



Instituto Politécnico  
de Castelo Branco  
Escola Superior  
de Tecnologia

## **Sistema de monitorização de estufa**

Miguel Nsanda Dunge,

Pedro João Daniel

### **Orientador**

Professor Adjunto Paulo Jorge Dias Torres

Relatório de Projeto apresentado à Escola Superior de Tecnologia Unidade Técnico Científica de Engenharia Eletrotécnica e Industrial do Instituto Politécnico de Castelo Branco, para cumprimento dos requisitos necessários à obtenção do grau de Licenciado em Engenharia Eletrotécnica e Telecomunicações, realizado sob a orientação científica da Professor Adjunto Doutor Paulo Jorge Dias Torres, do Instituto Politécnico de Castelo Branco.

**Outubro de 2024**



## **Composição do júri**

Presidente do júri

Paulo Jorge Dias Torres (orientador)

Professor Adjunto do Instituto Politécnico de Castelo Branco

Vogais

Hugo Rafael de Almeida e Marques

Professor Adjunto do Instituto Politécnico de Castelo Branco

Paula Cristina Alves Pereira

Prof. Adjunta do Instituto Politécnico de Castelo Branco



## **Dedicatória**

### **Pedro Daniel**

Aos meus tios, Alfredo Daniel Lina e Maria Matondo Toko, aos meus pais, João Mazebo e Celina Daniel, e aos meus irmãos, que com imenso carinho e apoio incansável, não pouparam esforços para que esta licenciatura se tornasse realidade.

### **Miguel Dunge**

Dedico este trabalho aos meus pais e irmãos, cujo amor e apoio constante me permitiram superar todos os desafios e chegar até este momento tão importante da minha vida.



## **Agradecimentos**

A Deus, pelo dom da vida, para que tudo fosse possível.

Foram várias as pessoas que contribuíram para que fosse possível a realização deste projeto e a quem queremos agradecer.

Gostaríamos de expressar a nossa sincera gratidão ao Professor doutor Paulo Jorge Dias Torres pelo apoio inestimável e pela orientação indispensável durante o desenvolvimento deste projeto final.

A todos os professores que contribuíram para a nossa aprendizagem e conhecimentos durante a licenciatura um muito obrigado.

Um agradecimento especial a todos os nossos colegas que nos ajudaram durante estes anos e pela compreensão e companheirismo demonstrados durante a execução deste projeto.



## Resumo

Este projeto apresenta um sistema de monitorização para estufas, utilizando o microcontrolador *ESP32* em conjunto com o minicomputador *Raspberry Pi 4*, a plataforma *MQTT* com o *broker Mosquitto*, e o *Node-RED* como interface de visualização e controlo. O objetivo é otimizar o cultivo em ambientes controlados, através da automação e monitorização de variáveis críticas para o desenvolvimento das plantas.

Sensores como o *DHT22*, *HD-38* e *HW-038* são utilizados para recolher dados em tempo real como a temperatura, humidade do ar, temperatura, humidade do solo e a qualidade da água usada na estufa. Estes parâmetros são analisados e utilizados para ajustar automaticamente as condições internas da estufa, garantindo um ambiente ideal para o cultivo e o desenvolvimento das plantas.

A integração eficiente entre os sensores e a plataforma visualização permitem um controlo preciso dos dados da estufa, logo, o sistema demonstrou ser uma solução eficaz para promover práticas agrícolas mais sustentáveis e produtivas, em contextos domésticos e de pequena escala.

## Palavras-chave

Sistema monitorização de estufa, *ESP32*, sensores, *Raspberry Pi*.



## Abstract

This project presents a monitoring system for greenhouses, using the ESP32 microcontroller in conjunction with the Raspberry Pi 4 minicomputer, the MQTT platform with the Mosquitto broker, and Node-RED as the visualization and control interface. The aim is to optimize cultivation in controlled environments by automating and monitoring critical variables for plant development.

Sensors such as the DHT22, HD-38 and HW-038 are used to collect real-time data such as temperature, air humidity, temperature, soil humidity and the quality of the water used in the greenhouse. These parameters are analyzed and used to automatically adjust the internal conditions of the greenhouse, ensuring an ideal environment for growing and developing plants.

The efficient integration between the sensors and the visualization platform allows precise control of the greenhouse data, so the system has proven to be an effective solution for promoting more sustainable and productive agricultural practices in domestic and small-scale contexts.

## Keywords

Greenhouse monitoring system, *ESP32*, sensors, *Raspberry Pi*.



## índice

1. Capítulo 1 - Introdução .....	1
1.1. Introdução.....	1
1.2. Diagrama em blocos do sistema de monitorização de estufa .....	2
1.3. Problema.....	3
1.4. Objetivos .....	3
1.4.1. Objetivo geral .....	3
1.4.2. Objetivos específicos .....	3
1.5. Solução.....	4
1.6. Estrutura do projecto.....	4
1.7. Enquadramento.....	4
1.7.1. Enquadramento territorial .....	5
2. Capítulo 2 - Fundamentos teóricos .....	6
2.1. Automação na agricultura.....	6
2.1.1. Importância da automação agrícola .....	6
2.1.2. Vantagens.....	6
2.2. Estufas e monitorização ambiental .....	6
2.2.1. Principais benefícios da estufa .....	7
2.2.2. Como funciona o cultivo em estufas .....	7
2.3. Internet das coisas (iot) aplicada à agricultura .....	8
2.3.1. Aplicações da internet das coisas (iot) na agricultura .....	8
2.4. Soluções existentes: explorar projetos e sistemas similares já utilizados no setor agrícola, comparando soluções e inovações.....	9
2.5. Comunicações <i>wireless fidelity (wi-fi)</i> .....	9
2.5.1. Conceito de <i>wi-fi</i> .....	9
2.6. Protocolos de comunicação .....	9
2.7. Sensores e atuadores .....	11
2.7.1. Sensores .....	11
2.7.2. Atuadores.....	11
2.8. Componentes de <i>hardware</i> do sistema .....	11
2.8.1. Módulo <i>esp-wroom-32</i> .....	11

2.8.2.	Sensor de humidade do solo <i>arduino hd-38</i> .....	14
2.8.3.	Sensor de humidade e temperatura <i>dht22</i> .....	15
2.8.4.	Sensor de temperatura <i>ds18b20</i> .....	16
2.8.5.	Sensor de condutividade da água <i>tds</i> .....	17
2.8.6.	Mini-bomba de água para <i>arduino</i> 12v rs385 2l/min.....	18
2.8.7.	Ventilador de alta velocidade .....	19
2.8.8.	Módulo relé 5v 10a 4 canais com optoacopladores.....	20
2.8.9.	<i>Display oled ssd1306</i> .....	21
2.8.9.1.	Características principais:.....	21
2.8.9.2.	Comunicação de dados:.....	22
2.8.9.3.	Controlador <i>ssd1306</i> : .....	22
2.9.	<i>Raspberry pi</i> .....	22
2.9.1.	<i>Dispositivos necessários para o uso do raspberry pi</i> .....	22
2.9.1.1.	Uma fonte de alimentação.....	22
2.9.1.2.	Um cartão <i>microsd</i> .....	23
2.10.	<i>Software</i> .....	25
2.10.1.	Protocolo <i>mqtt</i> .....	25
2.10.2.	Conceito de <i>mqtt</i> .....	25
2.10.3.	Definição de <i>broker mqtt</i> .....	25
2.10.4.	<i>Node-red</i> .....	25
2.10.5.	<i>Visual studio code</i> .....	26
3.	Capítulo 3 – Implementação do sistema .....	27
3.1.	Conceitos práticos.....	27
3.2.	Tipo de solo para estufas em portugal.....	27
3.3.	Placa eletrônica <i>esp32</i> .....	29
3.3.1.	Instalação do ambiente de desenvolvimento .....	29
3.3.2.	Integração dos sensores e atuadores com o <i>esp32</i> .....	30
3.3.2.1.	Sensor de humidade do solo <i>arduino hd-38</i> .....	30
3.3.2.2.	Sensor de humidade e temperatura <i>dht22</i> .....	31
3.3.2.3.	Sensor de temperatura <i>ds18b20</i> .....	32
3.3.2.4.	Sensor de condutividade da água <i>tds</i> .....	33

3.3.2.5.	Funcionamento:.....	33
3.3.2.6.	Atuadores do sistema.....	34
3.4.	<i>Raspberry pi, mqtt (mosquitto broker) e node-red</i> .....	35
3.4.1.	Instalando o sistema operacional do <i>raspberry pi 4, via raspberrry pi imager</i> 35	
3.4.2.	<i>Mqtt (mosquitto broker)</i> .....	37
3.4.3.	<i>Instalação do broker mqtt mosquitto no raspberrry pi</i> .....	37
3.4.4.	<i>Ativar o acesso remoto/autenticação</i> .....	39
3.4.5.	<i>Modelo publicação/assinatura e tópicos em mqtt</i> .....	41
3.4.6.	<i>Instalação do node-red</i> .....	41
3.4.6.1.	<i>Instalar o node-red</i> .....	42
3.4.6.2.	<i>Configurar as definições do node-red</i> .....	43
3.4.6.3.	<i>Iniciar o node-red</i> .....	43
3.4.6.4.	<i>Nó de acesso node-red</i> .....	44
3.4.6.5.	<i>Painel de controlo do node-red</i> .....	45
3.4.6.6.	<i>Instalando o painel de controle do node-red</i> .....	46
3.4.6.7.	<i>Resultado da dashboard</i> .....	47
3.4.6.8.	<i>Aba 2: dados do ar</i> .....	48
3.4.6.9.	<i>Aba 3: dados do solo</i> .....	48
3.4.7.	<i>Esquema elétrico do sistema de monitorização de estufas</i> .....	49
3.5.	Fluxograma .....	51
4.	Capítulo 4 – Testes e resultados.....	52
4.1.	Cenários de teste e resultados.....	52
4.2.	Cenários.....	53
5.	Conclusão .....	68
6.	Trabalhos futuros.....	69
7.	Referências bibliográficas.....	70



## Índice de figuras

<b>Figura 1-</b> Diagrama de blocos do sistema de monitorização de estufa. ....	2
<b>Figura 2-</b> Enquadramento territorial. [F16].....	5
<b>Figura 3 -</b> ESP32. [F1] .....	11
<b>Figura 4 -</b> Arquitetura do ESP32. [F2]. ....	12
<b>Figura 5 -</b> Diagrama de blocos do ESP32. [F3]. ....	13
<b>Figura 6 -</b> Pinagem da Placa ESP32 DevKit. [F4] .....	14
<b>Figura 7 -</b> Sensor de humidade do Solo <i>Arduino HD-38</i> . [F6] .....	15
<b>Figura 8 -</b> Sensor DHT22/AM2302 – pinout. [F5]. ....	16
<b>Figura 9 -</b> Sensor de Temperatura DS18B20. [F7].....	17
<b>Figura 10 -</b> Sensor de condutividade da água TDS. [F8]. ....	18
<b>Figura 11 -</b> Mini-Bomba de Água para Arduino 12V RS385 2L/Min. [F10].....	19
<b>Figura 12-</b> Ventilador [F19] .....	20
<b>Figura 13 -</b> Módulo Relé 5V 10A 4 Canais com Optoacopladores [F9].....	21
<b>Figura 14 -</b> Display OLED SSD1306.....	21
<b>Figura 15 -</b> Raspberry pi 4. [F11] .....	22
<b>Figura 16 -</b> Fonte de alimentação. [F12].....	23
<b>Figura 17 -</b> Inserção do cartão <i>microSD</i> . [F12]. ....	23
<b>Figura 18 -</b> Raspberry Pi 4, portas e componentes. [F13]. ....	24
<b>Figura 19 -</b> Tipos de solos em Portugal, para diferentes cultivos. [F17]. ....	27
<b>Figura 20 -</b> Embalagem e terra da sira germinação Bio. [F18] .....	28
<b>Figura 21 -</b> Montagem do sensor de humidade do solo .....	30
<b>Figura 22 -</b> Montagem do sensor de humidade/Temperatura do ar <i>DHT22</i> . ....	31
<b>Figura 23 -</b> Montagem do sensor de Temperatura <i>DS18B20</i> .....	32
<b>Figura 24 -</b> Montagem do sensor de condutividade da água TDS .....	33
<b>Figura 25 -</b> Montagem do Ventilador, mini-bomba de água e o módulo relés de 4 canais .....	35
<b>Figura 26 -</b> Raspberry Pi Imager. [F12]. ....	35
<b>Figura 27 -</b> Escolhendo o sistema a instalar, neste caso será o <i>Raspberry pi OS (32bit)</i> . [F12].....	36
<b>Figura 28 -</b> Instalando o sistema operacional. [F12]. ....	36
<b>Figura 29 -</b> Cliente <i>MQTT: Smartphone (assinante) -&gt; MQTT broker -&gt; MQTT client:</i> Sensor de temperatura. [F14] .....	37
<b>Figura 30 -</b> Instalação do mosquitto mqtt broker no Raspberry pi 4. ....	38
<b>Figura 31 -</b> Configuração do mosquitto mqtt broker no Raspberry pi 4. ....	40
<b>Figura 32 -</b> Instalação do mosquitto mqtt broker no Raspberry pi com autenticação. .....	40
<b>Figura 33 -</b> Node-red. [F15]. ....	41
<b>Figura 34 -</b> Instalação do node-red no Raspberry pi. ....	42
<b>Figura 35 -</b> Resolvendo alguns erros. ....	43
<b>Figura 36 -</b> Inicialização do node-red.....	44
<b>Figura 37 -</b> Nó de acesso node-red.....	45

<b>Figura 38</b> - Página Node-red.....	45
<b>Figura 39</b> - Instalando a node-red-dashboard.....	46
<b>Figura 40</b> - Painel de controlo dashboard.....	46
<b>Figura 41</b> - dashboard node-red.....	47
<b>Figura 42</b> - Fluxos criados no node-red para sistema de monitorização de estufas....	47
<b>Figura 43</b> - Aba painel principal da <i>dashboard</i> .....	48
<b>Figura 44</b> - Aba 2 dados do ar da <i>dashboard</i> .....	48
<b>Figura 45</b> - Aba 3 dados do solo da <i>dashboard</i> .....	49
<b>Figura 46</b> - Esquema elétrico logico do sistema de monitorização de estufas.....	49
<b>Figura 47</b> - Esquema elétrico de componentes reais do sistema de monitorização de estufas.....	50
<b>Figura 48</b> - Fluxograma do sistema de monitorização de estufas.....	51
<b>Figura 49</b> - Estufa.....	53
<b>Figura 50</b> - Potes com água para experiência.....	54
<b>Figura 51</b> - Resultados da água do pote branco.....	55
<b>Figura 52</b> - Água usada.....	56
<b>Figura 53</b> -Resultados da água do pote branco.....	56
<b>Figura 54</b> - Sensor de humidade do solo em solo seco.....	57
<b>Figura 55</b> - Valor da humidade no solo apresentado no monitor serial.....	57
<b>Figura 56</b> - Sensor de humidade do solo emergido em água.....	58
<b>Figura 57</b> - Valor da humidade na água apresentado no monitor serial.....	58
<b>Figura 58</b> - Valor da humidade do solo na dashboard.....	59
<b>Figura 59</b> - Sensor DHT22 sem qualquer alteração.....	60
<b>Figura 60</b> - Valor da humidade e temperatura do ar apresentados no monitor serial.....	60
<b>Figura 61</b> - Valor da humidade e temperatura do ar apresentados na dashboard.....	60
<b>Figura 62</b> - Sensor DHT22 sendo aquecido.....	61
<b>Figura 63</b> - Valor no monitor serial da humidade e temperatura do ar sendo o sensor aquecido.....	61
<b>Figura 64</b> - Valor na dashboard da humidade e temperatura do ar sendo o sensor aquecido.....	61
<b>Figura 65</b> - Sensor de humidade do solo no solo húmido.....	62
<b>Figura 66</b> - Valor no monitor serial da humidade do solo húmido.....	63
<b>Figura 67</b> - Valor na dashboard da humidade do solo húmido.....	63
<b>Figura 68</b> - Sensor de humidade do solo no solo seco.....	63
<b>Figura 69</b> - Valor no monitor serial da humidade do solo seco.....	64
<b>Figura 70</b> - Valor na dashboard da humidade do solo seco.....	64
<b>Figura 71</b> - Bomba de água ligada.....	64
<b>Figura 72</b> - Estufa sendo regada.....	65
<b>Figura 73</b> - Valor no monitor serial da temperatura no momento do teste.....	65
<b>Figura 74</b> - Aquecendo o DHT22 para ligar o ventilador.....	66
<b>Figura 75</b> - Valor no monitor serial da temperatura apos o aquecimento.....	66
<b>Figura 76</b> - Valor na <i>dashboard</i> da temperatura apos o aquecimento.....	66

**Figura 77 - Ventilador ligado ..... 67**

## Lista de tabelas

<b>Tabela 1</b> - Especificações Sensor de Humidade do solo HD-38 .....	15
<b>Tabela 2</b> - Especificações Sensor de Humidade/Temperatura DHT22.....	16
<b>Tabela 3</b> - Sensor de Temperatura DS18B20 .....	17
<b>Tabela 4</b> - Sensor de condutividade da água TDS.....	18
<b>Tabela 5</b> - Mini-Bomba de Água para Arduino 12V RS385 2L/Min .....	19
<b>Tabela 6</b> - Ventilador.....	20
<b>Tabela 7</b> - Módulo Relé 5V 10A 4 Canais com Optoacopladores .....	21
<b>Tabela 8</b> - Raspberry Pi 4, portas e componentes.....	24
<b>Tabela 9</b> - Lista dos materiais .....	52
<b>Tabela 10</b> - Referências de qualidade de água .....	55



## Lista de abreviaturas, siglas e acrónimos

μC	Microcontrolador
ADC	Analog-to-Digital Converter (Conversor analógico para digital)
AES	Advanced Encryption Standard (Padrão de criptografia avançada)
APP_CPU	Application CPU (CPU para aplicações)
CAN	Controller Area Network (Rede de comunicação para automóveis)
DAC	Digital-to-Analog Converter (Conversor digital para analógico)
DHT22	Temperature and humidity sensor (Sensor de temperatura e humidade)
ESP32	Microcontroller with Wi-Fi and Bluetooth connectivity (Microcontrolador com conectividade Wi-Fi e Bluetooth)
ESTCB	Escola Superior de Tecnologia de Castelo Branco
GND	Ground (Terra)
GPIO	General Purpose Input/Output (Pinos de entrada/saída de propósito geral)
HD-38	Soil moisture sensor (Sensor de humidade do solo)
I2C	Inter-Integrated Circuit (Comunicação serial de baixa velocidade)
IoT	Internet of Things (Internet das Coisas)
IPCB	Instituto Politécnico de Castelo Branco
MQTT	Lightweight communication protocol for IoT (Protocolo de comunicação leve para IoT)
OLED	Organic Light-Emitting Diode (Diodo emissor de luz orgânico)
PRO_CPU	Main CPU (CPU principal)
PWM	Pulse Width Modulation (Modulação por largura de pulso)
RAM	Random-Access Memory (Memória temporária)
RTC	Real-Time Clock (Relógio de tempo real)
RX	Receive (Receber)
SCL	Serial Clock Line (Linha de relógio I2C)
SD	Secure Digital (Cartão de memória)
SPI	Serial Peripheral Interface (Interface serial de periféricos)
TX	Transmit (Transmitir)
UART	Universal Asynchronous Receiver-Transmitter (Comunicação serial assíncrona)

ULP	Ultra-Low Power (Ultra-baixo consumo de energia)
USB	Universal Serial Bus
VCC	Voltage Common Collector (Tensão de alimentação)
VIN	Voltage Input (Tensão de entrada)
Wi-Fi	Wireless Fidelity (Rede sem fios)



# 1. Capítulo 1 - Introdução

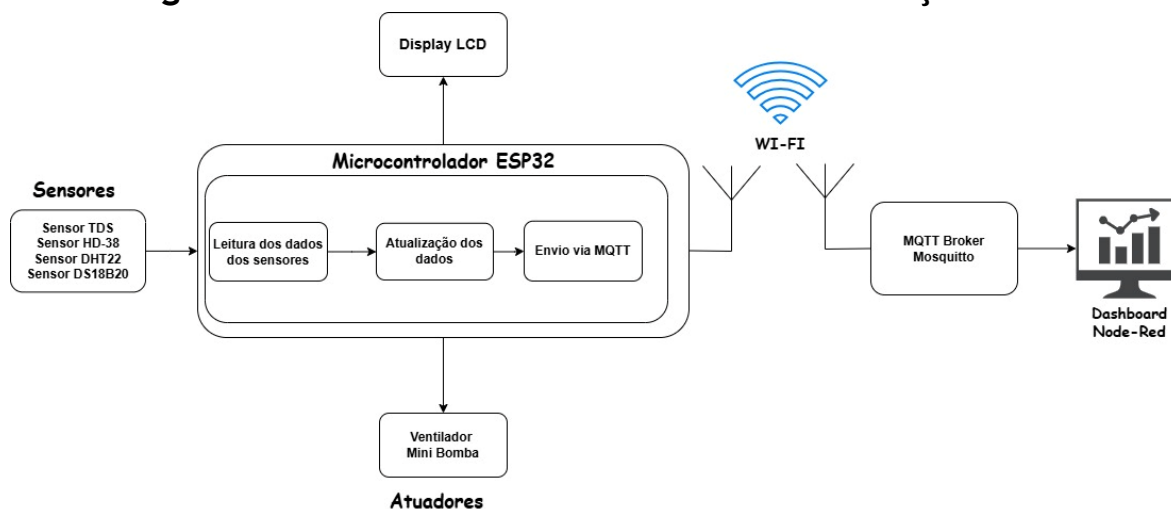
## 1.1. Introdução

Este relatório foi desenvolvido no âmbito do Projeto Final de Licenciatura em Engenharia Eletrotécnica e das Telecomunicações, curso oferecido pela Escola Superior de Tecnologia de Castelo Branco (ESTCB) do Instituto Politécnico de Castelo Branco (IPCB). O principal objetivo deste projeto é consolidar e aplicar os conhecimentos adquiridos ao longo do curso, através do desenvolvimento de um sistema de monitorização de estufas, uma solução prática e inovadora para o setor agrícola.

A agricultura moderna enfrenta desafios complexos, especialmente no que diz respeito à eficiência no uso de recursos naturais, como água e energia, e à adaptação às mudanças climáticas. Estufas são uma solução comum para proteger as plantas das variações climáticas e criar ambientes controlados que promovem o crescimento saudável das culturas. No entanto, o controle manual das variáveis ambientais, como temperatura, humidade e irrigação, torna-se ineficaz e sujeito a erros humanos, comprometendo a produtividade e a qualidade das plantas.

A automatização dos processos agrícolas por meio da Internet das Coisas (*IoT*) tem se mostrado uma inovação promissora, fornecendo soluções que melhoram a eficiência e reduzem o desperdício de recursos. O uso de microcontroladores como o *ESP32* e minicomputadores como o *Raspberry Pi* possibilita a criação de sistemas de monitorização remota e controle automático de estufas, utilizando sensores que coletam dados em tempo real e ajustam automaticamente o ambiente interno. Este tipo de sistema é capaz de regular variáveis essenciais para o cultivo, garantindo condições ideais de crescimento sem a necessidade de intervenção humana contínua.

## 1.2. Diagrama em Blocos do sistema de monitorização de estufa



**Figura 1-** Diagrama de blocos do sistema de monitorização de estufa.

O projeto centra-se na criação de um sistema de monitorização de estufas, utilizando o microcontrolador *ESP32*, amplamente reconhecido pelas suas capacidades de processamento, eficiência energética e conectividade. O sistema é complementado por um minicomputador *Raspberry Pi 4*, que, em conjunto com o protocolo de comunicação *MQTT* e a interface *Node-RED*, permite a recolha, análise e visualização de dados em tempo real. Estes dados são obtidos por meio de uma rede de sensores de alta precisão fundamentais para ajustar automaticamente as condições internas da estufa, garantindo um ambiente ideal para o crescimento das plantas, especialmente em regiões portuguesas onde o clima é variável.

