



# Controlo de Sistema de Rega Baseada em Tecnologia IoT

**Adilson Mendes Silva**

Dissertação apresentada à Escola de Tecnologia e Gestão de Bragança para a obtenção  
do Mestrado em Engenharia Industrial.

Trabalho orientado por:  
José Augusto Almeida Pinheiro de Carvalho

Bragança

2022





# Controlo de Sistema de Rega Baseada em Tecnologia IoT

**Adilson Mendes Silva**

Dissertação apresentada à Escola de Tecnologia e Gestão de Bragança para a obtenção  
do Mestrado em Engenharia Industrial.

Trabalho orientado por:  
José Augusto Almeida Pinheiro de Carvalho

Bragança

2022



“Nesse mundo só os fortes sobrevivem.  
Eu sei que a batalha é dura, mas acredite  
quando eu digo que as coisas  
sempre melhoram.”  
*(Tupac Amaru Shakur).*



# Dedicatória

Dedico esta tese para os meus pais. O que sinto por eles é maior que qualquer palavra. É maior que qualquer texto. Eles são a razão de eu ser. De eu existir. De eu continuar a viver. Eles são tudo para mim!



# Agradecimentos

Agradeço imensamente a Deus, por ter-me concedido saúde, força e disposição para fazer esse trabalho, sem ele nada disso seria possível, também sou grato a ele por tranquilizar o meu espírito nos momentos mais difíceis da minha trajetória acadêmica até então.

Também sou muito grato a todos os professores que contribuíram na minha trajetória acadêmica especialmente ao meu orientador, José Augusto Almeida Pinheiro de Carvalho que contribuiu bastante na orientação do meu projeto. Manifesto aqui a minha gratidão eterna por compartilhar a sua sabedoria, o seu tempo e a sua experiência.

Também agradeço aos meus pais que me deram apoio e incentivo nas horas difíceis. Sou grato aos meus familiares e amigos que não deixaram-me ser vencido pelo cansaço.

Os meus agradecimentos aos meus irmãos, tia que de alguma forma também contribuíram para que tudo corresse bem.

Sem esquecer do Instituto Politécnico de Bragança por proporcionar-me um ambiente criativo e amigável para os estudos. Sou grato a cada membro do corpo docente, à direção e a administração dessa instituição de ensino.



# Resumo

A Internet das Coisas (IoT) é descrita na rede de objetos físicos, agregados com sensores, software e algumas outras tecnologias com a intenção de conectar e possibilitar a troca de dados com outros dispositivos e sistemas pela Internet. Com o passar do tempo apercebe-se que a utilização da Internet das Coisas aumenta de uma forma exponencial, visto que é possível conectar com objetos utilizados no dia a dia e também devido as suas inúmeras vantagens, como, por exemplo, melhoria da eficiência energética, redução do custo das instalações e facilidade na utilização entre outros. A IoT procura controlar e integrar ambientes em automações residenciais, áreas agrícolas e não só, visando obter conforto, segurança e economia das residências.

Este projeto consiste em desenvolver um sistema de monitoramento e controlo de uma dada área agrícola com uma dimensão consideravelmente grande e constituído com diferentes tipos de produtos agrícolas, baseado no sistema IoT. Para ser mais específico, o projeto foi direcionado ao uso de protocolo Pub/Sub MQTT, com o uso do Mosquitto *Broker* (uma combinação de um router para fornecer a rede e um Raspberry Pi) o *Broker* por sua vez possibilita que haja total comunicação entre os clientes dentro da mesma rede.

Os clientes utilizados na realização deste projeto foram os microcontroladores ESP32 e ESP8266, também criou-se uma interface (*dashboard*) no JavaScript para que o utilizador pudesse obter as leituras dos dados e controlar os atuadores. No desenvolvimento deste projeto foi montado um sistema que consegue controlar a iluminação *on/off* e *dimming*, controlar a temperatura e fazer a leitura dos dados de temperatura e humidade fornecidos pelo sensor de temperatura e humidade.

**Palavras-chaves:** Internet das Coisas, Protocolo MQTT, ESP32, ESP8266.

# Abstract

As is known, the internet of things (IoT) is described in the network of physical objects, which are aggregated with sensors, software and some other technologies with the intention of connecting and enabling the exchange of data with other devices and systems over the internet. Over time, it can be seen that the use of the Internet of Things is increasing exponentially, as it can connect to objects used in everyday life and also due to its numerous advantages, such as , improving energy efficiency, reducing the cost of installations and ease of use, among others. IoT seeks to control and integrate environments in residential automation, agricultural areas and beyond, in order to obtain comfort, safety and economy in homes.

This project consists of creating a monitoring and control system for a given agricultural area with a considerably large dimension and consisting of different types of agricultural products, based on the IoT system. To be more specific, the project was directed to the use of the Pub/Sub MQTT protocol, with the use of the Mosquitto Broker (a combination of a router to provide the network and a Raspberry Pi) the Broker in turn allows for total communication communication between clients within the same network. The clients that were used in this project were the ESP32 and ESP8266 microcontrollers, and an interface (dashboard) was also created in JavaScript so that the user could get the data readings and control the actuators. In the construction of this project, a system was built that is capable of controlling the on/off and dimming lighting, controlling the temperature and reading the temperature and humidity data provided by the temperature and humidity sensor.

**Keywords:** Internet of Things, MQTT Protocol, ESP32, ESP8266.

# Índice

<b>Dedicatória</b>	<b>vii</b>
<b>Agradecimentos</b>	<b>ix</b>
<b>Resumo</b>	<b>xi</b>
<b>Abstract</b>	<b>xiii</b>
<b>Acrónimos</b>	<b>xxiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	2
1.2 Objetivos . . . . .	3
1.3 Estrutura do documento . . . . .	3
<b>2 Revisão bibliográfica</b>	<b>5</b>
2.1 Tecnologias . . . . .	7
2.1.1 Identificação por radiofrequência (RFID) . . . . .	7
2.1.2 Sistema global para comunicação móvel (GSM) . . . . .	8
2.1.3 Bluetooth de baixa energia (BLE) . . . . .	9
2.1.4 X10 . . . . .	10
2.1.5 Insteon . . . . .	11
2.1.6 Wavenis . . . . .	12
2.1.7 ZigBee . . . . .	13

2.1.8	Z-Wave . . . . .	15
2.1.9	MQTT . . . . .	17
2.2	Ferramentas e <i>software</i> . . . . .	21
2.2.1	ESP8266 . . . . .	21
2.2.2	ESP32 . . . . .	23
2.2.3	Node-RED . . . . .	26
2.2.4	Arduino IDE . . . . .	27
2.2.5	<i>Dashboard</i> . . . . .	28
2.2.6	Raspberry Pi . . . . .	29
2.3	Eficiência energética . . . . .	30
2.4	Aplicações de Tecnologia IoT . . . . .	34
2.5	Definição do problema . . . . .	41
<b>3</b>	<b>Sistema de rega eficiente</b>	<b>43</b>
3.1	Caso de estudo . . . . .	43
3.2	Solução proposta . . . . .	44
3.3	Módulo de monitorização e configuração (1) . . . . .	45
3.3.1	Interface gráfica ( <i>dashboard</i> ) . . . . .	45
3.4	Módulo intermediário (2) . . . . .	47
3.5	Módulo de atuação (3) . . . . .	49
3.5.1	Controlo da iluminação e a temperatura . . . . .	49
3.5.2	Controlo da iluminação e da temperatura . . . . .	52
3.6	Módulo de recolha de informação (4) . . . . .	54
3.6.1	Leitura de temperatura e humidade . . . . .	55
3.7	Partilha de informação entre os módulos (5, 6, 7) . . . . .	56
3.8	Descrição do sistema . . . . .	56
<b>4</b>	<b>Conclusão e Trabalho Futuro</b>	<b>61</b>
4.1	Trabalho Futuro . . . . .	62
4.1.1	Proposta 1 . . . . .	62

4.1.2 Proposta 2 . . . . .	63
<b>A Apêndice</b>	<b>69</b>
A.1 Código 1 . . . . .	69
A.2 Código 2 . . . . .	78

# Lista de Tabelas

2.1	Comparações entre ESP8266, ESP32 e Arduino UNO [31] . . . . .	25
3.1	Tópicos. . . . .	57



# Lista de Figuras

2.1	Modelo básico de funcionamento de um sistema de RFID [7]	8
2.2	Modelo Bluetooth de baixa energia [11]	10
2.3	Comunicação da rede Zigbee [19]	14
2.4	Exemplo de comunicação Z-Wave [23]	17
2.5	Protocolo MQTT [25]	18
2.6	Qualidade de serviço nível 0: entrega no máximo uma vez [26]	19
2.7	Qualidade de serviço nível 1: entrega pelo menos uma vez [26]	19
2.8	Qualidade de serviço nível 2: entrega exatamente uma vez [26]	20
2.9	A estrutura do módulo ESP8266 [33]	22
2.10	Esquema interna do ESP32 [37]	24
2.11	<i>Layout</i> de Node-RED	27
2.12	Arduíno IDE. [40]	28
2.13	Raspberry Pi [40]	30
2.14	Comparação de ZigBEE, Wi-Fi e Bluetooth [48]	33
2.15	Técnicas de eficiência energética em IoT [48]	33
3.1	Arquitetura da solução.	45
3.2	Interface.	47
3.3	Fluxograma para leitura de status da iluminação.	50
3.4	<i>Dashboard</i> controlo "on" ou "off".	51
3.5	<i>Dashboard dimming</i> .	52
3.6	Sistema para controlar a iluminação.	53

3.7	Fluxograma para leitura de dados no sensor. . . . .	54
3.8	Sistema para a leitura de temperatura e humidade. . . . .	55
3.9	Protocolo de comunicação. . . . .	56
3.10	Diagrama de blocos do sistema proposto. . . . .	57
3.11	Implementação do sistema. . . . .	58



# Acrónimos

**API** *Application Programming Interfaces.*

**AVAC** Aquecimento, Ventilação e Ar Condicionado.

**BLE** Bluetooth de baixa energia.

**CDMA** *Code Division Multiple Access.*

**FDMA** *Frequency Division Multiple Access.*

**FSK** Modulação de mudança de frequência.

**GPIO** *General Purpose Input/Output.*

**GSM** Sistema global para comunicação móvel.

**HTML** *Hyper Text Markup Language.*

**I2C** *Inter-Integrated Circuit.*

**IoT** Internet das Coisas.

**LED** *Light-emitting diode.*

**MQTT** *Message Queuing Telemetry Transport.*

**PHP** *Hypertext Preprocessor.*

**Pub/Sub** *Publisher/Subscriber.*

**QoS** *Qualidade de serviço, Quality of Service.*

**RF** *Radiofrequência.*

**RFID** *Identificação por radiofrequência.*

**RX** *Recetor.*

**SCADA** *Supervisory Control and Data Acquisition.*

**SCL** *Serial Clock Line.*

**SDA** *Serial Data Line.*

**TCP/IP** *Transmission Control Protocol/Internet Protocol.*

**TDMA** *Time Division Multiple Access.*

# Capítulo 1

## Introdução

Atualmente, a tecnologia evoluiu exponencialmente, onde a exploração da realidade virtual tem aumentado. Diferentes aparelhos que conectam entre si, podem ser operados a partir de um dispositivo conectado à Internet, como um *smartphone* ou telefones inteligentes. Isto é conhecido como a Internet das Coisas, que consiste em conectar todos os tipos de objetos à Internet, que por sua vez possibilita a sincronização entre si e pode ser utilizado remotamente. Internet das Coisas é uma conexão entre o mundo digital e o mundo físico. Dessa forma, pode-se afirmar que a Internet das Coisas está cada vez mais presente no nosso dia a dia, sem mencionar que muda a maneira com que nos relacionamos com os diferentes elementos e contextos.

A Internet das Coisas não limita a qualquer tipo de produtos de consumo, ou seja, ela pode interagir com qualquer produto [1].

A plataforma possibilita aos seus utilizadores que desenvolvam e controlem projetos na área e, uma das vantagens mais importante disto é um melhor gestão de tempo e permite um menor gasto com receita nas instalações [2].

No que diz respeito a segurança, ainda são uma das principais questões a serem tratadas e resolvidas, sabendo que muitos dos estudos e trabalhos a serem realizadas são direcionados para o lado da comunicação e distribuição das informações e poucos em tópicos de segurança, onde as soluções já existentes tornam o projeto inviável, devido à

necessidade de um alto grau de processamento de troca de informações. Para obter maior aceitação, o tópico de segurança tem de ser aperfeiçoado. [2].

Na questão de privacidade, existem alguns dispositivos que, para o utilizador, não trazem grandes vantagens em estarem ligados à Internet, mas, para os fabricantes, as ações de *marketing* procuram saber onde estão os seus produtos, quem os comprou, como e quando são utilizados. E quem possui este tipo de dados passa a saber o estilo de vida do consumidor e dados pessoais [3].

O sistema Internet das Coisas (IoT) deve ter três requisitos importantes: sentido abrangente: uso de sensores para obter informações do mundo circundante; Transmissão confiável: entrega em tempo real de informações precisas; Processamento inteligente: um mecanismo para analisar e processar grandes quantidades de dados e informações para adicionar controlo inteligente aos objetos [4].

O objetivo principal deste trabalho é desenvolver um sistema de comunicação sem fio usando o protocolo *Message Queuing Telemetry Transport* (MQTT), microcontroladores ESP32 e ESP8266 para controlar o sistema de rega numa zona agrícola com uma dimensão consideravelmente grande, onde possui diferentes tipos de produtos agrícolas, deverá ser comandado a partir de qualquer dispositivo eletrónico como *smartphone*, *tablet*, computadores, etc.

## 1.1 Motivação

Um aspeto da Internet das Coisas tem estado a oferecer uma visão abrangente de cenários com foco em benefícios privados, sociais e empresariais. É crucial desenvolver este desafio proposto. Ainda há bastantes pesquisas indispensáveis sobre desafios técnicos que muitas vezes são dadas como resolvidos (especialmente por pesquisadores e profissionais sem conhecimento técnico). A privacidade e segurança são fatores-chave para proporcionar uma

Internet confiável. São necessários novos mecanismos de compartilhamento de custos e benefícios que possibilitam a criação de oportunidades para novos participantes no mercado. Novos interessados precisam estar motivados para auxiliar o futuro da Internet das Coisas. Facilidade de participação, colaboração e geração de benefícios são os principais encargos para trazer novos participantes à Internet das Coisas. Estruturas abertas e ambientes de programação do utilizador final podem capacitar os indivíduos a criar micros serviços gratuitos e faturáveis, como guias de produtos e análises.

## 1.2 Objetivos

O objetivo deste trabalho é desenvolver um sistema de automação baseado em tecnologia IoT que consiga controlar o sistema de rega de forma automático e prático numa zona agrícola com uma dimensão consideravelmente grande, onde possui diferentes variedades de produtos agrícolas. Para alcançar o objetivo deste trabalho será requerido: utilização de elementos de sensorização e atuadores que serão instalados dentro das estufas, nos lugares mais estratégicos para fazer as leituras de temperatura e humidade, também para atuar sobre os aspersores de rega, iluminação e ventilação; desenvolvimento de uma lógica de controlo através do qual será controlado todo o sistema de rega; e a utilização de tecnologia IoT a que será mais viável para resolver o problema em questão.

## 1.3 Estrutura do documento

Esta tese está estruturada da seguinte forma:

**Capítulo 1** - Refere-se basicamente a alguns tópicos como a introdução, motivação e objetivo do trabalho.

**Capítulo 2** - No capítulo dois, numa fase inicial, descreveram-se algumas tecnologias que podem ser utilizadas no projeto, citando os seus modos de funcionamento, vantagens e desvantagens, logo a seguir fez-se uma comparação entre essas tecnologias e escolheu-se a mais vantajosa. Além disso, foi descrito as ferramentas e *softwares* aplicados no projeto,

fez-se resumo de alguns temas relacionados e por fim a descrição do problema.

**Capítulo 3** - No capítulo 3, é realizada uma descrição de toda a arquitetura do problema em foco. É exibida uma visão geral de todo o trabalho prático desenvolvido, também é explicado como funciona a troca de informação do projeto e a funcionalidade dos componentes no projeto.

**Capítulo 4** - Por fim serão apresentadas as principais conclusões e os trabalhos futuros.

# Capítulo 2

## Revisão bibliográfica

A Internet das Coisas (IoT) é um sistema de dispositivo inter-relacionados e tem a capacidade de transferir dados numa rede sem a necessidade de humanos para a integração. Até o dia de hoje, está a ter uma grande canalização em diversas áreas utilizando a IoT para operar com mais eficiência, compreender melhor as necessidades dos seus clientes, fornecer o melhor serviço, fazer com que os seus negócios cresçam cada vez mais e melhorar a tomada de decisão [5].

O sistema da IoT compreende-se em dispositivos autónomos que podem ser controlados pela *web*, que possam usar sistemas agregando sensores, microcontroladores, *hardware* de comunicação como, por exemplo, Arduíno, ESP8266, ESP32, para coletar dados, enviar e agir sobre os dados adquiridos aos seus ambientes [6] [5]. Os dispositivos de IoT coletam os seus dados e enviam para as nuvens [6] [5] para serem analisados ou dependendo da aplicação podem ser cessados localmente dependendo dos requisitos de segurança, as vezes esses dispositivos se comunicam com outros dispositivos relacionados, e agem com base nas informações obtidas do outro. Os dispositivos fazem o trabalho a maior parte do tempo sem a intervenção humana, embora a interação humana seja necessária, como, por exemplo, para configurá-los, fornecer instruções ou cessar os dados para uma tomada de decisão [5].

A Internet das Coisas ajuda as pessoas a viverem e trabalharem de maneira mais

inteligente além de obter controlo do que estão controlando, uma grande aplicação para Internet das Coisas é no uso residencial onde várias aplicações para automação residencial estão disponíveis, por exemplo uma empresa de automação residencial pode utilizar a IoT para monitorar e manipular sistemas mecânicos e elétricos em um edifício remotamente.

