem->execute(query);

    // Porcentagem de água lida no sensor de
umidade do solo 2:
    sprintf(query,
INSERT_SQL_porcentagem_sensor2,
percent_sensor2);
    cur_mem->execute(query);

    delay(200);

// DHT11:
// Ventoinha:
    sprintf(query, INSERT_SQL_ventoinha,
digitalRead(VENTOINHA_PIN));
    //Serial.print("sprintf: ");
    //Serial.println(query);
    cur_mem->execute(query);

    // Temperatura externo:
    sprintf(query, INSERT_SQL_temp_externo,
temperatureExterno3);
    cur_mem->execute(query);

    // Temperatura interno 1:
    sprintf(query, INSERT_SQL_temp_interno1,
temperatureInterno1);
    cur_mem->execute(query);

    // Temperatura interno 2:

```

```

    sprintf(query, INSERT_SQL_temp_interno2,
temperatureInterno2);
    cur_mem->execute(query);

    // Umidade externo:
    sprintf(query, INSERT_SQL_umid_externo,
humidityExterno3);
    cur_mem->execute(query);

    // Umidade interno 1:
    sprintf(query, INSERT_SQL_umid_interno1,
humidityInterno1);
    cur_mem->execute(query);

    // Umidade interno 2:
    sprintf(query, INSERT_SQL_umid_interno2,
humidityInterno2);
    cur_mem->execute(query);

    delete cur_mem;
    Serial.println("Informações Enviadas");
}

void setup() {
    // Inicializa a comunicação serial
    Serial.begin(9600);

    // Inicializa os sensores DHT11
    dhtInterno1.begin();
    dhtInterno2.begin();
    dhtExterno3.begin();

    // Inicializa a conexão WiFi
    Cayenne.begin(username, password, clientID,
ssid, wifiPassword);

    Serial.println("Conectado ao WiFi");
    Serial.println("Endereço IP: " +
WiFi.localIP().toString());
}

```

```

server.onNotFound([] (AsyncWebServerRequest
*request) {
    request->send(404, "text/plain", "Página
não encontrada");
});

server.onRequestBody([] (AsyncWebServerRequest
*request, uint8_t *data, size_t len, size_t
index, size_t total) {
    // Manipula o corpo da requisição, se
necessário
});

server.on("/", HTTP_GET, handleRequest);

server.begin();

// CONECTA NO MySQL
while (!conn.connect(server_addr, 3306,
login_banco_sql, senha_banco_sql)) {
    Serial.println("Conexão SQL falhou.");
    conn.close();
    delay(1000);
    Serial.println("Conectando SQL
novamente.");
}
Serial.println("Conectado ao servidor
SQL.");

// Configuração dos pinos
pinMode(BOMBA_PIN, OUTPUT);
pinMode(VENTOINHA_PIN, OUTPUT);

// Define o estado inicial dos pinos
digitalWrite(BOMBA_PIN, LOW);
digitalWrite(VENTOINHA_PIN, LOW);

delay(200);

// Define o estado inicial dos pinos
digitalWrite(BOMBA_PIN, HIGH);
digitalWrite(VENTOINHA_PIN, HIGH);

delay(200);

```

```

// Define o estado inicial dos pinos
digitalWrite(BOMBA_PIN, LOW);
digitalWrite(VENTOINHA_PIN, LOW);

delay(200);

// Define o estado inicial dos pinos
digitalWrite(BOMBA_PIN, HIGH);
digitalWrite(VENTOINHA_PIN, HIGH);

}

//botão para ligar/desligar a bomba de água
manualmente
CAYENNE_IN(12) {
    int value_bomba = getValue.asInt();
    CAYENNE_LOG("Canal %d, pino %d, valor %d",
12, BOMBA_PIN, value_bomba); // Escreva o
valor recebido no pino digital.

    digitalWrite(BOMBA_PIN, value_bomba);
    delay(200);

}

//slider para definir a temperatura limite
CAYENNE_IN(13) {
    int value_slide = getValue.asInt();
    CAYENNE_LOG("Canal %d, pino %d, valor %d",
13, TEMPERATURA_LIMITE, value_slide); //
Escreva o valor recebido no pino digital.