O sistema proposto neste projeto visa automatizar completamente o controlo das condições dentro da estufa, promovendo uma utilização mais eficiente dos recursos e um ambiente estável para o desenvolvimento das plantas. A integração de sensores com o *ESP32* e o *Raspberry Pi 4*, comunicando via protocolo *MQTT*, permite uma gestão inteligente das variáveis internas da estufa. O sistema ajusta automaticamente parâmetros como a rega e ventilação, com base nos dados recolhidos em tempo real, o que reduz a necessidade de intervenção humana e maximiza a eficiência do cultivo.

Este projeto não só demonstra a viabilidade técnica de sistemas de monitorização baseados em microcontroladores como o *ESP32*, mas também reforça o seu potencial para promover práticas agrícolas mais sustentáveis e eficientes. A combinação de tecnologia acessível, como o *ESP32* e o *Raspberry Pi*, com sensores de precisão e plataformas de controlo automatizado, representa um passo em direção à modernização do setor agrícola em Portugal. A sua aplicação pode beneficiar pequenos e grandes produtores, oferecendo soluções escaláveis para melhorar a produtividade e minimizar o desperdício de recursos.

### 1.3. Problema

A gestão eficiente de uma estufa requer uma monitorização constante e rigorosa das condições ambientais, de forma a garantir o crescimento saudável das plantas. No entanto, o controlo manual dessas variáveis, como a temperatura, a humidade do ar, a temperatura, humidade do solo e a qualidade de água, é um processo lento e propenso a erros. Estes erros podem causar flutuações nas condições ideais, resultando em temperaturas desadequadas, níveis de humidade incorretos e falta de água, comprometendo seriamente a produção agrícola. Além disso, a necessidade de respostas rápidas às alterações ambientais representa um grande desafio, especialmente quando o operador não está presente para intervir de imediato, o que pode agravar os impactos negativos sobre as culturas.

### 1.4. Objetivos

#### 1.4.1. Objetivo geral

- ❖ Desenvolver um sistema de monitorização de estufas, utilizando o microcontrolador *ESP32*, o minicomputador *Raspberry Pi* e a plataforma *node-RED*, com o propósito de otimizar as condições de cultivo em ambientes controlados. O sistema visa promover práticas agrícolas sustentáveis, melhorando a eficiência no uso de recursos como água e energia, e aumentando a produtividade das culturas em Portugal.

#### 1.4.2. Objetivos específicos

- ❖ **Integrar sensores com o *ESP32* e configurar *node-RED*:** Implementar uma rede de sensores *DHT22*, *HD-38* e *HW-038* conectados ao *ESP32* para monitorizar variáveis ambientais como temperatura, humidade do ar, temperatura, humidade do solo e a qualidade da água. Utilizar o *Raspberry Pi*, em conjunto com o ambiente *node-red*, para recolher, processar e visualizar os dados em tempo real, permitindo a visualização das condições internas da estufa.
- ❖ **Desenvolver, calibrar e avaliar o sistema para eficiência agrícola:** Desenvolver o sistema utilizando o Visual Studio Code, ajustando os parâmetros de funcionamento e realizando a calibração dos sensores para garantir a precisão dos dados recolhidos. Avaliar o desempenho do sistema em termos de produtividade agrícola e sustentabilidade, com foco na otimização do uso de recursos e minimizar desperdícios.
- ❖ **Avaliar o impacto do sistema na produtividade e na sustentabilidade agrícola:** Medir o efeito do sistema no crescimento das plantas, monitorizando a eficiência no uso de recursos como água e energia, e analisando a redução de desperdícios. Demonstrar os benefícios do sistema para a adoção de práticas agrícolas mais sustentáveis, com impacto positivo na gestão de estufas.

## 1.5. Solução

O projeto de um sistema automatizado de monitorização de estufas se justifica pela necessidade de otimizar o controle ambiental em ambientes de cultivo controlado. Ao automatizar o monitoramento e a regulação das variáveis internas da estufa, como temperatura, umidade do ar e umidade do solo, é possível reduzir o uso de recursos, melhorar a produtividade das culturas e minimizar a intervenção manual. Isso também proporciona um sistema mais resiliente, capaz de responder rapidamente às mudanças climáticas e ajustar-se conforme necessário para garantir o desenvolvimento ideal das plantas.

A crescente demanda por soluções tecnológicas no setor agrícola, juntamente com as preocupações com sustentabilidade e eficiência no uso de recursos, torna este projeto relevante. A adoção de tecnologias acessíveis, como o *ESP32* e o *Raspberry Pi*, também possibilita que sistemas automatizados sejam implementados tanto em pequenas quanto em grandes estufas, permitindo uma aplicação escalável.

## 1.6. Estrutura do projecto

O presente relatório é composto por quatro (4) capítulos. No primeiro capítulo, abordamos o enquadramento territorial do projeto, apresentamos um breve histórico e discutimos os principais parâmetros que orientam a sua execução.

O segundo capítulo, descrevemos os componentes de “hardware” do sistema, detalhando os diferentes sensores e dispositivos utilizados. Entre eles, destacamos o ESP32, um microcontrolador versátil com conectividade Wi-Fi e Bluetooth, que atua como o cérebro do sistema. Também abordamos a utilização de sensores específicos, como os de humidade do solo, temperatura e humidade do ar, a qualidade da água, essenciais para monitorar as condições ambientais. Além disso, mencionamos dispositivos auxiliares, como uma mini-bomba de água, um módulo de relé de 4 canais (5V, 10A) para controle de dispositivos elétricos, um ventilador para arrefecimento dos componentes, garantindo o funcionamento estável do sistema, descrevemos o Raspberry Pi, um minicomputador escolhido por sua versatilidade e baixo custo, que serve como núcleo tecnológico do projeto e por fim falamos dos softwares usados para programar o ESP32 e os sensores.

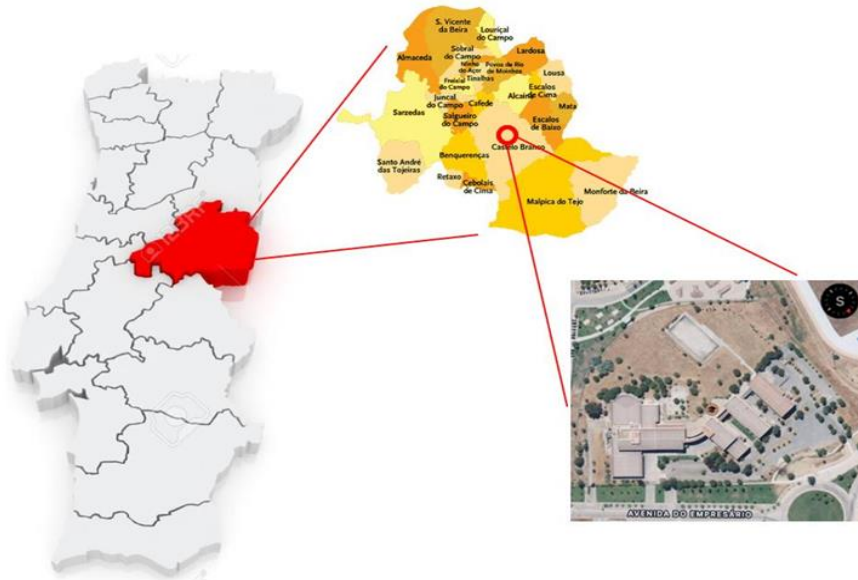
O terceiro capítulo centra-se na implementação prática do projeto de monitorização de estufas. Nesta secção, apresentamos a elaboração dos esquemas elétricos, a criação de fluxogramas que descrevem o funcionamento do sistema e a programação dos dispositivos. Esta fase é crucial para transformar a teoria num sistema operativo, testar o seu desempenho e assegurar que todos os componentes funcionem de forma integrada e eficiente. Adicionalmente, tratamos da instalação do Broker MQTT utilizando o Mosquitto, assim como da configuração do Node-RED.

## 1.7. Enquadramento

O enquadramento territorial da área de cultivo, com especial atenção às características do solo e às condições necessárias para o cultivo dentro da estufa. Discutiremos o tipo de solo que será utilizado, bem como as condições ideais para o desenvolvimento das plantas. Além disso, faremos uma breve introdução ao tipo de cultivo planeado, destacando os fatores que influenciam diretamente a produtividade e a saúde das culturas em ambiente controlado.

### 1.7.1. Enquadramento territorial

Este projeto vai incidir numa pequena amostra na Escola Superior de Tecnologia de Castelo Branco, no concelho de Castelo Branco, distrito de Castelo Branco (figura 1) pertencente à NUT III da Beira Baixa. O município de Castelo Branco tem 1438.19 km<sup>2</sup>, 52 272 habitantes (2021) e 19 freguesias o que se traduz numa densidade populacional de 36.4 hab/km<sup>2</sup>. Confinava com o concelho do Fundão a norte, Idanha-a-Nova a leste, Vila Velha de Rodão a sudoeste, Proença-a-Nova e Oleiros a Oeste e Espanha a Sul. [17].



**Figura 2-** Enquadramento territorial. [F16]

## 2. Capítulo 2 - Fundamentos teóricos

Neste capítulo, descrevemos os principais componentes de *hardware* do sistema, como o ESP32, que atua como o cérebro do projeto, e sensores para monitorizar a humidade do solo, a temperatura, a humidade do ar e a qualidade da água para a estufa. Também mencionamos dispositivos auxiliares, como a mini-bomba de água e o módulo de relé para controlo de dispositivos elétricos, além de um ventilador para arrefecimento. Abordamos ainda o Raspberry Pi, escolhido pela sua versatilidade e baixo custo, bem como os softwares utilizados para programar o ESP32 e os sensores.

### 2.1. Automação na agricultura

A automação agrícola é um processo mecânico que utiliza tecnologias para a gestão, monitorização e acionamento automático das operações de uma produção. Através da automação agrícola, é possível obter maior agilidade, eficiência e controlo das atividades de cultivo e extração de produtos agrícolas, uma vez que estas podem ser realizadas remotamente e de forma programada.

#### 2.1.1. Importância da Automação agrícola

A automação no setor agrícola permite otimizar os processos produtivos, promovendo maior eficiência e reduzindo custos, riscos, tempo e desperdícios.

O uso de tecnologias de automação no agronegócio impulsiona a evolução deste setor, trazendo benefícios tanto para o produtor como para o consumidor, com ganhos em desempenho, aumento da produção e melhoria da qualidade do produto final.

#### 2.1.2. Vantagens

- ❖ Aumento da produtividade e desempenho
- ❖ Economia de Custos
- ❖ Maior controle sobre a produção
- ❖ Precisão e qualidade nos produtos

### 2.2. Estufas e Monitorização Ambiental

Após a Segunda Guerra Mundial, a agricultura em estufas para fins comerciais teve um grande desenvolvimento, iniciado no norte da Europa nas primeiras décadas do século XX. As estufas foram criadas com o objetivo de ultrapassar as dificuldades de cultivo de plantas sensíveis ao frio. O cultivo em estufas expandiu-se para regiões com invernos mais amenos, como a Bacia do Mediterrâneo, especialmente após a introdução do plástico nos anos 60. Dado que as estufas permitem uma utilização racional da água, a produção alimentar em ambiente protegido expandiu-se para alguns países onde a água destinada à agricultura é escassa. [24]

O cultivo protegido, vulgarmente conhecido como cultivo em estufa, é uma técnica que consiste no crescimento de plantas dentro de uma estrutura fechada e controlada. Esta abordagem permite aos agricultores criar condições ambientais ideais, como temperatura e humidade específicas, essenciais para o desenvolvimento saudável das plantas. No contexto da agricultura moderna, em que a procura por alimentos é constante e crescente, as estufas oferecem uma forma fiável e eficiente de produzir alimentos ao longo de todo o ano, independentemente das variações climáticas. [25]

### 2.2.1. Principais benefícios da estufa

- ❖ **Controle climático:** A capacidade de controlar o ambiente interno é um dos principais benefícios das estufas. Isto permite aos agricultores cultivar plantas que normalmente não sobreviveriam no clima local. Além disso, o cultivo pode ocorrer durante todo o ano, independentemente das estações.
- ❖ **Proteção contra pragas e doenças:** Num ambiente fechado, as plantas estão menos expostas a insetos e agentes patogénicos externos, o que reduz significativamente a necessidade de aplicação de pesticidas e herbicidas, resultando em produtos mais limpos e seguros para o consumo.
- ❖ **Produção otimizada:** As estufas permitem o cultivo intensivo, maximizando a produção em áreas limitadas. Isto é especialmente vantajoso em zonas urbanas ou locais com espaço de cultivo restrito.

### 2.2.2. Como funciona o cultivo em estufas

O cultivo em estufas envolve a implementação de diversas técnicas e tecnologias avançadas para otimizar o crescimento das plantas, tais como:

- **Sistemas de aquecimento e ventilação:** Essenciais para manter uma temperatura constante e adequada, especialmente em climas mais frios ou durante a noite.
- **Sistemas de irrigação:** A irrigação por gotejamento é uma técnica comum em estufas, pois permite uma distribuição precisa e eficiente de água diretamente às raízes das plantas, minimizando o desperdício.
- **Controle de luz:** A luz é um fator fundamental para a fotossíntese. Em muitas estufas, luzes artificiais complementam a luz natural, garantindo que as plantas recebam a quantidade ideal de luz para o crescimento, independentemente das condições externas.
- **Monitorização e controlo automatizado:** Sensores avançados e sistemas automatizados permitem o monitoramento constante das condições internas e ajustam automaticamente os parâmetros para manter o ambiente ideal.

## 2.3. Internet das Coisas (IoT) Aplicada à Agricultura

A Internet das Coisas (*IoT*, do inglês *Internet of Things*) refere-se a uma rede de dispositivos físicos, como veículos, eletrodomésticos e outros objetos, equipados com sensores, software e conectividade de rede, permitindo a recolha e partilha de dados.

A *IoT* possibilita que estes dispositivos inteligentes comuniquem entre si e com outros dispositivos com acesso à Internet, como smartphones e tablets, criando uma vasta rede de dispositivos interligados capazes de trocar dados e realizar várias tarefas de forma autónoma.

À medida que o número de dispositivos conectados à Internet continua a aumentar, a *IoT* desempenhará um papel cada vez mais importante na transformação do nosso mundo, alterando a forma como vivemos, trabalhamos e interagimos uns com os outros.

### 2.3.1. Aplicações da Internet das Coisas (IoT) na Agricultura

De um modo geral, a Internet das Coisas abre caminho para a chamada agricultura inteligente, uma ideia revolucionária que alia as práticas agrícolas ao poder das tecnologias, com o objetivo de melhorar os processos produtivos. [26]

A *IoT* oferece uma série de vantagens neste contexto, destacando-se as seguintes:

#### ❖ **Automatização da irrigação**

Com a instalação de sensores automatizados, os agricultores conseguem otimizar os seus sistemas de irrigação. Na prática, isto significa que passam a utilizar os recursos hídricos de forma mais eficiente: além de saberem quando e em que quantidade regar o solo, podem adaptar a irrigação consoante o tipo de cultura, uma vez que cada uma tem exigências hídricas específicas. Como resultado, reduz-se o desperdício e promove-se a sustentabilidade.

Outra vantagem significativa é a diminuição do uso excessivo de agroquímicos no solo, uma vez que se torna mais fácil fazer as adaptações e correções necessárias na plantação, utilizando recursos naturais.

#### ❖ **Acesso a dados climáticos e meteorológicos precisos**

Com a *IoT* a possibilitar a recolha de dados em tempo real, os agricultores têm acesso a leituras e previsões climáticas altamente precisas e fiáveis, obtidas a partir de diferentes estações meteorológicas, que podem ser integradas para gerar resultados ainda mais rigorosos. Desta forma, torna-se mais fácil ajustar as rotinas de plantio, irrigação e colheita, assegurando a máxima produtividade possível. Quanto maior for esta preparação antecipada, menores serão as perdas nas colheitas.

## 2.4. Soluções Existentes: Explorar projetos e sistemas similares já utilizados no setor agrícola, comparando soluções e inovações.

Diversas soluções tecnológicas já estão a ser implementadas com sucesso no setor agrícola. Entre elas, destacam-se os sistemas de automatização em estufas, robôs para colheita, drones para monitorização aérea e plataformas de software que integram dados da *IoT* para análise preditiva.

Comparar estas soluções permite compreender os benefícios e limitações de cada uma. Por exemplo, enquanto as grandes explorações agrícolas utilizam drones para monitorizar vastas áreas de cultivo, os pequenos agricultores podem beneficiar de sensores de baixo custo para monitorizar o solo e ajustar automaticamente a irrigação.

A inovação contínua reside na criação de soluções mais acessíveis e de fácil integração, permitindo que até os pequenos agricultores possam tirar partido destas tecnologias.

## 2.5. Comunicações *Wireless Fidelity* (*Wi-Fi*)

### 2.5.1. Conceito de *Wi-Fi*

*Wi-Fi* é a abreviação de *Wireless Fidelity*, que em português significa "fidelidade sem fios". O *Wi-Fi*, é uma tecnologia de comunicação que não faz o uso de cabos, sendo geralmente transmitida através de frequências de rádio.

Em outras palavras, o *Wi-Fi* permite que dispositivos eletrónicos se conectem à internet sem necessidade de fios. Esta tecnologia utiliza ondas de rádio para transmitir dados entre um router e os dispositivos conectados a ele, proporcionando uma ligação rápida e estável.

O *Wi-Fi* não requer licenças para instalação ou operação. Para aceder a uma rede *Wi-Fi*, é necessário estar dentro da área de cobertura de um ponto de acesso, que pode estar presente em locais públicos que ofereçam *Wi-Fi*. Para utilizar esta tecnologia, é indispensável dispor de um dispositivo móvel, como um computador portátil, tablet ou telemóvel, permitindo o acesso à internet de forma prática e eficiente.

## 2.6. Protocolos de Comunicação

Os protocolos de comunicação facilitam a troca de dados entre os vários componentes do sistema, garantindo a integridade e eficiência das comunicações. Cada protocolo será descrito em termos de funcionalidades e características e a forma como são inseridos no projeto.

### ***MQTT (Message Queuing Telemetry Transport)***

**Descrição:** O *MQTT* é um protocolo de mensagens leve e baseado em publicação/assinatura. É ideal para aplicações de Internet das Coisas (IoT) devido à sua eficiência em redes com largura de banda limitada.

#### **Uso no Sistema:**

O *ESP32* pode atuar como um cliente *MQTT*, enviando dados dos sensores (como temperatura, humidade, etc.) para um *broker MQTT*.

O *Raspberry Pi* pode ser configurado como um *broker MQTT* (usando softwares como Mosquitto) ou como um cliente que recebe dados do *ESP32* e os processa ou armazena.

### ***HTTP/HTTPS (Hypertext Transfer Protocol)***

**Descrição:** Um protocolo de comunicação para a transferência de dados na web. O *HTTPS* é a versão segura do *HTTP*.

#### **Uso no Sistema:**

O *ESP32* pode fazer requisições *HTTP* para enviar dados para uma API ou um servidor web no *Raspberry Pi*, que pode, então, armazenar ou processar os dados.

O *Raspberry Pi* pode também servir uma interface web para visualizar os dados da estufa em tempo real.

### ***WebSockets***

**Descrição:** Um protocolo de comunicação que permite uma comunicação bidirecional entre cliente e servidor através de uma única conexão *TCP*.

#### **Uso no Sistema:**

Permite que o *ESP32* e o *Raspberry Pi* se comuniquem em tempo real, transmitindo dados de forma mais eficiente do que o *HTTP*.

O ***TCP (Transmission Control Protocol)*** é um dos principais protocolos de comunicação utilizados na Internet e em redes de computadores. Aqui estão alguns detalhes sobre como o *TCP* pode ser integrado em um sistema de monitorização de estufas com *ESP32*, *MQTT*, *Raspberry Pi 4* e sensores:

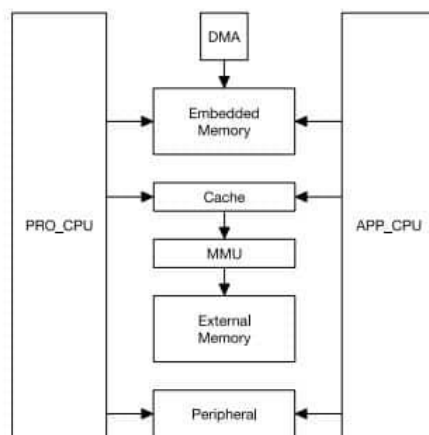
**Descrição:** O *TCP* é um protocolo de comunicação orientado à conexão que garante a entrega confiável de dados entre dispositivos. Ele é responsável por segmentar os dados em pacotes, controlar a ordem de entrega e assegurar que os pacotes cheguem ao destino sem erros.

**Camada:** O *TCP* opera na camada de transporte do modelo *OSI*, acima do *IP (Internet Protocol)*.



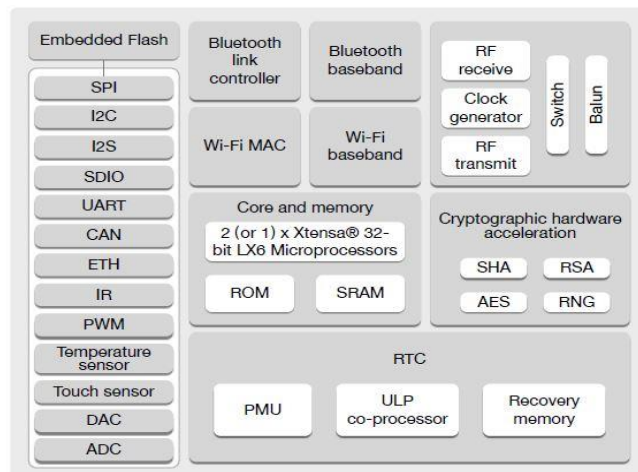
Em 2016 a *Espressif Systems* lançou o *ESP32*, um microcontrolador de baixo custo e alta eficiência energética, reconhecido por sua versatilidade. Ele é equipado com um processador dual-core de até *240 MHz*, *520 KB* de *RAM*, *4 MB* de memória flash interna e suporta uma ampla gama de periféricos, como *UART*, *SPI*, *I2C* e até mesmo interfaces para câmaras. Com essa combinação de recursos, o *ESP32* é a escolha perfeita para projetos que exigem conectividade e processamento eficientes, especialmente em ambientes onde a automação e o monitoramento são cruciais. [2].

O *ESP32* apresenta um sistema dual core: o *PRO\_CPU* para o protocolo e o *APP\_CPU* para aplicação. A memória interna, memória externa e periféricos localizam-se no barramento de dados e/ou barramento de instruções. Projetado para trabalhar até 32-bit e frequências de *clock* de 80 a 240 MHz ajustáveis. Apresenta um espaço de endereço para dados e barramento de instruções de 4 GB e uma *RAM* de *512 KB*. Exibe memórias internas *ROM* de *448 KB*, *SRAM* de *520 KB* e duas memórias *RTC* de *8 KB*. A memória externa suporta até quatro vezes *Flash* de *16 MB*. [4].



**Figura 4** - Arquitetura do *ESP32*. [F2].

O *ESP32* é um microcontrolador altamente versátil para projetos de Internet das Coisas (*IoT*), graças à sua ampla gama de periféricos e interfaces de comunicação, incluindo *SPI*, *UART*, *I2C*, *IR*, *SDIO*, *CAN*, *Ethernet*, *DAC*, sensores de toque e *I2C*. Além disso, possui *hardware* de criptografia embutido com suporte a *SHA1* e *AES*, aumentando a segurança das comunicações. O *ESP32* destaca-se pelo coprocessador de consumo ultrabaixo (*ULP*), que permite executar tarefas simples enquanto os processadores principais estão desligados, proporcionando modos de *sleep* mais avançados que o *ESP8266*. Em termos de conectividade, o *ESP32* suporta *Wi-Fi* e *Bluetooth 4.2* com *BLE*, ideal para dispositivos e aplicações que requerem baixo consumo de energia. A combinação dessas características torna o *ESP32* uma excelente escolha para o desenvolvimento de *IoT*.