Em uma escala mais ampla, cidades inteligentes podem ajudar os seus cidadãos a reduzir o desperdício e o consumo de energia. A IoT é essencial para os negócios, ela fornece às empresas uma visão em tempo real de como os seus sistemas realmente funcionam, fornecendo informações sobre tudo, desde desempenho das máquinas, operações de cadeia em suprimentos e logística. Ela permite que as empresas automatizam processos e reduzem os custos de mão de obra, também reduz o desperdício e melhora a prestação de serviços tornando mais barato a fabricação e a entrega de mercadorias além de oferecer transparência nas transações dos clientes [6] [5].

Alguns dos benefícios comuns da IoT permite que as empresas monitoram seus processos comerciais, melhoram as experiências dos clientes, economizam tempo e dinheiro, aumentem a produtividade dos funcionários, façam a integração e adaptação de modelos de negócio tomem melhores decisões de negócios e acabam gerando mais receita, também incentiva as empresas a repensarem a maneira como abordam seus negócios e fornecem as ferramentas para melhorar suas estratégias [6] [5].

A IoT é muito utilizado em organização de manufaturas, serviços públicos e transportes fazendo uso de sensores e outros dispositivos de IoT. Também tem-se utilizados em setores da agricultura, infraestrutura e automação residencial levando algumas organizações a transformações digitais. Como por exemplo, IoT pode beneficiar os agricultores na área da agricultura tornando os seus trabalhos muito mais fácil, os sensores do campo podem coletar dados sobre precipitação, humidade, temperatura e análise do solo além de outros fatores que ajudaria a automatizar as técnicas agrícolas [5].

## 2.1 Tecnologias

Existem inúmeras tecnologias para trabalhar com a IoT e nas seções a seguir, será apresentado diferentes tecnologias que podem ser utilizados para criar um sistema inteligente de arquitetura do sistema doméstico. Para cada tecnologia apresentada, serão discutidas as suas características, as suas vantagens, desvantagens e se eles são adequados para atingir os objetivos do projeto a ser realizado.

### 2.1.1 Identificação por radiofrequência (RFID)

Identificação por radiofrequência (RFID) usa indução eletromagnética ou propagação eletromagnética para identificar automaticamente objetos ou corpos humanos. RFID possui um mecanismo de segurança para evitar a cópia, combinando criptografia e dados de segurança para evitar falsificações [4]. Ele existe desde a década de 1940 e se tornou um suplemento à tecnologia de código de barras, amplamente utilizada em todo o mundo [7].

A RFID funciona da seguinte forma: A RFID é constituída por dois componentes, um dos componentes está situado no objeto a ser reconhecido e é chamado de transponder (tag) e o outro componente é chamado de leitor. O leitor depende da tecnologia a ser utilizada, e dependendo da tecnologia utilizada ela pode ser um dispositivo de captura ou captura/ transmissão de dados. Os leitores ou antenas podem usar sinais de rádio como um meio para ativar tags, trocar e enviar informações. Na fabricação das antenas são levados em consideração diferentes tamanhos e formas, diferentes características e configurações, para cada tipo de antena fabricado, é utilizado para diferentes tipos de aplicações [7].

As principais vantagens da RFID estão relacionadas com a sua utilização e instalação simples: apenas o sistema precisa de etiquetas RFID e leitores RFID, e esta é uma tecnologia adequada para apoiar o rastreamento físico [4].

RFID é limitado em relação às capacidades de detecção e flexibilidade de implantação,

uma vez que é uma tecnologia de proximidade tecnológica. Assim, a tecnologia RFID é mais orientada para identificar objetos ou pessoas ao invés de controlar dispositivos e suas propriedades. É possível ser utilizado em um sistema doméstico inteligente para identificar utilizadores [4], porém esse recurso está fora do objetivo deste projeto.

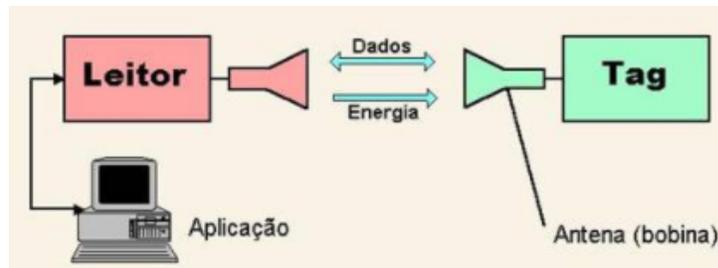


Figura 2.1: Modelo básico de funcionamento de um sistema de RFID [7]

### 2.1.2 Sistema global para comunicação móvel (GSM)

Este é um sistema de comunicação móvel 2G. Sistema global para comunicação móvel (GSM) é o primeiro sistema de comunicação móvel digital que suporta comunicação de voz e dados (até cerca de 130 kbit/s) [8]. É o padrão global para comunicações móveis com mais de 90% de participação de mercado, operando em mais de 219 países e regiões.

GSM é um dos sistemas de telecomunicações móveis mais seguros disponíveis atualmente, que pode manter a segurança de ponta a ponta e manter a confidencialidade das chamadas e o anonimato dos utilizadores GSM. O protocolo de segurança GSM atribui um identificador temporário ao número dos utilizadores para manter a privacidade dos utilizadores. Através do aplicativo, algoritmos de criptografia e salto de frequência habilitados por sistemas digitais e sinalização podem ser usados para obter privacidade de comunicação [4].

A nível do funcionamento o sistema GSM 900 utiliza duas classes de frequências na banda dos 900 MHz, numa classe a frequência varia de 890-915 MHz sendo utilizado para as transmissões do terminal e o outro nos 935-960MHz, para as transmissões da rede. O

recurso usado pelo GSM para gerir as frequências é uma combinação de duas tecnologias: o *Time Division Multiple Access* (TDMA) e o *Frequency Division Multiple Access* (FDMA). Uma ou mais destas frequências concedidas a cada estação-base que também é dividida, em termos de tempo, utilizando o TDMA, em oito espaços de tempo (*timeslots*). O terminal utiliza um *timeslot* para receção e outro para emissão. Eles encontram-se separados temporalmente para que o *smartphone* não esteja a receber e enviar os dados simultaneamente. Esta divisão de tempo é designado por *full rate*. Na rede também é possível dividir as frequências em 16 espaços, este processo é chamado de *half-rate*, mas neste processo a qualidade da transmissão é inferior.

A vantagem do GSM está na sua ampla cobertura, mesmo quando a Internet não está disponível, o que praticamente mantém todo o sistema online o tempo todo, e sua alta segurança. Infraestrutura, que fornece confiabilidade máxima, para que as pessoas não autorizadas não possam monitorar as informações enviadas ou recebidas [4].

As desvantagens do uso do GSM como protocolo de comunicação em um sistema doméstico inteligente são os custos adicionais de implementação e uso, exigindo que um provedor de serviços GSM externo seja capaz de interconectar todos os dispositivos. Além disso, a cobertura GSM é limitada em locais subterrâneos [4] e também o sistema GSM é baseado na rede TDMA, que é considerada menos avançada que a concorrente *Code Division Multiple Access* (CDMA) [9]. Por esta causa não será aplicado esta tecnologia no projeto.

### 2.1.3 Bluetooth de baixa energia (BLE)

Bluetooth de baixa energia (BLE), parte do padrão Bluetooth v4.0, é uma das novas tecnologias mais promissoras nos dispositivos, incluídos em quase todos os *smartphones* mais recentes. Ele permite que *tags* pequenas e de baixo custo anunciem a sua presença, transmitindo um pacote de publicidade uma vez por segundo [10], que lhes permite operar

durante até um ano com uma bateria de célula tipo moeda de lítio. Esta nova tecnologia cria mais oportunidades para implantação universal, leituras de *tag* de maior precisão e a capacidade de se misturar de forma invisível com um produto [4]. E, em termos de energia, o termo BLE carrega no nome baixa energia, ou seja, foi feito para funcionar em equipamentos móveis, longe de tomadas, simplesmente com pequenas baterias e com alta duração [10]. Esta tecnologia pode ser usada para comunicação limitada entre dispositivos. Se bem que o BLE forneça maior cobertura do que a RFID, mas não será utilizado, por motivos idênticos aos da RFID.



Figura 2.2: Modelo Bluetooth de baixa energia [11]

#### 2.1.4 X10

O padrão da indústria X10, desenvolvido em 1975 para comunicação entre dispositivos eletrônicos, é um dos padrões mais antigos para controlar e gerir dispositivos domésticos. Esta tecnologia fornece controlo sobre os dispositivos domésticos por meio das linhas de energia da casa. As linhas de força são utilizadas para sinalização e controlo, onde os sinais envolvem breves rajadas de radiofrequência representando informação digital [4].

As grandes vantagens do X10 estão relacionadas ao seu custo reduzido de implementação e à sua simplicidade e facilidade de instalação. Não há necessidade de profissionais qualificados para a instalação de um sistema X10 [12]. O X10 permite que os utilizadores controlem até 256 dispositivos e há uma grande variedade de dispositivos disponíveis

para comprar em lojas especializadas em eletricidade. Além disso, X10 é uma tecnologia comprovada que tem sido aplicada em sistemas reais há mais de 20 anos [4].

As principais desvantagens da tecnologia X10 estão relacionadas à sua largura de banda limitada. Desde X10 sinais só podem transmitir um comando de cada vez, primeiro endereçando o dispositivo para controlar e, em seguida, enviar uma operação para aquele dispositivo realizar, se dois sinais X10 forem transmitidos ao mesmo tempo, eles podem colidir, o que pode levar a comandos que não podem ser decodificados ou acionar operações erradas [4]. Como consequência das baixas velocidades de transmissão, os transmissores apenas são capazes de realizar operações simples que envolvem poucos dados (ligar e desligar aparelhos e luzes e regular a intensidade luminosa de lâmpadas) [13].

Outra desvantagem do X10 está relacionada ao seu baixo ritmo de comunicação, o protocolo X10 é lento, baixa fiabilidade do protocolo e também a número limitado de dispositivos [12]. O X10 como utiliza a rede elétrica para comunicar, não serve para lidar com sinais digitais de alta resolução como sinais de vídeo, televisão e Wi-Fi [13]. Devido a todas as desvantagens tratadas, não será aplicado o X10 no projeto.

### 2.1.5 Insteon

Insteon é uma tecnologia desenvolvida especialmente para a automação residencial para o controle de luzes, eletrodomésticos, termostatos, sensores de movimento e outros dispositivos eletrônicos encontrados na casa. Esta tecnologia usa a tecnologia de radiofrequência sem fio (RF) e a fiação elétrica existente de um edifício para se comunicar com dispositivos [14].

Os sinais de RF Insteon usam Modulação de mudança de frequência (FSK) em 904 MHz, com um valor bruto de 38,4 kb / s [4]. Dispositivos Insteon são pares, o que significa que qualquer um deles pode desempenhar o papel de remetente, recetor, ou retransmissor [4].

Existem inúmeras vantagens relacionadas a essa tecnologia, onde serão citadas algumas delas como: fácil de instalar: A instalação do Insteon nas casas existentes não requer nenhuma nova fiação porque os produtos Insteon se comunicam sobre os fios de linha de alimentação existentes, bem como sem fio; resposta instantânea: a mensagem de Insteon passa em menos de 0,05 segundos - mais rápido do que o olho humano pode detetar, portanto, não há atraso visível. Não há atraso significativo na resposta do dispositivo Insteon ao comando. A velocidade de sinalização do Insteon é otimizada para controlo residencial e é rápida o suficiente para responder rapidamente; confiável: O Insteon é uma das tecnologias de rede de controlo doméstico sem fio mais confiáveis porque os dispositivos Insteon repetem mensagens recebidas de outros dispositivos Insteon. Além disso, a sua comunicação de malha dupla garante que haja vários caminhos para as mensagens viajarem, é muito simples de usar e acessível [14].

A principal desvantagem desta tecnologia está relacionada com os seus custos de implementação e dispositivos caros, por isso não será utilizada no projeto.

### 2.1.6 Wavenis

Wavenis é uma pilha de protocolo sem fio, desenvolvida pela Coronis Systems e promovida pela Wavenis *Open Standards Alliance* (Wavenis OSA) para aplicações de controlo e monitoramento [4]. Opera nas bandas ISM em diferentes regiões do mundo, como 433 MHz (China), 868 MHz (Europa) e 915 MHz (EUA). Ele utiliza tecnologia de espectro de propagação de frequência para transmitir sinais de RF [15]. A subcamada Wavenis MAC fornece soluções síncronas e assíncronas. Numa rede síncrona, um nó é atribuído a um intervalo de tempo calculado pseudo-aleatoriamente com base no seu endereço. Antes da transmissão, o nó verifica se o canal está ocupado. Se estiver ocupado, ele calculará um novo intervalo de tempo e adicionará um atraso na transmissão da mensagem.

Wavenis é um protocolo de rede sem fio interessante, mas não é popular, reduzindo assim o número de dispositivos disponíveis que suportam a tecnologia [4], pode ser aplicado

na leitura remota de medidores multi-fluidos, na implementação de soluções eficientes em termos de energia na rastreabilidade RF de objetos. A nível de benefícios ele reduz o custo de manutenção, otimização da eficiência energética, previsão do consumo multi-fluido [16].

### 2.1.7 ZigBee

O Zigbee é um protocolo de comunicação sem fios atribuído a aparelho IoT, com o centro em dispositivos de baixa potência que não requer muita largura de banda. Esses, por sua vez, são dispositivos que apresentam um serviço energético limitado, com autonomia que pode durar muito tempo com uma bateria [17] [18]. Usa um padrão de comunicação de Radiofrequência (RF) baseado em IEEE 802.15.4 [17] [4].

A rede Zigbee ao nível de funcionamento, em vez de ligar os dispositivos diretamente ao router, que está ligado à Internet, os dispositivos serão ligados a um dispositivo chamado *hub* Zigbee que é um concentrador ou coordenador, ele vai conectar-se com o router que por sua vez vai conectar com a Internet. O Zigbee cria uma nova rede e os dispositivos além de comunicarem com o *hub* Zigbee, eles podem comunicar entre si, que é uma característica dessa rede e tem uma configuração de malha porque temos dispositivos que fazem o papel de nós na rede, e cada *hub* Zigbee pode gerir até 240 dispositivos, mas é importante consultar a recomendação do fabricante que indique a quantidade que pode garantir um bom desempenho [19].

O Zigbee tem três tipos de dispositivos diferentes [19]:

- **Coordenador ou *hub*** - É um dispositivo que inicia a rede e mantém informação de todos os dispositivos conectados, fazendo a conexão com a rede Wi-Fi e a Internet;
- **Router** - São dispositivos intermediários, que realizam a transmissão de dados para outros dispositivos próximos, como lâmpadas, tomadas e interruptores, funcionam como roteadores.
- **Dispositivos finais** - São dispositivos operados por baterias, como sensores, que contém a funcionalidade apenas o suficiente para falar com o hub ou router. Estes

dispositivos podem ficar sem trabalhar uma parte do tempo, o que reflete em baixo consumo de bateria.

O ZigBee tem uma vantagem sobre outros protocolos visto que é um protocolo completamente aberto, possui custo muito baixo de dispositivos, baixo consumo de energia, tem uma cobertura mais ampla sendo de fácil aplicação, maior velocidade de reação dos dispositivos, quando comparada ao Wi-Fi. Conforme a configuração da rede e dos dispositivos, é possível a solicitação de confirmação de recebimento de dados, o reenvio de dados não recebidos e o uso de diferentes caminhos para que as mensagens cheguem ao endereço final [20] [4] [19]. Também possuem algumas desvantagens como uma baixa potência de transmissão que pode ser um problema se a rede não for bem configurada, e, ainda não há garantia de interoperabilidade de equipamentos de marcas diferentes [19].



Figura 2.3: Comunicação da rede Zigbee [19]

### 2.1.8 Z-Wave

Z-Wave é um protocolo para automação doméstica desenvolvida pela *ZenSys* e promovida pela *Z Wave Alliance*, de baixo custo, completamente *wireless* e é implementável em qualquer estrutura, nova ou antiga, sem exigências ao nível de infraestrutura prévia [21] [4].

Este protocolo destaca-se pela fiabilidade, baixo consumo, rapidez e alcance das suas redes. O fato de trabalhar numa banda estreita, que está fora dos 2,4 GHz, faz com que não sofra interferências de redes *wireless*, sistemas Bluetooth, telefones sem fio ou dispositivos de transmissão áudio/vídeo [21]. O objetivo principal é permitir a transmissão confiável de mensagens curtas de uma unidade de controlo para um ou mais nós na rede.

O Z-wave tem um mecanismo para evitar colisões que verifica se o canal está disponível antes de iniciar uma transmissão. Se o canal estiver ocupado ou indisponível, a transmissão é adiada por um período aleatório de tempo [4].

Cada rede Z-Wave é identificada por uma ID de rede (identificação de rede), e cada dispositivo é identificado por uma ID de nó (identificação de nó). O ID da rede é a identificação comum de todos os nós pertencentes a uma rede lógica de Z-Wave. O ID de rede tem um comprimento de 4 bytes (32 bits) sendo atribuído a cada dispositivo, pelo controlador primário, quando o dispositivo está "incluído" na rede. Os nós com diferentes ID's de rede não podem comunicar entre si. O ID de nó é o endereço de um único nó na rede e tem um comprimento de 1 byte (8 bits) e deve ser exclusivo dentro da sua rede [21].

O Z-Wave mostra muitas vantagens, que o conduz a uma grande aceitação de mercado. E serão citadas alguns como [22]:

- **Instalação simples** - Para começar a utilizar um dispositivo Z-Wave eletrónico, como, por exemplo, um interruptor para ligar e desligar uma iluminação ou termostato, é necessário apenas de ligar, adicionar o dispositivo à rede Z-Wave e começar a controlá-lo imediatamente do seu PC, *tablet* ou telefone. A instalação da maioria dos dispositivos Z-Wave é “tão simples quanto substituir uma lâmpada”.

