

PM

Estudo e projecto de transformação de uma casa tradicional numa casa inteligente, baseada em IoT

PROJETO DE MESTRADO

Joheván de Gois Andrade

MESTRADO EM ENGENHARIA ELETROTÉCNICA - TELECOMUNICAÇÕES



UNIVERSIDADE da MADEIRA

A Nossa Universidade

www.uma.pt

maio | 2024

Estudo e projecto de transformação de uma casa tradicional numa casa inteligente, baseada em IoT

PROJETO DE MESTRADO

Joheván de Gois Andrade

MESTRADO EM ENGENHARIA ELETROTÉCNICA - TELECOMUNICAÇÕES

ORIENTAÇÃO

Luís Armando de Aguiar Oliveira Gomes

Resumo

As casas inteligentes estão a ganhar popularidade devido aos seus inúmeros benefícios. Atualmente, é cada vez mais crucial simplificar o quotidiano das pessoas. Os dispositivos inteligentes têm conquistado um lugar proeminente nas nossas vidas e estão cada vez mais integrados nos sistemas domésticos, automatizando processos como o controlo da temperatura ambiente e da iluminação, através de deteção de presença de pessoas, entre outras funções.

A visão da Internet das Coisas (IoT) é criar um mundo mais conectado e inteligente, onde os diversos dispositivos podem recolher informações, processar esses dados e tomar decisões de forma autónoma, melhorando a eficiência. No entanto, muitas das tecnologias utilizadas para criar casas inteligentes dependem de dispositivos de uma única marca e protocolo, o que limita a integração e utilização de outros dispositivos de marcas e tecnologias diferentes, que podem ser do interesse dos utilizadores, seja pela funcionalidade que oferecem, pela disponibilidade em loja ou pelo preço.

Neste projeto, foram estudadas e aplicadas soluções para integrar diversos dispositivos com diferentes tecnologias numa única plataforma *open source*. O objetivo era promover a interoperabilidade entre as mais diversas soluções de dispositivos, apresentando soluções que simplificam a criação da rede para que possa ser integrada em qualquer ambiente doméstico, e utilizando hardware de custo reduzido.

Foram objeto de estudo as *gateways* Hubitat e a Silvercrest da Lidl e uma plataforma instalada numa Raspberry Pi chamada Home Assistant, procurando-se explorar nas referidas soluções as suas diversas funcionalidades, a integração de dispositivos, a compatibilidade com vários dispositivos e a criação de cenários domóticos. Constatou-se que a plataforma Home Assistant, em comparação com o Hubitat e o Silvercrest, evidenciou maior compatibilidade com os diferentes sensores/atuadores disponíveis no mercado, maior flexibilidade na utilização de diversos protocolos ou tecnologias de comunicação, mas também uma maior complexidade para utilizadores sem experiência prévia com a domótica.

Palavras Chave: Domótica; Hubitat; Silvercrest; Home Assistant.

Abstract

Smart homes are gaining popularity due to their numerous benefits. Currently, it is increasingly crucial to simplify people's daily lives. Smart devices have earned a prominent place in our lives and are becoming more integrated into home systems, automating processes such as temperature control and lighting, through motion detection, among other functions.

The vision of the Internet of Things (IoT) is to create a more connected and intelligent world, where devices can collect information, process that data, and make autonomous decisions, improving efficiency. However, many of the technologies used to create smart homes rely on devices from a single brand and using a single protocol, limiting the integration and use of other devices from different brands and different technologies, that may be of interest to the users, because of the functionality they offer, availability in stores, or their price.

In this project, solutions were studied and applied to integrate various devices with different technologies into a single open-source platform. The goal was to promote interoperability among the various device solutions, presenting solutions that simplify the creation of the network, so that it can be integrated into any home environment, using low-cost hardware.

The Hubitat and Lidl's Silvercrest gateways were studied along with the Home Assistant platform, aiming to explore their various functionalities, device integration, compatibilities, and the creation of home automation scenarios on these platforms. It was found that the Home Assistant platform, compared to Hubitat and Silvercrest, showed greater compatibility with the different sensors/actuators available on the market, greater flexibility in using various communication protocols or technologies, but also greater complexity for users without prior experience with home automation.

Keywords: Home automation; Hubitat; Silvercrest; Home Assistant

Agradecimentos

Gostaria de expressar os meus mais sinceros agradecimentos a todos aqueles que contribuíram para a realização deste trabalho. Em primeiro lugar, gostaria de expressar o meu profundo apreço pela minha família cujo apoio, sacrifício e encorajamento constante foram fundamentais para concluir o meu percurso académico, especialmente à minha mãe licenciada Isabel Andrade, ao meu pai Alfredo De Gois e à Mimi.

Não posso deixar de expressar a minha gratidão ao meu orientador, o Professor Dr. Luís Armando de Aguiar Oliveira Gomes, por todo o apoio, disponibilidade e paciência demonstrada, orientação valiosa e conhecimentos transmitidos que definitivamente foram fundamentais para a conclusão desta tese

Gostaria de reconhecer a Universidade da Madeira, pela excelente formação académica proporcionada e pela oportunidade de crescimento intelectual e pessoal que me foi concedida. Agradeço também a todos os professores da Faculdade de Ciências Exatas e da Engenharia que contribuíram para a minha formação, partilhando os seus conhecimentos e experiências, e desafiando-me a alcançar o meu melhor.

Sem o apoio e contribuição de todos os mencionados, este trabalho não teria sido possível. A todos um muito obrigado.

Índice

Resumo.....	i
Abstract.....	iii
Agradecimentos.....	v
Índice.....	vii
Índice de figuras.....	xi
Índice de tabelas.....	xv
1. Introdução.....	1
1.1. Objetivos.....	2
1.2. ORGANIZAÇÃO DE CONTEUDOS.....	2
2. Revisão da bibliografia.....	5
2.1. Conceito de casa inteligente.....	5
2.1.1. Funcionalidades.....	5
2.2. Rede de uma casa inteligente.....	6
2.3. Classificação de um sistema de domótica.....	7
2.3.1. Meio para transmissão de dados.....	7
2.3.2. Arquiteturas de sistemas domóticos.....	8
2.4. Internet of things (IoT).....	10
2.4.1. Principais tecnologias.....	11
2.4.1.1. Wi-Fi.....	12
2.4.1.2. ZigBee.....	13
2.4.1.3. Bluetooth Low Energy (BLE).....	17
2.4.1.4. Z-Wave.....	22
2.4.1.5. MQTT.....	23
2.4.1.6. Comparação entre tecnologias.....	27
3. Equipamentos e software utilizados.....	29
3.1. Equipamentos.....	29
3.1.1. Raspberry Pi.....	29
3.1.2. ESP8266.....	30
3.1.3. ESP32.....	31
3.1.4. CC2531.....	32
3.1.5. Arduino UNO.....	33
3.1.6. Hubitat.....	34
3.1.7. Gateway Silvercrest.....	35
3.2. Sensores e atuadores.....	36