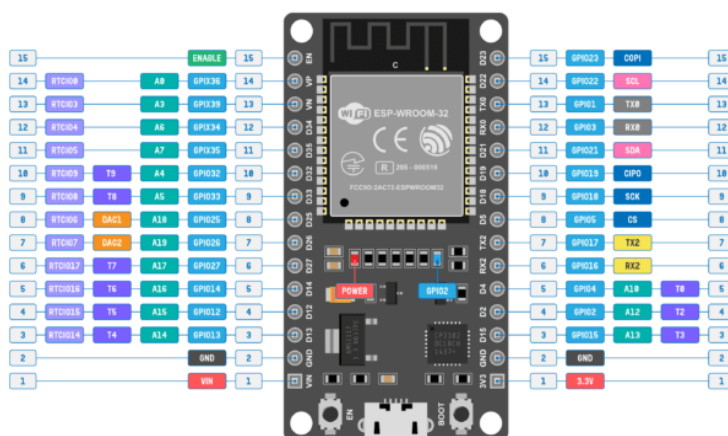


**Figura 5** - Diagrama de blocos do *ESP32*. [F3].

O *ESP32* é uma evolução significativa em relação ao *ESP8266*, disponibilizando melhorias e recursos adicionais. Já em relação ao *Arduíno*, podemos encontrar algumas semelhanças com o *Arduíno Uno*, visto que ambos são microcontroladores. No entanto o *ESP32* apresenta uma grande vantagem ao possuir *Wi-Fi* e *Bluetooth BLE* em oposição ao *ESP8266* que apenas possui *Wi-Fi* e ao *Arduíno* que não possui nenhuma dessas conectividades incorporadas.

Em termos de processador, o *ESP8266* possui um único núcleo de 32 bits, a uma frequência de até 160 MHz, enquanto o *ESP32* tem um sistema *dual core* de 32 bits cada um e um *clock* ajustável de 80 a 240 MHz, proporcionando um melhor desempenho. Já o *Arduíno Uno* é composto por um microcontrolador *ATmega328P* com um núcleo de 8 bits a operar a 16MHz, resultando num desempenho muito inferior ao *ESP32*. Relativamente à memória, o *ESP8266* dispõe de uma *RAM* de 160 KB enquanto o *ESP32* conta com *520KB de SRAM*, *448 KB de ROM*, duas memórias *RTC* de 8KB e suporte para memória externa de até 16MB de Flash e o *Arduíno* 2KB de SRAM e 32 KB de Flash. O microcontrolador *ESP32* apresenta uma vasta gama de periféricos, interfaces de comunicação e *hardware* de criptografia em relação ao seu antecessor. O *Arduíno Uno* apesar de apresentar algumas interfaces como *UART*, *I2C*, *SPI* e *PWM*, fica aquém da gama apresentada pelo *ESP32*. Os microcontroladores da família *ESP* destacam-se também pelo suporte de múltiplas linguagens de programação ao contrário do *Arduíno* que é limitado ao *Arduíno IDE* e à linguagem baseada em C/C++. [5].

O *ESP32* possui pinos de saída (*pinout*), tendo sido escolhida para este projeto uma placa ***ESP32 DEVKIT V1 DOIT*** de 30 pinos e 3.3V. Esta placa é composta por um pino de saída de 3.3V para alimentar periféricos externos, dois pinos de *GND*, um pino VIN que serve como alimentação externa e um pino EN (*enable*) usado para habilitar e desabilitar o chip. [5].



**Figura 6** - Placa ESP32 DevKit. [F4]

Os pinos GPIO (*General Purpose Input/Output*) permitem ser configurados tanto como entrada ou saída de leituras digitais e emissão de sinais de controlo de dispositivos externos. Possui também pinos ADC (*Analog-to-Digital Converter*) que permitem a conversão de sinais analógicos para digitais, essencial para alguns sensores que emitem sinais analógicos, ao contrário dos pinos DAC (*Digital-to-Analog Converter*) que convertem os sinais digitais em analógicos. Os pinos UART (*Universal Asynchronous Receiver/Transmitter*) possibilitam uma comunicação série, bastante importante na programação e comunicação com outros dispositivos serial. Os pinos I2C (*Inter-Integrated Circuit*) são utilizados para a comunicação com periféricos (sensores, *displays*, ...) com protocolo I2C. Existe pinos PWM (*Pulse Width Modulation*) capazes de controlar dispositivos como leds, modulando a largura de pulso dos sinais.

### 2.8.2. Sensor de Humidade do Solo *Arduino HD-38*

O sensor de Humidade do Solo *Arduino HD-38* é um dispositivo eletrónico projetado para medir com precisão os níveis de humidade no solo. Este sensor é ideal para monitorizar um jardim ou plantas em casa. Funciona através de duas ponteiros que enviam uma corrente elétrica pelo solo, medindo a resistência para determinar o nível de humidade. Quanto mais água o solo contiver, mais facilmente conduzirá eletricidade (menor resistência). Por outro lado, um solo mais seco conduzirá menos eletricidade (maior resistência). [6].



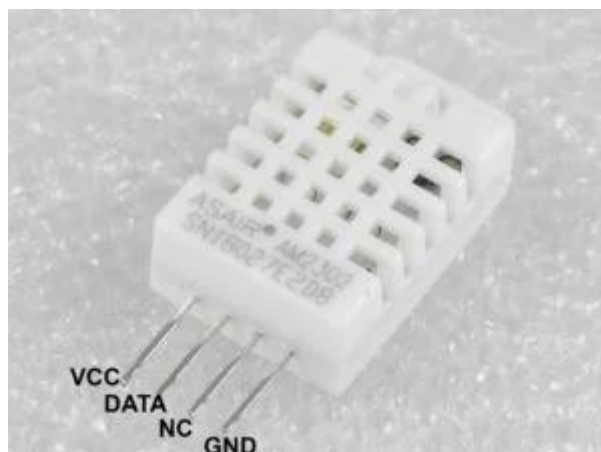
**Figura 7** - Sensor de humidade do Solo *Arduino HD-38*. [F6]

**Tabela 1** - Especificações Sensor de Humidade do solo *HD-38*

<b>Especificação</b>	<b>Descrição</b>
Modelo	HD-38
Tipo de Sensor	Sensor de humidade do solo
Tensão	5V a 12V DC
Corrente de trabalho	<20mA
Interface	Digital/Analógica
Comprimento do cabo da sonda	1m
Comprimento da sonda metálica	~79mm
Dimensões do módulo (CxLxE)	~35x15x7mm
Peso	21g

### 2.8.3. Sensor de humidade e temperatura *DHT22*

O sensor DHT22, também conhecido como AM2302, é um dispositivo que integra um sensor de humidade e um sensor de temperatura num único módulo, tornando-o ideal para monitorização ambiental em projetos eletrónicos. Ele fornece medições rápidas e precisas tanto da temperatura como da humidade do ar. O sensor utiliza um sensor capacitivo de humidade para medir a quantidade de vapor de água no ar e um termistor para medir a temperatura. Estes dois elementos estão ligados a um controlador de 8 bits, que processa as leituras e gera um sinal digital no pino de dados, facilitando a comunicação com microcontroladores como o Arduino, Raspberry Pi e outros. [7].



**Figura 8** - Sensor DHT22/AM2302 – pinout. [F5].

**Tabela 2** - Especificações Sensor de Humidade/Temperatura *DHT22*

<b>Especificação</b>	<b>Descrição</b>
Modelo	<i>AM2302 DHT22</i> Original
Tipo de Sensor	Sensor de temperatura e Humidade
Tensão	3.3 - 5V DC
Corrente de Operação	0,5 ~ 2,5 mA
Faixa de operação de umidade	0 ~ 100%
Margem de Erro ou Precisão	$\pm 2 \sim 5\%$ RH / $\pm 0,5^\circ\text{C}$
Faixa de operação da temperatura	-40 a 80° celsius
Dimensões (CxLxA)	~34x16x8mm (com pinos)
Peso	2,2g

#### **2.8.4. Sensor de Temperatura *DS18B20***

O sensor de Temperatura *DS18B20* é um componente eletrónico digital desenvolvido para ser utilizado em diversos ambientes, sendo capaz de medir a temperatura em locais húmidos, incluindo debaixo de água. Neste projeto, este sensor foi utilizado para monitorizar as condições ambientais do solo. Para que o sensor de temperatura funcione, é necessário conectá-lo a uma plataforma de prototipagem, como um *ESP32* ou *Raspberry Pi*. [8].



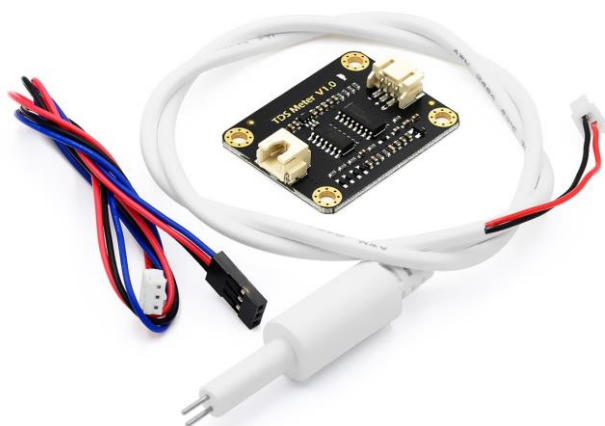
**Figura 9** - Sensor de Temperatura *DS18B20*. [F7]

**Tabela 3** - Sensor de Temperatura *DS18B20*

<b>Especificação</b>	<b>Descrição</b>
Modelo	<i>DS18B20</i>
Tipo de Sensor	Sensor de temperatura
Tensão da operação	3-5,5V
Faixa de medição	-55°C a +125°C
Precisão	±0.5°C entre -10°C e +85°C
Ponta de aço	inoxidável
Dimensão ponta de aço	6 x 50mm
Dimensão do cabo	100cm
Interface	de 1 fio

### 2.8.5. Sensor de condutividade da água *TDS*

O medidor *TDS* é um dispositivo eletrónico que atua como um sensor de condutividade da água doce, sendo amplamente utilizado nas áreas da agricultura, incluindo a hidropónica e a aquaporina, para controlar a qualidade da água. O sensor de condutividade da água, também conhecido simplesmente como medidor de *TDS*, possui uma sonda impermeável que pode ser submersa na água, permitindo que os seus elétrodos meçam a condutividade num determinado local. É importante salientar que as sondas não devem ser utilizadas em água com temperaturas superiores a 55 °C, nem nas proximidades das bordas dos recipientes, para evitar alterações nos valores medidos. [9].



**Figura 10** - Sensor de condutividade da água TDS. [F8].

**Tabela 4** - Sensor de condutividade da água TDS

<b>Especificação</b>	<b>Descrição</b>
Modelo	TDS
Tipo de Sensor	Sensor de condutividade da água
Tensão de entrada	3.3 ~ 5.5 V
Tensão de saída	0 ~ 2.3V
Corrente de trabalho	3 ~ 6mA
Faixa de Medição TDS	0 ~ 1000ppm
Precisão de medição TDS	± 10% FS (25 °C)
Tamanho do módulo	42 * 32mm
Interface do módulo	PH2.0-3P
Interface do eléctrodo	XH2.54-2P
<b>Sonda TDS</b>	
Número de agulhas	2
Comprimento total	83 cm
Interface de conexão	XH2.54-2P
Cor	Branca
Outros	Sonda à prova d'água.

#### 2.8.6. Mini-Bomba de Água para Arduino 12V RS385 2L/Min.

Projetada para operar em sistemas de baixa pressão, a mini-bomba é ideal para aplicações em estufas, onde a precisão na distribuição da água é crucial para o crescimento saudável das plantas. Com um design compacto e uma funcionalidade fiável, a mini-bomba facilita a gestão da humidade do solo, garantindo que a irrigação seja realizada de forma consistente e eficiente. [10]



**Figura 11** - Mini-Bomba de Água para Arduino 12V RS385 2L/Min. [F10]

**Tabela 5** - Mini-Bomba de Água para *Arduino* 12V RS385 2L/Min

<b>Especificação</b>	<b>Descrição</b>
Modelo	RS385
Materiais	metal e plástico
Tensão nominal	12V
Voltagem adequada	DC 9 a 15V
Corrente sem carga	0,6 <sup>a</sup>
Corrente em máxima eficiência	~2 <sup>a</sup>
Altura de aspiração máxima	2m
Elevação máxima	3m
Vazão de água máxima	~1,5 a 2 l/m
Diâmetro de entrada e saída	~7,6mm
Diâmetro do motor	28,6mm
Comprimento da bomba	90mm
Peso	100g

### 2.8.7. Ventilador de alta velocidade

O Ventilador Ventoinha FAN 12025 24V 2 fios MEC0252V1-000U-A99 da marca SUNON é um ventilador de tamanho padrão 120x120x25mm, utilizado para arrefecimento em sistemas eletrónicos, caixas de computadores, servidores e equipamentos industriais. Neste projeto, utilizamos o ventilador para arrefecer os componentes eletrónicos quando estes estiverem sobreaquecidos ou quando os sensores atingirem uma determinada temperatura significativa.



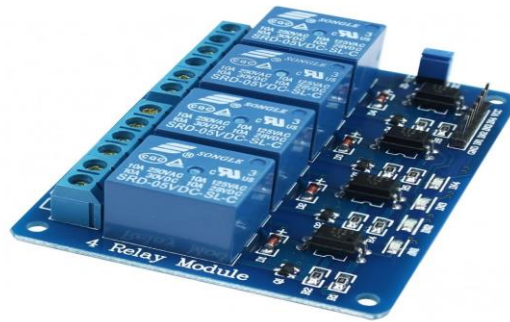
**Figura 12-** Ventilador [F19]

**Tabela 6** – Ventilador

<b>Características</b>	<b>Especificações</b>
Modelo	MEC0252V1-000U-A99
Tensão de alimentação	24V DC
Tensão de trabalho	10...27.6V
Tamanho do ventilador	120x120x25mm
Consumo de potência	5W
Corrente nominal	0.207A
Eficiência dos ventiladores	108.2 CFM (3.03m <sup>3</sup> /min)
Nível de barulho	45.5dBA
Tipo de rolamento	Vapo-Bearing™
Velocidade giratória	3100 (±15%) rot./min
Peso	162g
Tolerância à potência e consumo de corrente	±15%

#### **2.8.8. Módulo Relé 5V 10A 4 Canais com Optoacopladores.**

O Módulo Relé 5V de 4 Canais é um dispositivo eletrônico usado para controlar cargas de alta potência (como lâmpadas, motores, ventiladores, entre outros) através de sinais de baixa potência emitidos por microcontroladores ou sistemas de computação, como Arduino, Raspberry Pi, ESP32, etc. Ele contém quatro relés, que são interruptores eletromecânicos. Cada relé pode ser ativado ou desativado por um sinal elétrico de baixa tensão (neste caso, 5V), permitindo controlar dispositivos que utilizam corrente alternada (AC) ou corrente contínua (DC), com uma capacidade de até 10A por canal. [10].



**Figura 13** - Módulo Relé 5V 10A 4 Canais com Optoacopladores [F9]

**Tabela 7** - Módulo Relé 5V 10A 4 Canais com Optoacopladores

Especificação	Descrição
Carga nominal	10A250VAC/ 10A 125VAC/ 10A 30VDC/ 10A 28VDC
Tensão de operação	5VDC (VCC e GND)
Tensão de sinal	TTL 5VDC (IN1, IN2, IN3, IN4)
Saídas	Contato reversível NA (normal aberto), NF (normal fechado), C (comum)
Corrente por canal	Até 10A
Dimensões totais (CxLxA)	7,5x5x1,8cm
Peso com embalagem	60g

### 2.8.9. Display OLED SSD1306

O ecrã *OLED SSD1306* é um monitor monocromático extremamente económico, ideal em comparação com outros tipos de displays. Com este ecrã, é possível desenhar formas e exibir textos e imagens bitmap de maneira eficiente.



**Figura 14** - Display OLED SSD1306

#### 2.8.9.1. Características principais:

- **Tamanho:** 0,96 polegadas
- **Resolução:** 128×64 píxeis
- **Retroiluminação:** Não necessita, proporcionando excelente contraste em ambientes escuros

- **Pinos:** 4 no total
- **Alimentação:** *VCC* e *GND*
- **Dados:** *SCL* (relógio de série) e *SDA* (dados de série)

#### 2.8.9.2. Comunicação de Dados:

- **I2C:** O modelo *SSD1306* utiliza este protocolo, que é ideal para quem precisa de simplicidade na fiação, já que requer apenas dois pinos de dados e pode ser partilhado com outros periféricos.

#### 2.8.9.3. Controlador *SSD1306*:

Este módulo integra o *SSD1306*, um controlador *CMOS OLED* de chip único. Ele é responsável por gerir a comunicação com o microcontrolador e pode operar tanto em *I2C* como em *SPI*. A escolha entre os dois modos depende das necessidades do projeto:

- **I2C:** Menos pinos, mais simplicidade na fiação.

## 2.9. *Raspberry pi*

O *Raspberry Pi* é uma pequena placa de computador, do tamanho de um cartão de crédito, desenvolvida no Reino Unido pela Fundação *Raspberry Pi*, com o objetivo de promover o ensino básico de informática nas escolas. Desde a sua primeira venda em 2012, foram comercializadas mais de 46 milhões de placas *Raspberry Pi* até fevereiro de 2022. A imagem abaixo mostra um *Raspberry Pi 4*, modelo B+. O modelo mais recente é o *Raspberry Pi 5*. Neste projeto, estamos a utilizar o *Raspberry Pi 4*. [11].

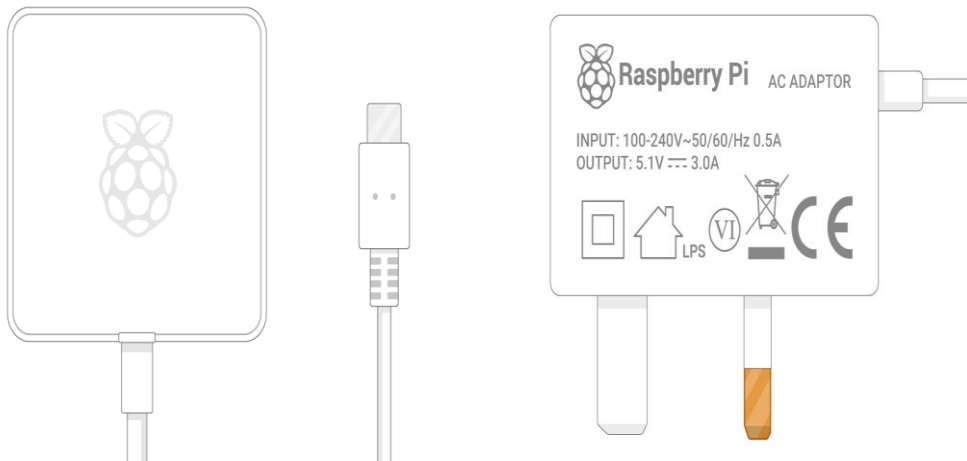


**Figura 15 - *Raspberry pi 4*. [F11]**

### 2.9.1. **Dispositivos necessários para o uso do *Raspberry pi***

#### 2.9.1.1. Uma fonte de alimentação

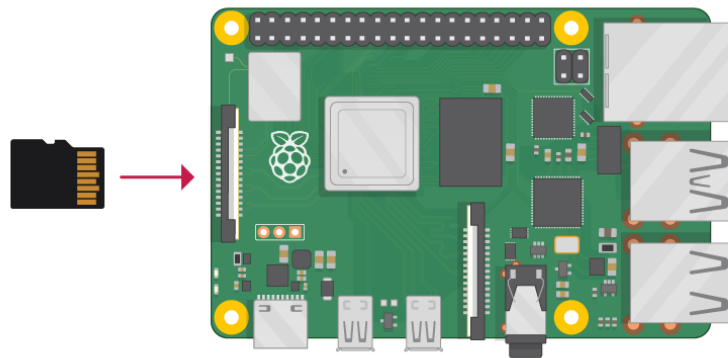
Para conectar a uma tomada, todos os modelos *Raspberry Pi* têm uma porta *USB* (a mesma encontrada em muitos celulares): *USB-C* para *Raspberry Pi 4* ou *micro USB* para *Raspberry Pi 3, 2 e 1*. [12].



**Figura 16** - Fonte de alimentação. [F12].

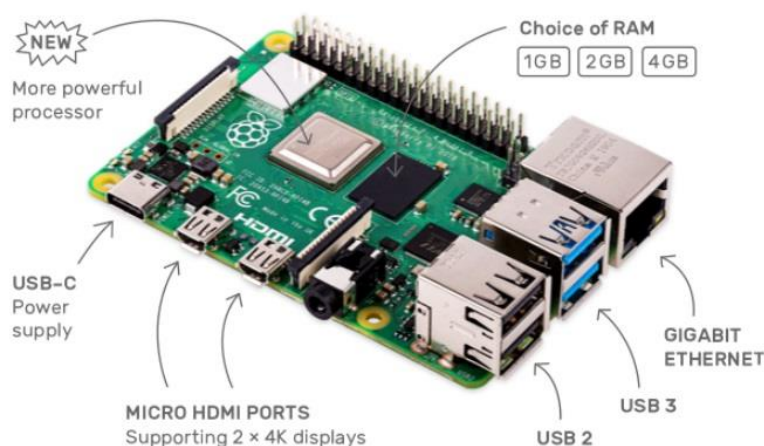
### 2.9.1.2. Um cartão microSD

Como qualquer computador, o *Raspberry pi 4*, usa um cartão *microSD*, como ilustra a imagem seguinte:



**Figura 17** - Inserção do cartão *microSD*. [F12].

Depois da configuração base do Raspberry, podemos usar periféricos como um teclado e um mouse (rato). Mas para visualizar o ambiente de desktop do Raspberry Pi OS, nós precisamos de um monitor ou de um computador, usando alguns softwares de acesso remoto como o “termius”. Abaixo temos uma ilustração de todas as portas do Raspberry pi:



**Figura 18** - Raspberry Pi 4, portas e componentes. [F13].

**Tabela 8** - Raspberry Pi 4, portas e componentes

Especificação	Descrição
Processador	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memória	4GB LPDDR4
Conectividade	Wireless 2.4 GHz e 5.0 GHz IEEE 802.11b/g/n/ac
	LAN, Bluetooth 5.0, BLE
	Gigabit Ethernet
	2× portas USB 3.0
	2× portas USB 2.0
GPIO	Standard 40-pin GPIO header
Video e Som	2× portas micro HDMI (suporta até 4Kp60)
	2-lane MIPI DSI display port
	2-lane MIPI CSI camera port
	Áudio estéreo de 4 pólos e porta de vídeo composto
Multimédia	H.265 (4Kp60 decode);
	H.264 (1080p60 decode, 1080p30 encode);
	OpenGL ES, 3.0 graphics
Suporte cartão SD	Micro SD para carregar o sistema operativo e armazenamento de dados
Alimentação	5V DC via USB-C (mínimo 3A)
	5V DC via GPIO header (mínimo 3A)
	Power over Ethernet (PoE) - (requer adaptador em separado PoE HAT)
Temperatura de funcionamento	0 a 50°C

## 2.10. Software

### 2.10.1. Protocolo MQTT

Este protocolo realiza a comunicação *publish/subscribe*, permitindo taxas de transmissão que possibilitam a utilização até de *hardware* com baixo poder de processamento, embora com um grau relativo de entrega de informações e confiabilidade. Essas características tornam o protocolo *MQTT* uma das melhores alternativas para projetos que envolvem IoT. O protocolo é amplamente utilizado por ser leve, exigir pouco do hardware, ter baixo consumo de dados e oferecer uma comunicação bilateral, ou seja, do cliente para o servidor e vice-versa.

### 2.10.2. Conceito de MQTT

O protocolo *MQTT* (*Message Queuing Telemetry Transport*), desenvolvido na década de 90 pela *IBM* e pela *Eurotech*, é um protocolo de mensagens que visa oferecer um baixo consumo de rede, largura de banda e outros recursos de software. É leve, aberto, simples e foi concebido para ser fácil de implementar. O formato utilizado no *MQTT* é de Cliente/Servidor.

O *MQTT* surgiu da necessidade de desenvolver um protocolo simples e leve que permitisse a comunicação entre máquinas. Atualmente, o protocolo é padronizado por uma organização dedicada ao avanço de normas de informação.

### 2.10.3. Definição de Broker MQTT

O broker é o elemento responsável por gerir as publicações e as subscrições no protocolo *MQTT*. Funciona como um mediador entre as máquinas, facilitando a comunicação entre elas. O broker permite um desacoplamento entre as partes, uma inovação neste tipo de sistema.

Definido como o servidor intermediário (broker, em inglês), ele gere as informações recebidas dos clientes que publicam dados (como sensores, por exemplo) e transmite-as aos clientes que solicitam esses dados.