    TEMPERATURA_LIMITE = value_slide;

}

//botão para ligar/desligar as ventoinhas
manualmente
CAYENNE_IN(15) {
    int value_ventoinha = getValue.asInt();

```

```

    CAYENNE_LOG("Canal %d, pino %d, valor %d",
15, VENTOINHA_PIN, value_ventoinha); //
Escreva o valor recebido no pino digital.

    digitalWrite(VENTOINHA_PIN,
value_ventoinha);
    delay(200);

}

float calcularPPMCO(float tensao) {
    // Curva de calibração do sensor MQ-9 para
o nível de CO em ppm
    float curva[3] = {2.3, 0.72, -0.34};

    // Cálculo do nível de CO em ppm
    float ppmCO = curva[0] * pow(tensao, 2) +
curva[1] * tensao + curva[2];

    return ppmCO;
}

void loop() {
    // Executa as funções de leitura dos
sensores, controle de irrigação, verificação
da qualidade do ar e concentração de água no
solo a cada 2 segundos
    static unsigned long lastMillis = 0;
    if (millis() - lastMillis > 2000) {
        lastMillis = millis();

        sensorReadings();
        Control();
        enviaDados();
        publishData();

        Serial.print("Temperatura limite
utilizada: ");
        Serial.println(TEMPERATURA_LIMITE);
        Serial.println();