3.2.1.	AM2302	36
3.2.2.	PIR.....	37
3.2.3.	Relé.....	39
3.2.4.	KY-021	40
3.2.5.	IKEA série TRÅDFRI	41
3.3.	Plataformas.....	41
3.3.1.	Home Assistant.....	41
3.3.2.	Hubitat Elevation	43
3.3.3.	Lidl Home.....	43
4.	Instalação e configuração dos sistemas utilizados	45
4.1.	Hubitat	45
4.1.1.	Dispositivos.....	46
4.1.1.1.	Tomada ZigBee	46
4.1.2.	Automatizações	48
4.1.2.1.	Automatização horária.....	48
4.2.	Silvercrest	49
4.2.1.	Dispositivos.....	51
4.2.1.1.	Tomada IKEA	51
4.2.2.	Automatizações	52
4.2.2.1.	Automatização por geolocalização	52
4.2.2.2.	Automatização horária.....	53
4.3.	Home Assistant	54
4.3.1.	<i>Add-ons</i>	56
4.3.2.	Sensores MQTT	57
4.3.2.1.	Sensor PIR.....	57
4.3.2.2.	Tomada inteligente	61
4.3.2.3.	Integração dos sensores MQTT no Home Assistant.....	63
4.3.3.	Sensores/Atuadores ZigBee	65
4.3.3.1.	Coordenador ZigBee	66
4.3.3.2.	Integração de sensores/atuadores ZigBee	68
4.3.4.	Sensores Wi-Fi	69
4.3.4.1.	Sensor AM2302	69
4.3.4.2.	Sensor de portas e janelas	72
4.3.5.	Automatizações	75
4.3.5.1.	Desenvolvimento de um sensor de garagem com notificação de porta aberta através de dispositivo móvel	75

4.3.5.2.	Desenvolvimento de um sistema de alarme inteligente.....	77
4.3.5.3.	Desenvolvimento de um termostato inteligente	80
5.	Resultados.....	83
5.1.	Hubitat Elevation.....	83
5.1.1.	Controlo do atuador através do <i>dashboard</i>	83
5.1.2.	Automação horária	84
5.2.	Silvercrest e Lidl Home	84
5.2.1.	Controle manual da tomada	84
5.2.2.	Automatização por geolocalização.....	85
5.2.3.	Automatização horária	86
5.3.	Home Assistant	86
5.3.1.	Sensor e atuador MQTT	86
5.3.1.1.	Sensor PIR.....	86
5.3.1.2.	Relé	88
5.3.2.	ZigBee.....	89
5.3.2.1.	Tomada ZigBee	89
5.3.3.	Sensores Wi-Fi	90
5.3.3.1.	Sensor de temperatura e humidade AM2302	90
5.3.3.2.	Sensor de portas e janelas	91
5.3.4.	Automações	92
5.3.4.1.	Sensor de porta de garagem com notificação através do dispositivo móvel	92
5.3.4.2.	Sistema de alarme.....	93
5.3.4.3.	Termostato inteligente.....	94
6.	Conclusões	97
6.1.	Trabalhos futuros	98
7.	Referências	101
Anexo I	105
Anexo II	109
Anexo III	113

Índice de figuras

Figura 1 - Arquitetura centralizada [7].	8
Figura 2 - Arquitetura descentralizada [7].	9
Figura 3 - Arquitetura distribuída [7].	10
Figura 4 - Topologia em estrela [15].	15
Figura 5 - Topologia em árvore [15].	16
Figura 6 - Topologia em malha [13].	16
Figura 7 - Configurações entre versões de Bluetooth e tipos de dispositivos [17].	18
Figura 8 - Canais de frequência [17].	18
Figura 9 - Topologia Broadcast [17].	19
Figura 10 - Topologia Connected[17].	20
Figura 11 - Eventos de conexão [17].	21
Figura 12 - Exemplo de uma tabela de roteamento Z-Wave [19].	22
Figura 13 - Esquema MQTT [21].	24
Figura 14 - Separação dos diferentes tópicos [22].	24
Figura 15 - QoS 0 no protocolo MQTT [24].	25
Figura 16 - QoS 1 no protocolo MQTT [24].	26
Figura 17 - QoS 2 no protocolo MQTT [24].	26
Figura 18 - Raspberry Pi modelo 3B+ e layout dos pinos GPIO [26], [27].	30
Figura 19 - NodeMCU ESP8266 com layout dos pinos [29].	31
Figura 20 - ESP32 com layout dos pinos [31].	32
Figura 21 - CC2531 com layout dos pinos [33].	33
Figura 22 - Kit completo para programar o CC2531 [34].	33
Figura 23 - Arduino UNO com layout das portas analógicas e digitais [36].	34
Figura 24 - Smart hub da Hubitat Elevation [38].	35
Figura 25 - Automação na nuvem vs automação local com o Hubitat [39].	35
Figura 26 - Central domótica Silvercrest da marca Lidl com as suas características [41].	36
Figura 27 - Sensor de temperatura e humidade AM2302.	37
Figura 28 - Princípio de funcionamento do sensor de movimento por infravermelho [44].	38
Figura 29 - Diagrama de blocos do PIR [45].	38
Figura 30 - Sensor utilizado. Grove - Adjustable PIR Motion Sensor [46], [47].	38
Figura 31 - Relé DFR0017 V3.1 [49].	39
Figura 32 - Sensor KY-021 [50].	40
Figura 33 - Tomada ZigBee da Ikea com comando próprio [51].	41
Figura 34 - Soluções integradas no Home Assistant [53].	42
Figura 35 - Gateway Silvercrest [40].	44
Figura 36 - Web Interface do Hubitat.	45
Figura 37 - Integração do atuador IKEA TRÅDFRI no Hubitat.	46
Figura 38 - Comandos e preferências da tomada inteligente disponíveis no Hubitat.	47
Figura 39 - Dashboard do Hubitat.	48
Figura 40 - Automatização de uma tomada no Hubitat.	49
Figura 41 - Integração do módulo Silvercrest na aplicação Lidl Home.	50
Figura 42 - Emparelhamento da tomada IKEA no Silvercrest.	51
Figura 43 - Criação de uma automatização por geolocalização com a central Silvercrest e a app Lidl Home.	53
Figura 44 - Criação de uma automatização por dia e hora na Silvercrest e na app Lidl Home.	54
Figura 45 - Página inicial do Home Assistant.	56