### 2.10.4. Node-red

O *Node-RED* é uma poderosa ferramenta de código aberto para a criação de aplicações da Internet das Coisas (*IoT*) com o objetivo de simplificar a componente de programação. O *Node-RED* é executado no navegador *Web* e utiliza programação visual que permite ligar blocos de código, conhecidos como nós, para executar uma tarefa. Os nós, quando ligados entre si, são designados por fluxos. [15].

O *Node-RED* é de código aberto e foi desenvolvido pela *IBM*. O *Raspberry Pi* funciona perfeitamente com o *Node-RED*. É uma ferramenta de programação visual, o que a torna mais acessível a um maior número de utilizadores. Com o *Node-RED* pode passar mais tempo a fazer coisas interessantes, em vez de passar inúmeras horas a escrever código. [15]

### **2.10.5. Visual Studio Code**

O *Visual Studio Code* é uma das *IDEs (Integrated Development Environment, ou ambiente de desenvolvimento integrado)* mais populares do mercado. Esta ferramenta é multiplataforma, estando disponível para os sistemas operativos *Windows, Linux e MacOS*, e pode ser adaptada às necessidades de quem a utiliza através da instalação de extensões.[23]

## 3. Capítulo 3 - implementação do sistema

### 3.1. Conceitos práticos

Neste capítulo iremos abordar os passos necessários para a realização do projeto

O projeto proposto consiste na implementação e monitoramento de um sistema de irrigação automatizado utilizando os conceitos de internet das coisas, onde foram utilizados sensores, microntroladores e atuadores, conectados em uma mesma rede *WI-FI*, onde os dados coletados foram transmitidos via protocolo de comunicação *MQTT*. A irrigação no sistema realizou-se de forma automática com condição estabelecida em código de humidade do solo menor 50%, mas é importante destacar que a escolha deste valor para a realização da irrigação foi apenas para este projeto proposto, não sendo utilizado a referência de umidade de nenhuma cultura de planta em específico, pois para cada cultura de plantas existe um valor de humidade adequado para irrigação bem específico e o monitoramento observou-se de forma remota, com a exibição dos dados por meio de interface gráfica em dashboard.

### 3.2. Tipo de solo para estufas em Portugal

Existe uma grande variedade de solos em Portugal, embora predominem os solos pobres e pouco profundos, nos quais o crescimento vegetal tem limitações. A sua fertilidade pode, no entanto, ser melhorada por práticas de gestão adequadas que, além de apoiarem a produtividade vegetal (agrícola e florestal), devem ter em conta a necessidade de equilíbrio e de preservação deste sistema natural.[19]



**Figura 19** - Tipos de solos em Portugal, para diferentes cultivos. [F17].

Neste projeto, a terra utilizada para a estufa é composta por um solo orgânico da “**Siro Germinação BIO**” com características de alta retenção de humidade, para proporcionar as condições ideais para o cultivo de hortaliças. O solo orgânico, rico em matéria orgânica, contribui significativamente para o crescimento saudável das plantas, fornecendo nutrientes essenciais e favorecendo a aeração e a drenagem

adequadas. A presença de humidade controlada no solo garante que as raízes das hortaliças possam se desenvolver de maneira uniforme, promovendo um crescimento mais robusto e vigoroso, ideal para a produção em ambiente protegido. [20]

O solo Siro germinação BIO, é um substrato especial para a germinação em - MPB, de todo o tipo de sementes. A sua textura fina de baixa salinidade proporciona uma adequada capacidade de retenção de água, facultando um teor de humidade constante, para promover uma elevada taxa de germinação com perfeito desenvolvimento da plântula.[20]

Este substrato é equilibrado e enriquecido com adubo orgânico biológico, favorecendo a sua componente nutricional na fase inicial de desenvolvimento das plantas. Composição: Turfa loira de Sphagnum, Siro Agro 1 (Húmus de casca de pinho - certificado pela RAL), coco peat e Adubo orgânico biológico de origem animal. Abaixo temos a imagem da embalagem: [20]



**Figura 20** - Embalagem e terra da sira germinação Bio. [F18]

### **Características Físico-químicas:**

pH em CaCl<sub>2</sub>: 5.5-6.5

Condutividade: 150 µs/cm

Granulometria 0-8mm

Matéria Orgânica > 60%

Fertilização biológica

NPK 9-2-2 - 2 kg/m<sup>3</sup>

### **Condições do solo**

### 3.3. Placa eletrônica ESP32

O projecto que desenvolvemos, utilizamos o *ESP32*, uma placa de desenvolvimento de código aberto baseada no poderoso chip ESP32. Assim como o *Arduino Uno*, o *ESP32* é um microcontrolador, porém, ele oferece vantagens significativas, como conectividade *Wi-Fi* e *Bluetooth* integradas, o que o torna ideal para aplicações em projetos de Internet das Coisas (IoT). As possibilidades de programação também são uma característica notável, já que é compatível com *C*, *C++*, *Python*, *Lua*, *MicroPython*, além de *JavaScript*. [1]

O *ESP32* é suportado por diversas plataformas de desenvolvimento, como *Arduino IDE*, *PlatformIO* e *Espressif IDF*.

Neste projeto a ferramenta de desenvolvimento usada para programar o *ESP32* foi o *Visual Studio Code* com a extensão *PlatformIO*

#### 3.3.1. Instalação do ambiente de desenvolvimento

Para a instalação do ambiente de desenvolvimento, foram realizados os seguintes passos:

1º) Download e Instalação:

- Acesso ao [site oficial do VS Code](#).
- Baixe o instalador apropriado para o seu sistema operacional (*Windows*, *macOS* ou *Linux*).
- Execute o instalador e siga as instruções na tela.

2º) Abrir o *VS Code*:

- Após a instalação, abra o *VS Code*.

3º) Instalar a Extensão do *PlatformIO*:

- Abra o *VS Code*.
- Vá para o painel de extensões clicando no ícone de extensões (quatro quadrados) na barra lateral esquerda ou pressione *Ctrl + Shift + X*.
- Pesquise por "*PlatformIO IDE*" e clique em "Instalar".

4º) Criar um Novo Projeto:

- Após a instalação, abra o *Command Palette* pressionando *Ctrl + Shift + P*.
- Digite "*PlatformIO: New Project*" e selecione a opção.
- Preencha os campos solicitados, como o nome do projeto, a placa de desenvolvimento que você está usando e a pasta onde deseja salvá-lo.
- Clique em "*Finish*" para criar o projeto.

5º) Configurar a Placa e a Biblioteca:

- O *PlatformIO* criará um arquivo *platformio.ini* na raiz do seu projeto. Você pode adicionar bibliotecas ou ajustar configurações de compilação nesse arquivo.
- Para instalar bibliotecas, use o *Command Palette* novamente e digite “*PlatformIO: Install Library*”.

### 3.3.2. Integração dos sensores e atuadores com o ESP32

#### 3.3.2.1. Sensor de humidade do Solo Arduino HD-38

Para a integração deste sensor com o *ESP32*, ligamos o fio positivo do sensor ao pino de alimentação do *ESP32* 3.3V, o fio preto à *GND* e o fio de dados do sensor ao pino 17 *GPIO 36*. Para o nosso projeto usamos este sensor para nos fornecer a humidade do solo na A figura 20 representa o esquema de ligações do sensor de humidade do solo, a figura 20 mostra a imagem do sensor na estufa.

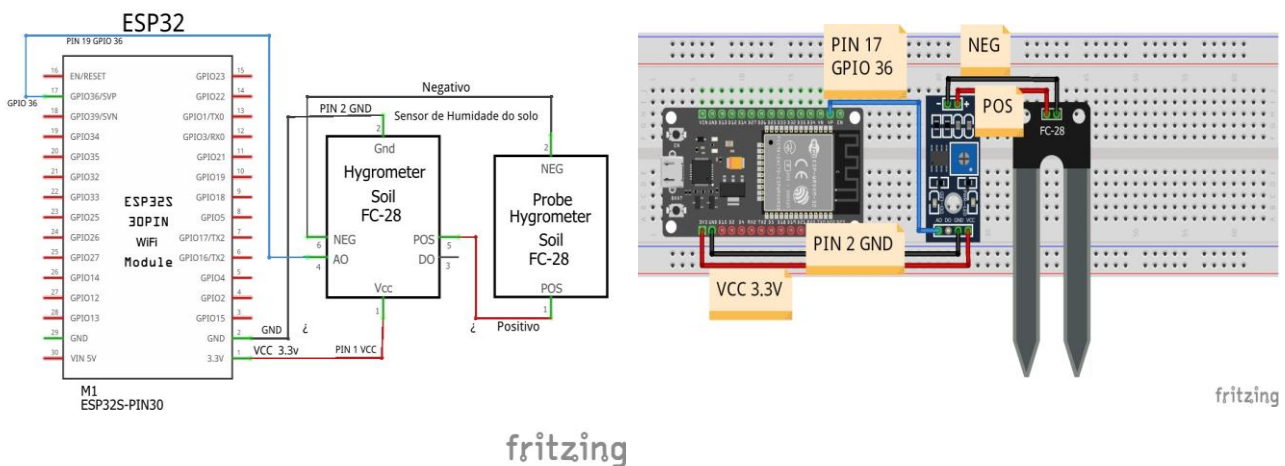


Figura 21 - Montagem do sensor de humidade do solo

```
int humidade,sensor_analogico;
const int sensor_pin = 36; /* Pino O/P do sensor de humidade do solo */

void setup(void){
  Serial.begin(115200); /* Definir a taxa de transmissão para 115200 */
}

void loop(void){
  sensor_analogico = analogRead(sensor_pin);
  humidade = ( 100 - ( (sensor_analogico/4095.00) * 100 ) );
  Serial.print("Humidade do solo = ");
  Serial.print(humidade);/*Impri a temperatura na janela de série */
  Serial.println("%");
  delay(1000); /* Esperar 1000ms */
}
```

Acima temos o código feito para o funcionamento do sensor de humidade do solo, a leitura da humidade é feita pelo comando “*analogRead(x)*”, em que *x* é o número do

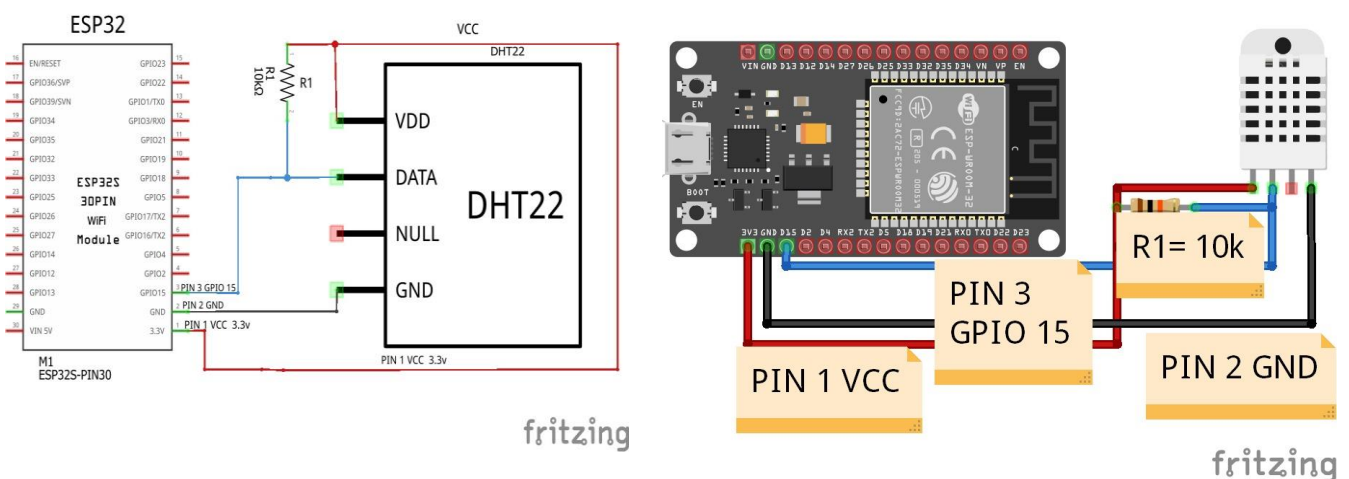
pino do *ESP32*, neste caso 0 pin 36. Como queríamos os valores em percentagem tivemos de fazer a seguinte operação

$$\text{humidade} = (100 - ((\text{valor vindo do sensor}/4095) * 100))$$

Desta forma, tivemos os valores desejados entre 0% e 100%, em que 0% corresponde ao solo seco e os 100% correspondem a um solo húmido.

### 3.3.2.2. Sensor de humidade e temperatura *DHT22*

O sensor *DHT22* permiti-nos saber a situação climática dentro da estufa, dando os dados da temperatura e humidade do ar. Este sensor tem 4 pinos, mas nós só usamos 3 pinos sendo o pino *VCC* ligado alimentação a 3.3V, o *GND* para a *GND* do *ESP32*, o pino dados para GPIO 15. Colocou-se uma resistência de 10kΩ entre a alimentação (*VCC*) e o pino de dados (*DATA*). Esta resistência é designada por resistência de *pull-up* garante que a linha de dados esteja num estado lógico alto quando não existe nenhum dispositivo a impor um nível baixo, prevenindo flutuações de dados e leituras incorretas. A figura 21 representa o esquema de ligações do sensor de temperatura e humidade do ar, a figura 21 mostra a imagem do sensor na estufa.



**Figura 22** - Montagem do sensor de humidade/Temperatura do ar *DHT22*.

```
#include <DHT.h>

hum_ar = dht.readHumidity();
temp_ar = dht.readTemperature();
```

Para programar este sensor foi necessário importar a biblioteca `#include <DHT.h>`, os valores da humidade é dado pela função `dht.readHumidity()`; valores da temperatura é dado pela função `dht.readTemperature()` em graus Celsius

### 3.3.2.3. Sensor de Temperatura DS18B20

Este sensor permite-nos saber a temperatura do solo da estufa. Para a integração deste sensor com o *ESP32*, ligamos o pino positivo do sensor ao pino de alimentação do *ESP32* 3.3V, o fio *GND* do sensor ao pino *GND* do esp32. O pino dados para GPIO 27. Colocou-se uma resistência de 4.7kΩ entre a alimentação (*VCC*) e o pino de dados (*DATA*). A esta resistência chama-se resistência *pull-up* que permite o correto funcionamento do protocolo *One-Wire* utilizado por este sensor, assim como a garantia de que o barramento de dados mantém o estado de repouso adequado, permitindo comunicação bidirecional confiável e prevenindo interferências. A figura 22 representa o esquema de ligações do sensor de temperatura e humidade do ar, a figura 22 mostra a imagem do sensor na estufa.

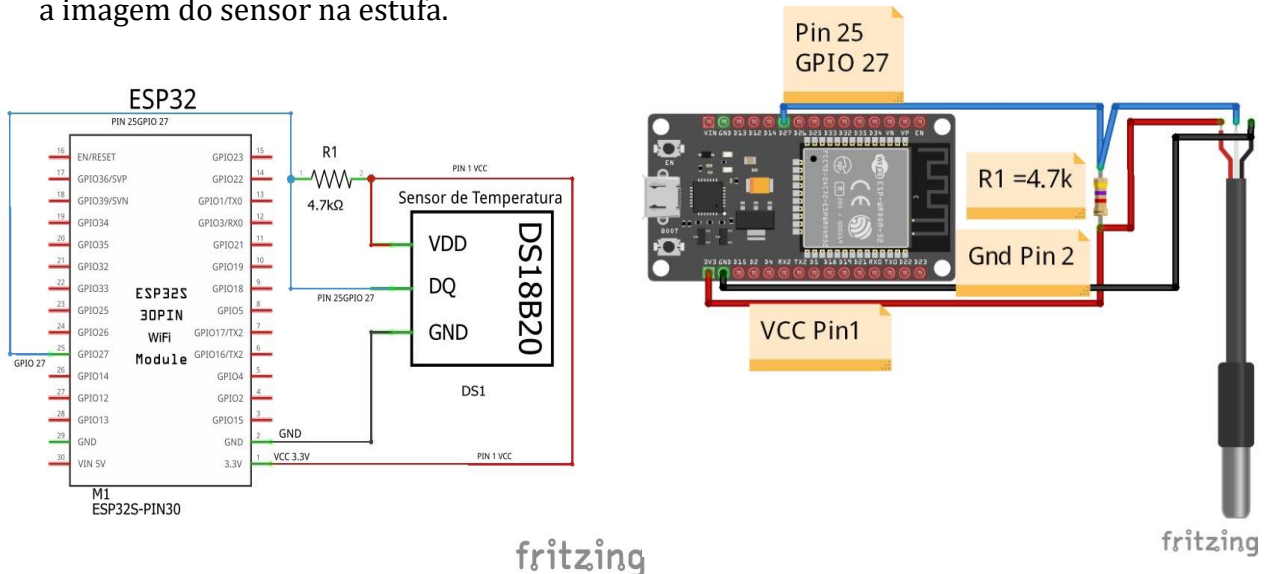


Figura 23 - Montagem do sensor de Temperatura DS18B20

A programação para o sensor de temperatura DS18B20 começa com a criação de um objeto da classe *OneWire*. Este objeto é responsável por gerenciar a comunicação no barramento 1-Wire, que é o protocolo utilizado pelo sensor DS18B20 para transmitir dados. O barramento 1-Wire permite conectar vários dispositivos em um único fio, sendo essencial para a interação com o DS18B20. Criação do objeto *OneWire*: Inicializa a comunicação com o sensor no barramento 1-Wire, conectado ao pino 27.

Criação do objeto *DallasTemperature*: Esse objeto, ligado ao *OneWire*, gerencia a comunicação com o sensor DS18B20. `DallasTemperature sensors(&oneWire);`

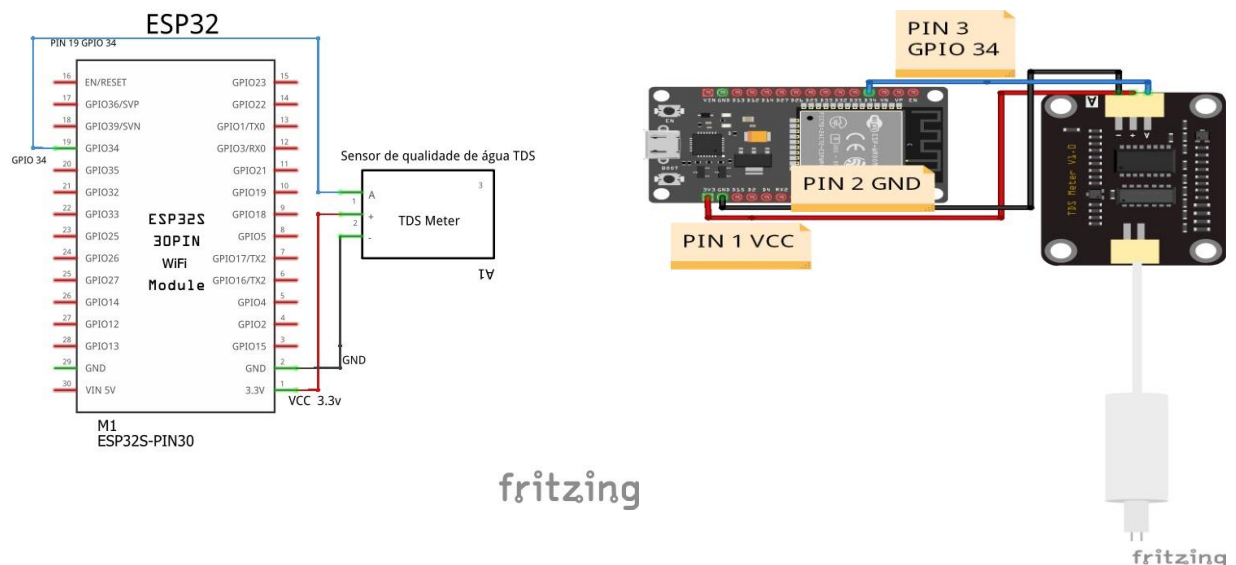
**Inicialização do sensor:** No `setup ()`, o sensor é preparado para medições com `sensors.begin()`.

`sensors.requestTemperatures()`: Solicita ao sensor que faça uma leitura de temperatura.

***sensors.getTempCByIndex(0)***: Obtém a temperatura em graus Celsius do sensor no índice 0 do barramento.

### 3.3.2.4. Sensor de condutividade da água TDS

Para a integração deste sensor com o esp32, ligamos o fio positivo do sensor ao pino de alimentação do ESP32 3.3V, o fio preto à GND e o fio de dados do sensor ao pino 17 GPIO 34. Para o nosso projeto usamos este sensor para vermos a qualidade a qualidade da água que estamos regando a nossa estufa. A figura 23 representa o esquema de ligações do sensor de humidade do solo, a figura 23 mostra a imagem do sensor na estufa.



**Figura 24** - Montagem do sensor de condutividade da água TDS

O sensor *TDS Meter* mede a quantidade de sólidos dissolvidos na água (*Total Dissolved Solids - TDS*), expressa em partes por milhão (ppm). Este valor reflete a presença de minerais, sais e outras substâncias dissolvidas, sendo um indicador da qualidade da água.

### 3.3.2.5. Funcionamento:

1. Princípio de Medição: O sensor mede a condutividade elétrica da água. Quanto mais sólidos dissolvidos, maior é a condutividade. O sensor converte essa condutividade num valor de *TDS* (em ppm).
2. Compensação de Temperatura: A temperatura da água afeta a condutividade, por isso, o sensor aplica um coeficiente de compensação para corrigir a medição, tendo como base uma temperatura de referência (geralmente 25°C).
3. Leitura dos Valores: O sensor lê a tensão elétrica, que é convertida num valor digital. Este valor é, então, processado para calcular o *TDS* com base numa fórmula específica.

4. Aplicações: O *TDS Meter* é usado para verificar a pureza da água, classificar a água como pura, aceitável, ou de má qualidade, dependendo do nível de *TDS*.

### 3.3.2.6. Atuadores do sistema

Neste sistema de monitorização de estufas implementado existem três atuadores, são: Módulo relé de 4, Mini-Bomba de Água para Arduino 12V para regar a estufa e um ventilador para o sistema de ventilação.

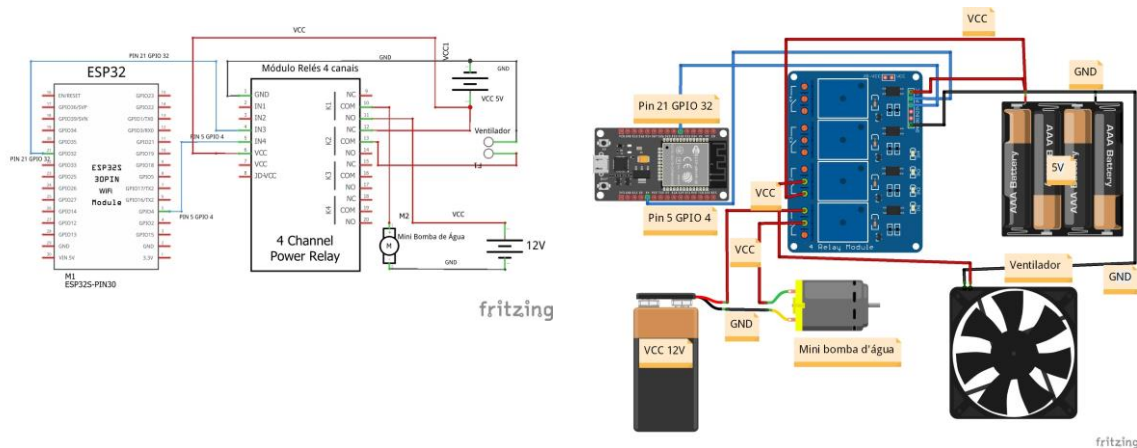
#### Sistema de Rega:

- Condições de funcionamento:
- Quando a humidade do solo é menor a 50% ( $\text{humidade\_solo} < 50$ ), o sistema entende que o solo está seco e precisa ser regado.
- Caso a humidade esteja acima de 50%, o sistema considera que o solo já está suficientemente regado.
- Ações executadas:
- Se o solo estiver seco ( $\text{humidade} < 50\%$ ):
- A bomba de água é ativada, ligando o Relé 4 ( $\text{digitalWrite}(\text{Rele\_4}, \text{LOW})$ ), o que aciona a bomba e começa a regar a estufa.
- Se o solo estiver húmido ( $\text{humidade} > 50\%$ ):
- A bomba é desligada, desativando o Relé 4 ( $\text{digitalWrite}(\text{Rele\_4}, \text{HIGH})$ ).