- **Acessibilidade** - Uma grande vantagem do Z-Wave em comparação com outras soluções de casa inteligente, como X10 e ZigBee, é sua acessibilidade. Como a instalação não requer um empreiteiro especializado ou trabalho de construção, pode optar por expandir o seu sistema a qualquer momento, as soluções de automação residencial Z-Wave são adequadas e estão disponíveis para todos.
- **Alcance do sinal sem fio** - A frequência Z-Wave usada na Europa é 868,42 MHz. Isso significa que, ao contrário de outras tecnologias sem fio usadas nas casas, edifícios, etc., ele não sofre interferência de telefones, redes Wi-Fi e similares. O alcance comum de um sinal Z-Wave é de 100 metros ao ar livre e 50 metros dentro dos edifícios.

Z-Wave também é uma tecnologia de malha, ou seja, todos os dispositivos Z-Wave podem se comunicar entre si e encaminhar o sinal de um dispositivo Z-Wave para outro, o que significa que as informações são roteadas do remetente para o destinatário em todo o Z-Rede Wave. Isto significa que, quanto mais dispositivos Z-Wave houver na sua casa, maior será o alcance e o tamanho da rede Z-Wave.

- **Interoperabilidade e disponibilidade de dispositivos** - Atualmente Z-Wave é uma das técnicas de automação residencial mais usadas no mundo. Existem mais de 900 produtos oferecidos por mais de 250 fabricantes em todo o mundo. Todos os dispositivos Z-Wave são totalmente interoperáveis, isso significa que não está vinculado a um único fabricante ao implementar uma solução de automação residencial. Portanto, o seu investimento fica mais protegido do que usar uma solução proprietária de um único fabricante.
- **Eficiência energética** - A capacidade de economizar energia é de extrema importância. Alguns dos dispositivos Z-Wave são alimentados por baterias, e esses dispositivos são otimizados para a eficiência energética com uma bateria de duração de um ano ou mais.
- **Confiabilidade e segurança** - O Z-Wave usa o mesmo padrão de segurança da

informação (criptografia) que os aplicativos de *Internet banking*.

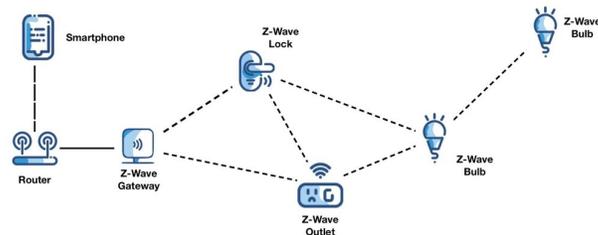


Figura 2.4: Exemplo de comunicação Z-Wave [23]

### 2.1.9 MQTT

O MQTT foi criado na década de 90 pela empresa IBM, com a intenção de desenvolver trabalhos de sistema de supervisão e aquisição de dados, este protocolo foi tão interessante que evoluiu até encontrar um espaço nesse vasto mercado de Internet das Coisas [24].

MQTT é uma arquitetura *Publisher/Subscriber* (Pub/Sub) sobre *Transmission Control Protocol/Internet Protocol* (TCP/IP) que permite a comunicação bidirecional entre um dispositivo e um Broker MQTT. É extremamente simples e leve [4].

No padrão pub/sub, os *subscriber* que também é conhecido como o cliente, quando ele deseja receber uma informação, dá um *subscribe* no tópico que o interessa através de uma requisição, para um elemento da rede designado de *Broker* que por sua vez trabalha como um intermediário no processo de comunicação, *subscriber* que querem publicar informações, o fazem através do *Broker*, também para rotear as mensagens até o destinatário é feito através do *Broker*. [24].

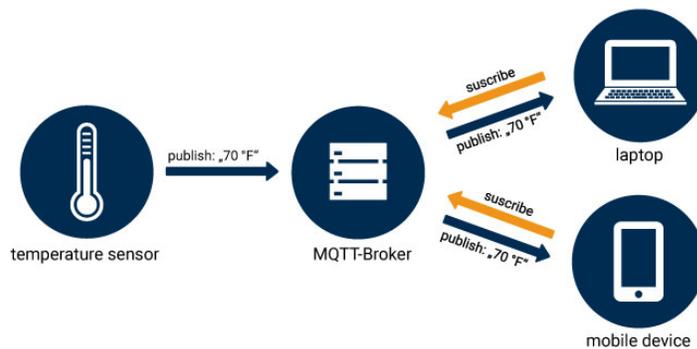


Figura 2.5: Protocolo MQTT [25]

O *Broker* é o coração de qualquer protocolo pub/sub. O *Broker* tem como principal função receber, filtrar e encaminhar as mensagens aos clientes. Além disso, o *Broker* é responsável por comandar os tópicos e receber as assinaturas dos clientes. Outra função que o *Broker* desempenha é de armazenamento de dados, tanto informações de registo dos assinantes, como as informações publicadas. A conexão do cliente ao *Broker*, seja ele *subscribe* ou *publish*, é originalmente feita via TCP, com opções de *login* (utilizador e senha) e uso de criptografia. Todo processo de conexão estabelece também um nível de Qualidade de serviço, *Quality of Service* (QoS) desejado, indicando como deve ser a relação entre os elementos comunicantes [24].

QoS é um recurso chave do protocolo MQTT. O QoS dá ao cliente o poder de escolher um nível de serviço que corresponda à confiabilidade da rede e à lógica do aplicativo. Como o MQTT gerência a retransmissão de mensagens e garante a entrega (mesmo quando o transporte subjacente não é confiável), o QoS torna a comunicação em redes não confiáveis muito mais fácil.

**QoS 0 no máximo uma vez** - Este nível de serviço garante uma entrega de melhor esforço. Não há garantia de entrega. O destinatário não confirma o recebimento da mensagem e a mensagem não é armazenada e retransmitida pelo remetente [26].



Figura 2.6: Qualidade de serviço nível 0: entrega no máximo uma vez [26]

**QoS 1 - pelo menos uma vez** - O nível 1 de QoS garante que uma mensagem seja entregue pelo menos uma vez ao recetor. O remetente armazena a mensagem até obter um pacote *PUBACK* do recetor que confirma o recebimento da mensagem. É possível que uma mensagem seja enviada ou entregue várias vezes [26].



Figura 2.7: Qualidade de serviço nível 1: entrega pelo menos uma vez [26]

**QoS 2 - exatamente uma vez** - QoS 2 é o nível de serviço mais alto no MQTT. Este nível garante que cada mensagem seja recebida apenas uma vez pelos destinatários pretendidos. QoS 2 é o nível de serviço de qualidade mais seguro e lento. A garantia é fornecida por pelo menos dois fluxos de solicitação / resposta (um *handshake* de quatro partes) entre o remetente e o destinatário. O remetente e o receptor usam o identificador de pacote da mensagem *publish* original para coordenar a entrega da mensagem [26].



Figura 2.8: Qualidade de serviço nível 2: entrega exatamente uma vez [26]

O protocolo MQTT tem muitas vantagens e serão citados algumas como: a codificação mais simples, que possibilita o protocolo a funcionar mesmo em sistemas pouco modernos ou com falhas de armazenamento. Além disso, será útil até em redes onde a Internet é limitada; É um sistema que não se sobrecarrega, só o necessário passa por um protocolo MQTT; Segurança e qualidade aliadas com a simplicidade do protocolo; Baixo consumo de memória, baixa necessidade de processamento para o envio de mensagem e baixo consumo de banda; O MQTT consegue ser bem mais leve e flexível devido ao modelo de *Publish/Subscribe*. Os *publishers* como os *subscribers* não precisam estar sincronizados para enviar e receber as mensagens, entretanto é possível enviar mensagens para o *Broker* de forma síncrona e somente continuar a execução após a confirmação e recebimento [27] [28] [29];

Há mais um que pode ser bastante útil para um sistema IoT: mensagens retidas. O protocolo MQTT inclui um mecanismo onde o *Broker* armazena a última mensagem retida para um tópico específico. Este recurso permite que um cliente que se inscreve num

tópico com mensagens retidas, para receber a última mensagem imediatamente após inscrevendo. Com este recurso, os clientes não precisam esperar até que uma nova mensagem seja publicada para saber o último *status* conhecido de outros dispositivos [4].

A escolha do servidor MQTT deve considerar as necessidades específicas das aplicações que estão em desenvolvimento, bem como os recursos financeiros disponíveis para o projeto. O protocolo MQTT foi visto como o protocolo mais viável para a resolução deste problema, conforme as vantagens que já foram citadas nas alíneas anteriores.

## 2.2 Ferramentas e *software*

Para realizar o sistema de controlo de uma zona agrícola, várias ferramentas e *softwares* foram utilizados, como ESP8266, ESP32, sensor DHT, Raspberry Pi, Arduino IDE, e *dashboard*.

### 2.2.1 ESP8266

O ESP8266 é um microcontrolador tipo sistema em chip (SoC) de uma fabricante chinesa chamado Espressif, esses módulos englobam capacidades para processamentos de dados, leituras e controlos das portas *General Purpose Input/Output* (GPIO), e tem protocolo de comunicação em rede TCP/IP sem fio (não precisando de nenhum módulo externo [30]), Wi-Fi de 2,4 GHz (no padrão 802.11 b/g/n, com suporte a criptografia de segurança WPA/WPA2) [31]. Pode atuar como uma aplicação *stand-alone*, onde ele não precisa de nenhum outro componente para funcionar [32] [31].

Em atuação, o ESP8266 consegue trabalhar em duas configurações: *access point* e *client*. Na configuração *access point*, ele funciona basicamente como um roteador, criando uma rede Wi-Fi restrita por *login* e senha. Neste modo o ESP8266 cria um servidor com protocolo de Internet aleatório ou predefinido dependendo da programação realizada. Este servidor pode conter uma página *web* com informações dos componentes ligados ao ESP8266 como *client*, ele estabelece uma conexão com a rede Wi-Fi escolhida, uma vez

conectado também cria um servidor e todos os dispositivos conectados na mesma rede Wi-Fi que o ESP8266 tem acesso a este servidor pelo endereço de IP. Este servidor também pode conter uma página *web* e o seu IP também pode ser aleatório ou predefinido na programação [32].

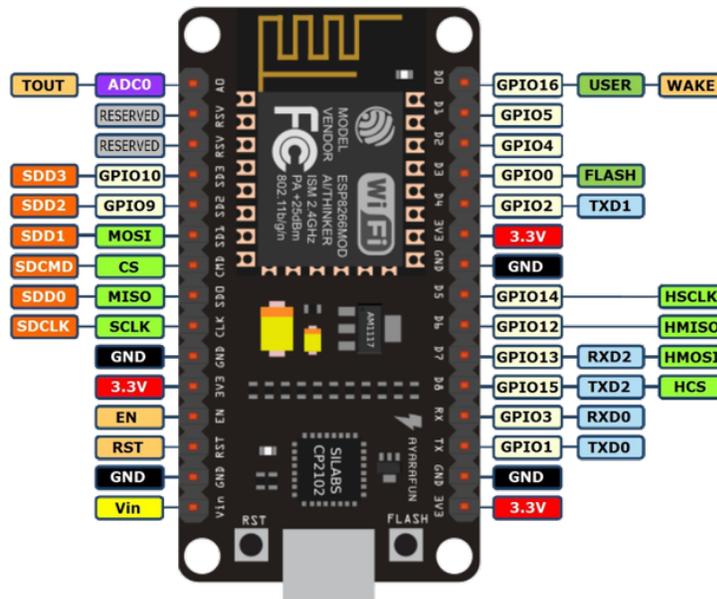


Figura 2.9: A estrutura do módulo ESP8266 [33]

O ESP8266 tem infinitas aplicações, mas serão citadas e esclarecidas de uma forma resumidas algumas das aplicações mais comuns, como:

- **Internet das Coisas** - O ESP8266 teve uma grande contribuição na IoT, pode ser utilizado para praticamente qualquer projeto de IoT que necessita de um cérebro simples e compacto [31] [30];
- **Automação e Controlo** – Seja industrial ou residencial, qualquer que seja esse módulo executa o trabalho;
- **Transmissão de dados** – Por ter a tecnologia Wi-Fi integrada, pode transmitir dados a uma rede *Wireless* de uma forma fácil e sem muito esforço;
- **Processamento de dados** – Processamento que vai de leitura de sensores analógicos digital até cálculos complexos para processar resultados, seja comparadores, seja

multiplicadores, somadores ou qualquer outro que o controlador necessite, poderá ser executado;

- **Gestão de dados** – O ESP8266 recebe os dados como, por exemplo, um sensor e gerência o que fazer entregando resultados dependendo do tipo de processamento à qual foi programado;
- **Web Server** – Permite fazer um servidor *Web* com ele, e pode cessar uma página escrita em *Hyper Text Markup Language* (HTML), *Hypertext Preprocessor* (PHP) ou qualquer outra linguagem de desenvolvimento *Web* suportado por ele.
- **Access Point** - Pode-se fazer comunicação entre ESPs, diretamente sem precisar de nenhuma rede externa, ou seja, comunicação ponto a ponto ou cessar com um *Smartphone*, computador [31].

### 2.2.2 ESP32

O ESP32 é uma série de microcontroladores com um preço muito baixo e um baixo consumo de energia, com Wi-Fi e Bluetooth integrado [34]. Teve a sua criação e desenvolvimento através da empresa chinesa chamado *Espressif Systems*, e é um substituto do microcontrolador ESP8266 [35]. O ESP32 tem antenas embutidas, amplificador de recepção de baixo ruído, filtros e módulos de gestão de energia. Está projetado para dispositivos móveis, sistemas de IoT entre outros aplicativos [36]. O ESP32 é efetuado nas placas de expansão e módulos entre [160, 240] MHz. Pode-se considerar uma boa velocidade para qualquer coisa que precisa de um microcontrolador com opções de conectividade [37].

O diagrama de blocos representado na figura 2.10 apresenta os componentes do ESP32.

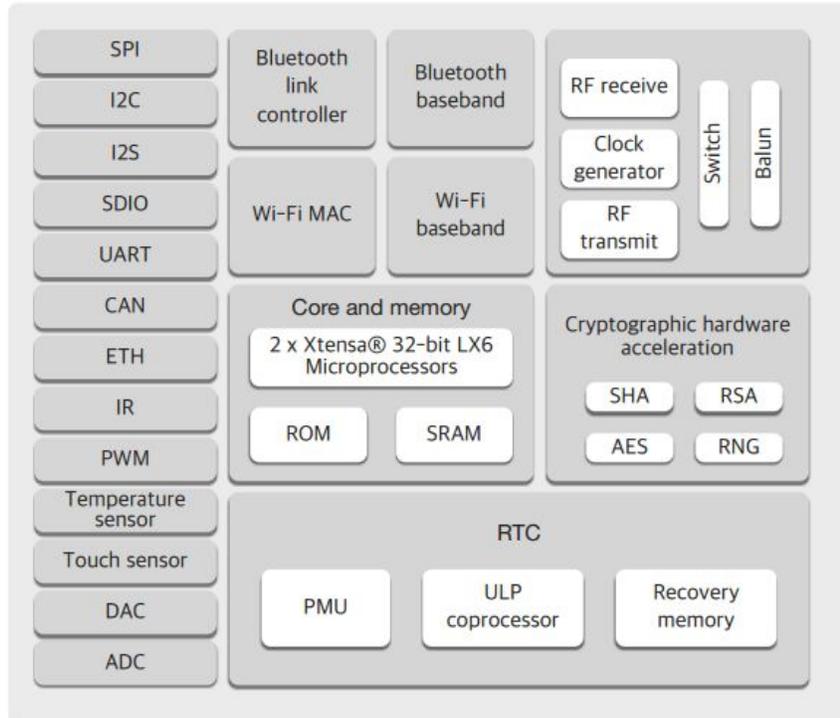


Figura 2.10: Esquema interna do ESP32 [37]

Os processadores de ESP32 tem memória interna ligados a seguintes usos [37]:

- ROM de 448 KBytes para partida e aplicações básicas;
- SRAM no chip de 520 KBytes para dados e instruções;
- SRAM de 8 KBytes em RTC, pode ser nomeado de memória RTC SLOW e consegue ser cessado por coprocessador;
- SRAM de 8 KBytes em RTC, que é nomeado de memória RTC FAST e pode ser utilizado para armazenamento de dados;
- 1 Kbit de EFUSE, de 256 bits é possível ser utilizado para o sistema (endereço MAC e configuração do chip) e os outros 768 bits são reservados para aplicativos do cliente, incluindo Flash-Encryption e Chip-ID.

### Comparações entre ESP8266, ESP32 e Arduino UNO

Na tabela 2.1 será feita uma comparação entre o ESP8266, ESP32 e o Arduino UNO R3 para ver as diferenças mais precisa entre eles.

	ESP8266	ESP32	Arduino UNO R3
Corrente	197 mA	220 mA	40 mA
Núcleo	1	2	1
Arquitetura	32 bits	32 bits	8 bits
Clock	80 - 160 MHz	160 - 240 MHz	16 MHz
Wi-Fi	SIM	SIM	Não
Bluetooth	Não	SIM Clássico e BLE	Não
RAM	160 KB	520 KB	2KB
FLASH	16 Mb	16 Mb	3KB
GPIO	11	22	12
DAC	0	2	0
ADC	1	18	6
Interfaces	SPI-I2C-UART-I2S	SPI-I2C-UART-I2S-CAN	SPI-I2C-UART

Tabela 2.1: Comparações entre ESP8266, ESP32 e Arduino UNO [31]

Para fazer uma boa comparação entre o ESP8266, ESP32 e o Arduino UNO serão apresentados alguns aspetos muito importante conforme ilustra a tabela 2.1 que ajuda os utilizadores a compreender as vantagens que uma tem sobre as outras.