Figura 46 - Add-ons utilizados.	57
Figura 47 - Esquemático da montagem do circuito.	58
Figura 48 - Instalação do gestor de placas esp8266.	58
Figura 49 - Seleção da placa NodeMCU.	59
Figura 50 - Instalação da biblioteca necessária para utilizar Wi-Fi nas placas de desenvolvimento ESP32 e ESP8266.	59
Figura 51 - Fluxograma do código para o sensor PIR com MQTT.	60
Figura 52- Circuito montado para o PIR.	61
Figura 53 - Esquemático do circuito utilizado.	61
Figura 54 - Fluxograma do código para o relé com MQTT.	62
Figura 55 - Circuito montado para o Relé.	63
Figura 56 - Configuração do broker MQTT.	64
Figura 57 - Configuração do ficheiro .yaml no file editor.	64
Figura 58 - Configuração dos cartões para a visualização das diferentes entidades na plataforma Home Assistant.	65
Figura 59 - CC Debugger para programar o FW no dongle CC2531.	66
Figura 60 - Flash Programmer com os parâmetros necessários para programar o CC2531.	67
Figura 61 - Endereço do dispositivo na Raspberry Pi.	68
Figura 62 - Configuração das diferentes entidades no cartão do painel principal.	69
Figura 63 - Esquema do circuito.	70
Figura 64 - Circuito montado para o sensor de temperatura e humidade.	70
Figura 65 - Visualização do dispositivo no ESPHome.	71
Figura 66 - Código inserido no ESPHome para o sensor de temperatura e humidade.	71
Figura 67 - Aditamento ao cartão do painel principal das novas entidades de interesse.	72
Figura 68 - Esquema de ligação do sensor KY-021 à ESP 8266.	73
Figura 69 - Circuito montado na protoboard para o KY-021.	73
Figura 70 - Código utilizado para programar a ESP8266 para usar o sensor KY-021 como sensor de uma porta de garagem.	74
Figura 71 - Configuração dos gatilhos da automação.	76
Figura 72 - Ações despoletadas na automação pelos gatilhos.	76
Figura 73 - Esquema do sensor da porta da garagem.	77
Figura 74 - Trigger e condições utilizadas para despoletar as ações dentro da automação.	78
Figura 75 - Ações desencadeadas.	79
Figura 76 - Esquema do sistema de alarme.	79
Figura 77 - Parâmetros utilizados para a constituição da automação.	81
Figura 78 - Parâmetros utilizados para a constituição da automação 2.	81
Figura 79 - Esquema do termostato inteligente.	82
Figura 80 - Controle do atuador TRÅDFRI IKEA através do Hubitat.	83
Figura 81 - Controle manual da tomada IKEA através do Silvercrest e da app Lidl Home.	85
Figura 82 - Dados enviados pelo dispositivo construído com o sensor PIR: (a) sem movimento; (b) com movimento.	87
Figura 83 - Visualização dos dados enviados pelo sensor de movimento PIR, através do software MQTT Explorer.	87
Figura 84 – Comportamento do relé: (a) sem movimento; (b) com movimento.	88
Figura 85 - Interruptor TRADFRI: (a) desativado; (b) ativado.	89
Figura 86 - Histórico do estado do atuador.	89
Figura 87 - Valores instantâneos de temperatura e humidade medidos com o AM2302.	90
Figura 88 - Histórico do sensor de temperatura e humidade.	90

Figura 89 - Valor de temperatura obtido no “The Weather Channel” [62].	91
Figura 90 – Resultados do sensor KY-021: (a) com campo magnético; (b) sem campo magnético.	91
Figura 91 - Abertura da porta da garagem para despoletar a notificação.	92
Figura 92 - Notificação despoletada pela automação passados os 3 minutos.	93
Figura 93 - Recorte de ecrã com as condições cumpridas para despoletar o alerta do alarme.	93
Figura 94 - Alerta de deteção de intruso enviado ao dispositivo móvel do utilizador.	94
Figura 95 - Mensagem publicado no tópico “Alarme” após ser detetado um intruso.	94
Figura 96 - Valores de temperatura registados pelo sensor AM2302.	95
Figura 97 - Estado da tomada TRÅDFRI.	95
Figura 98 - Valores de temperatura registados.	96
Figura 99 - Estado do atuador IKEA TRÅDFRI.	96

Índice de tabelas

Tabela 1 – Tabela comparativa entre cada tecnologia apresentada.	27
---	----

1. Introdução

Ao longo dos anos a tecnologia tem evoluído de uma forma imparável, com estes avanços a impactarem a sociedade de forma a melhorarem a qualidade de vida dos cidadãos, e é por esta mesma razão que esses fenómenos são refletidos no entorno mais próximo e importante: o lar.

O ser humano precisa de um espaço confortável, eficiente, seguro, inteligente e personalizável, de tal forma que possa ser adaptado as necessidades de cada pessoa, otimizando tarefas rotineiras, reduzindo esforços, poupando tempo e recursos, e tornando a experiência dentro da casa o mais satisfatória possível. Uma casa inteligente é precisamente isso, um sistema que contém um conjunto de dispositivos que trabalham e comunicam entre si para melhorar a qualidade de vida dos seus ocupantes.

Este projeto visava resolver os principais desafios presentes na transformação de uma casa tradicional numa casa inteligente. Esses desafios estão relacionados com:

- Instalação de cablagem, tomadas e quadros de telecomunicações, que aumentam os custos de implementação e a necessidade de profissionais qualificados para a instalação do sistema, e que geralmente obrigam a obras incomodas em habitações já habitadas;
- Inflexibilidade das interfaces: as interfaces com os utilizadores geralmente são limitadas a um único método de controle e não permitem expandir e criar diferentes interfaces de utilizador;
- Protocolos proprietários: estes sistemas são projetados com fins comerciais, para serem lucrativos, e portanto, utilizam protocolos proprietários que só permitem a conexão de produtos licenciados, obrigando geralmente à compra de soluções “integradas” que raramente respondem a todos os desejos dos clientes.

Esta é uma área em forte crescimento e o presente trabalho surge como mais uma contribuição, na medida em que apresenta o desenvolvimento de uma *gateway* multiprotocolo que permite a um utilizador comum, transformar, de forma facilitada e mais económica, uma casa tradicional numa casa inteligente, integrando várias das tecnologias de transmissão e soluções específicas existentes no mercado. Pretendia-se assim construir uma solução que permitisse integrar diversos sensores, atuadores e

controladores existentes no mercado, que funcionam com diferentes protocolos de comunicação e, por isso, geralmente, não podem ser interligados.