#### Sistema de Ventilação:

- Condições de funcionamento:
- Quando a temperatura do ar atinge ou excede 25°C ( $\text{temp\_ar} \geq 25$ ), o sistema considera que é necessário ligar o ventilador para controlar a temperatura.
- Se a temperatura estiver abaixo de 25°C, o ventilador é desligado.
- Ações executadas:
- Se a temperatura estiver alta ( $\text{temp\_ar} \geq 20^\circ\text{C}$ ):
- O ventilador é ativado, ligando o Relé 3 ( $\text{digitalWrite}(\text{Rele\_3}, \text{LOW})$ ), o que aciona o ventilador.
- Se a temperatura estiver baixa ( $\text{temp\_ar} < 20^\circ\text{C}$ ):
- O ventilador é desligado, desativando o Relé 3 ( $\text{digitalWrite}(\text{Rele\_3}, \text{HIGH})$ ).

A figura 24 representa o esquema de ligações dos atuadores, a figura 24 mostra a imagem dos atuadores na estufa.



**Figura 25** - Montagem do Ventilador, mini-bomba de água e o módulo relés de 4 canais

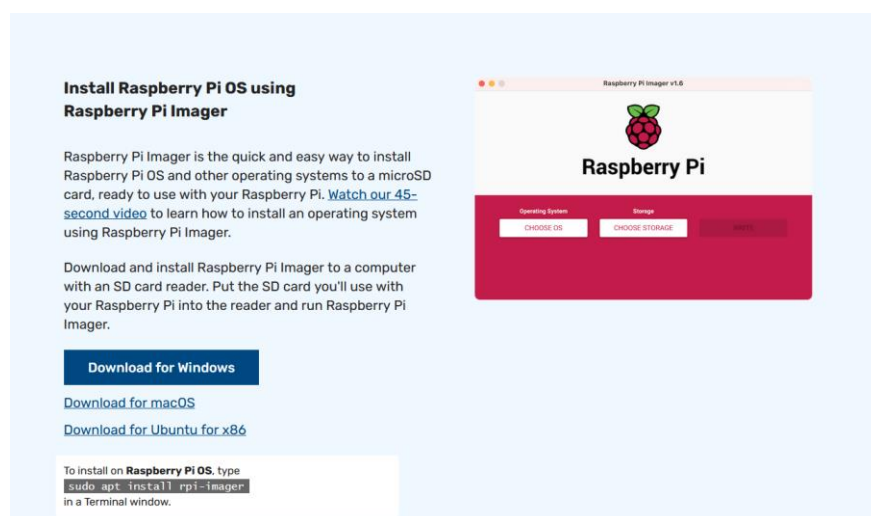
### 3.4. Raspberry pi, mqtt (mosquitto broker) e node-red

O *Raspberry Pi* será utilizado neste projeto para gerir a página web, uma vez que é um mini-computador versátil. Com o *Node-RED*, iremos criar *dashboard* que permitirá monitorizar os fatores ambientais da estufa, possibilitando a visualização dos dados recolhidos, como a temperatura do ar e do solo, a humidade do solo e do ar, e a qualidade da água.

#### 3.4.1. Instalando o sistema operacional do *Raspberry pi 4*, via *Raspberry Pi Imager*

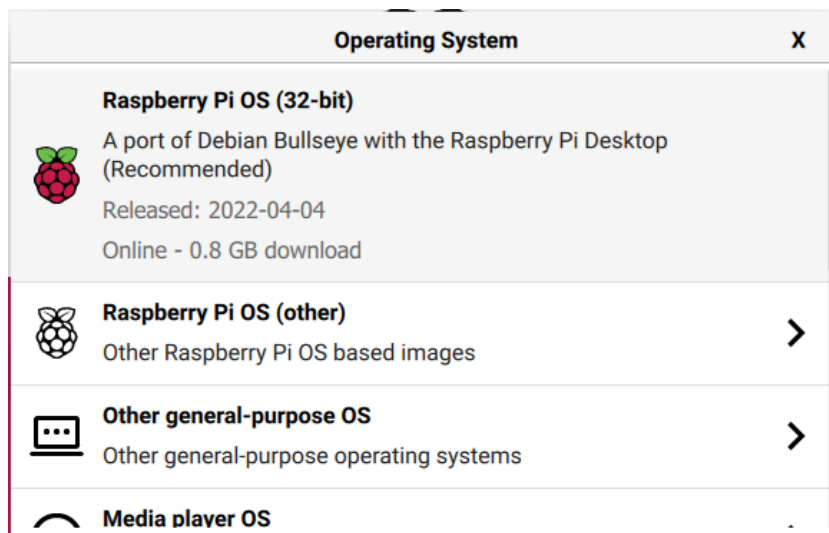
O *Raspberry Pi Imager* é a maneira mais fácil de instalar o *Raspberry Pi OS* no seu cartão SD.

1º) Baixar a imagem do sistema operacional do *Raspberry pi 4*, para as versões correspondentes aos nossos computadores, como ilustra a imagem a seguir:



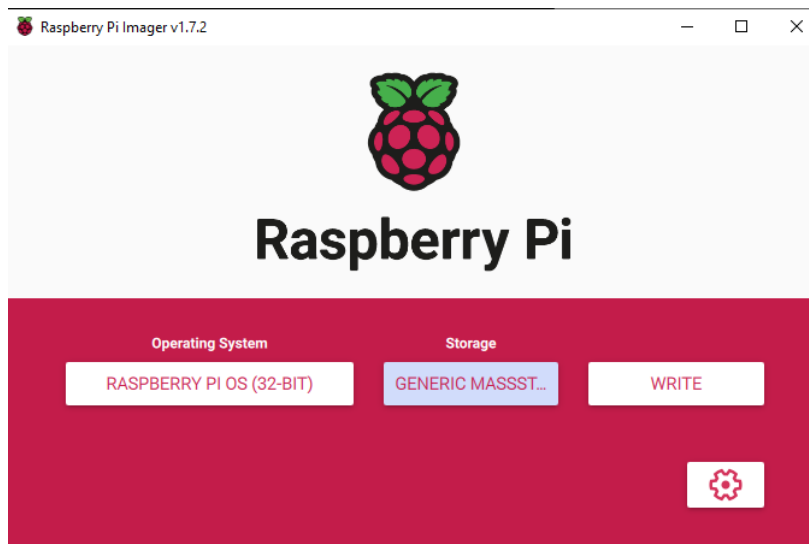
**Figura 26** - *Raspberry Pi Imager*. [F12].

Foi necessário estarmos conectados à internet na primeira vez para que o *Raspberry Pi Imager* baixe o SO que escolher. Esse SO será então armazenado para uso *offline* futuro. Estar online para usos posteriores significa que o *Raspberry Pi Imager* sempre lhe dará a versão mais recente.



**Figura 27** - Escolhendo o sistema a instalar, neste caso será o *Raspberry pi OS (32bit)*. [F12].

Por fim clicar em *Write* para instalar o sistema.



**Figura 28** - Instalando o sistema operacional. [F12].

2º) Após instalar o sistema, tem de iniciar sessão no seu Raspberry Pi via ssh utilizando o nome de utilizador e a palavra-passe que definiu durante o processo de instalação.

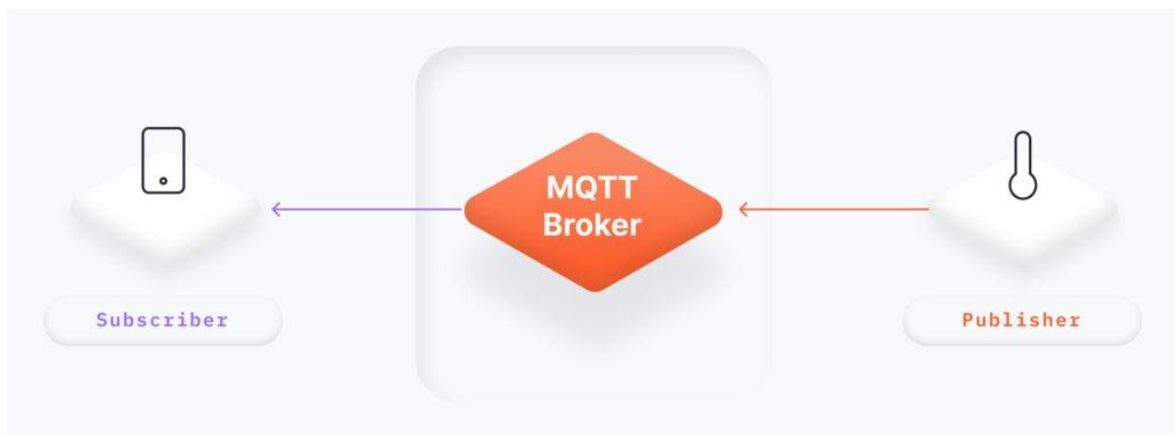
4) Abra a linha de comando do computador ssh pi@" ip do rasp". Prima enter e será pedido o seu nome de utilizador a seguir prima Enter novamente.

5) Em seguida, introduza a sua palavra-passe e prima Enter. Não verá quaisquer caracteres que aparece na janela enquanto digita a palavra-passe

### 3.4.2. MQTT (mosquitto broker)

A instalação do *mosquitto broker* é necessário de modo que a comunicação seja viabilizada a partir do protocolo *MQTT*. *MQTT* (Transporte de telemetria de fila de mensagens), desenvolvido pela *IBM*. Este protocolo é simples e leve, usado para enviar e receber mensagens pela internet e é projetado para dispositivos que têm baixa largura de banda. Atualmente, esse protocolo é frequentemente usado nos dispositivos *IoT* para enviar e receber os dados dos sensores. [14].

Abaixo temos uma ilustração de como funciona o *Mosquitto MQTT Broker*. O Mosquitto atua como o servidor central numa arquitetura de comunicação do tipo 'publicador/subscritor' (*publish/subscribe*), onde os dispositivos podem publicar mensagens ou subscrever-se a tópicos para receber dados.



**Figura 29** - Cliente *MQTT*: Smartphone (assinante) -> *MQTT broker* -> *MQTT client*: Sensor de temperatura. [F14]

### 3.4.3. Instalação do Broker MQTT Mosquitto no Raspberry PI

Para procederem à instalação do *Broker no Raspberry PI* devem realizar os seguintes passos.

1º) Aceder ao *Raspberry PI* via *SSH* e fazer *update e upgrade* ao sistema para que todos os repositórios sejam atualizados. Para isso vamos executar os seguintes comandos:

```
sudo apt update
```

```
sudo apt upgrade
```

O comando ***sudo apt update*** atualiza a lista de pacotes disponíveis no sistema, enquanto o ***sudo apt upgrade*** instala as atualizações para os pacotes já instalados, garantindo que o sistema está atualizado com as últimas correções e melhorias de software.

*sudo apt update && sudo apt upgrade*

2º) Quando solicitado, prima **Y** e **Enter**. Demorará algum tempo a atualizar (neste caso, demorou cerca de 5 minutos, mas pode demorar mais tempo). Após as atualizações vamos à instalação do *broker Mosquitto no Raspberry Pi*. Para isso usamos o seguinte comando:

*sudo apt install -y mosquitto mosquitto-clients*

O comando *sudo apt install -y mosquitto mosquitto-clients* instala o *Mosquitto Broker* (para gerir a comunicação via *MQTT*) e os clientes *MQTT* (para publicar e subscrever mensagens) no sistema. A opção *sudo* concede privilégios administrativos e o *-y* automatiza o processo de instalação sem pedir confirmação.

3º) Para ativarem o serviço, executamos o seguinte comando:

*sudo systemctl enable mosquitto.service*

O comando *sudo systemctl enable mosquitto.service* configura o *Mosquitto Broker* para iniciar automaticamente durante o arranque do sistema, garantindo que o serviço estará sempre em execução após um reinício.

4º) Agora, teste a instalação executando o seguinte comando:

*mosquitto -v.*

Isto retorna a versão do *Mosquitto* que está atualmente a correr no seu *Raspberry Pi*. Será 2.0.11 ou superior.

Atenção: Após seguir os passos, você verá uma janela idêntica à que está na imagem abaixo.

```
Setting up libdlt2:armhf (2.18.6-1) ...
Setting up libwebsockets16:armhf (4.0.20-2) ...
Setting up mosquitto (2.0.11-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/mosquitto.service -
/lib/systemd/system/mosquitto.service.
Processing triggers for man-db (2.9.4-2) ...
Processing triggers for libc-bin (2.31-13+rpt2+rpil+deb11u2) ...
pi@raspberrypi:~ $ sudo systemctl enable mosquitto.service
Synchronizing state of mosquitto.service with SysV service script with /lib/syst
emd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mosquitto
pi@raspberrypi:~ $ mosquitto -v
1639226740: mosquitto version 2.0.11 starting
1639226740: Using default config.
1639226740: Starting in local only mode. Connections will only be possible from
clients running on this machine.
1639226740: Create a configuration file which defines a listener to allow remote
access.
1639226740: For more details see https://mosquitto.org/documentation/authenticat
ion-methods/
1639226740: Opening ipv4 listen socket on port 1883.
1639226740: Error: Address already in use
1639226740: Opening ipv6 listen socket on port 1883.
1639226740: Error: Address already in use
```

**Figura 30** - Instalação do *mosquitto mqtt broker* no *Raspberry pi 4*.

### 3.4.4. Ativar o acesso remoto/autenticação

Para ativar o acesso remoto de modo a podermos comunicar com outros dispositivos IoT, temos de editar/criar um ficheiro de configuração. Pode editar o ficheiro de configuração com uma das seguintes opções:

- Autenticação com utilizador e palavra-passe.

Vamos adicionar autenticação com utilizador e palavra-passe.

1º) Execute o seguinte comando, mas substitua YOUR\_USERNAME pelo nome de utilizador que pretende utilizar:

ATT: Utilizamos o utilizador MQTT “iot”,

```
sudo mosquitto_passwd -c /etc/mosquitto/passwd iot
```

Quando executar o comando anterior com o nome de utilizador pretendido, ser-lhe-á pedido que introduza uma palavra-passe. Não serão apresentados quaisquer caracteres enquanto introduz a palavra-passe. Introduza a palavra-passe e memorize a combinação utilizador/palavra-passe, nós usamos a palavra-passe “iot”. Este comando anterior cria um ficheiro de palavras-passe chamado password no diretório /etc/mosquitto. Agora, precisamos editar o arquivo de configuração do mosquitto para que ele permita apenas com o nome de utilizador e a palavra-passe que definimos.

2º) Execute o seguinte comando para editar o ficheiro de configuração:

```
sudo nano /etc/mosquitto/mosquitto.conf
```

3) Adicione a seguinte linha no topo do ficheiro (certifique-se de que está no topo do ficheiro), caso contrário não funciona)

```
per_listener_settings true
```

4) Adicione as três linhas seguintes para permitir a ligação de utilizadores autenticados e diga ao Mosquitto onde se encontra o ficheiro de nome de utilizador/palavra-passe

- ❖ *allow\_anonymous false*
- ❖ *listener 1883*
- ❖ *ficheiro\_senha /etc/mosquitto/passwd*

O seu ficheiro de configuração terá o seguinte aspeto (as novas linhas estão a negrito):

```

GNU nano 5.4 /etc/mosquitto/mosquitto.conf *
# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

per_listener_settings true
pid_file /run/mosquitto/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d
allow_anonymous false
listener 1883
password_file /etc/mosquitto/passwd

^G Help      ^O Write Out  ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File  ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line

```

**Figura 31** - Configuração do mosquitto mqtt broker no Raspberry pi 4.

- 7) Prima CTRL-X, depois Y e, por fim, prima Enter para sair e guardar as alterações.
- 8) Reinicie o Mosquitto para que as alterações tenham efeito.

**`sudo systemctl restart mosquitto`**

Aguarde alguns segundos. Em seguida, para verificar se o *Mosquitto* está a funcionar, pode digitar o seguinte comando:

**`sudo systemctl status mosquitto`**

Agora, tem a autenticação com nome de utilizador e palavra-passe ativada.

```

pi@raspberrypi:~$ sudo mosquitto_passwd -c /etc/mosquitto/passwd sara
Password:
Reenter password:
pi@raspberrypi:~$ sudo nano /etc/mosquitto/mosquitto.conf
pi@raspberrypi:~$ sudo systemctl restart mosquitto
pi@raspberrypi:~$ sudo systemctl status mosquitto
● mosquitto.service - Mosquitto MQTT Broker
   Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor pre>
   Active: active (running) since Mon 2022-08-29 18:22:55 WEST; 27s ago
     Docs: man:mosquitto.conf(5)
           man:mosquitto(8)
   Process: 13791 ExecStartPre=/bin/mkdir -m 740 -p /var/log/mosquitto (code=e>
   Process: 13792 ExecStartPre=/bin/chown mosquitto /var/log/mosquitto (code=e>
   Process: 13793 ExecStartPre=/bin/mkdir -m 740 -p /run/mosquitto (code=exite>
   Process: 13794 ExecStartPre=/bin/chown mosquitto /run/mosquitto (code=exite>
   Main PID: 13795 (mosquitto)
     Tasks: 1 (limit: 4164)
           CPU: 81ms
   CGroup: /system.slice/mosquitto.service
           └─13795 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf

Aug 29 18:22:55 raspberrypi systemd[1]: Starting Mosquitto MQTT Broker...
Aug 29 18:22:55 raspberrypi systemd[1]: Started Mosquitto MQTT Broker.
lines 1-17/17 (END)

```

**Figura 32** - Instalação do mosquitto mqtt broker no Raspberry pi com autenticação.

### 3.4.5. Modelo Publicação/Assinatura e Tópicos em MQTT

O protocolo *MQTT* opera no modelo de publicação/assinatura (*publish/subscribe*), onde os dispositivos podem publicar mensagens num tópico e os interessados em receber essas mensagens podem inscrever-se nesse tópico. Quando uma mensagem é publicada, todos os dispositivos inscritos nesse tópico recebem a mensagem. Isso permite que os dispositivos se comuniquem de forma assíncrona, sem a necessidade de conexões diretas entre eles. O *MQTT* também suporta diferentes níveis de qualidade de serviço (*QoS*), que determinam o grau de confiabilidade na entrega das mensagens.

O modelo de publicação/assinatura separa o cliente que publica a mensagem do cliente que a solicita. Os clientes não precisam comunicar diretamente, e o modelo permite que vários clientes publiquem ou recebam informações simultaneamente. As informações são enviadas por meio de tópicos, que podem ser entendidos como endereços para os quais uma mensagem é direcionada. O cliente que publica cria este endereço, e o cliente que se inscreve requisita as informações. Um assinante pode inscrever-se em vários tópicos, enquanto um cliente que publica pode criar diversos tópicos.

Isso é possível graças a um servidor intermediário conhecido como *broker*, que organiza e classifica os dados ao colocá-los em tópicos correspondentes. Por exemplo, o tópico *pressao\_ar\_data*, publicado por um sensor que mede a pressão atmosférica, conterá apenas dados relativos à pressão do ar. Por outro lado, o tópico *humidade\_data*, publicado por um sensor de umidade, incluirá apenas dados de umidade. Os dados serão requisitados por um cliente *subscriber* (como uma aplicação ou *dashboard*) que se inscreveu em ambos os tópicos.

### 3.4.6. Instalação do Node-RED

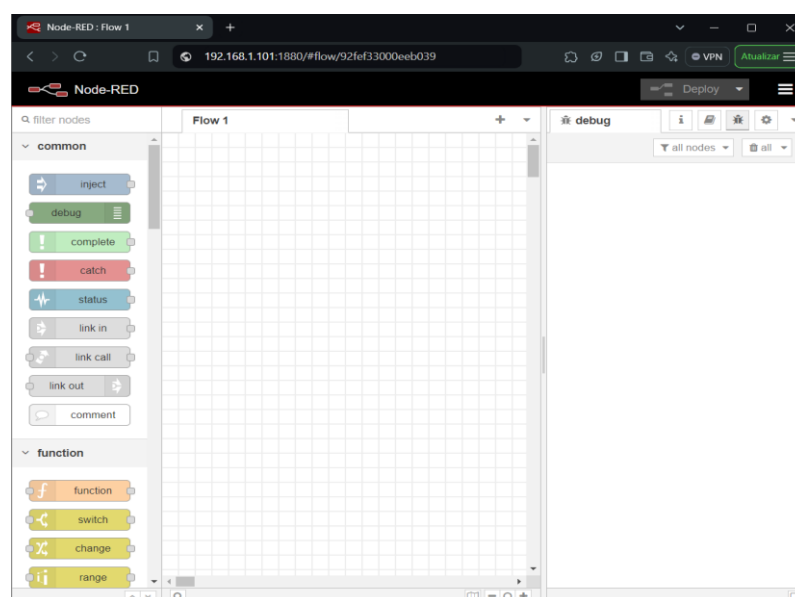


Figura 33 - Node-red. [F15].

### 3.4.6.1. Instalar o Node-RED

Tendo uma ligação SSH estabelecida com o seu *Raspberry Pi*, siga os próximos passos para instalar o *Node-RED*.

Em primeiro lugar, actualize e faça o upgrade do seu sistema com o seguinte comando:

```
sudo apt update && sudo apt upgrade
```

A certa altura, ser-lhe-á perguntado se pretende continuar. Clique em Y e depois em Enter. Este processo pode demorar alguns minutos.

Quando estiver concluído, introduza o seguinte comando para instalar o *Node-RED*:

- ❖ `bash < (curl -sL`
- ❖ `https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered [16].`

Depois de executar o comando, ser-lhe-á perguntado: "Deseja instalar nós específicos do Pi?" Prima Y e *Enter*. Levará alguns minutos para instalar o *Node-RED*. No final, deverá aparecer uma mensagem semelhante na janela do Terminal (ver página seguinte).

```

Running Node-RED install for user mypt at /home/mypt on debian

This can take 20-30 minutes on the slower Pi versions - please wait.

Stop Node-RED                               ✓
Remove old version of Node-RED               ✓
Node option not specified                    : --node18 or --node20
Leave existing Node.js                        : v20.17.0 Npm 10.8.2
Clean npm cache                              -
Install Node-RED core                        ✓ 4.0.2
Move global nodes to local                   -
Leave existing nodes                          -
Install extra Pi nodes                       ✓
Add shortcut commands                        ✓
Update systemd script                        ✓

Any errors will be logged to /var/log/nodered-install.log
All done.
You can now start Node-RED with the command node-red-start
or using the icon under Menu / Programming / Node-RED
Then point your browser to localhost:1880 or http://{your_pi_ip-address}:1880

Started : Tue 27 Aug 22:28:59 WEST 2024
Finished: Tue 27 Aug 22:43:26 WEST 2024

Just FYI : Your settings.js file is different from the latest defaults.
You may wish to run
diff -y --suppress-common-lines /usr/lib/node_modules/node-red/settings.js /home/mypt/.node-red/settings.js
to compare them and see what the latest options are.

*****
### WARNING ###
DO NOT EXPOSE NODE-RED TO THE OPEN INTERNET WITHOUT SECURING IT FIRST

Even if your Node-RED doesn't have anything valuable, (automated) attacks will
happen and could provide a foothold in your local network

Follow the guide at https://nodered.org/docs/user-guide/runtime/securing-node-red
to setup security.

### ADDITIONAL RECOMMENDATIONS ###
- Remove the /etc/sudoers.d/010_pi-nopasswd file to require entering your password
when performing any sudo/root commands:

sudo rm -f /etc/sudoers.d/010_pi-nopasswd

*****

```

**Figura 34** - Instalação do *node-red* no *Raspberry pi*.

### 3.4.6.2. Configurar as definições do Node-RED

Ser-lhe-á perguntado se pretende personalizar as definições agora. Prima **Y** e, em seguida, prima **Enter** (em algumas versões, este passo é omitido e passa-se para a captura de ecrã na página seguinte).

```
*****
**
Would you like to customise the settings now (y/N) ?
pi@raspberrypi:~ $ █
```

Durante as configurações, se receber um erro, execute o seguinte comando para começar a personalizar as definições:

#### ***node-red admin init***

```
Any errors will be logged to /var/log/nodered-install.log
All done.
You can now start Node-RED with the command node-red-start
or using the icon under Menu / Programming / Node-RED
Then point your browser to localhost:1880 or http://{your_pi_ip-address}:1880

Started : Wed 27 Apr 11:43:13 BST 2022
Finished: Wed 27 Apr 11:47:27 BST 2022

You may want to run node-red admin init
to configure your initial options and settings.

pi@raspberrypi:~ $ node-red admin init

Node-RED Settings File initialisation
=====
This tool will help you create a Node-RED settings file.

? Settings file █ /home/pi/.node-red/settings.js
```

**Figura 35** - Resolvendo alguns erros.

### 3.4.6.3. Iniciar o Node-RED

Execute o seguinte comando para iniciar o *Node-RED*: `node-red`

Deverá aparecer uma mensagem semelhante na janela do Terminal:

```

Settings file written to /home/pi/.node-red/settings.js
pi@raspberrypi:~ $ node-red-start

Start Node-RED

Once Node-RED has started, point a browser at http://192.168.1.106:1880
On Pi Node-RED works better with the Firefox or Chrome browser

Use node-red-stop to stop Node-RED
Use node-red-start to start Node-RED again
Use node-red-log to view the recent log output
Use sudo systemctl enable nodered.service to autostart Node-RED at every boot
Use sudo systemctl disable nodered.service to disable autostart on boot

To find more nodes and example flows - go to http://flows.nodered.org

Starting as a systemd service.
27 Nov 11:23:45 - [info]
Welcome to Node-RED
=====
27 Nov 11:23:45 - [info] Node-RED version: v3.0.2
27 Nov 11:23:45 - [info] Node.js version: v16.18.1
27 Nov 11:23:45 - [info] Linux 5.15.61-v8+ arm64 LE
27 Nov 11:23:46 - [info] Loading palette nodes
27 Nov 11:23:49 - [info] Settings file : /home/pi/.node-red/settings.js
27 Nov 11:23:49 - [info] Context store : 'default' [module=memory]
27 Nov 11:23:49 - [info] User directory : /home/pi/.node-red
27 Nov 11:23:49 - [warn] Projects disabled : editorTheme.projects.enabled=false
27 Nov 11:23:49 - [info] Flows file : /home/pi/.node-red/flows.json
27 Nov 11:23:49 - [info] Creating new flow file
27 Nov 11:23:49 - [warn] Encrypted credentials not found
27 Nov 11:23:49 - [info] Server now running at http://127.0.0.1:1880/
27 Nov 11:23:49 - [info] Starting flows
27 Nov 11:23:49 - [info] Started flows

```

**Figura 36** - Inicialização do node-red.

Arranque automático do *Node-RED* no arranque Para executar automaticamente o *Node-RED* quando a Pi arranca, é necessário introduzir o seguinte comando (prima CTRL + C para parar a tarefa anterior antes de introduzir o novo comando).