**Corrente de consumo:** A nível de corrente de consumo o Arduino tem uma vantagem sobre os ESPs, mas dependendo do modo que está a ser usado os ESPs.

**Núcleo de processamento:** O ESP8266 e o Arduino tem um núcleo de processamento, só o ESP32 que tem 2 núcleos de processamento.

**Arquitetura:** Os ESPs trabalham com arquitetura de 32 bits enquanto que o Arduino trabalha com 8 bits.

**Clock de processamento:** O ESP8266 trabalha com 80MHz ou 160MHz pode-se

selecionar qual a velocidade que clock deseja que seu módulo trabalhe, o ESP32 também trabalha com duas opções, contudo é de 160 MHz ou 240MHz, que é bastante rápido para um microcontrolador, já o Arduíno trabalha em 16 MHz, que é muito inferior aos ESPs.

**Wi-Fi:** Os ESPs já vem embarcado neles um sistema *Wireless* sem fio, e o Arduíno não tem.

**Comunicação Bluetooth:** Só o ESP32 que já vem com o Bluetooth embarcado.

**Memória RAM:** O ESP8266 tem 160KB, o ESP32 tem 520KB e o Arduíno tem 2KB, isso é uma grande diferença de memoria RAM entre os ESPs e o Arduíno.

**Memória Flash:** O ESPs tem 4MB e o Arduíno tem 32KB,

**GPIOs:** No ESP8266 são 11 GPIO, no ESP32 são 22GPIO e o Arduíno são 12GPIO.

**DAC:** O único que tem o conversor Digital Analógico é o ESP32 que tem 2 GPIO com conversor Digital Analógico.

**ADC:** O ESP8266 só tem um conversor analógico digital, o Arduíno UNO que tem 6 GPIO e o ESP32 que tem 18 GPIO.

**Interfaces:** O ESP8266 suporta SPI - I2C – UART – I2S; Já O ESP32 Suporta SPI – I2C – UART – I2S – CAN; O Arduíno suporta: SPI – I2C e UART.

### 2.2.3 Node-RED

Node-RED é uma plataforma que se baseia em fluxos, foi concebido pela IBM *Emerging Technology*, é um recurso de desenvolvimento *Open-Source*. A sua criação foi para comunicar com aparelhos de *hardwares*, *Application Programming Interfaces* (API) e sistemas online com o objetivo de tornar os sistemas de IoT mais fácil. O Node-RED possibilita a programação a partir de interfaces gráficas muito simples e claro. Para o desenvolvimento de aplicações com recursos ao Node-RED, o utilizador só tem de conectar nós entre si, em que cada nó tem uma função. O Node-RED também concede a escrita de códigos mais organizados usando o JavaScript [38].

O Node-RED é composto por três tipos de nós [39]:

- **Nós da entrada** - Para uma dada aplicação, estes nós concedem autorização para que sejam introduzidos dados;
- **Nós de saída** - Dá-se a autorização para envio de dados para fora da aplicação;
- **Nós de processamento** - Estes permitem a manipulação de dados com o objetivo de possibilitar novos dados para os seguintes estágios.

### Node-RED-Dashboard

É um tipo de modelo que oferece um conjunto de nós no Node-RED, para que possa criar um painel de dados ao vivo muito rápido, elas devem ser instaladas e configuradas no Node-RED visto que, ao abrir o Node-RED pela primeira vez, ela não estará disponível.

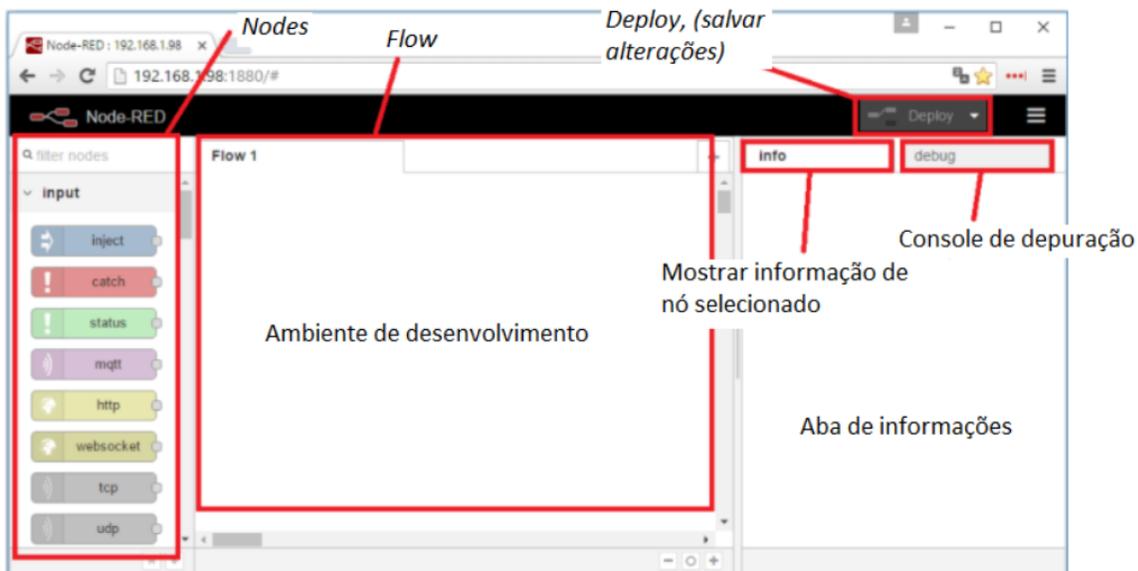


Figura 2.11: *Layout* de Node-RED

### 2.2.4 Arduino IDE

Arduino *Integrated Development Environment* (IDE) é utilizado para escrever e fazer *upload* de código para os microcontroladores ESP32 e ESP8266. É um *software* de código

aberto e trabalha em várias plataformas, como Windows, Mac OS X e Linux. O Arduino IDE oferece um editor de texto para escrever código. Estes programas são conhecidos como esboços e são salvos com a extensão de arquivo.ino. A linguagem de programação Arduino é uma excelente versão simplificada do C ++.

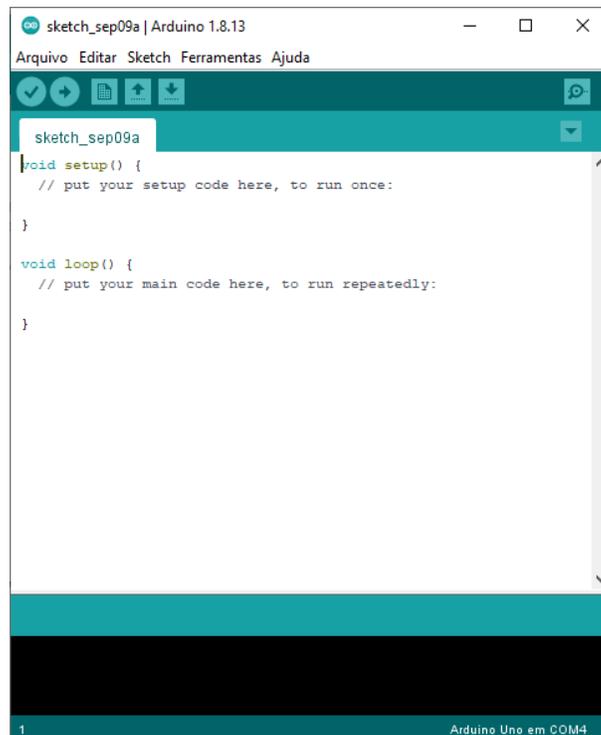


Figura 2.12: Arduino IDE. [40]

### 2.2.5 *Dashboard*

É um painel que mostra as métricas e indicadores importantes para alcançar os objetivos desejados de forma visual, ajudando a compreensão das informações. Criou-se um *dashboard* para ler dados do sensor, enviar e receber comandos do controlo dos atuadores, também tem muitas vantagens como a sua leveza, por ser muito leve é muito difícil de ter crash e capta informações em tempo real, tem uma instalação muito fácil em qualquer dispositivo (móvel, computadores), foi criado com uma interface de fácil utilização.

Tendo as duas interfaces analisadas (Node-RED-dashboard e dashboard-JavaScript)

optou-se por utilizar o dashboard-JavaScript por ser mais vantajoso do que o Node-RED-dashboard. Visto que o Node-RED-dashboard, no controlo da temperatura, atrasa alguns segundos para enviar a mensagem para a interface, e é menos leve do que o dashboard-JavaScript.

### 2.2.6 Raspberry Pi

O Raspberry é um computador de placa única que possui um tamanho de um cartão de crédito criado no Reino Unido pela *Raspberry Pi Foundation* com o intuito de favorecer o ensino de ciência da computação básica nas escolas [41].

Pode ser visto como um computador, uma vez que possui um processador, RAM, portas USB que possibilita a conexão do rato e teclado, porta HDMI que permite conectar uma TV ou monitor e também pode ser conectado à Internet. Com o Raspberry Pi é possível fazer muitas coisas que é feito com um computador, como, por exemplo, programar, fazer edição de documentos, jogar, navegar na *web* entre outras funcionalidades [41].

A placa Raspberry Pi possui alguns recursos que o computador normal não possui como pinos de entrada e saída (GPIOs). Os pinos GPIOs possibilitam a interação com o mundo real (desenvolvendo grandes projetos) [41].

O Raspberry Pi precisa ser acompanhado de alguns acessórios, se não, será apenas uma placa eletrónica sem muitas funcionalidades. Precisa de pelo menos uma fonte de alimentação e também um cartão MicroSD com pelo menos 8 GB, classe 10, para poder armazenar os arquivos e o sistema operacional do Pi. O Pi por não possuir um disco rígido, então todas as atividades realizadas têm de ser salvas no cartão MicroSD incluindo o sistema operacional [41].



Figura 2.13: Raspberry Pi [40]

## 2.3 Eficiência energética

A eficiência energética é uma área que visa a continuidade da operação na mesma medida em que há redução de custos acionados. Pode-se afirmar que um equipamento é eficiente, exemplificando, quando a bateria tem uma maior duração enquanto gasta menos energia do que dispositivos idênticos [42].

A Internet das Coisas tem uma grande competência de apoiar as organizações a alcançarem a eficiência energética dentro das suas estruturas, é por essa razão que o recurso capaz de otimizar diversos outros processos, está a tornar-se cada vez mais utilizado. Consoante a pesquisa realizada em [42] conforme *Juniper Research* revelou, o número de sensores IoT está a crescer num número considerável, até 2022 esses números devem ultrapassar os 50 bilhões. Em 2018, os objetos conectados tinham uma estimativa de 21 milhões [42].

Segundo Bruno Eduardo Medina [43], a Comissão Europeia tem reconhecido o uso das tecnologias da informação e comunicação como um dos recursos essenciais capazes de aprimorar a eficiência energética dos edifícios. O uso dessas tecnologias tem sido

visto como o principal meio pelo qual os países europeus irão conseguir chegar às metas determinadas para 2020, de entre elas o aumento em 20% da eficiência energética e a redução de 20% na emissão de gases do efeito estufa em relação a 1990 [43].

Em algumas ocorrências, a redução chegou a até 50% do consumo inicial do prédio, como, por exemplo, o projeto *HosPilot*, implementado na cidade de Logrono. O hospital San Pedro adotou medidas estratégicas de detecção de presença e monitoramento de estado das salas, otimização da ventilação com base na detecção de presença e no controlo de CO<sub>2</sub>, tecnologia de iluminação, sensores de iluminação, e ajuste dinâmico do ponto esperado de temperatura com pontos de ajuste de inverno/verão. Todas essas medidas foram capazes de reduzir em 49% o consumo energético [43].

Segundo Pérez [44], em 2004 o consumo de energia nos edifícios residenciais e comerciais, nos Estados Unidos, atingia um valor de 37% do consumo total de energia do país, excedendo outros grandes setores como indústria 28% e transporte 32%, essa taxa foi ainda maior no Reino Unido, alcançando um valor de 39% [44].

Segundo adene [45], o setor da indústria, constitui um dos setores com maiores consumos energéticos e com forte dependência de recursos energéticos não renováveis. O consumo energético do sector da indústria representa cerca de 30% do consumo final de energia do País (Portugal) [45].

De acordo com o último relatório da Mordor Intelligence, a tecnologia de IoT aplicado ao setor de energia alargará nos próximos cinco anos a uma taxa média de 17,24%, chegando a 39.090 milhões de dólares em 2025 [46].

O uso de energia residencial em países desenvolvidos é muito grande e com o passar do tempo tende a crescer cada vez mais, devido ao aumento do uso de eletrodomésticos e também o sistema Aquecimento, Ventilação e Ar Condicionado (AVAC). Essas residências são responsáveis por 22% da energia consumida nos EUA, enquanto, na União Europeia, representam 26% do consumo total [44].

Os sistemas AVAC representam um valor perto dos 40% do consumo de energia de um edifício. Para aproximar do objetivo de eficiência energética para 2030 de (maior ou igual) a 27%, a regulamentação europeia incentiva os edifícios a instalarem sistemas de

controle e monitorização, entre outras medidas [47].

### **Comparação da eficiência energética entre ZigBEE, Wi-Fi e Bluetooth em sistema de estacionamento**

Segundo o Robert Cabral, utilizaram-se simuladores para os testes do sistema e fizeram-se algumas analogias utilizando os protocolos de comunicação, ZigBEE, Wi-Fi e Bluetooth. A Figura 2.14 apresenta o comportamento do sistema utilizando as diversas comunicações. A coluna em azul representa a potência requerida do sistema para cada tipo de comunicação, e a coluna vermelha do alcance de comunicação representa a distância em que cada nó pode estar do outro [48].

Pode-se ver a diferença do Wi-Fi em comparação com as outras comunicações em relação ao consumo de energia, vindo a ser um problema para a vida útil da bateria, já o Bluetooth gasta bem menos que os outros protocolos, porém perde no alcance da comunicação, assim sendo para cada sistema deve-se ser estudado a melhor maneira de fazer a comunicação[48].

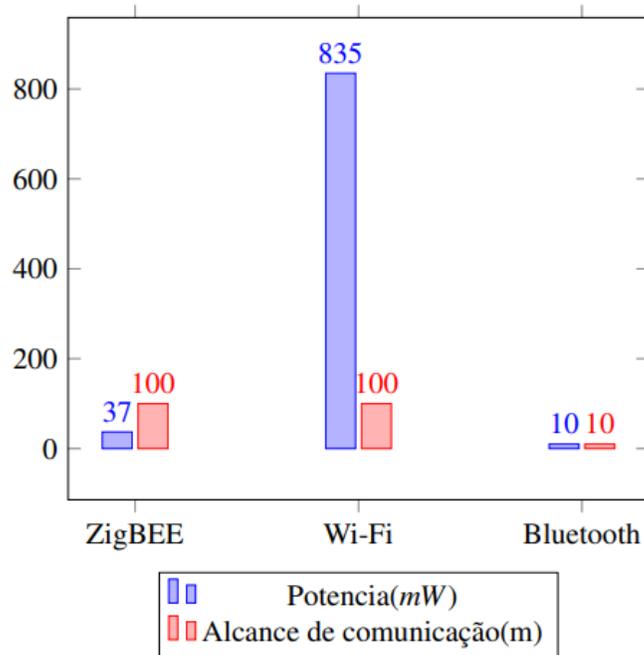


Figura 2.14: Comparação de ZigBEE, Wi-Fi e Bluetooth [48]

Boas práticas que fazem sistemas utilizar menos bateria.



Figura 2.15: Técnicas de eficiência energética em IoT [48]

## 2.4 Aplicações de Tecnologia IoT

### Sistema de automação residencial baseado em MQTT usando ESP8266

Na pesquisa realizada em [49], os autores pretendiam montar um sistema de automação residencial usando o protocolo de comunicação MQTT e o ESP8266. Pretendiam monitorar remotamente as condições dentro de casa e controlar os eletrodomésticos. Devido ao consumo de energia e a largura de banda serem uma preocupação, os autores precisavam de um dispositivo de baixa potência que transmita por meio de um protocolo menos detalhado, entretanto acabaram por escolher o protocolo MQTT. No protótipo, tentaram implementar o MQTT em ESP8266. Conectaram sensores e atuadores no ESP8266, e um corredor MQTT baseado em Mosquitto que foi estabelecido para monitoramento e controle.

Segundo os autores detalharam a implementação do protótipo, na configuração da rede, a intensidade da luz detetada usando o sensor LDR que está conectado na placa de desenvolvimento ESP8266, que por sua vez, processa os dados do sensor e executa a atuação. Atua como um *gateway* para transmissão de dados por Wi-Fi. ESP8266 é configurado como cliente MQTT publicando os dados do sensor para o MQTT Broker e assinatura dos comandos para controlar a atuação. *Light-emitting diode* (LED) e lâmpada foram usados como atuadores no protótipo. O módulo ESP8266 publica os dados do sensor sob o tópico. Segundo os autores, os clientes MQTT, como PCs e dispositivos móveis, podem conectar-se ao servidor MQTT por meio de tecnologias de comunicação como Ethernet, 2G, 3G, Wi-Fi. Na configuração de *software* para programar o módulo ESP8266 utilizaram o Arduíno IDE [49].