1.1. Objetivos

O desenvolvimento do presente trabalho teve como principais objetivos os seguintes tópicos:

- Realizar a revisão bibliográfica sobre os conceitos das casas inteligentes e da *IoT* (Internet das coisas);
- Estudar as várias tecnologias de transmissão que podem ser utilizadas numa casa inteligente. (funcionamento, normas, cobertura, débito, etc.)
- Efetuar uma análise aos diversos tipos de sensores e atuadores normalmente utilizados neste ambiente aplicacional;
- Analisar as várias soluções existentes no mercado;
- Realizar o desenvolvimento de uma *gateway* multiprotocolo conjuntamente com alguns sensores e atuadores;
- Criar cenários e regras de controlo para a casa conforme as necessidades ou preferências dos utilizadores para otimizar e automatizar as tarefas;
- Testar a solução implementada;
- Comparar as diferentes plataformas.

1.2. ORGANIZAÇÃO DE CONTEUDOS

O presente trabalho de mestrado encontra-se estruturado em 6 capítulos. As páginas pre-textuais apresentam o resumo, os agradecimentos, e os índices de conteúdo e figuras.

O primeiro e atual capítulo apresenta a introdução e os objetivos propostos para a realização do trabalho.

No segundo capítulo é realizado um estudo aprofundado ao estado da arte, onde são abordados todos os conceitos fundamentais da domótica, como os diferentes meios para a transmissão dos dados, arquiteturas, tecnologias, protocolos, topologias e aplicações.

No terceiro capítulo é apresentada uma breve descrição e as características principais de todos os equipamentos, sensores, atuadores e plataformas utilizadas para o desenvolvimento deste trabalho, sendo que nos equipamentos podem ser referidos a Raspberry Pi, as placas ESP8266 e ESP32, e as diferentes *gateways* utilizadas. Em relação aos sensores e atuadores, são discutidos o AM2302, o PIR, o KY-021, o Relé e a tomada IKEA. Na secção das plataformas utilizadas são abordadas o Home Assistant, o Hubitat Elevation e o Lidl Home.

O quarto capítulo indica os passos necessários para a instalação, configuração e integração dos diferentes dispositivos utilizados de forma que seja possível a sua utilização com a plataforma e a tecnologia ou protocolo de interesse, assim como também a criação de cenários de automação onde possam interagir com outros sensores/atuadores.

Os resultados são finalmente apresentados no capítulo 5. Esses resultados estão organizados por *gateways* e subcategorias para cada dispositivo e cenário doméstico utilizado.

O capítulo 6 engloba as conclusões gerais e sugestões de trabalhos futuros para a continuação e implementação de melhorias no âmbito deste estudo. Seguem-se as referências bibliográficas e, por último, os anexos, onde são disponibilizados os códigos de programação implementados.

2. Revisão da bibliografia

Neste capítulo serão abordados todos os conceitos relacionados com as casas inteligentes, desde a sua definição, até às diferentes soluções que podem ser encontradas no mercado, passando pelos protocolos e topologias que podem ser usadas.

2.1. Conceito de casa inteligente

A expressão “casa inteligente” está associada ao conceito de automação residencial ou domótica [1]. As diversas definições empregues no conceito de automação residencial focam-se quase na sua totalidade num sistema integrado que permite, simplesmente, controlar tudo o que diz respeito a uma habitação.

A “domótica” é o ramo da ciência que integra tecnologias de computadores e robótica com os eletrodomésticos presentes no nosso domicílio. Em termos etimológicos, “domótica” é, a junção de “*domus*”, que significa casa, com “robótica”, deixando bem claro que o controlo das funções dos aparelhos residenciais pode ser uma realidade bem presente no quotidiano dos utilizadores.

Esta ciência, ligada ao domínio da automação de habitações, tem como objetivos fundamentais oferecer uma maior segurança e maior conforto, seja a nível da deteção de emergências, tais como incêndios ou fugas de gás ou água, seja a nível da deteção e sinalização de situações de intrusão, ou mesmo ao simples controlo de uma divisão em termos de temperatura ou iluminação, por exemplo quando os ocupantes estejam em repouso [1].

2.1.1. Funcionalidades

Uma vez que a domótica é uma área bastante abrangente, esta pode ser dividida em diversos setores [2]:

- **Conforto:** Área responsável por aumentar os níveis de qualidade de vida dos seus utilizadores, através da automação de determinadas rotinas, tais como: Controlo da climatização, iluminação, abertura e fecho de estores, entre outros.
- **Segurança:** A segurança é um dos principais benefícios da domótica, pois permite dotar a casa com sistemas para a deteção de vários cenários indesejados, como,

intrusões, inundações, fugas de gás e incêndios. Caso seja detetada alguma destas situações, a automação permite atuar de forma a tentar resolver o problema ativando algum atuador e/ou avisando os utilizadores e as autoridades.

- **Eficiência Energética:** Esta área é considerada crucial no cenário contemporâneo, especialmente tendo em conta as novas diretrizes energéticas e ambientais em escala global. Esta área é responsável pela otimização e monitorização dos consumos de energia numa variedade de sistemas.
- **Comunicação:** Engloba as comunicações internas, como aquelas entre o utilizador e o sistema de automação residencial, bem como as comunicações externas. Algumas das vantagens são as seguintes [3]:
 - Oferece informações em tempo real sobre a residência a qualquer momento, possibilitando um controle remoto total, independentemente da localização do utilizador;
 - Envio de alertas e notificações tanto ao proprietário quanto às autoridades competentes, quando necessário e conforme programado;
 - Integração e comunicação com todos os dispositivos e serviços eletrónicos incluído no sistema.

2.2. Rede de uma casa inteligente

Os sistemas domóticos baseiam-se no uso de 5 tipos de dispositivos: sensores, controladores, atuadores, dispositivos específicos e interfaces, todos eles ligados a uma rede, chamada de rede de controlo [4]. Ao agir em conjunto, criando automações e seguindo algum protocolo, estes dispositivos podem comportar-se de forma “inteligente”. As funções dos diferentes dispositivos são as seguintes [4]:

- **Sensores:** Dispositivos responsáveis por efetuar a monitorização dos valores das diferentes informações da casa, tais como, luminosidade, temperatura, pressão, fugas de água, gás, etc.
- **Controladores:** São dispositivos encarregues de efetuar a gestão e controlo do sistema através da informação recebida dos sensores, efetuando o posterior processamento e comando dos atuadores. A decisão dos controladores após todo

o processamento da informação dos sensores depende da programação neles inserida e/ou das solicitações requisitadas pelo utilizador através do uso de uma aplicação ou servidor.