***sudo systemctl enable nodered.service***

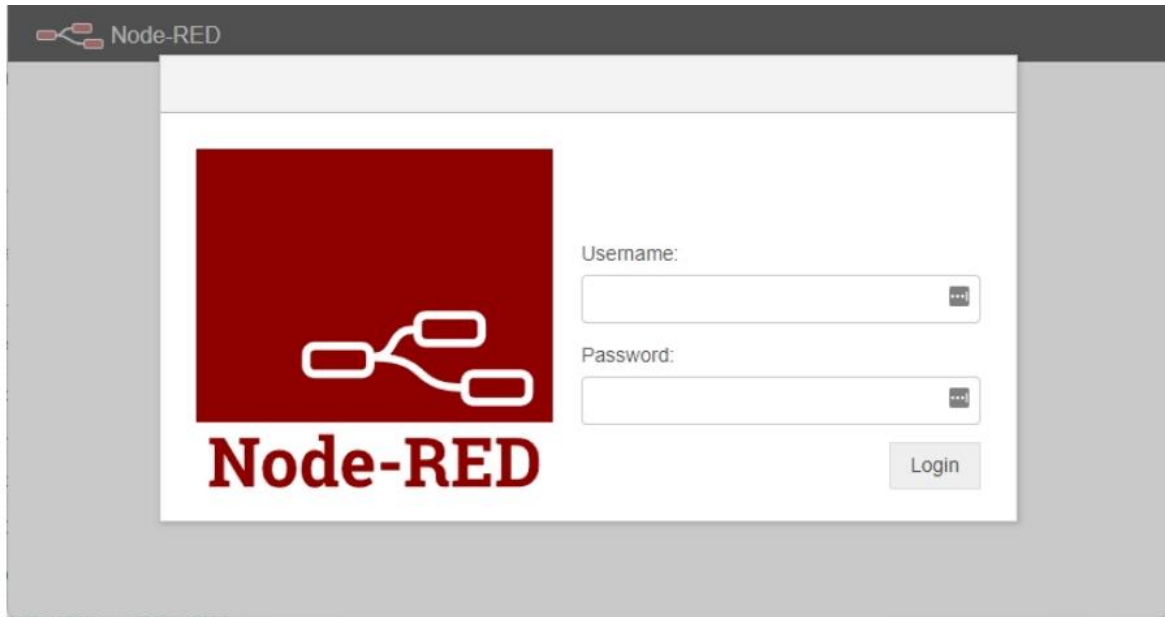
Isto significa que enquanto o seu *Raspberry Pi* estiver ligado, o *Node-RED* estará a funcionar. Agora, reinicia a tua Pi para que o arranque automático tenha efeito. Da próxima vez que a *Raspberry Pi* reiniciar, o *Node-RED* já estará a funcionar.

***sudo reboot***

#### **3.4.6.4. Nó de acesso node-RED**

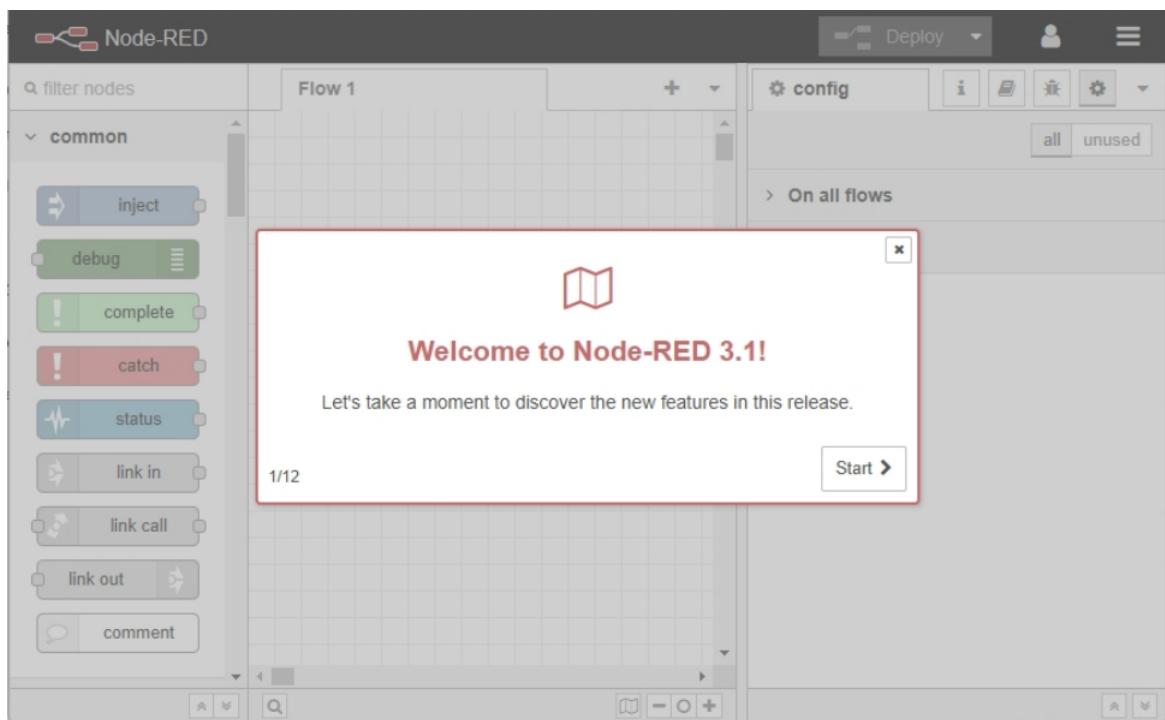
O *Node-RED* é executado na porta 1880. Para aceder ao *Node-RED*, abra um browser no seu computador e escreva o endereço IP do Raspberry Pi seguido de :1880. Por exemplo, no nosso caso 10.6.64.159:1880

Depois de introduzir o endereço IP do Raspberry Pi seguido de :1880 no navegador de Internet, a página de início de sessão do *Node-RED* deve ser carregada. Insira o seu nome de utilizador e palavra-passe do *Node-RED* que definiu ao configurar as definições do *Node-RED*.



**Figura 37** - Nó de acesso *node-red*

Agora, tem acesso ao *Node-RED*. Pode começar a construir os fluxos.



**Figura 38** - Página *Node-red*.

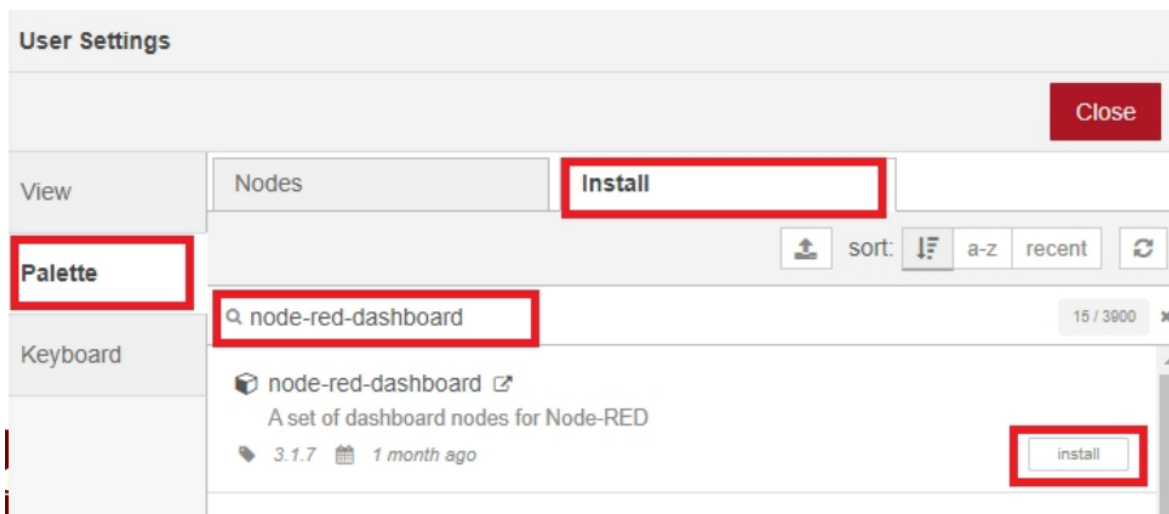
#### 3.4.6.5. Painel de controlo do *Node-RED*

Esta unidade é uma introdução ao painel de controlo *node-red* com *Raspberry Pi*. Cobriremos como instalar o *Node-RED Dashboard* e exemplificaremos como construir uma interface gráfica de utilizador para os seus projecto de *IoT* e de Domótica.

O *Node-RED Dashboard* é um módulo que fornece um conjunto de nós no *Node-RED* para criar rapidamente um painel de dados em direto. Por exemplo, fornece nós para criar rapidamente uma interface de utilizador com botões, barras deslizantes, gráficos, medidores, etc.

### 3.4.6.6. Instalando o painel de controlo do Node-RED

Pode instalar os nós do painel de controlo do *Node-RED* utilizando o Menu > Gerir paleta. Em seguida, procure o *node-red-dashboard* e instale-o.



**Figura 39** - Instalando a *node-red-dashboard*.

### Figura 38 - Instalando a *node-red-dashboard*.

Após a instalação, os nós do painel de controlo aparecerão na paleta.

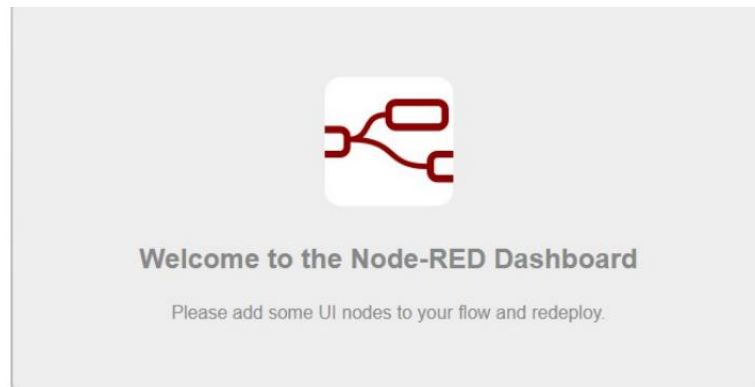


**Figura 40** - Painel de controlo dashboard.

Os nos do *Dashboard* fornecem *widgets* que aparecem na interface de utilizador (*IU*) da aplicação. A interface de utilizador está acessível no seguinte URL:

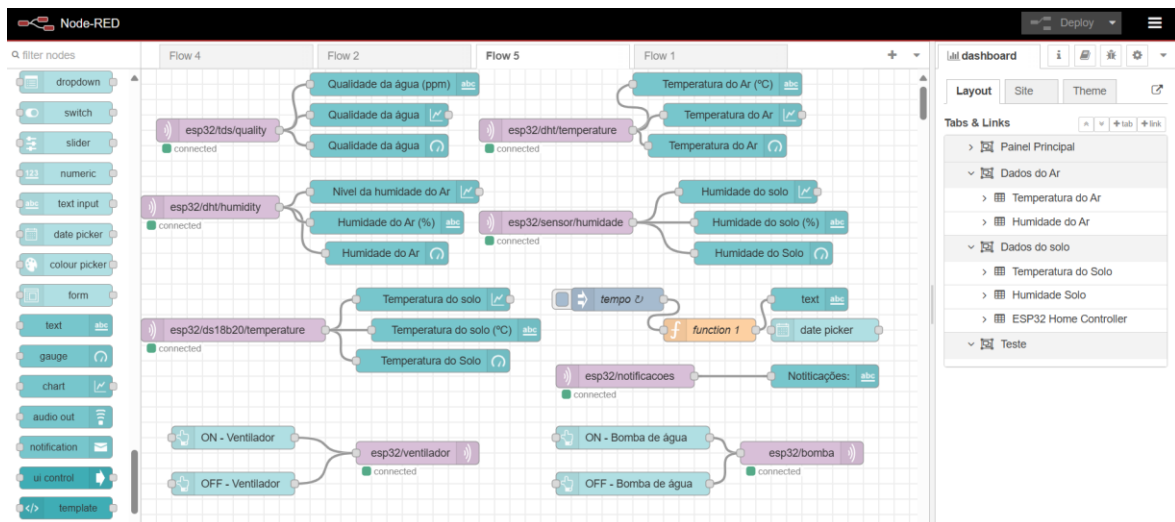
[http://Seu\\_RPi\\_IP\\_address:1880/ui](http://Seu_RPi_IP_address:1880/ui)

Neste caso: [http:// 10.6.64.159:1880/ui](http://10.6.64.159:1880/ui)



**Figura 41** - dashboard node-red.

Apos a instalação do node-red criamos os fluxos para o funcionamento da dashboard



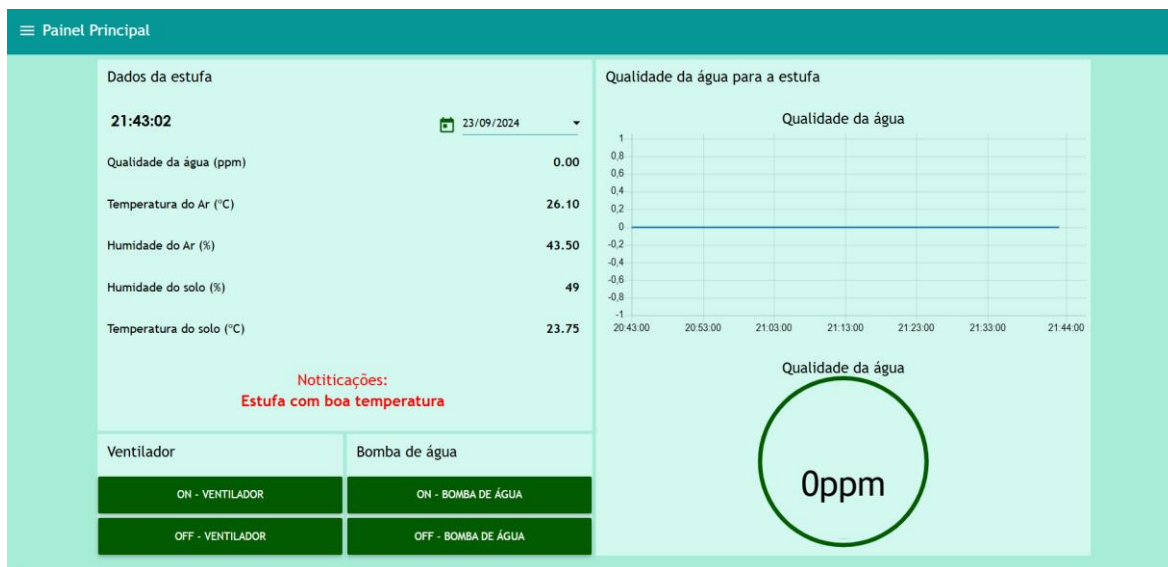
**Figura 42** - Fluxos criados no node-red para sistema de monitorização de estufas

### 3.4.6.7. Resultado da dashboard

Apos a instalação do *node-red* criamos a nossa *dashboard*. A *dashboard* é constituída por três abas que são:

#### Aba 1: Painel principal

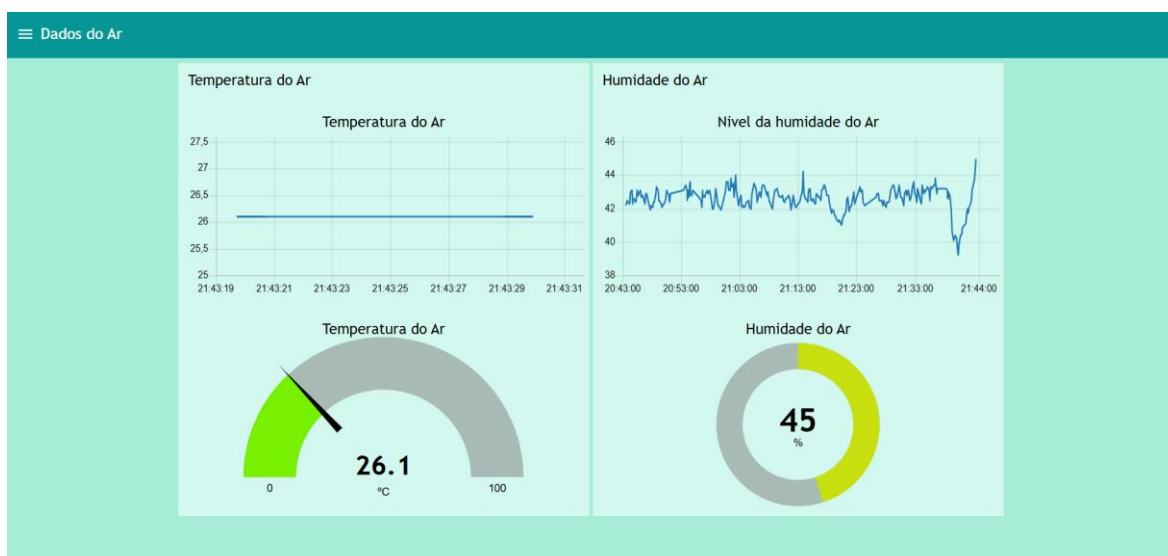
Uma que denominamos Painel principal, nesta aba encontramos os dados da estufa, ou seja, as condições climáticas da estufa como a temperatura, a humidade do ar, temperatura, a humidade do solo, a qualidade da água que regamos a estufa em forma de gráficos, e também os comandos manuais que nos permitem a atuar na estufa como ligar o ventilador e a bomba de água.



**Figura 43** - Aba painel principal da *dashboard*

### 3.4.6.8. Aba 2: Dados do ar

Esta aba permiti-nos visualizar as condições do ar na estufa em forma de gráficos, a temperatura do ar e a sua humidade



**Figura 44** - Aba 2 dados do ar da *dashboard*

### 3.4.6.9. Aba 3: Dados do solo

Esta aba permiti-nos visualizar as condições do ar na estufa em forma de gráficos, a temperatura do ar e a sua humidade

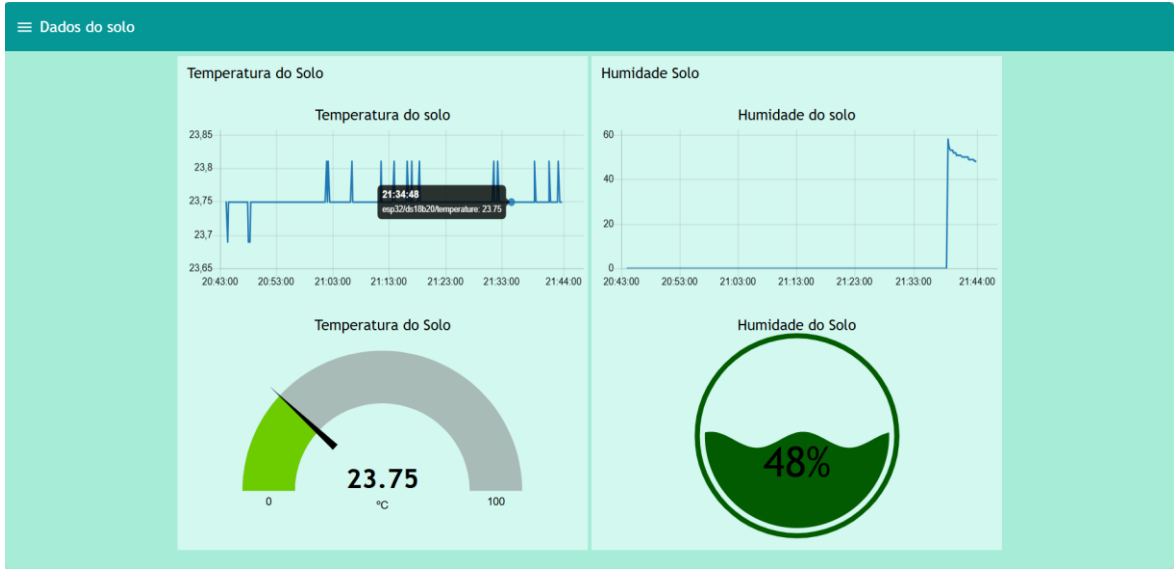
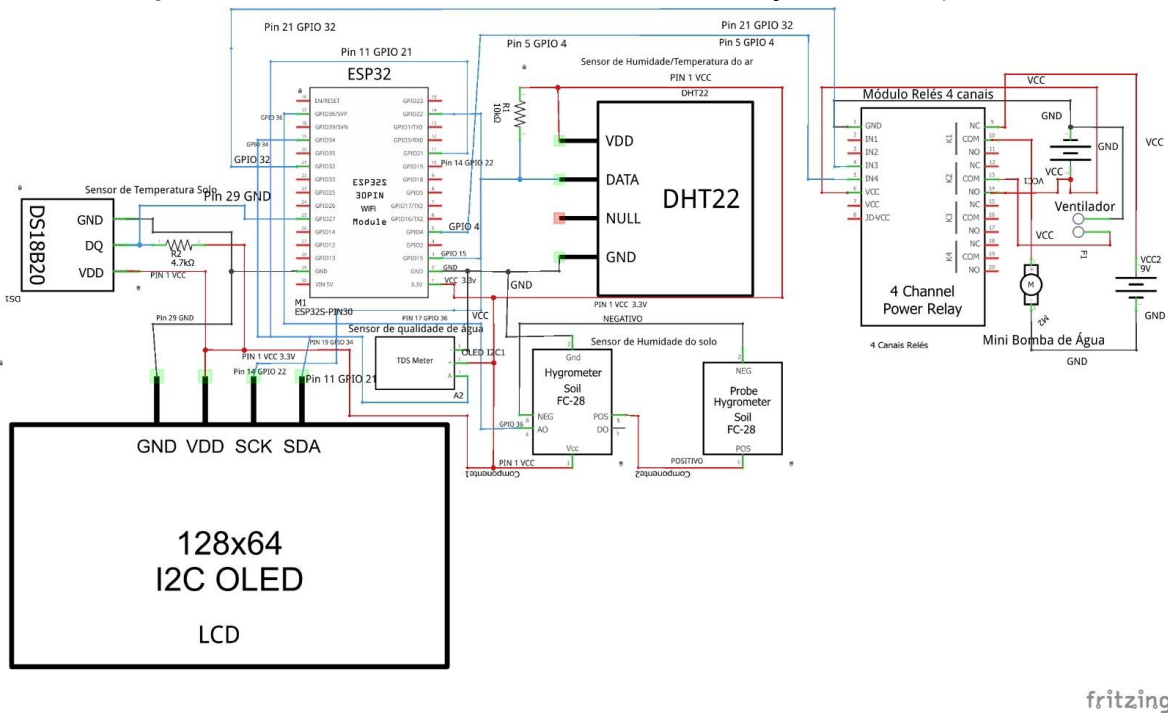