### Monitoramento de ambiente de baixo custo usando ESP8266

Segundo as pesquisas realizadas em [49], os autores pretendiam criar um programa para sentir as condições ambientais, como temperatura e humidade. Utilizaram linguagem de alto nível que possui códigos claros e expressivos projetados para responder às ações

(linguagem Micropython). [50]

De acordo com o [50] Micropython é uma implementação muito eficiente de código aberto da linguagem de programação Python. Esta linguagem de programação é transportada para a placa ESP8266, que por sua vez está interligada com o sensor DHT11, para medir as condições ambientais, como a temperatura e humidade. A placa ESP8266 tem duas conexões, uma com o sensor DHT11 para medir a temperatura e humidade, e o segundo com Display OLED para mostrar as informações coletadas de temperatura e humidade do sensor DHT11 na tela e todos são implementados usando a linguagem Micropython. A placa ESP8266 é alimentado por USB, no entanto, o sensor DHT22 e o display tem interface com a placa ESP8266. O sensor DHT22 usa protocolo de comunicação de 1 fio em que o único barramento para a linha de dados troca os dados entre o sensor e a placa ESP8266 para completar a comunicação. O display OLED tem interface com a placa ESP8266 com a aplicação do protocolo *Inter-Integrated Circuit* (I2C) padrão projetado para dispositivos de baixa velocidade.

Segundo o [50], o I2C é o protocolo serial de 2 fios que vem com *Serial Data Line* (SDA) e *Serial Clock Line* (SCL) permitindo a transmissão serial entre o display OLED e o Placa ESP8266 para que um bit seja enviado em cada pulso de *clock*. As condições ambientais como temperatura e humidade são continuamente detetadas pelo sensor DHT22 e as mesmas informações de parâmetros são recuperadas para exibição na tela OLED [50].

### **Sistema de medição sem fio de consumo de água utilizando o protocolo MQTT**

Segundo o [51], desenvolveram uma tese em que o objetivo principal era desenvolver um sistema de comunicação sem fio para medição do consumo de água no empreendimento Sapiens Parque Procederam com pesquisas que envolviam temas que cercavam medição remota de serviços públicos tais como, água, energia e gás, sobre os conceitos e utilização de tecnologias da Internet das Coisas [51].

De acordo com [51], precisavam atender as possíveis perdas no sistema de distribuição apresentando o mérito do acesso às leituras de forma rápida e precisa, a partir daqui pensaram num sistema de monitoramento que deveria fornecer informações em tempo

real para rápida intervenção da equipa de manutenção. Podendo assim ser conformado através de uma rede inteligente de monitoramento remoto, que em posse das informações sobre o consumo o próprio consumidor age de forma preventiva para economia de energia [51].

Para o [51], o uso de tecnologias online é considerado como uma possibilidade para o problema apresentado. Também considerando a evolução dos serviços de armazenamento de dados em nuvem junto com o avanço da tecnologia da Internet das Coisas, a utilização de equipamentos que consigam conversar e dispor as informações para acesso remoto de forma contínua e que possuam baixo consumo de energia, podem ser alternativas para a proposta de solução [51].

A proposta de solução escolhida na dissertação prevê o uso do protocolo MQTT. Também a utilização de um kit de desenvolvimento CC1200DK que está junto a um módulo Wi-Fi ESP8266, e juntos formam o concentrador de dados para envio das informações ao servidor e também outro kit CC1200DK instalado junto a um medidor de água com saída pulsada para coleta e transmissão de dados dos volumes consumidos para o concentrador[51].

Como a medição do líquido só é possível apenas de maneiras indiretas, os equipamentos dispõem um sistema de conversão apoiados, ou na medição da velocidade do fluido, por meio de uma área de seção transversal conhecida ou na adição de unidades de volumes conhecidos em intervalos de tempos (medidores) [51].

O processo de leitura é feito com a medição através de um registador mecânico velocimétrico de jato único que em um dos seus ponteiros possui um pequeno ímã. A cada volta efetuada por este ponteiro (corresponde a medição do valor de um litro) ocorre uma interação com um sensor interruptor acoplado na parte externa do medidor. Esta interação oferece uma condição de curto-circuito nos terminais do cabo conector ligado a uma das portas GPIO38 do módulo CC1200DK e também à terminal terra [51].

Para monitorar os sinais de tensão provenientes do sensor interruptor instalado no medidor de consumo, foi operado a porta P2.5 do módulo CC1200DK, ela está associada

ao botão “*down button*” e com ele é possível simular o sensor interruptor sem necessariamente existir um medidor de água acoplado à placa eletrônica. Neste processo permanece acumulando os dados conseguidos na memória do CC1200DK e, a cada 1 hora, estes dados são transferidos primeiro entre as placas do MSP430 e CC1200 via comunicação SPI39 e desta última, os dados são enviados para o módulo concentrador via rádio frequência. O módulo concentrador faz uma técnica oposta, ao receber os dados por meio da radiofrequência, transfere os dados para o microcontrolador MSP430 (via SPI). A comunicação serial RX/TX é utilizada em conjunto com um protocolo próprio, desenvolvido para a aplicação com o propósito de transferir as informações para o ESP8266 [51].

O ESP8266, atendendo o seu próprio protocolo, os dados serão enviados para o MQTT Broker e as informações serão tratadas corretamente para a visualização do utilizador [51].

Conforme o [51], na implementação do trabalho tiveram algumas dificuldades como o desenvolvimento dos códigos precisou de grande tempo de estudo mesmo saindo de um modelo existente. Também a utilização do ESP8266 mostrou problemas ao ser utilizado para envio dos dados recebidos no terminal Recetor (RX), via comunicação UART/serial do conjunto CC1200DK. O primeiro *Broker* a ser usado nos testes foi o *cloud* MQTT, mas mostrava-se muito instável durante os testes com frequentes quedas de conexão. O motivo era o tempo sem comunicação com o *Broker*. Para contornar este problema utilizou-se o *Broker* do projeto Eclipse que permitia tempos maiores sem comunicação [51].

### **Controlador de luz doméstica baseado em ESP8266 com protocolo MQTT**

Segundo a pesquisa realizada em [52] o objetivo deste trabalho era desenvolver um *design* para controlar equipamentos elétricos (as lâmpadas) de um dado apartamento usando o microcontrolador ESP8266 com o protocolo MQTT. Os autores pretendiam substituir qualquer equipamento convencional para um *switch* sem fio (controlador de luz), onde o utilizador poderia controlar os equipamentos e dentro ou fora da casa automaticamente.

A plataforma de programação ESP8266 era o Arduíno que foi implementado na biblioteca pub/Sub *Client* para comunicar no protocolo MQTT. Com o protocolo MQTT o sistema era criado dando ordens através do tópico “/ order / (ruang)” e para ver os

status das luzes utilizou-se o tópico `"/ status / (ruang)"`, também alguns métodos de comando e status foi aplicado ao aplicativo *Android* "Kontrol Rumah MQTT" para que a sua utilização possa ser mais acessível.

O controlador de luz tinha de estar conectado à Internet, porque o Broker MQTT é usado online.

O controlador de lâmpada ESP8266 pode comandar a luz do botão de toque / sensor, do comando *Android SmartPhone* e do cronograma. Se o botão for tocado, as luzes anteriormente desligadas acenderão, e vice-versa, então o módulo ESP8266 dará status para *SmartPhone* para que a tela mude de acordo com as condições da lâmpada. *SmartPhone* instalado de aplicativo pode controlar as luzes no ESP8266, definir a configuração do ESP8266 e exibir a programação tempo que pode ser definida pelo utilizador [52].

Segundo o [52], o *design* de um programa *Android* para controlar o ESP8266 foi feito com um aplicativo *Android Studio* com a linguagem Java. A comunicação entre o *SmartPhone* e o ESP8266 ocorre onde o *SmartPhone* publicará o tópico com uma carga útil de "0" ou "1". Entrega de carga útil "0" ou "1" dependendo da condição da lâmpada. Se a luz estiver acesa, a carga útil "0" é enviada em ordem ESP8266 apaga as luzes e vice-versa.

Conforme o [52], durante a realização do trabalho realizaram alguns testes do sistema para descobrir se o sistema poderia funcionar como pretendido. O teste do sistema foi dividido em várias partes, como: O teste do dispositivo ESP8266 com MQTT Box; Testando o dispositivo ESP8266 com o aplicativo "*MQTT Home Control*"; Testar o tempo de resposta do controlador ESP8266 (obtiveram o tempo de resposta de 0,0832 segundos) e testar a programação da lâmpada no ESP8266. Os resultados deste teste são monitorados pela observação do estado das luzes de acordo com o horário programado. E os autores concluíram o trabalho com sucesso [52].

### Abordagem orientada para a Internet das Coisas para monitoramento e controlo da concessionária de água

Este é um trabalho desenvolvido pela [29], onde o objetivo era propor uma abordagem mais eficiente de monitoramento e controlo distribuído de água de modo a reduzir a perda de água atual e ajudar os operadores de serviços públicos a melhorar a água e o sistema de gestão. O monitoramento do consumo de água pode ser usado para ensinar os cidadãos como reduzir custos e recursos.

Consoante [29], o sistema de abastecimento de água inclui diferentes níveis como equipamento de campo, controlo de processo, gestão e aplicações e o seu *design* adequado pode ser visto na forma como garante a interoperabilidade de controlo, funções em todos os seus níveis de componentes. Para monitorar os dados de consumo de água, é necessário que esses contadores forneçam dados de saída específicos para o processo de medição.

Para monitoramento das concessionárias de água, utilizaram o *Supervisory Control and Data Acquisition* (SCADA). O principal objetivo dos sistemas SCADA é monitorar e controlar diferentes equipamentos e dispositivos. O sistema SCADA são usados numa ampla gama de processos como [29]:

- Processos de infraestrutura (petróleo e gás oleodutos, gestão de água, gestão de resíduos, transmissão de energia elétrica e distribuição, etc.);
- Processos industriais (manufatura, refino, geração de energia, etc.);
- processos de instalações (edifícios, terminais de autocarro, aeroportos, etc.).

Também utilizaram a tecnologia de identificação por radiofrequência (RFID) utilizando sensores junto aos pontos de consumo, que é considerada para identificar coisas de forma única (dispositivos, sensores, etc. específicos para monitoramento de utilidade de água e controlo), que deve ser conectado à Internet das Coisas.

A tecnologia de identificação por radiofrequência RFID segundo [29], é considerada para este trabalho baseado em IoT abordagem para monitoramento de utilidade de água

e controlo, provando uma solução de baixo custo para a identificação única de coisas que deveriam ser conectados à Internet das Coisas.

O uso desta tecnologia é justificado por algumas das características principais e funcionalidades dos sistemas RFID como [29]:

- Atingindo custos baixos e eficiência energética;
- Sem contato e sem linha de visão, funcionalidades que permitem o acesso a dados em condições ambientes e através de várias substâncias;
- Permitindo a integração de tecnologia RFID com sensor.

Da mesma forma fizeram o uso de tecnologias multiagente para garantir a troca de informações entre coisas diferentes.

Um agente é um componente de *software* que tem uma função bem definida na operação de um sistema. Além disso, um agente deve ter a capacidade de se comunicar com outros agentes ou utilizadores humanos. Um sistema multiagente é uma coleção dessas entidades que cooperam com uns aos outros [29].

Apresentaram uma arquitetura baseada em IoT para monitoramento e controlo de concessionárias de água onde foi dividida em vários níveis:

- **nível de sensor** - O nível do sensor inclui diferentes dispositivos de campo, alguns deles dotados de computação e capacidades de comunicação. Alguns dispositivos podem agir como atuadores e com base nas condições locais detetadas ou um comando remoto pode iniciar ações adequadas.
- **nível de comunicação** - É muito importante para IoT. Este nível inclui fio (xDSL, fibras óticas) e redes móveis (3G, 3G +, GPRS). Neste os aspetos de nível de segurança e privacidade devem ser tratados.
- **gerenciamento** - Inclui *software* inteligente, instrumentos usados para gerir, controlar e operar dispositivos IoT. Pode incluir estruturas, monitoramento e controlo de água, etc.

- **nível de utilizador** - Consiste em diferentes dispositivos usados por utilizador em nível de utilizador.

A proposta de solução deste trabalho prevê a utilização de um ambiente de supervisão e aquisição de dados num ambiente online, onde inicialmente estarão disponíveis informações básicas de consumo por unidade e posteriormente os valores globais do empreendimento[29].

## 2.5 Definição do problema

A agricultura é a atividade humana que gasta mais água. Num contexto de alterações climáticas no qual os períodos de seca são cada vez mais frequentes e mais severos, torna-se necessário aplicar medidas de mitigação ao nível da utilização da água. Assim é importante que os recursos água e energia sejam utilizados de forma mais sustentável e eficiente. Desta forma pretende-se desenvolver uma solução de IoT numa zona agrícola garantindo uma boa produção dos produtos, economizando tempo dos proprietários, melhorando as receitas e controlando todo o sistema de rega.

A solução deve ser adaptável a diferentes áreas agrícolas, designadamente a zonas agrícolas com grande dimensão e divididos por estufas, em que cada estufa tem um tipo de plantação diferente. Precisa de alguns cuidados com as plantações para que o utilizador possa ter bons rendimentos, ou seja, deste modo não será gasta a água de forma desnecessária nas regas, controlar a temperatura nas plantações conforme o tipo de plantação e controlar a ventilação possibilitando a circulação do ar dentro da estufa, tudo isso de forma fácil e prático.

Neste contexto, pretende-se desenvolver um sistema de automatização que possibilita o utilizador da aplicação controlar todo o sistema a distância. Neste controlo deve ser possível saber as temperaturas dentro de todas as estufa em tempo real e também deve ser possível controlá-lo, ou seja, se a temperatura dentro da estufa está muito alto, o utilizador pode abrir as áreas de ventilação, caso contrário, o utilizador deverá fechar as áreas de ventilação e controlar a temperatura de modo a deixar as plantações a uma temperatura

muito confortável. Este sistema também deve informar o utilizador o nível de humidade no terreno e a partir desta informação, estará com as informações necessárias para controlar os aspersores de rega. Para além das iluminações para ajudar no desenvolvimento das plantações também pretende-se desenvolver outro sistema de iluminação para poder fornecer ao utilizador o conforto de visitar as suas plantações a hora que desejar.

# Capítulo 3

## Sistema de rega eficiente

Neste capítulo, após da descrição do caso de estudo será apresentado a solução proposta para o problema, centrando-se na apresentação da arquitetura, também serão discutidas as soluções quer ao nível do *hardware*, quer ao nível das comunicações.

### 3.1 Caso de estudo

No contexto do problema definido no capítulo anterior, o caso de estudo consiste na existência de três estufas agrícolas. Cada estufa é provida de capacidade de controlo da luminosidade, da humidade e de temperatura para criar condições propícias à produção agrícola.

Consideram-se três culturas (cenoura, tomate e morango) que podem ser realizadas em qualquer uma das estufas. Estes três cultivos necessitam de cuidados diferentes, no caso da cultura de cenoura, requer uma temperatura ideal definida no intervalo de  $[15, 21]^{\circ}\text{C}$  para poderem ter um bom desenvolvimento, a temperatura pode ser fornecida pela luz do dia ou pelas luzes artificiais em que podem ser ajustados até obter a temperatura desejada. Para controlar o valor da humidade do solo adequado no caso da produção das cenouras é necessário usar o sistema de rega (regar 3 vezes por semana) para melhorar a qualidade das raízes das mesmas.

Quanto à cultura de tomate a temperatura ideal durante o dia tem de estar entre  $[19, 21]^{\circ}\text{C}$  e  $[12, 14]^{\circ}\text{C}$  durante a noite, para a humidade o valor recomendado tem de estar num intervalo  $[75, 85]\%$  no período diurno e noturno. Relativamente a iluminação é exigida no mínimo 6 horas de luz solar, visto que o tomate é muito sensível à baixa iluminação e que também pode rachar no caso de intensidade luminosa muito alta.

Por fim, na cultura de morango, tem de ser fornecido uma temperatura máxima de  $27^{\circ}\text{C}$  e relativamente à humidade, pode ser controlada com a irrigação não deixando o valor do mesmo muito alto porque pode causar a criação de mofo que afeta a produção.

## 3.2 Solução proposta

Este projeto visa controlar as três estufas que são: a produção de cenoura, tomate e morango conforme referido anteriormente. Para isso, é necessário a implementação de uma lógica de controlo de rega inteligente que requer o conhecimento de parâmetros importantes como o caso da temperatura, a humidade e a luminosidade, para se poder decidir consoante as soluções mais ecológicas (poupança de água) e eficazes. Dado que um sistema de rega é um sistema distribuído que envolve pontos de rega (aspersores), aquisição de grandezas físicas, é proposta uma arquitetura modular constituída por um conjunto de módulos com funções específicas que se comunicam entre si.

Na figura 3.1 está representada a arquitetura proposta, composta por um conjunto de módulos que comunicam entre si.

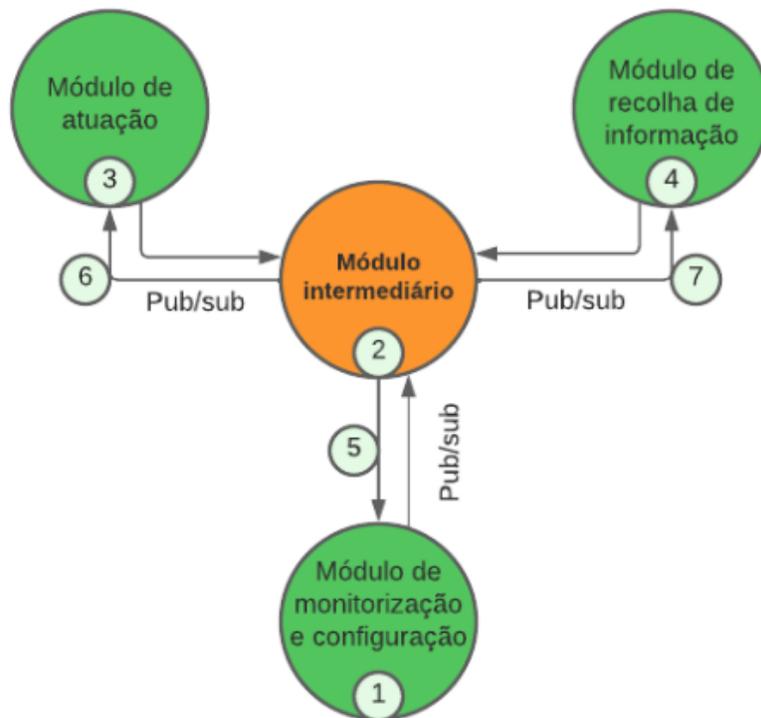


Figura 3.1: Arquitetura da solução.