- Atuadores: Os atuadores são os aparelhos responsáveis por realizar a ação solicitada. Estes são dispositivos de saída, uma vez que a informação sai do sistema para o equipamento que se deseja controlar.
- Dispositivos específicos: São necessários ao funcionamento do sistema, como por exemplo, *modems*, *routers* ou *gateways*, que permitem o envio de informação entre os diversos meios de transmissão.
- Interfaces: Permitem que o utilizador visualize e efetue o envio de informação de controlo e gestão para os diferentes controladores; a informação é apresentada normalmente em *websites*, *displays*, aplicações para telemóveis, *tablets*, etc.

2.3. Classificação de um sistema de domótica

Os sistemas de automação podem ser repartidos em três categorias distintas, de acordo com o meio utilizado para a transmissão de dados, arquitetura dos sistemas e topologia de rede [5].

2.3.1. Meio para transmissão de dados

A interconexão de dispositivos numa habitação requer um meio de comunicação, isto é, uma infraestrutura que possibilite a transmissão eficiente de informações entre os diversos dispositivos da rede de automação residencial. Nesse contexto, destacam-se três abordagens distintas para esse propósito [5]:

- As soluções com fios: Envolvem a implementação de cabos dedicados, os quais requerem instalação prévia durante a construção da residência ou ao longo de um processo de renovação. Esta abordagem, geralmente mais dispendiosa, é predominantemente adotada em instalações comerciais e em residências de elevado padrão construtivo, devido aos custos associados.

- A tecnologia PLC (*Power Line Communication*): Facilita a transmissão simultânea de energia e dados por meio da rede elétrica. Essa tecnologia viabiliza a prestação de serviços de controle de equipamentos através da infraestrutura elétrica já existente, representando a sua principal vantagem na redução dos custos de instalação, uma vez que aproveita a infraestrutura elétrica preexistente.
- As soluções sem fios: Apresentam, atualmente, um notável crescimento no mercado, impulsionadas pelo aumento de confiabilidade e acessibilidade económica em comparação com as soluções com fios. A sua aplicação em residências já ocupadas não requer grandes remodelações, contribuindo assim para a crescente demanda do mercado por esses dispositivos.

2.3.2. Arquiteturas de sistemas domóticos

A classificação da arquitetura dos sistemas de automação depende do local onde se localiza a “inteligência” do sistema domótico. Pode-se dispor de uma arquitetura centralizada, de uma arquitetura descentralizada, de uma arquitetura distribuída ou de uma arquitetura que é um misto das anteriores [5],[6].

- Arquitetura centralizada: Numa arquitetura centralizada todos os elementos, incluindo sensores, atuadores e interfaces, estão interligados a um único sistema de controle central. Esse controlador desempenha o papel de recetor das entradas dos sinais provenientes dos sensores e, após o processamento desses sinais, é o responsável por enviar os comandos e ajustes aos atuadores, para que executem as operações necessárias [5], [6]. Na Figura 1 é possível observar um esquema deste tipo de arquitetura.

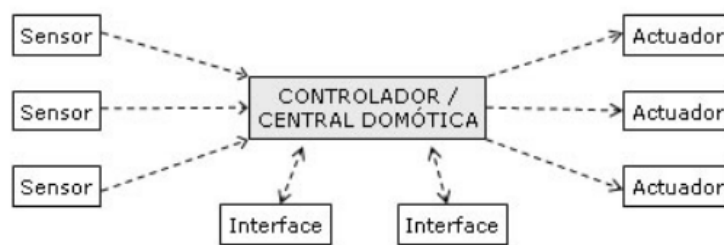


Figura 1 - Arquitetura centralizada [7].

As principais vantagens destes sistemas assentam no facto de normalmente os sensores e atuadores utilizados serem universais. Como tal, a sua instalação e posterior utilização é consideravelmente mais simples do que a dos restantes sistemas. Por outro lado, estes são sistemas mais económicos porque os sensores universais são consideravelmente mais baratos, e também porque se recorre à utilização de apenas um controlador central para gestão e controlo total do sistema domótico.

As desvantagens que se podem encontrar nestes sistemas são a possibilidade de ocorrência de algum tipo de falha na unidade central da gestão e controlo deste sistema, sendo que, caso tal aconteça, todo o sistema domótico da habitação se encontra comprometido.

- **Arquitetura descentralizada:** Numa arquitetura descentralizada observa-se a presença de diversos dispositivos com capacidade de processamento inteligente, cada qual desempenhando funções específicas pertinentes às exigências do sistema de automação, estando esses equipamentos interligados por uma rede, permitindo que comuniquem entre si e recebam e enviem sinais aos sensores e atuadores [6]. Na Figura 2 é representado um sistema com uma arquitetura descentralizada.

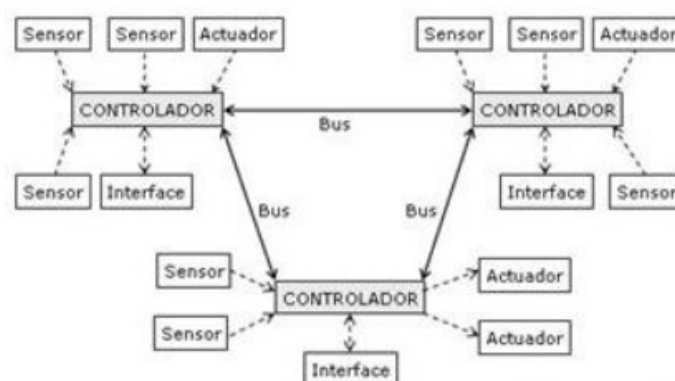


Figura 2 - Arquitetura descentralizada [7].

As vantagens desta arquitetura baseiam-se na grande flexibilidade devido à utilização e distribuição de diversos controladores permitindo assim uma expansão mais eficiente do sistema comparativamente a um sistema centralizado. De igual forma, existe uma maior robustez graças à utilização de diversos

controladores, pois em caso de falha apenas é afetado o funcionamento do conjunto de elementos interligados ao controlador em que ocorreu a falha, o que, dependendo da sua programação, pode ou não comprometer o resto do sistema.

Contudo, este tipo de sistemas também possui desvantagens, uma vez que a utilização de diversos controladores distribuídos pela rede torna o custo deste sistema mais elevado.

- **Arquitetura distribuída:** Nesta arquitetura, os sensores, atuadores e interfaces estão ligados a um barramento comum, sem necessidade de um elemento central, cada subsistema encarrega-se de realizar a sua tarefa. Com este tipo de ligação a resiliência é maior, uma vez que se um nó falhar não irá afetar os restantes de dispositivos [5].



Figura 3 - Arquitetura distribuída [7].

Algumas das vantagens que este tipo de sistemas oferece são uma maior flexibilidade devido à utilização e distribuição de diversos controladores e uma maior robustez.