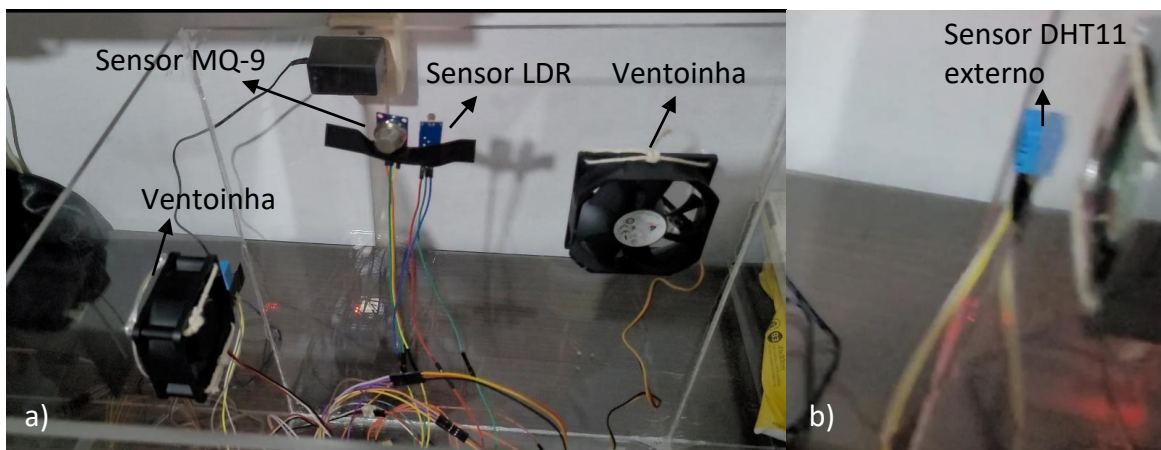
Serial.println("
_____
_____");
    }
}

```

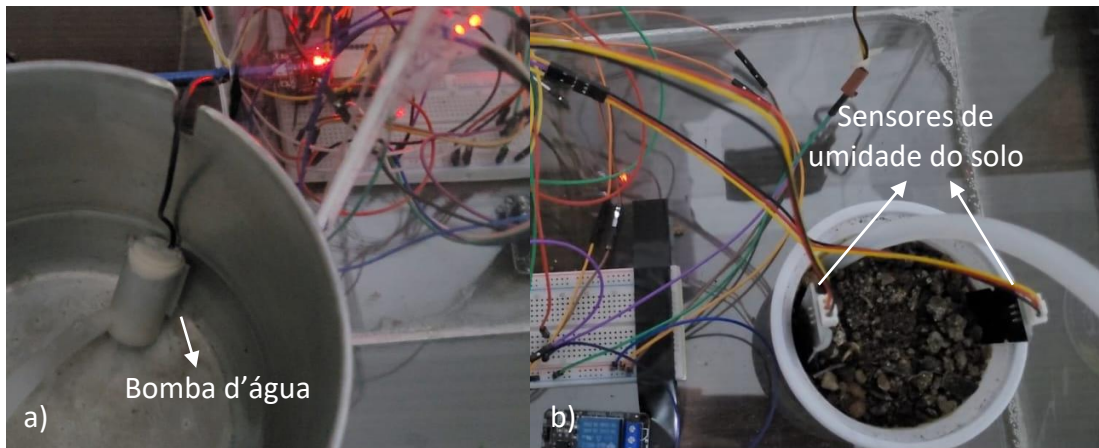
```
// Mantém a conexão com o Cayenne MQTT  
Cayenne.loop();  
}
```

## Anexo C. Vídeo com explicações sobre o projeto

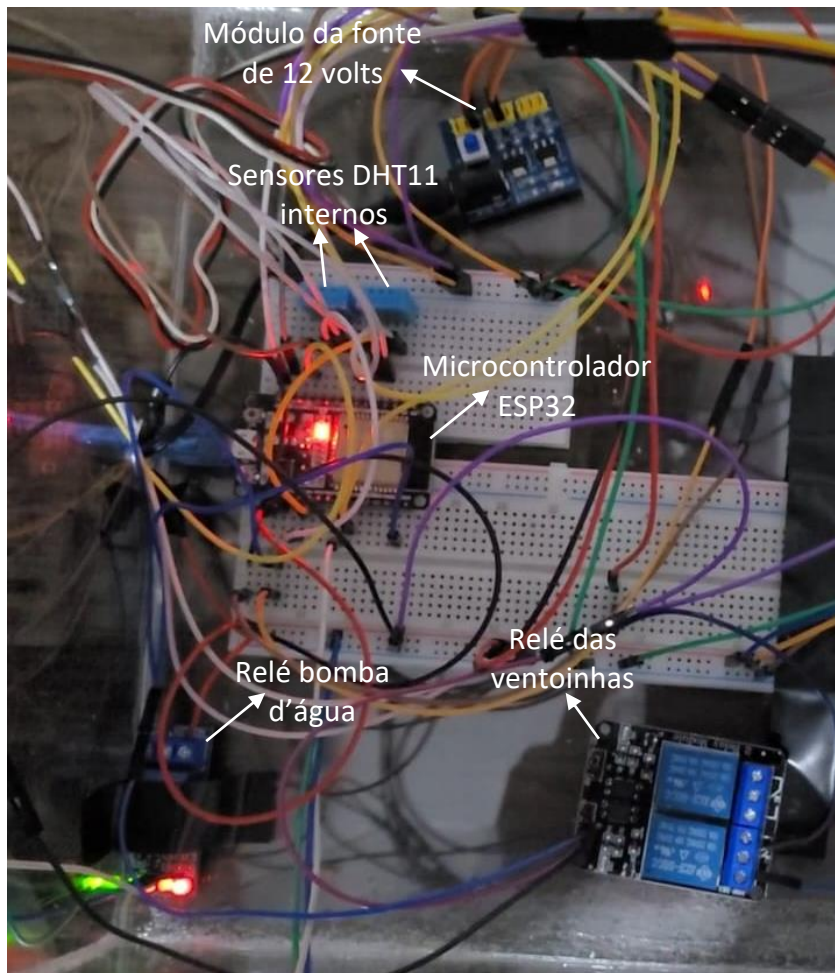
O vídeo realizado tem por objetivo explicar e mostrar o funcionamento do projeto, e pode ser acessado através do seguinte link: <https://youtu.be/cBXEFNTu5DM>. É importante ressaltar que este vídeo pode encontrar-se futuramente indisponível para o acesso. Na figura 39 é possível visualizar a localização do sensor de umidade e temperatura externo (DHT11), do sensor de luminosidade (LDR), do sensor de gás monóxido de carbono (MQ-9) e das ventoinhas. Na Figura 40 é apresentada a localização da bomba d'água e dos sensores de umidade do solo, enquanto que na Figura 41 é apresentada a localização dos sensores de umidade e temperatura internos (DHT11), do relé das ventoinhas e do relé da bomba d'água, a localização do microcontrolador e do módulo da fonte de 12 volts que é responsável por controlar a tensão enviada à partir da fonte de 12 volts.



**Figura 39** Na imagem a) é apresentada a localização das ventoinhas, do sensor LDR e do sensor MQ-9, enquanto na imagem b) é apresentada a localização do sensor DHT11 [Autor]



**Figura 40** Na figura a) está sendo apresentada a localização da bomba d'água, enquanto na figura b) mostra-se a localização dos sensores de umidade de solo [Autor]



**Figura 41** Localização dos relés dos atuadores, do microcontrolador, dos sensores DHT11 internos e do módulo da fonte de 12 volts [Autor]