### 3.3 Módulo de monitorização e configuração (1)

Este é o módulo que efetua a configuração de todo o sistema. Através do qual são definidos os *setpoints* de temperatura, humidade e de todos os parâmetros importantes para uma dada cultura, permite configurar uma estufa ou escalar o seu número em caso de necessidade. Através deste módulo o utilizador consegue monitorizar o estado do sistema. É suportado por uma aplicação móvel usada pelo utilizador da aplicação, agrega as funções de configuração e de monitorização.

#### 3.3.1 Interface gráfica (*dashboard*)

O *dashboard* desempenha um papel muito importante no controlo de rega na zona agrícola dentro e fora da estufa, ele consegue controlar os módulos de atuação conforme as regras de rega implementadas para o sistema agrícola e faz todo o trabalho sem precisar

de interação humana direta, ou seja, conforme as grandezas convertidas pelo sensor para serem medidas (temperatura e humidade), essas mesmas grandezas são requisitadas pelo módulo de monitorização e configuração (*dashboard*). Neste contexto, o *dashboard* necessita de ligar ou desligar os aspersores de rega, ligar ou desligar o sistema de aquecimento com base nas grandezas que foram requisitados por ele mesmo e controlar a iluminação utilizando sensores, tudo de uma forma autónoma.

Para que o sistema seja configurado é preciso do utilizador para fazer as configurações para a monitorização automática. Também, visto que no *dashbord* existem *slider* e botões para controlar o sistema de aquecimento e controlar a iluminação, o utilizador da aplicação também consegue controlar o sistema agrícola apesar de não ter essa necessidade visto que o *dashboard* pode executar todo a processo.

Foi desenvolvido uma interface simples que possibilita qualquer utilizador comum a compreender a leitura de dados e o funcionamento. Neste enquadramento, escolheu-se o JavaScript para desenvolver a interface visto que através do tal consegue-se obter vários tipos de interfaces simples e modernas em que poderão ser criadas conforme as necessidades do projeto.

JavaScript é uma linguagem de programação de utilização geral em que é bastante utilizado para desenvolvimento de *software* e desenvolvimento de *web*. Pormenorizando, o JavaScript opera mais na programação de *front-end*, ou seja, na parte visual dos *sites* ou aplicação. Neste contexto o JavaScript é utilizado com CSS e HTML que são a parte estática, o CSS serve para dar estilo, HTML é para inserir os conteúdos e o JavaScript é responsável pela parte da interação da interface.

A figura 3.2 mostra uma interface, criada no JavaScript direcionado para este projeto em que é muito simples e moderno. Esta interface pode ser manuseada por qualquer pessoa familiarizada ou não com a tecnologia de IoT, ela permite controlar a iluminação, controlar a temperatura e fazer leitura dos dados da temperatura e humidade.

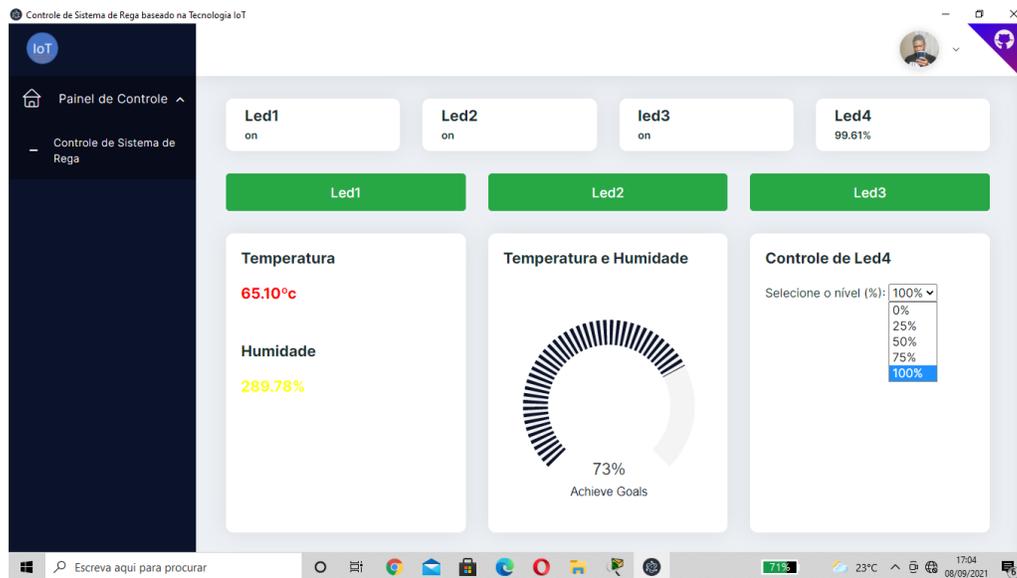


Figura 3.2: Interface.

Para ajudar o utilizador com as informações a respeito dos dados da temperatura e da humidade, os indicadores da temperatura e da humidade foram programados de modo a apresentarem várias cores de acordo com um intervalo de valores. Para o indicador da temperatura é apresentado uma cor amarela no intervalo de 0°C até 10°C, cor azul num intervalo de 11°C até 30°C e cor vermelha para uma faixa de 31°C até 70°C. A cor amarela é para alertar temperaturas muito baixas, a cor azul para indicar temperaturas favoráveis e a cor vermelha para alertar temperatura muito alta. Para o indicador da humidade usou-se a cor azul para a faixa de 0% até 33%, a cor verde para uma faixa de 34% até 66% e por fim cor amarelo para uma faixa de 67% até 100%.

### 3.4 Módulo intermediário (2)

O módulo intermediário é o componente central da arquitetura proposta que coordena a comunicação entre os módulos que integram. Recorrendo a este módulo é possível esconder a constituição da arquitetura aos módulos que a constituem. Assim, permite a escalabilidade da arquitetura de forma simples. Funciona como um repositório de informação ao qual os módulos que queiram partilhar informação a enviam e os módulos que

pretendem aceder a um tipo de informação, solicitam.

Foi utilizado um Raspberry Pi e instalou-se Broker mosquitto no Raspberry Pi para obter o módulo intermediário conforme apresentado na solução proposta, deste modo foi realizado algumas configurações como:

Primeiramente, deve ser definido um endereço IP fixo para o Raspberry Pi, com a finalidade dos outros módulos presentes na arquitetura, possam ver o Broker que está no Raspberry Pi. Para que isso seja possível, há dois caminhos a seguir. Ver qual endereço IP que o router vai atribuir para o Raspberry Pi ou pode ser feito através do comando “ifconfig” na linha de comando do Raspberry Pi.

Para que o Raspberry Pi disponibilize um ambiente para fazer as configurações necessárias para definir o IP fixo do Raspberry Pi, recorreu-se ao comando "sudo nano/etc/dhcpd.conf". Após isso, insere-se os dados abaixo e pode ser guardado usando o comando (ctrl + x):

- “interface wlan0”
- “static ip\_address=192.168.\*.\*”
- “static router = 192.168.\*.\*”

Antes de ser instalado e configurado o MQTT Broker mosquitto no Raspberry Pi é muito importante fazer uma atualização geral, em que pode ser utilizado o comando “sudo apt-get upgrade”. Em seguida, começa o processo de instalação do Broker no qual recorreu-se ao comando “sudo apt-get install mosquitto -y”.

Posteriormente, vai ser instalado o cliente do mosquitto responsável por realizar o processo de *publish* e *subscriber*, onde foi operado com o comando “sudo apt-get install mosquitto-clients -y”.

Por fim, poderá ser feito o nível de autenticação, criando o nome do utilizador e a senha, impedindo que ninguém faça *publish* ou *subscriber* sem estar autorizado para tal.

## 3.5 Módulo de atuação (3)

Os módulos de atuação desempenham funções de regulação, como por exemplo atuação nos aspersores de rega, controlo do sistema de temperatura e controlo do sistema de iluminação. Para a atuação no sistema de rega, o módulo de atuação estabelece a comunicação com o módulo intermediário para obter os parâmetros de rega. O módulo intermediário por sua vez também realiza a comunicação com o módulo de configuração e monitorização para obter os parâmetros de rega com base nos dados obtidos pelo módulo de recolha de informação e enviar para o módulo de atuação para que possa controlar a humidade. Para o controlo da temperatura e iluminação o processo é semelhante.

O módulo de monitorização e configuração ao comunicar com o módulo intermediário publicando informação de monitorização desejado, seja para controlar a iluminação, seja a controlar a temperatura ou controlar os aspersores de rega de uma estufa em específico, o módulo intermediário vai comunicar com o módulo de atuação para executar informação de monitorização desejado.

O módulo de atuação é responsável por atuar sobre o sistema de iluminação, temperatura e aspersores de rega em todas as estufas, mas de forma separada, visto que existem três estufas com plantas diferentes, que necessitam de cuidados diferentes.

### 3.5.1 Controlo da iluminação e a temperatura

O fluxograma representado na figura 3.3, mostra o modo como decorre a comunicação entre os módulos (módulo de monitorização e configuração, módulo intermediário e módulo de atuação) sendo os responsáveis para o controlo da iluminação, dando uma boa visibilidade ao utilizador, permitindo que visite as estufas a hora que desejar e também são responsáveis por controlar a temperatura e os aspersores de rega das estufas aumentando a produtividade.

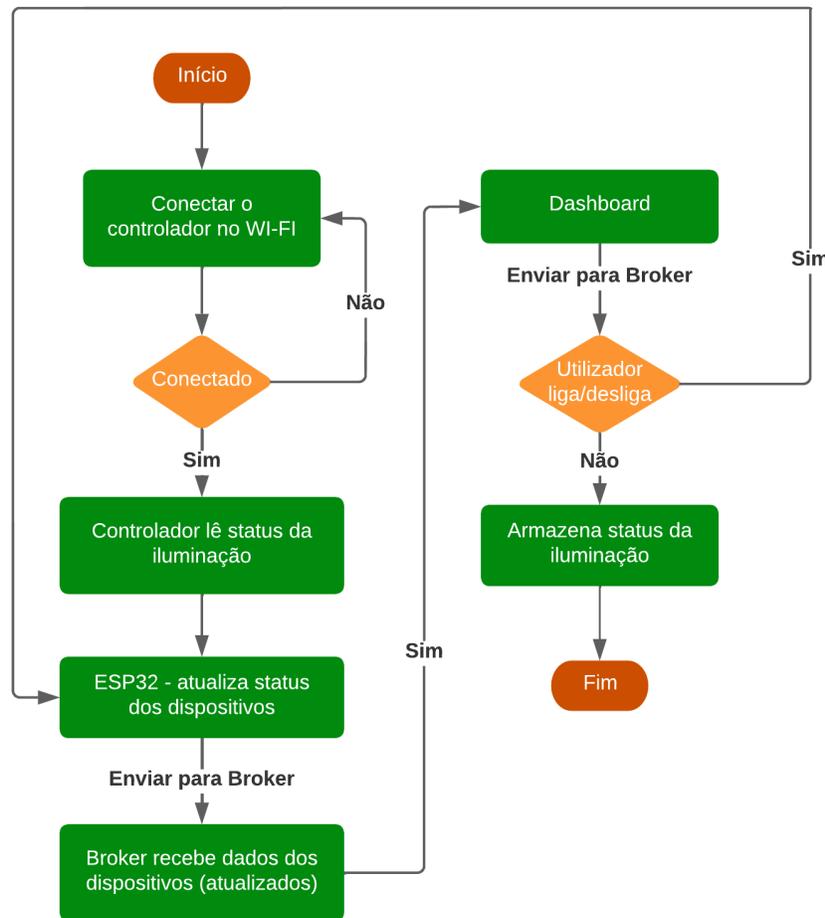


Figura 3.3: Fluxograma para leitura de status da iluminação.

Para controlar a iluminação, o módulo atuador lê os estados da iluminação, em seguida publica as informações para o módulo intermediário, que, vai publicar a mesma informação para o módulo de monitorização, deixando o utilizador com a informação no que diz respeito ao estado na iluminação.

Caso o utilizador deseje mudar o estado na iluminação, será enviado uma informação de monitorização através do módulo de monitorização (*dashboard*), que fará o mesmo caminho (passa para o módulo intermediário, depois para o módulo de atuação) ao chegar no módulo de atuação é executado o comando que foi enviado. Por fim, o módulo de atuação envia a informação do novo estado da iluminação para o módulo de monitorização

(faz o caminho inverso) para informar o utilizador do novo estado na iluminação.

Relativamente ao *dashboard* destinado para controlar a iluminação, foram criados três botões em que é possível mudar de posição conforme programado pelo utilizador, também foram criados três modos de sinalização onde muda o nome para *on* ou *off* consoante a posição do botão.

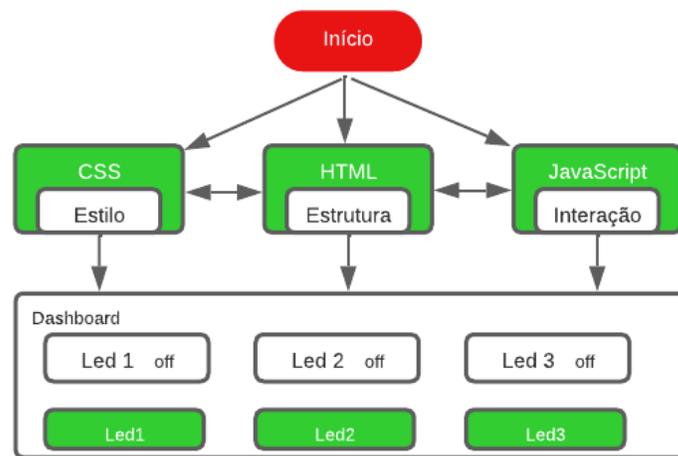


Figura 3.4: *Dashboard* controlo "on" ou "off".

Para controlar a temperatura todos os processos de comunicação entre os módulos são praticamente iguais ao controlo da iluminação, exceto no módulo de monitorização (*dashboard*). No módulo de monitorização é disponibilizado um *slider* que permite regular a intensidade da iluminação no intervalo de 0 a 255, ao regular a intensidade da iluminação regula a temperatura conforme a necessidade das plantações perante cada estufa aumentando assim a produtividade.

Na criação do *dashboard* para controlar o *dimming* começou por instalar o pacote *pigpio* no terminal, em seguida começou-se a programação do *dashboard* definindo a variável duty Cycle num intervalo de 0 a 255, ou seja, com o valor de duty Cycle = 0 a iluminação deve estar apagada e com o valor de 255 a iluminação deverá estar com o máximo do brilho e o duty Cycle poderá ser variado neste intervalo [0, 255] variando a intensidade luminosa.

O *dashboard* criado foi uma aplicação em que poderá ser transferido e instalado em qualquer dispositivo. Para o controlo do *dimming* foi criado cinco botões para a variação do duty Cycle [0, 25, 50, 75, 100], consoante o valor do duty Cycle variado pelos botões é apresentado o respetivo valor em percentagem na iluminação.

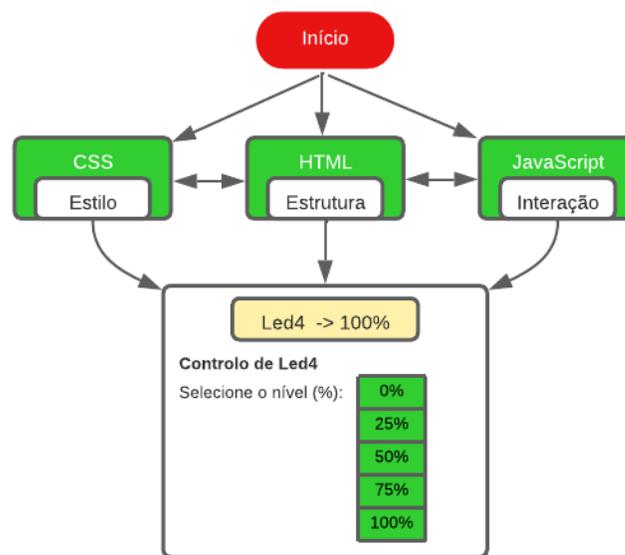


Figura 3.5: *Dashboard dimming*.

### 3.5.2 Controlo da iluminação e da temperatura

Em concordância com a imagem 3.6, foi montado uma estrutura, onde é possível controlar a iluminação e o sistema de aquecimento. O controlo decorre da seguinte forma:

No *dashboard* criaram-se botões para controlar a iluminação e, para controlar a temperatura, foi criado um *slider* para desempenhar esta função.

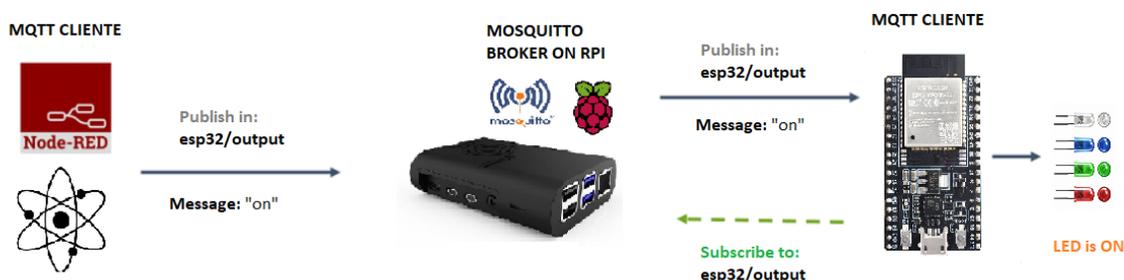


Figura 3.6: Sistema para controlar a iluminação.

Para que o utilizador esteja sempre bem informado no que diz respeito ao estado na iluminação, o *dashboard* solicita todos os dados do último estado da iluminação e guarda essas informações para o utilizador de modo a representar o que acontece.