Como desvantagens deve ser mencionado o custo, que é significativamente mais elevado e a necessidade de um maior tempo de planeamento, devido à separação de funcionalidades por controladores, dificultando a programação individual de cada um dos respetivos controladores ou interfaces utilizados.

2.4. *Internet of things (IoT)*

O avanço tecnológico tem vindo a acelerar a conectividade sem fios e a miniaturização, permitindo a extensão do conceito da domótica para muitos outros aspetos, dando origem ao que em inglês é chamado de *Internet of Things* (Internet das Coisas), abreviado como IoT [8].

A IoT pode ser definida como uma rede de dispositivos eletrônicos pequenos, de baixo custo e de baixo consumo energético, na qual as comunicações ocorrem sem intervenção humana. Cada dispositivo atua como um “nó inteligente” da rede, detetando informações e executando um processamento de sinal de baixo nível para filtrar sinais de ruído e reduzir a largura de banda necessária para as comunicações de nó para nó. Finalmente, de referir que os nós enviam essa informação para a nuvem, de forma segura e para proteger, armazenar e processar os dados [8], [9].

A IoT está intimamente ligada à automação, pois permite que qualquer máquina comunique com outra máquina, o que é designado por *Machine to Machine* (M2M), de forma automatizada para um determinado fim. Existem inúmeros exemplos de automação associados à IoT, pois tudo o que possa ter um sensor, pode ser automatizado com um determinado objetivo, facilitando e melhorando a qualidade de vida das pessoas, o que permitiu o lançamento de vários conceitos e serviços como as cidades inteligentes, as casas inteligentes ou a telemedicina [10].

Para tornar possível a implementação da internet das coisas é necessário resolver questões referentes aos protocolos de comunicação utilizados, a forma de fornecer energia aos dispositivos, minimizar os seus consumos de energia, entre outras questões que serão abordadas de seguida.

2.4.1. Principais tecnologias

Com a exponencial evolução da IoT e o aumento da taxa de aceitação por parte da sociedade sobre este tema, todos os dias novas tecnologias são desenvolvidas, e para as acompanhar, novos protocolos são criados. Serão de seguida descritos os protocolos mais utilizados e mais conhecidos para aplicações em casas inteligentes, sendo que este resumo é apenas uma pequena parte desse grande universo que é a IoT, até porque na IoT não existe uma solução única para um projeto, sendo sempre necessário encontrar o protocolo que melhor se adapte à situação.

Algumas das características mais relevantes para seleccionar o protocolo adequado são a taxa de transmissão, o alcance, o consumo de energia, as topologias que podem ser implementadas, a qualidade e confiabilidade, a segurança, a capacidade da rede e o custo dos equipamentos necessários [11].

Existem várias tecnologias sem fios que podem ser utilizadas em projetos de IoT, nomeadamente: Wi-Fi, ZigBee, Bluetooth *Low Energy*, MQTT, Z-Wave, Sigfox e LoRa.

2.4.1.1. Wi-Fi

Wi-Fi é uma tecnologia de rede sem fios que permite a transmissão de dados por meio de ondas de rádio. É amplamente utilizado para que dispositivos móveis e outros equipamentos se liguem à Internet e troquem informações entre si, criando uma rede WLAN (*Wireless Local Area Network*), baseada no padrão IEEE 802.11. "Wi-Fi" é uma abreviatura do termo inglês *Wireless Fidelity* [12].

Ao longo dos anos têm sido criadas diferentes versões desta norma, tendo em vista a otimização da largura de banda e o aumento dos débitos, além de melhorar a compatibilidade e segurança. As principais versões que merecem ser destacadas são as seguintes:

- 802.11**b** - Lançada em 1999 foi o primeiro padrão popular e barato. Utiliza o espectro não licenciado a 2,4 GHz, permite débitos até 11 Mbit/s, usa *Direct Sequence Spread Spectrum* (DSSS) na camada física e utiliza 11 canais de frequência (3 não sobrepostos).
- 802.11**a** - Foi uma versão lançada no mesmo ano que à anterior. Opera a 5 GHz, permite débitos até 54 Mbit/s e usa modulação OFDM (*Orthogonal Frequency Division Multiplexing*).
- 802.11**g** - Lançada em 2003 e opera a 2,4 GHz. Esta norma, permite débitos até 54 Mbit/s e usa modulação OFDM.
- 802.11**n** - No ano de 2009, foi lançada esta versão que permite a utilização de MIMO (*Multiple Input Multiple Output*) que refere-se a uma técnica em que múltiplos transmissores e receptores são usados para melhorar o desempenho da comunicação sem fios. A principal ideia por trás do MIMO é aumentar a capacidade e a confiabilidade da transmissão de dados sem fios, explorando a diversidade espacial [13]. Esta versão do Wi-Fi usa uma modulação OFDM, opera nas duas bandas (a 2,4 e 5 GHz) e permite débitos até 150 Mbit/s.
- 802.11**ac** - Lançada no ano 2014, opera a 5 GHz, permite débitos até 1 Gbit/s (Gigabit Wi-Fi).

O Wi-Fi é amplamente utilizado em casas inteligentes e em aplicações de IoT devido à sua conveniência, flexibilidade e elevada velocidade de transmissão de dados. Esta tecnologia permite que diversos dispositivos se conectem a uma rede sem a necessidade de cabos físicos, o que facilita a instalação e a mobilidade dos dispositivos. Além disso, o Wi-Fi é capaz de transmitir dados em alta velocidade, permitindo uma comunicação eficiente entre os dispositivos conectados.

Nas casas inteligentes, o Wi-Fi é usado para conectar uma ampla variedade de dispositivos, como termostatos, câmaras de segurança, lâmpadas, eletrodomésticos, sistemas de entretenimento e muitos outros. Isso permite que esses dispositivos comuniquem entre si e com o utilizador por meio de uma rede doméstica. Com o uso de aplicações móveis ou interfaces de voz, os utilizadores podem controlar e monitorizar os seus dispositivos conectados remotamente, proporcionando maior conveniência, eficiência energética e segurança nas suas residências.

Como desvantagens, pode-se referir que a confiabilidade do Wi-Fi pode variar dependendo da qualidade do *router*, e que o uso do Wi-Fi em casas inteligentes pode também apresentar riscos de segurança, se a rede Wi-Fi não estiver adequadamente protegida, fazendo com que *hackers* possam ter acesso aos dispositivos inteligentes. Outra desvantagem é que à medida que o número de dispositivos aumenta, a largura de banda disponível é dividida entre eles, o que pode resultar numa conexão mais lenta.