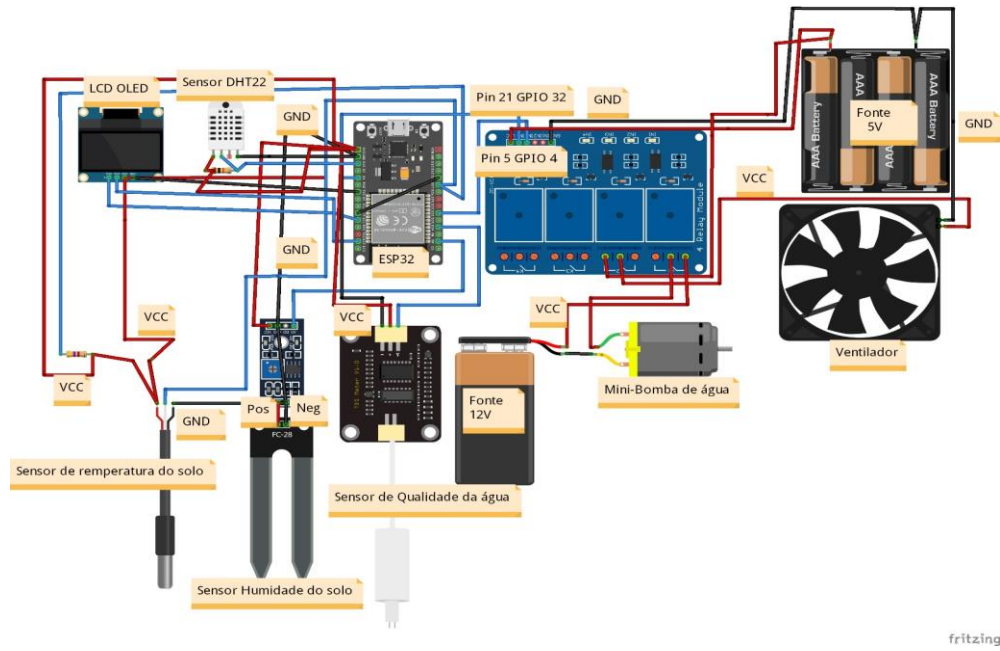
Figura 45 - Aba 3 dados do solo da dashboard

### 3.4.7. Esquema elétrico do sistema de monitorização de estufas



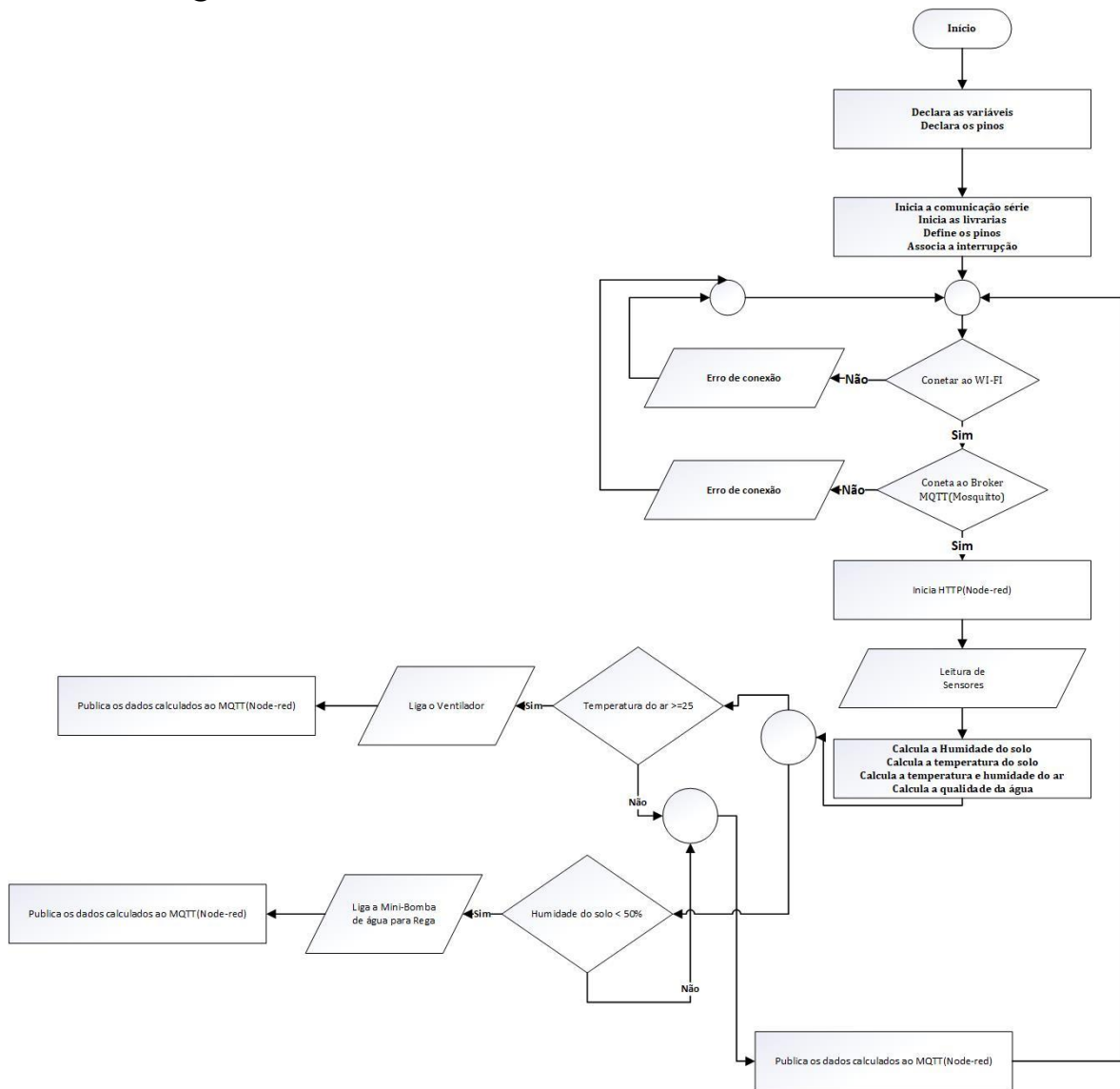
fritzing

Figura 46- Esquema elétrico lógico do sistema de monitorização de estufas



**Figura 47** - Esquema elétrico físico de componentes do sistema de monitorização de estufas

### 3.5. Fluxograma



**Figura 48 - Fluxograma do sistema de monitorização de estufas**

## 4. Capítulo 4 - Testes e resultados

### 4.1. Cenários de Teste e Resultados

Neste capítulo é apresentado todo o material utilizado no desenvolvimento do projeto, na **tabela 8** é referimos o preço dos materiais, onde foram e as suas quantidades. Também serão apresentadas as experiências realizados para medir a humidade e a temperatura do sistema proposto, bem como os resultados obtidos, com o intuito de avaliar o desempenho e as características do sistema implementado.

E também apresentamos as imagens da estufa montada, foi nesta estufa que implementamos o sistema de monitorização de estufas que desenvolvemos.

Quant.	Componente	Preço (€)	Distribuidor
1	ESP32	2.51€	<i>Aliexpress</i>
1	Módulo Detector Sensor de Umidade do Solo HD-32	2,15€	<i>Aliexpress</i>
1	Sensor Humidade/ Temperatura Ar DHT22	5€	<i>Aliexpress</i>
1	Sensor de Temperatura DS18B20	2.5€	<i>Aliexpress</i>
1	Sensor de Qualidade de águaTSD Meter	1.7€	<i>Aliexpress</i>
1	Ventilador	11,78€	<i>Aliexpress</i>
1	Mini-Bomba de água	4.15€	<i>Aliexpress</i>
1	Estufa	44€	Leroy Merlin
2	<i>Breadboard</i>	10.12€	<i>Aliexpress</i>
1	<i>Raspberry pi 4</i>	50€	Worten
2	Resistências (10kΩ, 4,7kΩ)	2€	<i>Aliexpress</i>
1	LCD "I2C OLED"	1,64€	<i>Aliexpress</i>
1	Módulo Relés 4 canais	3,05€	<i>Aliexpress</i>
1	Terra para o cultivo	8,99€	Leroy Merlin
	<b>Total</b>	<b>149,59€</b>	

**Tabela 9** - Lista dos materiais

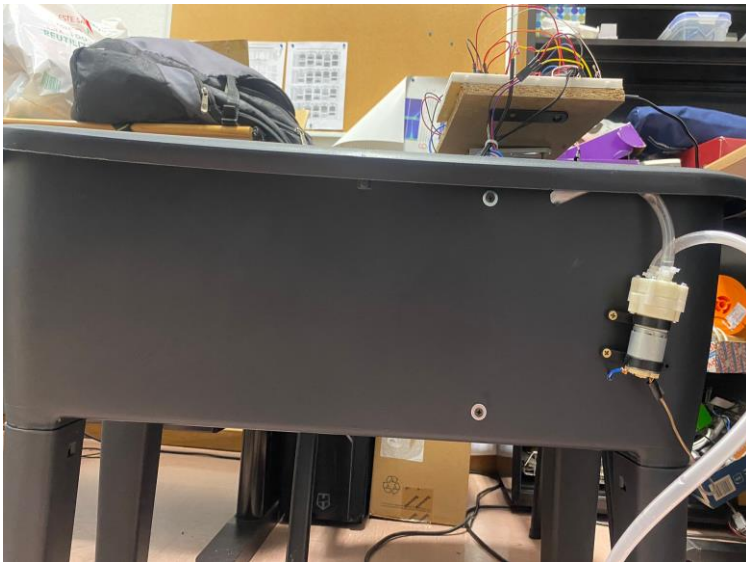
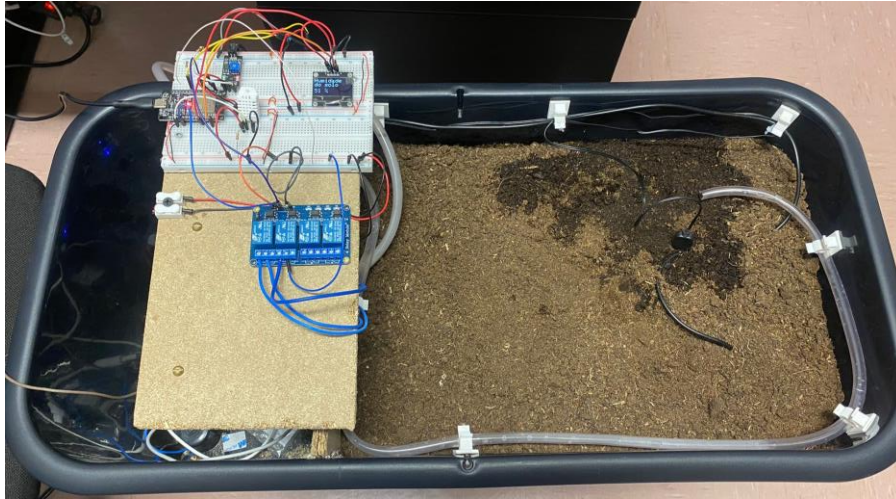


Figura 49- Estufa

## 4.2. Cenários

### Cenário 1 - Testes do sensor de qualidade de água (*Tds*)

Neste cenário de teste usamos o sensor *Tds*, usamos dois recipientes com água onde o primeiro pote colocamos a água da fonte (esta no pote branco) e o segundo pote colocamos água com detergente (esta no pote transparente).



**Figura 50** - Potes com água para experiência

O objetivo desta experiência foi calibrar o sensor TDS e visualizar os dados da água no monitor serial. Os resultados esperados eram que a calibração fosse precisa tanto para a água proveniente da fonte como para a água com outras substâncias adicionadas.

**Primeira etapa da calibração:** Na primeira etapa, compilámos o código no VSC para o ESP32, de modo a visualizar o valor de referência da qualidade da água no pote branco, exibido no monitor serial, com o sensor totalmente submerso na água.

ATT: Segundo informações tiradas no *chatgpt*:

### **Entendimento do TDS (Sólidos Totais Dissolvidos)**

O sensor TDS mede a concentração de sólidos dissolvidos na água, como sais minerais, metais, e outras substâncias. A medição é dada em **ppm** (partes por milhão).

- **Água potável** geralmente tem um valor de TDS abaixo de 500 ppm.
- Valores de **TDS elevados** podem indicar impurezas ou contaminação, enquanto valores muito **baixos** podem significar que a água carece de minerais importantes.

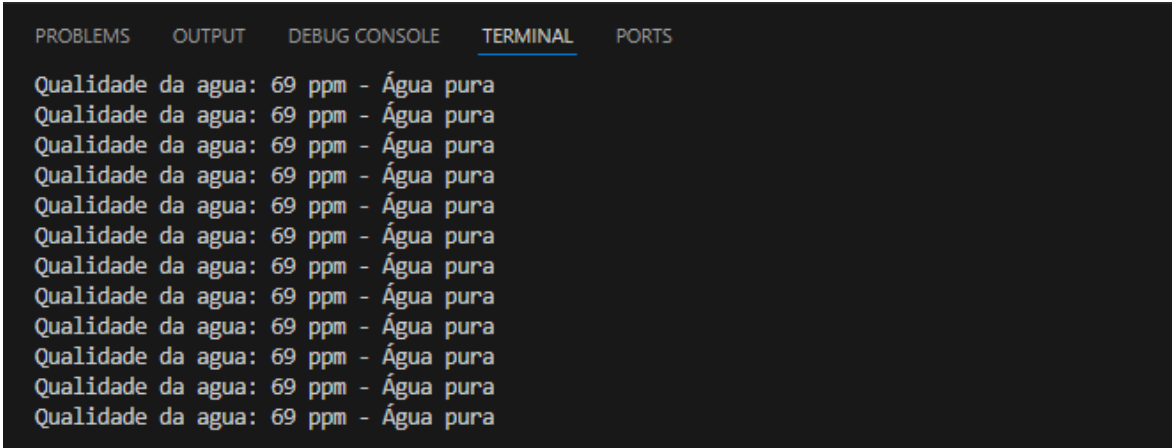
## Referências de Qualidade da Água com Base no TDS

Aqui está uma tabela comum de classificação da água com base nos valores de TDS.

**Tabela 10-** Referências de qualidade de água

Faixa de TDS (ppm)	Qualidade da Água
0 - 50	Água ultrapura (normalmente usada em laboratórios)
50 - 150	Água pura (de boa qualidade)
150 - 300	Água de qualidade aceitável
300 - 500	Água potável, mas com minerais dissolvidos elevados
500 - 1000	Água considerada menos adequada para consumo (pode ser salobra)
1000+	Água inadequada para beber (geralmente água salina ou contaminada)

Logo apos a investigação dos TDS conseguimos verificar que o sensor dá-nos valores aceitáveis 69 ppm, de boa qualidade.



```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS
Qualidade da agua: 69 ppm - Água pura
Qualidade da agua: 69 ppm - Água pura
Qualidade da agua: 69 ppm - Água pura
Qualidade da agua: 69 ppm - Água pura
Qualidade da agua: 69 ppm - Água pura
Qualidade da agua: 69 ppm - Água pura
Qualidade da agua: 69 ppm - Água pura
Qualidade da agua: 69 ppm - Água pura
Qualidade da agua: 69 ppm - Água pura
Qualidade da agua: 69 ppm - Água pura
Qualidade da agua: 69 ppm - Água pura
Qualidade da agua: 69 ppm - Água pura
Qualidade da agua: 69 ppm - Água pura

```

**Figura 51** - Resultados da água do pote branco

**Segunda etapa da calibração:** Nesta fase, analisámos a água com detergente que estava no pote transparente. A água apresentou 477 ppm, sendo considerada potável, mas com uma elevada concentração de minerais dissolvidos.



**Figura 52-** Água usada  
Potes de água com detergente

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
Qualidade da água: 477 ppm - Potável, mas com muitos minerais dissolvidos
Qualidade da água: 477 ppm - Potável, mas com muitos minerais dissolvidos
Qualidade da água: 477 ppm - Potável, mas com muitos minerais dissolvidos
Qualidade da água: 477 ppm - Potável, mas com muitos minerais dissolvidos
Qualidade da água: 477 ppm - Potável, mas com muitos minerais dissolvidos
Qualidade da água: 477 ppm - Potável, mas com muitos minerais dissolvidos
Qualidade da água: 479 ppm - Potável, mas com muitos minerais dissolvidos
Qualidade da água: 479 ppm - Potável, mas com muitos minerais dissolvidos
Qualidade da água: 478 ppm - Potável, mas com muitos minerais dissolvidos
Qualidade da água: 478 ppm - Potável, mas com muitos minerais dissolvidos
Qualidade da água: 477 ppm - Potável, mas com muitos minerais dissolvidos
Qualidade da água: 477 ppm - Potável, mas com muitos minerais dissolvidos
```

**Figura 53** -Resultados da água do pote branco

## Cenário 2 - Testes do sensor de humidade do solo

Neste teste, foi utilizado o sensor de humidade do solo (HD-38), recorrendo a um pote com água e uma esponja para introduzir o sensor, assim como terra na estufa. A experiência foi realizada em duas etapas: a primeira com o sensor totalmente submerso em água, dentro da esponja, e a segunda com o sensor colocado no solo seco dentro da estufa.

O objetivo desta experiência foi de realizar a calibração do sensor, verificar se os dados de humidade eram diferentes ao nível de sonda introduzida em água e no solo e visualizar os dados de humidade no monitor serial.

Os resultados esperados eram que a calibração fosse válida tanto para água como para o solo e que os dados coletados fossem coerentes.

• **Primeira etapa da calibração:** Compilar o código para o ESP32, de modo a visualizar o valor de referência da humidade exibido no monitor serial, com o sensor colocado no solo seco da estufa.



**Figura 54** - Sensor de humidade do solo em solo seco

O valor da humidade no solo posto em solo seco foi de 14% apresentados no monitor serial

```
Humidade do solo = 14%  
Humidade do solo = 14%  
Humidade do solo = 14%  
Humidade do solo = 14%  
Humidade do solo = 14%  
Humidade do solo = 14%  
Humidade do solo = 14%  
Humidade do solo = 14%  
Humidade do solo = 14%  
Humidade do solo = 14%  
Humidade do solo = 14%  
Humidade do solo = 14%  
Humidade do solo = 14%  
Humidade do solo = 14%  
Humidade do solo = 14%
```

**Figura 55** - Valor da humidade no solo apresentado no monitor serial

**Segunda etapa da calibração:** Nesta fase, foi realizada a experiência para obter o valor de referência da humidade com o sensor totalmente submerso em água, dentro da esponja.



**Figura 56** - Sensor de humidade do solo emergido em água

O valor da humidade no solo está o sensor emergido em água estava nos 70% apresentados no monitor serial

```
Humidade do solo = 71%  
Humidade do solo = 76%  
Humidade do solo = 74%  
Humidade do solo = 70%
```

**Figura 57** - Valor da humidade na água apresentado no monitor serial

Na imagem abaixo é possível ver que os dados percentuais da humidade do solo que estão sendo publicados na *dashboard* do *node-red*. Os dados são publicados por meio do tópico *esp32/sensor/humidade*.

ATT: Os processos de publicação dos tópicos foram explicados no capítulo 3 na parte do MQTT (*broker mosquitto*)



**Figura 58** - Valor da humidade do solo na dashboard

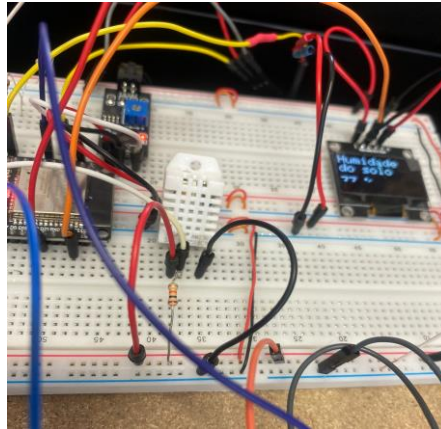
### **Cenário 3 - Testes do sensor de temperatura e humidade do ar**

Para este teste usamos o sensor DHT22 que nos fornece os valores de temperatura e humidade do ar dentro da estufa. Fizemos o teste duas etapas, que são: o sensor dando os valores reais da tempera e da humidade sem qualquer alteração e o sensor sendo aquecido.

O objetivo desta experiência foi de realizar a calibração do sensor, verificar se os dados ambientais sem qualquer alteração eram diferentes dos dados com o sensor aquecido.

Os resultados esperados eram que a calibração fosse válida para ambas as situações e que os dados coletados fossem coerentes.

- **Primeira etapa da calibração:** Conectar o sensor ao ESP32 e compilar o código, de modo a visualizar os valores de humidade e temperatura sem qualquer intervenção no sensor, tanto no monitor serial como na dashboard.



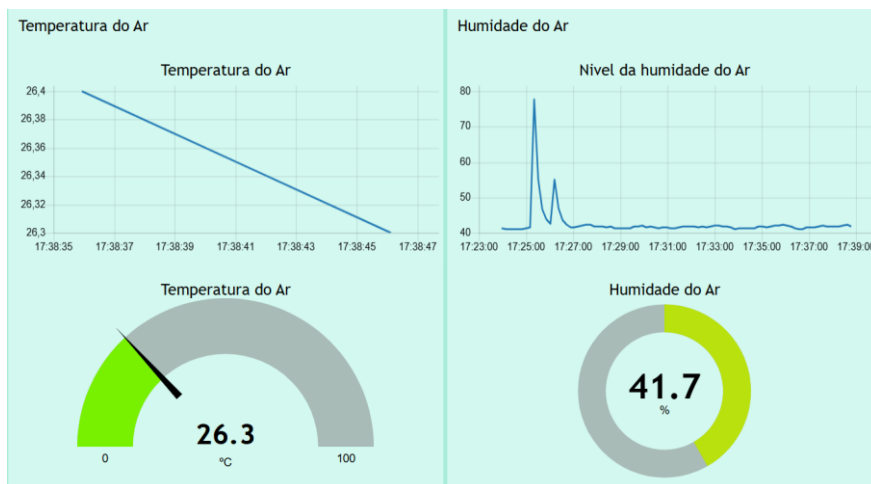
**Figura 59** - Sensor DHT22 sem qualquer alteração

Os valores da humidade e temperatura do ar na estufa, são: temperatura 26.10 graus e da humidade de 41.60 % apresentados no monitor serial

```

Temperatura do ar = 26.10
Humidade do ar = 41.60
Temperatura do ar = 26.10
Humidade do ar = 41.60
    
```

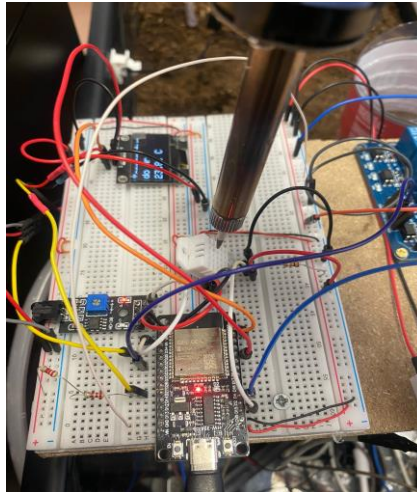
**Figura 60** - Valor da humidade e temperatura do ar apresentados no monitor serial



**Figura 61** - Valor da humidade e temperatura do ar apresentados na *dashboard*

- Segunda etapa da calibração: Visualizar os valores de humidade e temperatura ao aquecer o sensor, tanto no monitor serial como na *dashboard*.

Aquecemos o sensor com um ferro de soldar



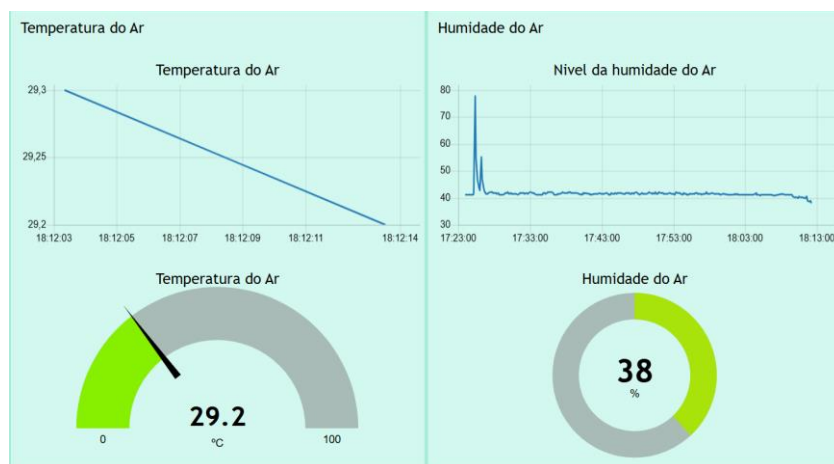
**Figura 62** - Sensor DHT22 sendo aquecido

Os valores da humidade e temperatura do ar sendo o sensor aquecido, são: temperatura 26.10 graus e da humidade de 41.60 % apresentados no monitor serial. Os valores tiveram um aumento com relação aos resultados da etapa 1

```

Temperatura do ar = 29.30
Humidade do ar = 38.80
Temperatura do ar = 29.30
Humidade do ar = 38.80
  
```

**Figura 63** - Valor no monitor serial da humidade e temperatura do ar sendo o sensor aquecido



**Figura 64** - Valor na dashboard da humidade e temperatura do ar sendo o sensor aquecido

#### Cenário 4 - Testes de irrigação do solo na estufa

Esta experiência foi realizada na estufa e foi analisado mediante duas possibilidades, solo seco e solo húmido.

Para este teste usamos os seguintes equipamentos: o sensor de humidade do solo, o rele, o esp32 e a mini-bomba de água.

O sensor de humidade foi utilizado para fazer a captação da humidade no solo para transmiti-las ao microcontrolador e por sua vez a *dashboard*, e simultaneamente havia uma condição sendo constantemente verificada pelo microcontrolador, que foi implementada em código C++ para que os atuadores fossem acionados caso o valor de humidade lido pelo sensor de humidade fosse menor que 50%, nesse momento o relé iria ser acionado e iria comutar de normalmente aberto para normalmente fechado, ao comutar o relé iria acionar a bomba de água para realizar a irrigação.

O objetivo do teste é realizar uma irrigação automatizada na estufa e o seu monitoramento. A saída esperada deveria ser uma leitura de humidade do solo maior que 50% de humidade em terra húmida, valores abaixo 20% de humidade em terra seca e a irrigação deveria acontecer quando o sensor estivesse imerso em terra seca, por satisfazer a condição de humidade  $< 50\%$ , que foi pré-estabelecida no código utilizado no projeto e os dados de temperatura e humidade lidos pelos sensores deveriam ser exibidos constantemente no *Dashboard*, independentemente da condição estabelecida.

**Primeira etapa:** Nesta etapa vamos analisar o sensor no solo húmido. Obedecendo a condição, se humidade no solo  $\geq 50\%$  presente os valores e da informação de estufa regada. Visualizar no monitor serial e na *dashboard*.



**Figura 65** - Sensor de humidade do solo no solo húmido

O valor da humidade do solo foi de 58% no solo húmido apresentados no monitor serial.