Para controlar o estado da iluminação conforme o modelo publicador subscritor, começa-se por alterar a posição do botão no *dashboard*, deste modo o *dashboard* publica uma informação de monitorização (para que os outros módulos possam saber e usar) para o módulo intermediário, ou seja, no *Broker* que por sua vez irá centralizar a mesma informação. Em seguida o *Broker* vai publicar a informação de monitorização para o módulo de atuação que necessita de saber as variáveis de processo de controlo, logo terá de subscrever essa informação e por fim o módulo de atuação vai executar a informação enviada.

Para que o utilizador esteja informado da execução da informação de monitorização, o *Broker* vai fazer um subscrito no módulo de atuação de modo a ter a nova informação do estado do aparelho controlado, em seguida o *dashboard* faz um subscrito no *Broker* que já tem disponível a informação do novo estado do aparelho e apresenta para o utilizador deixando-o informado.

### 3.6 Módulo de recolha de informação (4)

O módulo de recolha de informação é um módulo que permite que o utilizador esteja informado dos dados da temperatura e da humidade dentro de cada estufa. Para isso é estabelecido a comunicação entre o módulo de monitorização e configuração, módulo intermediário e o módulo de recolha de informação. A figura 3.7 apresenta um fluxograma que mostra como decorre a comunicação entre os módulos para obter os dados da temperatura e da humidade.

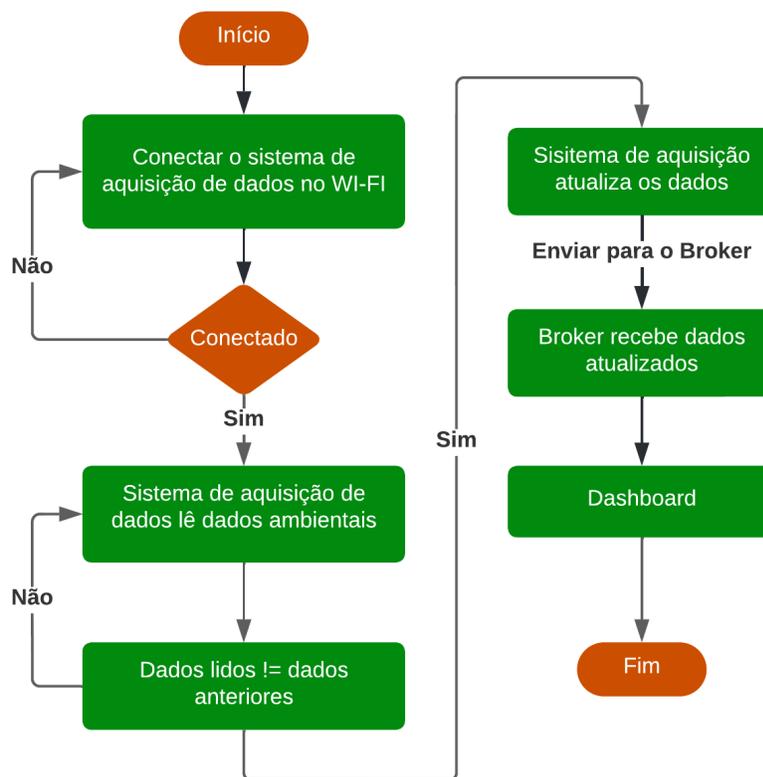


Figura 3.7: Fluxograma para leitura de dados no sensor.

Conforme representado na figura 3.7, para que o utilizador possa ter os dados à sua disponibilidade, no que diz respeito à temperatura e humidade, a comunicação decorre da seguinte forma:

Os dados primários ao serem convertidos pelo sensor é lido pelo módulo de recolha de informação, ou seja, pelo sistema de aquisição da grandeza, em seguida os dados da

temperatura e humidade adquiridos são analisadas pelo módulo de recolha de informação, se for igual aos dados anteriores, é descartado e será feito uma nova leitura, caso contrário, se as leituras dos dados da temperatura e humidade diferirem dos dados anteriores o sistema de aquisição de grandezas atualiza os dados e publica-os no módulo intermediário. Com os dados no módulo intermediário, serão publicados para o módulo de monitorização e configuração que está devidamente organizado (*dashboard*) para receber os dados e apresentar para o utilizador.

### 3.6.1 Leitura de temperatura e humidade

A figura 3.8 mostra como decorre o funcionamento da obtenção dos dados de temperatura e humidade até ser disponibilizado no *dashboard*:

O sensor visa converter as grandezas primárias (temperatura e humidade) dentro das estufas (plantação das cenouras, morangos e tomates), em seguida o módulo de recolha de informação faz uma requisição no sensor para obter os dados disponíveis no sensor. Este módulo, publica os dados no módulo intermediário (*Broker*). As informações disponíveis no *Broker* serão publicadas no módulo de monitorização (*dashboard*). Estando os dados no *dashboard* pode ser acedido pelo utilizador.

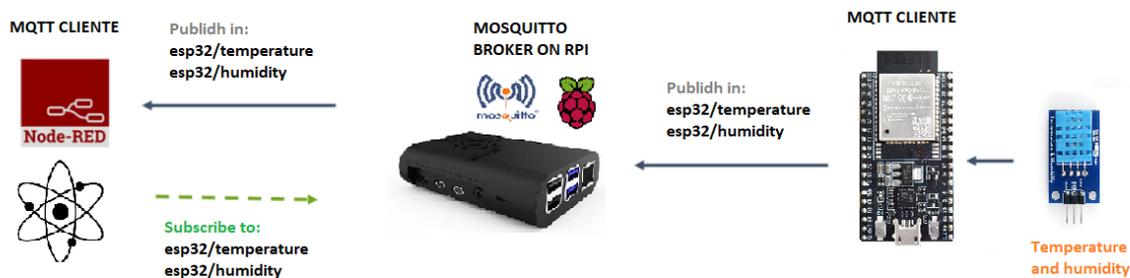


Figura 3.8: Sistema para a leitura de temperatura e humidade.

### 3.7 Partilha de informação entre os módulos (5, 6, 7)

A partilha de informação entre os módulos que também pode ser designado como troca de informação entre os módulos é a comunicação de dados que é efetuado através de uma arquitetura publicador/subscritor (pub/sub) sob o protocolo TCP/IP, que permite a comunicação bilateral entre o *dashboard*, o módulo central e os outros módulos. É um protocolo de troca de mensagens M2M (*Machine to Machine*).

### 3.8 Descrição do sistema

Para controlar todas as grandezas nas estufas, foi utilizado o protocolo MQTT para efetuar a comunicação de dados.

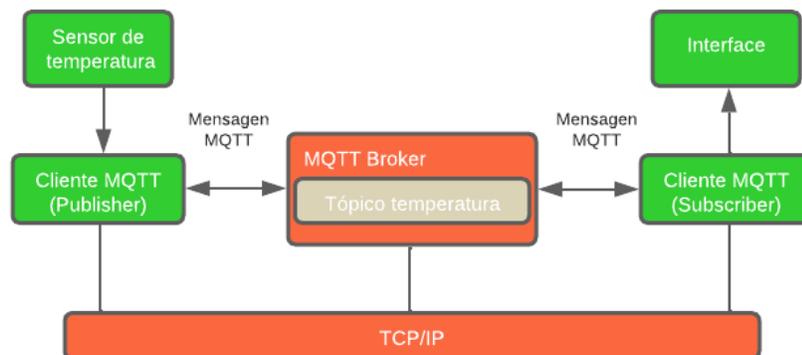


Figura 3.9: Protocolo de comunicação.

O MQTT é um protocolo de troca de mensagens entre máquinas, em que utiliza uma arquitetura chamada pub/sub (*publish/subscribe*) onde todas as mensagens são realizadas por tópicos. Todas as mensagens publicadas ou subscritas passam para o *Broker* que é um servidor central, o *Broker*, consegue pegar nas mensagens disponibilizadas e redistribuir para os clientes desejados. Os clientes são destinados às máquinas conectadas neste protocolo como, por exemplo, sensores, *dashboard*, etc. O protocolo MQTT suporta uma

grande quantidade de mensagens (tópicos) e também dispositivos conectados em simultâneo no *Broker*.

Na arquitetura deste projeto, todos os módulos presentes são tanto *publishers* como *subscribers* visto que todos tem a necessidade de estabelecer comunicações entre si, publicando e subscrevendo tópicos.

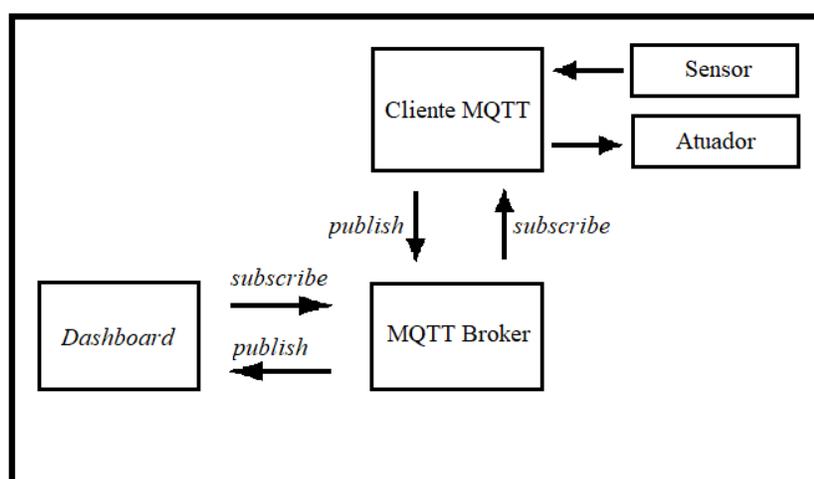


Figura 3.10: Diagrama de blocos do sistema proposto.

A comunicação no protocolo MQTT é realizada por tópicos, e para isso foram criados tópicos para controlar todo o sistema de rega a ser desenvolvido, controlando cada *item* conforme mostrado na tabela 3.1.

Tópicos	Funções
esp32/output1	Controlar iluminação 1 “on” ou “off”
esp32/output2	Controlar iluminação 2 “on” ou “off”
esp32/output3	Controlar iluminação 3 “on” ou “off”
esp32/output4	Controlar nível de iluminação 4
esp32/temperature	Fazer leitura da temperatura
esp32/humitidy	Fazer a leitura da humidade

Tabela 3.1: Tópicos.

Foi utilizado o nível de QoS 2 (qualidade do serviço), que é o nível de qualidade que garante que a mensagem seja entregue uma única vez, ou seja, ele envia a mensagem para o *Broker* e a mensagem será reservada, eventualmente caso seja necessário fazer o reenvio da mensagem, estará disponível. Este nível de qualidade é o mais seguro visto que garante que a mensagem seja enviada pelos dois lados e também possibilita ter a confirmação nos dois lados (*Broker*/publicador ou *Broker*/subscritores).

Em princípio a leitura dos dados dos sensores, controlo dos atuadores e de sistema de iluminação seria desenvolvido num lugar apropriado (estufa) e com materiais apropriados, entretanto, por falta de materiais necessários para a implementação desejado, optou-se por desenvolver todo o sistema num pequeno circuito que foi montado numa placa conforme mostrado na figura 3.11 sendo utilizado LEDs em vez de lâmpadas.

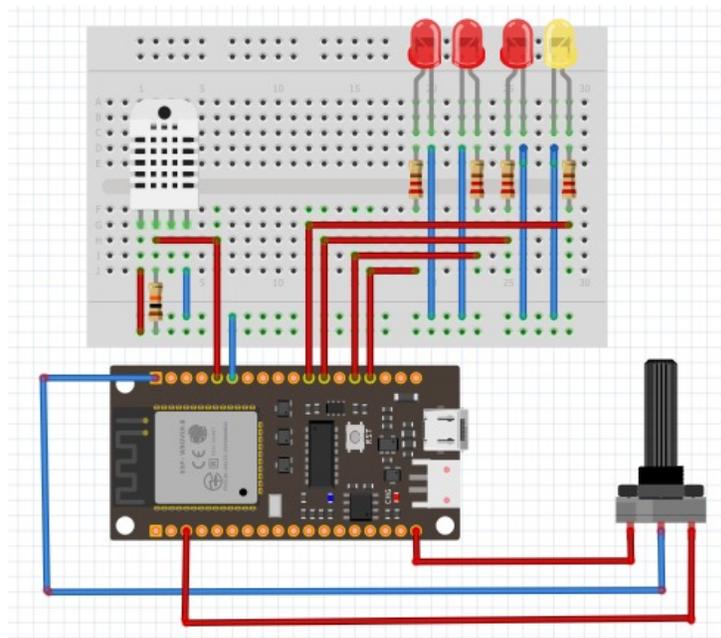


Figura 3.11: Implementação do sistema.

Foi implementado três iluminações vermelhas para serem controlados com o comando *on/off*, uma iluminação amarela onde será controlado com o comando *dimming* que neste enquadramento serve para controlar a temperatura e por fim um sensor para fazer a

leitura de temperatura e humidade.



# Capítulo 4

## Conclusão e Trabalho Futuro

A agricultura é atividade que mais água consome a nível mundial, sendo a água um recurso finito e com as mudanças climáticas se torna cada vez mais escasso. Neste contexto é necessário usar soluções para uma utilização mais eficaz e mais sustentável deste recurso vital.

A importância de implantação de redes IoT no âmbito da agricultura sustentável de forma a alcançar os resultados desejados em um curto espaço de tempo e conservar recursos por uma combinação de dispositivos eficientes e de baixo custo.

A partir deste trabalho, foi possível desenvolver uma solução para a área da agricultura usando o protocolo MQTT incluindo a utilização da interface HTML, Mosquitto *Broker*, microcontroladores ESP8266, ESP32 e por fim sensores e atuadores.

Durante a implementação do trabalho consegue-se compreender as vantagens do protocolo MQTT na prática, como por exemplo a sua leveza, flexibilidade e fácil de implementar. A troca de mensagem neste protocolo (tópicos) é muito fácil, e também possibilita a utilização de microcontroladores diferentes e permite a comunicação entre eles.

A interface desenvolvida em JavaScript também provou ser satisfatória para o sistema utilizado neste trabalho. Com a sua programação, aplicação desenvolvida no contexto da área agrícola apresentou-se uma interação muito boa com o protocolo MQTT. Envia

e recebe dados pela Internet gerando valores da variação da temperatura mudando de cores de acordo com o intervalo de valores apresentado (apresenta uma interface muito moderna). E também constatou-se que a sua programação está num nível normal (em termos de complexidade).

Entretanto, durante o desenvolvimento da presente dissertação, foram concluídos todos os objetivos propostos com sucesso. Em contrapartida, apareceram alguns problemas durante o desenvolvimento do código para a criação da interface para controle dos componentes.

## 4.1 Trabalho Futuro

Por último, tendo concluído todos os desenvolvimentos e implementações do projeto para controlo de temperatura, leitura de temperatura e humidade e controlo de iluminação, este poderá ser utilizado em edifícios (restaurantes, hotéis) e também em estabelecimento para criação de gado. Posteriormente, serão apresentados os exemplos das situações propostas.

### 4.1.1 Proposta 1

Nos edifícios como restaurantes e hotéis, este projeto pode ser muito bem-adaptado e complementado de acordo com a necessidade de cada compartimento. Visto que pode ser reutilizado outros tipos de sensores para coletar dados de temperatura ou coletar dados a partir da Internet em seguida controlar a temperatura nos compartimentos como por exemplo quartos dos hotéis. E no controlo da iluminação, que é muito importante e têm uma fácil adaptação visto que pode ser necessário a sua implementação em todos os compartimentos e é crucial nas zonas das refeições (restaurante) sendo que pode ser controlado o nível da iluminação do espaço de acordo com a luz que entra no local.

### **4.1.2 Proposta 2**

Para a criação de gados é muito importante ter uma temperatura controlada de acordo com o tipo a animal presente um determinado estabelecimento dado que é de extrema importância na reprodução dos animais. E o controle da iluminação também tem as suas vantagens no crescimento dos animais, sendo assim este projeto pode ser adaptado e complementado para essa proposta. E para o acasalamento de animais de grande porte podem ser utilizados atuadores para abrir as portas e a grande vantagem disso é que as pessoas não correm riscos de ter lesões causadas pelos animais.

# Bibliografia

- [1] *GCF global Internet das coisas*, <https://edu.gcfglobal.org/pt/dicas-tecnologicas/o-que-e-internet-das-coisas/1/>, Accessed: 2021-03-06.
- [2] R. M. R. Felipe Canal Mantovani, «internet das coisas cliente em tempo real,» rel. téc., dez. de 2018.
- [3] L. L. N. L. Blaise Barney, «POSIX Threads Programming,» rel. téc., ago. de 2012.
- [4] t. L. Marco Soudo Cunha, «Domotic Module for the Internet-of-Things,» rel. téc., mai. de 2017.
- [5] *youtube IoT conserto geral*, <https://www.youtube.com/watch?v=PDfns7a9X2w>, Accessed: 2021-03-14.
- [6] *tridens technology all about the internet of thing (IoT)*, <https://tridens technology.com/all-about-the-internet-of-things-iot/>, Accessed: 2021-03-14.
- [7] *Ncontrol RFID radio frequency identification*, <https://www.ncontrol.com.pt/o-que-e-rfid.html>, Accessed: 2021-03-06.
- [8] *faqtos sistema GSM*, <https://faqtos.pt/faq-items/o-que-e-o-sistema-gsm/>, Accessed: 2021-03-07.
- [9] *oficina da net O que é GSM e como funciona*, [https://www.oficinadanet.com.br/artigo/733/gsm\\_o\\_que\\_e\\_e\\_como\\_funciona](https://www.oficinadanet.com.br/artigo/733/gsm_o_que_e_e_como_funciona), Accessed: 2021-03-06.
- [10] *segurança eletrônica Bluetooth Low Energy*, <https://revistasegurancaeletronica.com.br/bluetooth-low-energy-o-que-essa-tecnologia-nos-reserva-para-o-futuro/>, Accessed: 2021-03-07.