2.4.1.2. ZigBee

A norma ZigBee é uma especificação baseada no padrão IEEE 802.15.4 para um conjunto de protocolos de comunicação de alto nível, utilizados para criar redes de área pessoal com pequenos rádios de baixa potência, utilizados na domótica, recolha de dados meteorológicos, médicos, etc. O *standard* foi desenhado pela ZigBee Alliance para aplicações que precisam de comunicações seguras, com baixas taxas de transmissão e com o objetivo de maximizar a vida útil das baterias dos dispositivos [14].

Este protocolo especifica que a comunicação pode ocorrer em 3 bandas diferentes, de acesso livre, destinadas a aplicações científicas, industriais e médicas (ISM):

- 868 – 868,8 MHz;
- 902 – 928 MHz;
- 2400 – 2483,5 MHz.

Apesar de qualquer uma destas bandas poder ser utilizada pelos dispositivos 802.15.4, a banda dos 2,4 GHz é a mais utilizada, uma vez que é uma banda livre na maioria dos países do mundo. A banda dos 868 MHz é específica para utilizações na Europa, e a banda dos 902-928 MHz só pode ser utilizada nos Estados Unidos, Canadá e em alguns outros países [15]. O *standard* 802.15.4 especifica que a comunicação deve ocorrer em canais com 5 MHz.

Na banda dos 2,4 GHz o débito de transmissão máximo especificado é de 250 kbit/s, com 16 canais disponíveis (desde os 2,280 GHz até os 2,405GHz). No entanto, devido à complexidade adicionada pelos mecanismos de segurança e encriptação dos dados, o ritmo de transmissão atingido é na realidade metade do especificado. Por sua vez, nas bandas dos 915 MHz e 868 MHz estão disponíveis taxas de transmissão de 40 kbit/s com 10 canais de comunicação e 20 kbit/s com um canal de comunicação, respetivamente [15].

Nas bandas dos 868 MHz e 915 MHz é utilizada modulação BPSK (*Binary Phase Shift Keying*), sendo que na banda dos 2,4 GHz o protocolo IEEE 802.15.4 utiliza modulação O-QPSK (*Offset Quadrature Phase Shift Keying*) com forma de meia sinusóide para modular a portadora de radiofrequência.

2.4.1.2.1. Tipos de dispositivos

Existem 3 tipos de dispositivos ZigBee com diferentes papéis na construção da rede [14]:

- **Coordenador:** É o encarregado de formar a rede, atribuir endereços, realizar a manutenção da rede e o encaminhamento da informação entre dispositivos. Existe apenas um por rede, mas pode servir de ponte entre várias redes, e como é quem gere toda a informação, deve estar sempre alimentado.
- **Router:** Permite que mais nós se juntem à rede, aumentando o seu alcance físico. Pode também efetuar funções de controlo ou monitorização, para além do reencaminhamento de dados. A sua existência é opcional.
- **Nó terminal:** É um nó sensor ou atuador, que devido à sua posição na rede, está apenas ligado a um nó *router* ou diretamente ao coordenador. Desempenha uma

ação de controlo ou monitorização através de dispositivos que lhe estejam associados (sensores, microcontrolador, atuadores, etc.). Geralmente, procura-se que estes nós consumam pouca energia, pois são normalmente alimentados por baterias e por isso, são muitas vezes colocados em modo de adormecimento.

2.4.1.2.2. Topologias de rede

Agora que já foram apresentados os diferentes tipos de dispositivos que podem coexistir numa rede ZigBee, serão de seguida abordadas sucintamente as diversas topologias de rede, dando atenção aos principais conceitos inerentes a cada uma delas [14].

- Topologia em estrela: Nesta configuração de rede, o Coordenador assume a responsabilidade integral pelo controlo da rede, desempenhando um papel central ao estabelecer comunicação direta com todos os nós terminais. Desta forma, o Coordenador é encarregado de iniciar toda a rede e de supervisionar todos os dispositivos a ela conectados. Entretanto, é importante notar que, nessa topologia, toda informação transmitida necessita de passar pelo nó central, o que pode ser considerado uma desvantagem devido à possível concentração de tráfego e à dependência do funcionamento contínuo desse nó central [15]. Na Figura 4 é apresentada uma topologia em estrela.

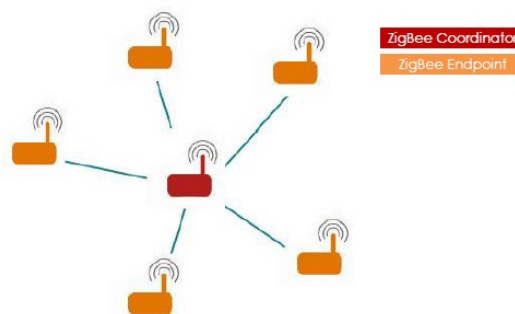


Figura 4 - Topologia em estrela [15].

- Topologia em árvore: É estabelecida uma estrutura hierárquica na qual os dados e as mensagens de controle são distribuídos. No topo da hierarquia existe o Coordenador que representa o núcleo da rede. A partir dele, originam-se diversas

ramificações primárias, que, por sua vez, se desdobram em ramificações secundárias. As ramificações secundárias têm a capacidade de comunicar entre si, utilizando um *Router*. No entanto, é crucial observar que as ramificações primárias só podem comunicar entre si ao passar a informação pelo Coordenador, reforçando uma estrutura de comunicação mais centralizada [15].

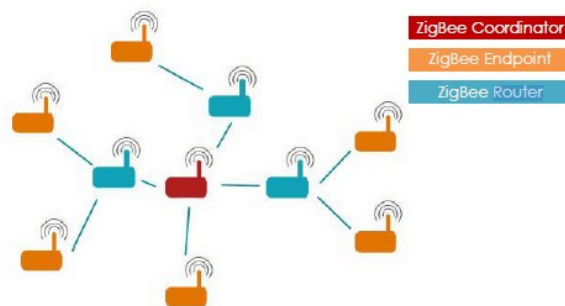


Figura 5 - Topologia em árvore [15].

- Topologia em malha: Neste tipo de topologia os dispositivos Coordenador e *routers* têm a liberdade de enviar informações para qualquer outro dispositivo na rede, indicando a ausência de centralização na circulação de dados. Nesta situação o Coordenador apenas registra a entrada e saída de dispositivos na rede, assumindo um papel passivo no que diz respeito ao fluxo da informação. Esta topologia é robusta e possui redundância, resolvendo na maioria das situações o problema da perda da ligação por parte de algum dos dispositivos [15]. Na Figura 6 visualiza-se um exemplo de uma topologia em malha.