```
Humidade do solo = 58
Humidade do solo ideal, estufa regada
Humidade do solo = 58
Humidade do solo ideal, estufa regada
```

**Figura 66** - Valor no monitor serial da humidade do solo húmido



**Figura 67** - Valor na *dashboard* da humidade do solo húmido

**Segunda etapa:** Nesta etapa vamos analisar o sensor no solo seco. Obedecendo a condição, se humidade no solo  $\leq$  a 50%, a bomba será ligada para regar a estufa. Apresentando os valores e da informação de que a estufa esta sendo regada. Visualizar no monitor serial e na *dashboard*.



**Figura 68** - Sensor de humidade do solo no solo seco

O valor da humidade do solo foi de 0 % no solo seco apresentados no monitor serial.

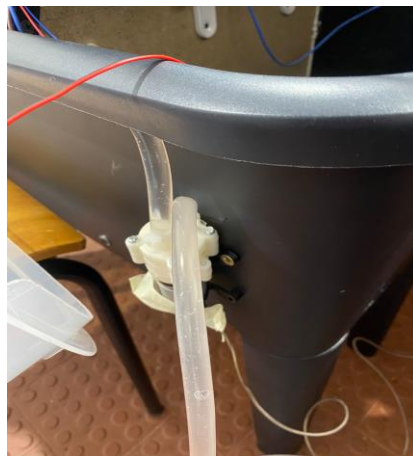
```
Humidade do solo = 0  
Humidade do solo baixa.Bomba ligada regando a estufa
```

**Figura 69** - Valor no monitor serial da humidade do solo seco

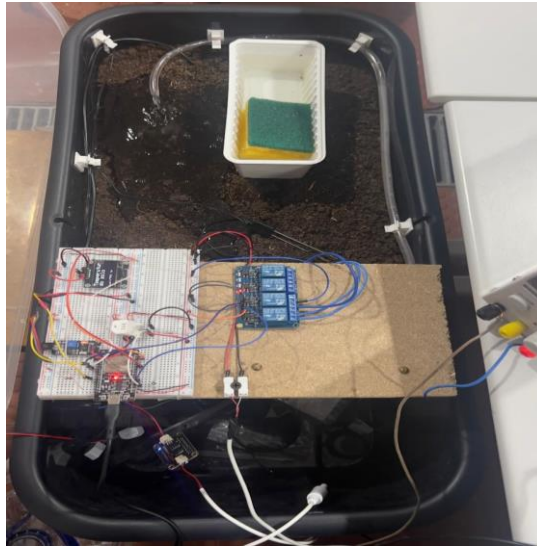


**Figura 70** - Valor na dashboard da humidade do solo seco

Apos a verificação o relé é acionado de seguida a bomba de água é ligada, logo o sistema de irrigação é ativado



**Figura 71** - Bomba de água ligada



**Figura 72 - Estufa sendo regada**

#### **Cenário 4 - Testes do sistema de ventilação na estufa**

Para este teste usamos os seguintes equipamentos: o sensor *DHT22*, o rele, o *ESP32* e o ventilador.

O sensor de *DHT22* foi utilizado para fazer a captação da temperatura do ar na estufa para transmiti-la ao microcontrolador e por sua vez a *dashboard*.

Para que este sistema funcione tem que ser obedecido uma condição sendo constantemente verificada pelo microcontrolador, que foi implementada em código C++ para que os atuadores fossem acionados caso a temperatura for menor que 20 graus o relé seria acionado e iria comutar de normalmente aberto para normalmente fechado, ao comutar o relé iria acionar o ventilador.

O objetivo do teste é demonstrar o sistema automático de ventilação da estufa que é ativado quando a temperatura do ar for maior que 20 graus.

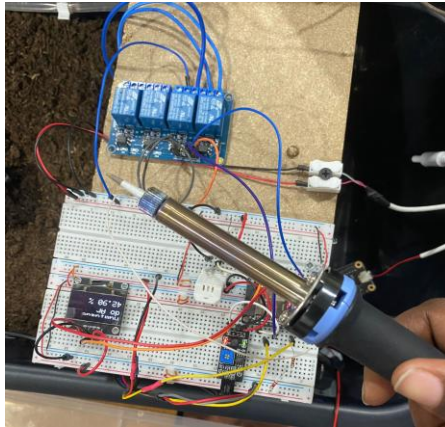
Quando realizamos este teste, obtivemos a temperatura de 17 graus, apresentados no monitor serial.

```
Temperatura do ar = 17 graus  
Temperatura ideal, ventilador desligado  
Temperatura do ar = 17 graus  
Temperatura ideal, ventilador desligado
```

**Figura 73 - Valor no monitor serial da temperatura no momento do teste**

#### **Realização do teste aquecendo o sensor de temperatura**

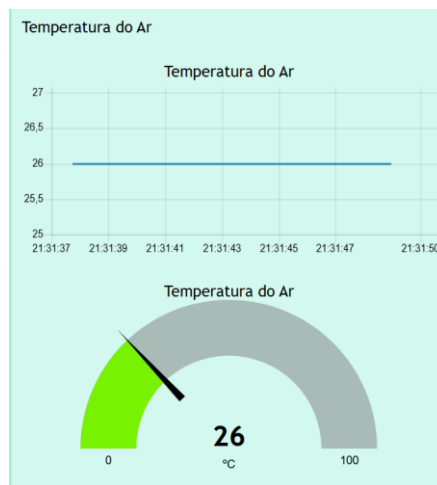
Aquecendo o sensor para aumentar a temperatura do ar e ativar o sistema de ventilação, verificando a condição de ligar o ventilador quando a temperatura for maior que 20 graus



**Figura 74** - Aquecendo o *DHT22* para ligar o ventilador

```
Temperatura do ar = 26.00  
Temperatura alta, ventilador ligado  
Temperatura do ar = 26.00  
Temperatura alta, ventilador ligado
```

**Figura 75** - Valor no monitor serial da temperatura apos o aquecimento



**Figura 76** - Valor na *dashboard* da temperatura apos o aquecimento

Apos aquecer o sensor de temperatura e aumentar a temperatura para 26 graus superior a temperatura da condiçao que é 20 graus. Logo a condiçao é verificada o relé é acionado de seguida o ventilador é ligado.



**Figura 77 - Ventilador ligado**

## 5. Conclusão

O desenvolvimento do Sistema de Monitorização de Estufas apresentado neste projeto evidenciou o potencial das tecnologias IoT em revolucionar a agricultura, proporcionando um controlo mais rigoroso e eficiente das condições ambientais. A combinação do microcontrolador ESP32 com o Raspberry Pi permitiu a criação de uma solução robusta que monitora, em tempo real, parâmetros críticos como a humidade e temperatura do solo, a qualidade da água e as condições atmosféricas, utilizando sensores avançados como o DHT22. A inclusão de um ecrã LCD facilita a visualização imediata dos dados, permitindo um controlo local eficiente, enquanto a integração com a plataforma Node-RED proporciona uma gestão centralizada e acessível remotamente.

A monitorização constante das condições ambientais é essencial para garantir que as plantas recebam a quantidade adequada de água e nutrientes. Os sensores de humidade e temperatura do solo, juntamente com o sensor de qualidade da água, asseguram que as plantas estejam em condições ideais para um crescimento saudável. Além disso, a capacidade do sistema de processar e armazenar dados em tempo real permite ajustes automáticos com base em alterações ambientais, melhorando a eficiência no uso de recursos naturais, especialmente a água, que é vital para uma agricultura sustentável. Este sistema não apenas otimiza a produtividade das culturas, mas também contribui para a conservação dos recursos hídricos.

Por fim, a flexibilidade do sistema permite a sua expansão através da adição de novos sensores ou módulos IoT, o que proporciona uma base sólida para futuras inovações. A interface intuitiva do Node-RED, juntamente com a monitorização em tempo real e a possibilidade de controlo remoto, torna esta solução não só prática, mas também adaptável às necessidades específicas de agricultores e gestores agrícolas. Em suma, este projeto representa um passo significativo em direção à modernização da agricultura, promovendo a eficiência, a sustentabilidade e a resposta aos desafios contemporâneos que o setor enfrenta.

## 6. Trabalhos futuros

Futuros trabalhos no sistema de monitorização de estufas podem focar na expansão para múltiplas culturas, com ajustes personalizados de irrigação, temperatura, luminosidade e humidade do solo para cada planta. A utilização de algoritmos de aprendizado de máquina permitiria otimizar automaticamente as condições de cultivo para diferentes culturas, promovendo o crescimento eficiente. Sensores mais avançados, como para monitoramento de nutrientes, pH e gases no solo, aumentariam a precisão do sistema, fornecendo dados em tempo real para ajustes imediatos, garantindo melhores condições para as plantas.

Além disso, a implementação de tecnologias de conectividade como *LoRa* ou *NB-IoT* possibilitaria o gerenciamento remoto de estufas em áreas amplas, expandindo o alcance do sistema e permitindo automação inteligente em locais isolados. A análise de dados em nuvem, aliada à inteligência artificial, permitiria a previsão de necessidades de irrigação, a deteção precoce de pragas e doenças, e o planejamento de intervenções proativas. Através de *dashboards* preditivos, os agricultores poderiam acompanhar o crescimento das plantas e otimizar o uso de recursos como água e energia.

Outra vertente importante seria o aumento da sustentabilidade, com a introdução de painéis solares e baterias, tornando o sistema autossustentável em termos energéticos, especialmente em áreas rurais com pouca infraestrutura. Estudos sobre o impacto ambiental, comparando métodos tradicionais com monitorização inteligente, e a análise da viabilidade econômica para diferentes tamanhos de produção agrícola, ajudariam a demonstrar o retorno sobre o investimento. Melhorias na interface de usuário, como controle remoto via aplicativos móveis, também são essenciais para tornar o sistema mais intuitivo e acessível a diversos perfis de usuários, maximizando a eficiência e o controle das estufas.

## 7. Referências bibliográficas

- [1] Victor Vision. “ESP32: Conheça a placa para IoT com Wi-Fi e Bluetooth.” Consultado em 15 de julho de 2024. <https://victorvision.com.br/blog/placa-esp32/>.
- [2] UsinaInfo. “ESP32S NodeMCU IoT com Wi-Fi e Bluetooth 38 Pinos com USB-C.” Consultado em 25 de agosto de 2024. [https://www.usinainfo.com.br/esp32/esp32s-nodemcu-iot-com-wifi-e-bluetooth-38-pinos-com-usb-c-8796.html?search\\_query=esp32&results=187](https://www.usinainfo.com.br/esp32/esp32s-nodemcu-iot-com-wifi-e-bluetooth-38-pinos-com-usb-c-8796.html?search_query=esp32&results=187).
- [3] Eletrogate. “Conhecendo o ESP32 - Introdução.” Consultado em 5 de setembro de 2024. <https://blog.eletrogate.com/conhecendo-o-esp32-introducao-1/>.
- [4] Z. Zhou et al., “Exploring the IoT-based Monitoring System in Smart Agriculture,” IEEE Internet of Things Journal, 2017. Consultado em 12 de julho de 2024. <https://ieeexplore.ieee.org/document/8101926/citations#citations>.
- [5] UsinaInfo. “Diferença entre ESP32 e ESP8266.” Consultado em 20 de julho de 2024. <https://www.usinainfo.com.br/esp32-611#:~:text=A%20diferen%C3%A7a%20do%20ESP32%20x%20ESP8266%20%C3%A9%20que,irm%C3%A3o%2C%20possuindo%20total%20compatibilidade%20do%20a%20IDE%20Arduino>.
- [6] UsinaInfo. “Sensor de Umidade do Solo Arduino HD-38.” Consultado em 15 de agosto de 2024. <https://www.usinainfo.com.br/sensor-de-solo/sensor-de-umidade-do-solo-arduino-hd-38-5475.html>.
- [7] ElectroFun. “Sensor de Umidade e Temperatura DHT22.” Consultado em 25 de agosto de 2024. <https://www.electrofun.pt/sensores-arduino/sensor-humidade-temperatura-dht22>.
- [8] Mauser.pt. “Sensor de Umidade do Solo V1.2.” Consultado em 10 de setembro de 2024. [https://mauser.pt/catalog/product\\_info.php?products\\_id=096-8756](https://mauser.pt/catalog/product_info.php?products_id=096-8756).
- [9] Mauser.pt. “Sensor de Temperatura LM35DZ.” Consultado em 20 de setembro de 2024. [https://mauser.pt/catalog/product\\_info.php?products\\_id=095-0745](https://mauser.pt/catalog/product_info.php?products_id=095-0745).
- [10] UsinaInfo. “Módulo Relé 5V 10A 4 Canais com Optoacopladores.” Consultado em 5 de setembro de 2024. [https://www.usinainfo.com.br/rele-arduino/modulo-rele-5v-10a-4-cana-is-com-optoacopladores-2534.html?search\\_query=rele+de+4+modulos&results=90](https://www.usinainfo.com.br/rele-arduino/modulo-rele-5v-10a-4-cana-is-com-optoacopladores-2534.html?search_query=rele+de+4+modulos&results=90).
- [11] Raspberry Pi. “Raspberry Pi 4 Model B (8GB).” Consultado em 15 de setembro de 2024. <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/?variant=raspberry-pi-4-model-b-8gb>.
- [12] Raspberry Pi Foundation. “Setting Up Your Raspberry Pi.” Consultado em 20 de setembro de 2024. <https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up/1>.
- [13] Sapo.pt. “Aprenda a Instalar o MQTT Broker (Mosquitto) no Seu Raspberry Pi.” Consultado em 5 de setembro de 2024. <https://sapo.pt/>.
- [14] Amen Technologies. “Instalando e Testando o Corretor Mosquitto MQTT no Raspberry Pi para Comunicação IoT.” Consultado em 10 de julho de 2024. <https://amen-technologies.com/>.

- [15] Node-RED. "Node-RED: Running on Raspberry Pi." Consultado em 20 de setembro de 2024. <https://nodered.org/>.
- [16] Wikipédia. "Distrito de Castelo Branco." Consultado em 10 de setembro de 2024. [https://pt.wikipedia.org/wiki/Distrito de Castelo Branco](https://pt.wikipedia.org/wiki/Distrito_de_Castelo_Branco).
- [17] Florestas.pt. "Estudo do Solo: Pedologia e Cartas do Solo em Portugal." Consultado em 15 de setembro de 2024. <https://florestas.pt/>.
- [18] Direção-Geral do Território. "V7\_NQ\_Folheto\_A4\_COS\_16Junho2020." Consultado em 20 de setembro de 2024. <https://www.dgterritorio.gov.pt/>.
- [19] Tipos de solo em Portugal. Consultado em 5 de setembro de 2024. <https://www.agroportal.pt/os-solos-em-portugal-conhecer-para-beneficiar/>.
- [20] Siro Germinação. "Siro Germinação." Consultado em 10 de setembro de 2024. <https://portal.siro.pt/pt/Store/Product?ref=SR10550000116&codeCat=>.
- [21] Raspberry Pi em: 60+ Raspberry Pi Projects, Tutorials and Guides | Random Nerd Tutorials. Consultado em 15 de setembro de 2024.
- [22] Display OLED SSD1306: Consultado em 1 de setembro de 2024. <https://www.kitsarduino.com.br/cmp/oled.html>.
- [23] Consultado em 10 de setembro de 2024. <https://blog.formacao.dev/visual-studio-code-vscode-instalacao-e-configuracao/>.
- [24] Consultado em 5 de setembro de 2024. [https://recipp.ipp.pt/bitstream/10400.22/24216/1/Tese\\_4780\\_v2.pdf](https://recipp.ipp.pt/bitstream/10400.22/24216/1/Tese_4780_v2.pdf).
- [25] Consultado em 20 de setembro de 2024. <https://masterplanti.com.br/beneficios-do-cultivo-prottegido-ou-em-estufas/>.
- [26] Consultado em 10 de setembro de 2024. <https://blog.sensix.ag/internet-das-coisas-iot-a-tecnologia-a-favor-de-uma-agricultura-inteligente/>.
- [27] Consultado em 15 de setembro de 2024. <https://pt.rfstariot.com/blog/generic-serial-communication-protocols-uart-spi-i2c-b64>.

---

## FONTES

## IMAGENS

- [F1] UsinaInfo. "ESP32S NodeMCU IoT com Wi-Fi e Bluetooth 38 Pinos com USB-C." Consultado em 15 de julho de 2024. [https://www.usinainfo.com.br/esp32/esp32s-nodemcu-iot-com-wifi-e-bluetooth-38-pinos-com-usb-c-8796.html?search\\_query=esp32&results=187](https://www.usinainfo.com.br/esp32/esp32s-nodemcu-iot-com-wifi-e-bluetooth-38-pinos-com-usb-c-8796.html?search_query=esp32&results=187).
- [F2] Embarcados. "ESP32: Lidando com Multiprocessamento - Parte I." Consultado em 22 de agosto de 2024. <https://embarcados.com.br/esp32-lidando-com-multiprocessamento-parte-i/>.
- [F3] Eletrogate. "Conhecendo o ESP32 - Introdução." Consultado em 10 de setembro de 2024. <https://blog.eletrogate.com/conhecendo-o-esp32-introducao-1/>.
- [F4] CircuitState. "DOIT ESP32 DevKit V1: Pinout Diagram and Reference." Consultado em 5 de setembro de 2024. <https://www.circuitstate.com/pinouts/doit-esp32-devkit-v1-wifi-development-board-pinout-diagram-and-reference/>.
- [F5] ElectroFun. "Sensor de Umidade e Temperatura DHT22." Consultado em 18 de

setembro de 2024. <https://www.electrofun.pt/sensores-arduino/sensor-humidade-temperatura-dht22>.

[F6] UsinaInfo. "Sensor de Umidade do Solo Arduino HD-38." Consultado em 25 de agosto de 2024. <https://www.usinainfo.com.br/sensor-de-solo/sensor-de-umidade-do-solo-arduino-hd-38-5475.html>.

[F7] Mauser.pt. "Sensor de Umidade do Solo V1.2." Consultado em 12 de setembro de 2024. [https://mauser.pt/catalog/product\\_info.php?products\\_id=096-8756](https://mauser.pt/catalog/product_info.php?products_id=096-8756).

[F8] Mauser.pt. "Sensor de Temperatura LM35DZ." Consultado em 28 de setembro de 2024. [https://mauser.pt/catalog/product\\_info.php?products\\_id=095-0745](https://mauser.pt/catalog/product_info.php?products_id=095-0745).

[F9] UsinaInfo. "Módulo Relé 5V 10A 4 Canais com Optoacopladores." Consultado em 15 de setembro de 2024. [https://www.usinainfo.com.br/rele-arduino/modulo-rele-5v-10a-4-canais-com-optoacopladores-2534.html?search\\_query=rele+de+4+modulos&results=90](https://www.usinainfo.com.br/rele-arduino/modulo-rele-5v-10a-4-canais-com-optoacopladores-2534.html?search_query=rele+de+4+modulos&results=90).

[F10] Eletrogate. "Mini Bomba de Água Motor DC RS-385." Consultado em 2 de setembro de 2024. <https://www.eletrogate.com/mini-bomba-de-agua-motor-dc-rs-385>.

[F11] Raspberry Pi. "Raspberry Pi 4 Model B (8GB)." Consultado em 20 de setembro de 2024. <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/?variant=raspberry-pi-4-model-b-8gb>.

[F12] Raspberry Pi Foundation. "Setting Up Your Raspberry Pi." Consultado em 12 de setembro de 2024. <https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up/1>.

[F13] Maker Advisor. "Best Raspberry Pi 4 Starter Kits 2023 – Buying Guide." Consultado em 5 de setembro de 2024. <https://makeradvisor.com/>.

[F14] Cedalo. "MQTT Subscribe and Publish Example with Mosquitto." Consultado em 15 de setembro de 2024. <https://cedalo.com/blog/mqtt-subscribe-publish-mosquitto-pub-sub-example/>.

[F15] Node-RED. "Node-RED: Flow-based Development Tool." Consultado em 30 de setembro de 2024. <https://nodered.org/>.

[F16] Wikipédia. "Distrito de Castelo Branco – Wikipédia, a enciclopédia livre." Consultado em 10 de setembro de 2024. [https://pt.wikipedia.org/wiki/Distrito\\_de\\_Castelo\\_Branco](https://pt.wikipedia.org/wiki/Distrito_de_Castelo_Branco).

[F17] Tipos de solo. Consultado em 15 de setembro de 2024. <https://eos.com/pt/blog/tipos-de-solo/>.

[F18] Siro Germinação. "Siro Germinação." Consultado em 22 de setembro de 2024. <https://portal.siro.pt/pt/Store/Product?ref=SR10550000116&codeCat=>.

[F19] <https://evolt.pt/produto/ventilador-ventoinha-fan-12025-24v-2-fios-mec0252v1-000u-a99-12012025mm-sunon/>, acesso 15 de Setembro de 2024