- [11] *The internet oh things* Modelo bluetooth de baixa energia, Accessed: 2021-03-25.
- [12] *wordpress sistemas X10 vantagens e desvantagens*, <https://sistemasx10.wordpress.com/sistemas-x10/vantagensdesvantagens-do-x10/>, Accessed: 2021-03-06.
- [13] *slideplayer X10*, <https://slideplayer.com.br/slide/281420/>, Accessed: 2021-03-06.
- [14] *RF what is insteon*, <https://www.everythingrf.com/community/what-is-insteon>, Accessed: 2021-03-07.
- [15] *RF wireless world what is wavenis*, <https://www.rfwireless-world.com/Terminology/Wavenis-basics.html>, Accessed: 2021-03-07.
- [16] *webdyn wavenis operação, aplicação*, <https://www.webdyn.com/en/products-2/enercontrol-gateways/energy-router-webdynrf-wavenis/>, Accessed: 2021-03-07.
- [17] *pocket zigbee*, <https://www.pocket-lint.com/pt-br/casa-inteligente/noticias/129857-o-que-e-zigbee-e-por-que-e-importante-para-sua-casa-inteligente>, Accessed: 2021-03-13.
- [18] *techtudo zigbee*, <https://www.techtudo.com.br/noticias/2019/12/o-que-e-zigbee-saiba-tudo-sobre-o-protocolo-para-iot-e-casa-conectada.ghtml>, Accessed: 2021-03-13.
- [19] *youtube zigbee funcionamento vantagens e desvantagens*, <https://www.youtube.com/watch?v=PDfns7a9X2w>, Accessed: 2021-03-13.
- [20] *educação e tecnologia zigbee vantagens*, Accessed: 2021-03-13.
- [21] J. P. A. GONÇALVES, «Protocolos de Automação Doméstica - Solução de Automação Residencial e Vigilância Baseada em Protocolo Z-Wave,» rel. téc., nov. de 2017.
- [22] *smarterHome Vantagens do Z-Wave*, <https://smarterhome.sk/en/informacie/advantages-of-z-wave-10>, Accessed: 2021-03-27.

- [23] *linkd home Os melhores hubs Z-wave*, <https://linkdhome.com/articles/best-zwave-hub>, Accessed: 2021-03-27.
- [24] *O protocolo MQTT MQTT*, <https://www.gta.ufrj.br/ensino/eel878/redes1-2018-1/trabalhos-vf/mqtt/>, Accessed: 2021-03-10.
- [25] *OPC Router MQTT*, <https://www.opc-router.com/what-is-mqtt/>, Accessed: 2021-03-10.
- [26] *hiveMQ qualite of service of MQTT*, <https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/>, Accessed: 2021-03-10.
- [27] *engprocess automação industrial MQTT*, <https://engprocess.com.br/mqtt-broker/>, Accessed: 2021-03-27.
- [28] *gta ufrij MQTT*, <https://www.gta.ufrj.br/ensino/eel878/redes1-2019-1/v1/mqtt/>, Accessed: 2021-03-27.
- [29] F. victor schreiber, «SISTEMA DE MEDIÇÃO SEM FIO DE CONSUMO DE ÁGUA DO EMPREENDIMENTO SAPIENS PARQUE UTILIZANDO PROTOCOLO MQTT,» rel. téc., set. de 2012.
- [30] *ATHOS electronics ESP8266 – O que é e para que serve?* <https://athoselectronics.com/esp8266-o-que-e/>, Accessed: 2021-03-29.
- [31] *FVM Learnig Conhecendo ESP8266, Especificações e comparações com ESP32 e Arduino*, <http://www.fvml.com.br/2018/12/conhecendo-esp8266-especificacoes-e.html>, Accessed: 2021-03-29.
- [32] J. D. M. FILHO, «SISTEMA DE AQUISIÇÃO DE DADOS UTILIZANDO O MÓDULO ESP8266 NodeMCU,» rel. téc., dez. de 2017.
- [33] *Automação em 5 minutos O que é o módulo ESP8266 NodeMCU?* Accessed: 2021-03-29.
- [34] *iot techtrends ESP32*, <https://www.iottechrends.com/what-is-esp32/>, Accessed: 2021-04-14.

- [35] *wikipedia ESP32*, <https://pt.wikipedia.org/wiki/ESP32>, Accessed: 2021-04-14.
- [36] *The internet of thing with ESP32 ESP32*, <http://esp32.net/>, Accessed: 2021-04-14.
- [37] *Explore Embedded ESP32*, Accessed: 2021-04-14.
- [38] *pplware Node-RED*, <https://pplware.sapo.pt/tutoriais/node-red-internet-of-things/>, Accessed: 2021-04-14.
- [39] *envisia Node-RED*, <https://www.envisia.com.br/2019/01/08/voce-sabe-o-que-e-tecnologia-node-red/>, Accessed: 2021-04-14.
- [40] *Raspberry Pi Raspberry Pi*, <https://www.electromaker.io/blog/article/best-raspberry-pi-4-cases-top-raspberry-pi-4-compatible-cases>, Accessed: 2021-07-12.
- [41] *Raspberry Pi Raspberry Pi*, <https://randomnerdtutorials.com/getting-started-with-raspberry-pi/>, Accessed: 2021-07-12.
- [42] *futurecom como a internet das coisas (IoT) pode ajudar na eficiência energética das empresas*, <https://digital.futurecom.com.br/transformao-digital/entenda-como-internet-das-coisas-iot-pode-ajudar-na-eficincia-energtica-das>, Accessed: 2021-03-26.
- [43] B. E. MEDINA, «Ajuste Dinâmico e Automático de Iluminação em Sistemas Industriais e Edifícios Usando Tecnologias de IoT,» rel. téc., jun. de 2017.
- [44] C. M. da Silva, «DESENVOLVIMENTO DE UMA REDE DE SENSORES E ATUADORES SEM FIO PARA O CONTROLE DE SISTEMAS DE CLIMATIZAÇÃO,» rel. téc., jun. de 2019.
- [45] *adene Eficiência Energética na Indústria*, <https://www.adene.pt/industria/>, Accessed: 2021-03-26.
- [46] *instalador Eficiência energética no setor industrial é a alavanca da mudança*, <https://www.oinstalador.com/Artigos/267197-Eficiencia-energetica-no-setor-industrial-e-a-alavanca-da-mudanca.html>, Accessed: 2021-03-26.

- [47] *carrier* *Sistemas de aquecimento, ventilação e ar condicionado - Portugal*, <https://www.carrier.com/commercial/pt/pt/sobre-carrier/centros-de-excelencia/venge/>, Accessed: 2021-03-26.
- [48] R. D. A. CABRAL, «ESTUDO SOBRE A EFICIÊNCIA ENERGÉTICA EM SISTEMAS IOT COM DIVERSIDADES DE DUPLICAÇÃO DE PACOTES,» rel. téc., jun. de 2019.
- [49] R. K. Kodali e S. Soratkal, «MQTT based Home Automation System Using ESP8266,» rel. téc., nov. de 2016.
- [50] —, «Low Cost Ambient Monitoring using ESP8266,» rel. téc., nov. de 2016.
- [51] V. G. CRISTINA TURCU CORNEL TURCU, «An Internet of Things Oriented Approach for Water Utility Monitoring and Control,» rel. téc., jun. de 2018.
- [52] R. P. Pratama, «PENGENDALI LAMPU RUMAH BERBASIS ESP8266 DENGAN PROTOKOL MQTT,» rel. téc., dez. de 2020.

# Apêndice A

## Apêndice

### A.1 Código 1

Refere-se ao código compilado no ESP32. Este microcontrolador foi programado de modo a controlar os estados das iluminações (*on/off* e *dimming*), controlo (variação) de temperatura e leitura dos dados do sensor de temperatura e humidade. Para isso foi necessário a comunicação por MQTT com o objetivo de enviar e receber dados.

```
#include <WiFi.h>
```

```
#include <PubSubClient.h>
```

```
#include <Wire.h>
```

```
#include "DHT.h"
```

```
//-----
```

```
WiFiClient espClient;
```

```
PubSubClient client(espClient);
```

```
long lastMsg = 0;
```

```
char msg[50];
```

```
int value = 0;
```

```

float temperatura = 0;    // pino L1 que queremos publicar no Brocker
float humedada = 0;      // pino L2 que queremos publicar no Brocker
const int ledPin1 = 2;   // LedPin GPIO2
const int ledPin2 = 4;   // LedPin GPIO4
const int ledPin3 = 15;  // LedPin GPIO15
#define DHTPIN 14        // Pino digital conectado ao sensor DHT

//-----PWM-----
const int ledPin4 = 16;  // LedPin GPIO16
const int freq = 5000;
const int ledChannel = 0;
const int resolution = 8;
#define DHTTYPE DHT11    // DHT 11
DHT dht(DHTPIN, DHTTYPE);

//-----SSID - password do Router servidor a Internet-----
const char* ssid = "pimpampum";    // nome da Internet
const char* password = "0987654321"; // password da Internet

//----- IP Address do MQTT Broker (Rasperry PI)-----
const char* mqtt_server = "192.168.1.33"; // endereço do servidor

//-----void setup()-----
void setup()
{
  Serial.begin(115200);
  setup_wifi();

```

```
client.setServer(mqtt_server, 1883);
client.setCallback(callback);

pinMode(ledPin1, OUTPUT);
pinMode(ledPin2, OUTPUT);
pinMode(ledPin3, OUTPUT);

ledcSetup(ledChannel, freq, resolution);
ledcAttachPin(ledPin4, ledChannel);

Serial.println(F("DHT11 test!"));

dht.begin();

}

//-----VOID SETUP_WIFI()----- realiza a conexão com o wifi da Internet-----
void setup_wifi()
{
  delay(10);
  Serial.println();
  Serial.print("connectando a");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED){
    delay(500);
    Serial.print(".");
  }
  Serial.print("");
```

```
Serial.println("WiFi conectado");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}

//-----void Callback-----realiza a conexão MQTT (subscreber: recebe dados)---
void callback(char* topic, byte* message, unsigned int length)
{
  Serial.print("Mensagem recebida no tópico: ");
  Serial.print(topic);
  Serial.print(", Message: ");
  String messageTemp;
  for (int i = 0; i < length; i++){
    Serial.print((char)message[i]);
    messageTemp = (char) message[i];
  }
  Serial.println();

  //-----primeiro Output (usando topic: esp32/output1)-----
  //se receber mensagem sobre o tópico 'esp32/output', verifica se "on" o "off"
  if (String(topic) == "esp32/output1")
  {
    Serial.print("mudando a saida: ");
    if(messageTemp == "on")
    {
      Serial.println("on");
      digitalWrite(ledPin1, HIGH);
    }
    else if(messageTemp == "off")
```

```
{
  Serial.println("off");
  digitalWrite(ledPin1, LOW);
}
}

//-----segundo output (usando topic: esp32/output2)-----
//se receber mensagem sobre o tópico 'esp32/output2', verifica se "on" o "off"
if (String(topic) == "esp32/output2")
{
  Serial.print("mudando a saida: ");
  if(messageTemp == "on")
  {
    Serial.println("on" );
    digitalWrite(ledPin2, HIGH);
  }
  else if(messageTemp == "off")
  {
    Serial.println("off");
    digitalWrite(ledPin2, LOW);
  }
}

//-----terceiro output (usando topic: esp32/output3)-----
//se receber mensagem sobre o tópico 'esp32/output3', verifica se "on" o "off"
if (String(topic) == "esp32/output3")
{
  Serial.print("mudando a saida: ");
  if(messageTemp == "on")
  {
```

```

    Serial.println("on" );
    digitalWrite(ledPin3, HIGH);
}
else if(messageTemp == "off")
{
    Serial.println("off");
    digitalWrite(ledPin3, LOW);
}
}

//-----quarto output (usando topic: esp32/output4)-----
//se receber mensagem sobre o tópico 'esp32/output3', tem de variar o PWM (0 - 255)
if (String(topic) == "esp32/output4")
{
    Serial.print("Alteracao de saida PWM: ");
    Serial.println("messageTemp");
    ledcWrite(ledChannel,messageTemp.toInt());
}
// Sentir à vontade em adicionar mais instruções if para controlar mais GPIO com MQTT
}

//-----VOID RECONNECT()----- realiza a conexão em caso de falha
void reconnect()
{
    //loop até reconectado
    while (!client.connected())
    {
        Serial.print("Tentando conexão MQTT...");
        if (client.connect("ESP8266Client")){

```

```
Serial.println("conectado");
client.subscribe("esp32/output1"); //-----topic:'esp32/output1'-----
client.subscribe("esp32/output2"); //-----topic:'esp32/output2'-----
client.subscribe("esp32/output3"); //-----topic:'esp32/output3'-----
client.subscribe("esp32/output4"); //-----topic:'esp32/output4'-----
}else{
  Serial.print("Fallo, rc=");
  Serial.print(client.state());
  Serial.println("Tente novamente em 5s");
  delay(5000);
}
}
}
//-----VOID loop()----- realiza conexão MQTT (publish: enviar dados)
void loop()
{
  if(!client.connected()){
    reconnect();
  }
  client.loop();

  long now = millis();
  if (now - lastMsg > 100) //100ms (tempo de amostragem)
  {
    lastMsg = now;

    //----- assinar1 que queremos enviar para o Broker-----
    temperatura = analogRead(36)*(100.0/4095.0);
```

```

//converter o valor para matriz char
char tempString[8];
dtostrf(temperatura, 1, 2, tempString);
Serial.print("temperatura: ");
Serial.println(tempString);
client.publish("esp32/temperature", tempString); //topic: 'esp32/temperature'

//-----assinar2 que queremos enviar para o Broker-----
humedad = 37.9;

// converter o valor para matriz char
char humString[8];
dtostrf(humedad, 1, 2, humString);
Serial.print("Humedad: " );
Serial.println(humString);
client.publish("esp32/humidity", humString); //Topic 'esp32/humidity'

//----- leitura de dados do sensor DHT -----

// Aguarde alguns segundos entre as medições.
delay(2000);

// A leitura da temperatura ou humidade leva cerca de 250 milissegundos!
// As leituras do sensor também podem ter até 2 segundos(é um sensor muito lento)
float h = dht.readHumidity();
// Ler a temperatura como Celsius (o padrão)
float t = dht.readTemperature();
// Ler a temperatura como Fahrenheit (isFahrenheit = true)

```

```
float f = dht.readTemperature(true);

// Verifique se alguma leitura falhou e saia mais cedo (para tentar novamente).
if (isnan(h) || isnan(t) || isnan(f)) {
  Serial.println(F("Failed to read from DHT sensor!"));
  return;
}

// Calcula o índice de calor em Fahrenheit (o padrão)
float hif = dht.computeHeatIndex(f, h);
// Computa o índice de calor em Celsius (isFahreheit = false)
float hic = dht.computeHeatIndex(t, h, false);

Serial.print(F("Humidity: "));
Serial.print(h);
Serial.print(F("% Temperature: "));
Serial.print(t);
Serial.print(F("°C "));
Serial.print(f);
Serial.print(F("°F Heat index: "));
Serial.print(hic);
Serial.print(F("°C "));
Serial.print(hif);
Serial.println(F("°F"));

}
}
```

## A.2 Código 2

Este trata-se do código que foi compilado para o ESP8266. Este microcontrolador foi programado com o objetivo de controlar o relé também foi necessário a comunicação por MQTT de modo a enviar e receber dados.

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// Esp8266 pins.
#define ESP8266_GPI02    2 // LED azul.
#define ESP8266_GPI04    4 // Controlo de relés.
#define ESP8266_GPI05    5 // Entrada Opto-acoplador.
#define LED_PIN          ESP8266_GPI02

// Wifi SSID.
const char ssid[] = "pimpampum";
// Wifi password.
const char password[] = "0987654321";
// Flag for sync on re-connection.
bool isFirstConnect = true;
volatile int relayState = LOW; //Status do botão de aplicação Blynk
volatile int inputState = LOW; //Estado do pino de entrada.

void setup() {
  pinMode( ESP8266_GPI04, OUTPUT ); // Pino de controlo do relé.
  pinMode( ESP8266_GPI05, INPUT_PULLUP ); // Pino de entrada.
  pinMode( LED_PIN, OUTPUT ); // Módulo ESP8266 LED azul.
  digitalWrite( LED_PIN, LOW ); // Liga o LED.
  Blynk.begin( auth, ssid, password ); // Iniciar a conexão do Blynk.
```

```
digitalWrite( LED_PIN, HIGH );           // Desligar o LED.
}

// Esta função funciona toda vez que a conexão Blynk é estabelecida.
BLYNK_CONNECTED() {
  if ( isFirstConnect ) {
    Blynk.syncAll();
    isFirstConnect = false;
  }
}

// LED de entrada de sincronização.
BLYNK_READ( V2 ) {
  CheckInput();
}

// Comando de relé (Blynk).
BLYNK_WRITE( V0 ) {
  if ( param.asInt() != relayState ) {
    relayState = !relayState;           // Toggle state.
    digitalWrite( ESP8266_GPI04, relayState ); // Relay control pin.
    Blynk.virtualWrite( V1, relayState*255 ); // Set Blynk app LED.
  }
}

int DebouncePin( void ) {
  // Read input pin.
  if ( digitalRead( ESP8266_GPI05 ) == HIGH ) {
```

```
// Debounce input.
delay( 25 );
if ( digitalRead( ESP8266_GPI05 ) == HIGH )
    return HIGH;
}
return LOW;
}

// Conjunto LED baseado no estado do pino de entrada.
void CheckInput( void ) {
    if ( DebouncePin() != inputState ) {
        Blynk.virtualWrite( V2, inputState*255 );
        inputState = !inputState;
    }
}

void loop() {
    Blynk.run();
    CheckInput();
}
```