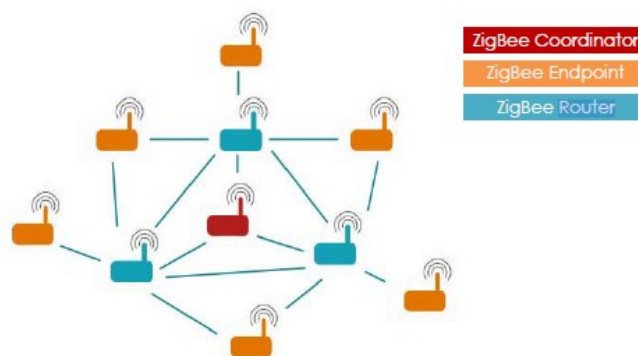


Figura 6 - Topologia em malha [13].

2.4.1.2.3. Vantagens e desvantagens

O protocolo ZigBee, além do baixíssimo consumo de energia, possui também outras vantagens, como por exemplo o baixo custo dos equipamentos, não usa endereçamento IP, tem grande flexibilidade com a possibilidade teórica de existirem até 65000 nós, baixo tempo de latência devido ao curto tempo de ligação e rápida transição para o modo de funcionamento [16].

Como qualquer outro protocolo, também existem desvantagens ou limitações, como as baixas taxas de transferência, o que mesmo para aplicações simples inviabiliza certas ações que precisam de taxas de transferência superiores [16]. O Zigbee opera usualmente na faixa de frequência de 2,4 GHz, que é suscetível a interferências de outros dispositivos que utilizam a mesma faixa, como dispositivos Wi-Fi e dispositivos Bluetooth. Outra das desvantagens está relacionada com a interoperabilidade entre dispositivos Zigbee de diferentes fabricantes porque embora existam padrões definidos, a implementação exata pode variar, o que pode resultar em incompatibilidade entre dispositivos de diferentes marcas.

2.4.1.3. Bluetooth Low Energy (BLE)

Bluetooth Low Energy (BLE) ou *Bluetooth Smart* é uma tecnologia de redes de área pessoal sem fios, concebida e comercializada pelo *Bluetooth Special Interest Group* [17]. O *Bluetooth Low Energy* tem uma abordagem diferente da apresentada pelo *Bluetooth* Clássico e não pretende substituir o seu antecessor, mas sim oferecer novas soluções para ampliar a sua utilização.

A sua arquitetura simples e barata, aliada ao seu baixo consumo energético e ao seu reduzido tamanho, tornam esta tecnologia ideal para a utilização em sensores e aplicações que exigem eficiência energética, pois a vida útil da bateria pode ser estendida.

Existem duas versões de dispositivos *Bluetooth Low Energy*, o *Single-mode* e o *Dual-mode*. Um dispositivo *Single-mode* pode comunicar com dispositivos *Single-mode* e *Dual-mode*, mas não com dispositivos que suportem apenas *Classic Bluetooth*. O *Dual-mode* já permite esse tipo de comunicação, pelo que esses dispositivos conseguem trocar dados com qualquer versão de *Bluetooth*, tal como é possível observar na Figura 7 [17].

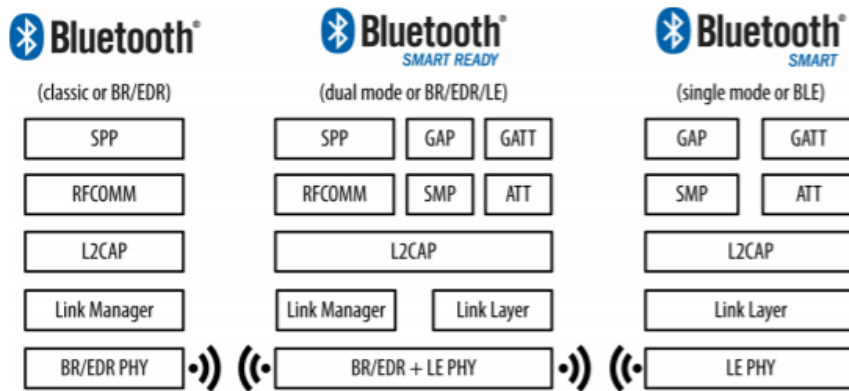


Figura 7 - Configurações entre versões de Bluetooth e tipos de dispositivos [17].

O BLE possui uma taxa de transmissão teórica de 1 Mbps, com um alcance de 50 metros em linha de vista. Este protocolo utiliza a faixa de frequências dos 2,4 GHz que é dividida em 40 canais com 2 MHz de largura de banda, que vão desde os 2,4000 GHz até aos 2,4835 GHz. Tal como pode ser visto na Figura 8, 37 dos canais são usados para a transmissão de dados, sendo que os últimos 3 canais (37, 38 e 39) são utilizados para realizar *Advertising* [17].

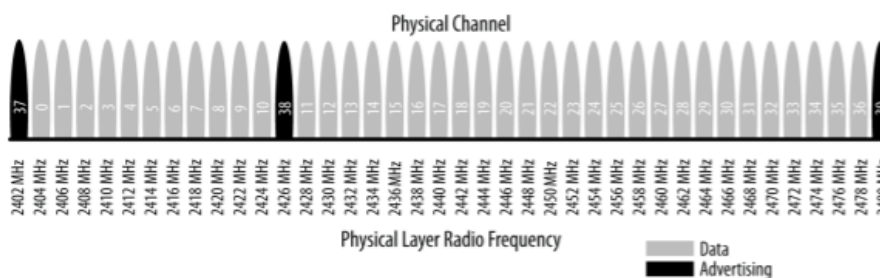


Figura 8 - Canais de frequência [17].

O BLE utiliza a técnica de salto em frequência (*Frequency Hopping Spread Spectrum - FHSS*) entre canais para evitar interferências com outros dispositivos que utilizam a mesma faixa de frequências. A modulação escolhida para codificar os dados enviados é a *Gaussian Frequency Shift Keying (GFSK)*, que é a mesma modulação utilizada pelo Bluetooth clássico.

2.4.1.3.1. Topologias de rede

Um dispositivo *Bluetooth* de baixa energia pode comunicar com o mundo exterior de duas maneiras: através de *broadcasting* ou através de *connections*. Cada mecanismo

tem suas próprias vantagens e limitações, e ambos estão sujeitos às diretrizes estabelecidas pelo GAP (*Generic Access Profile*) [17].

- *Broadcasting and Observing*: Utilizando ligações do tipo *broadcasting* é possível enviar dados para qualquer dispositivo que esteja como recetor dentro do alcance [17]. Na Figura 9 pode-se observar uma ilustração deste mecanismo.

Existem dois perfis de dispositivos neste método:

- *Broadcaster*: Envia pacotes não conectáveis de *advertising* periodicamente para qualquer dispositivo disposto a receber.
- *Observer*: Verifica repetidamente as frequências predefinidas para receber qualquer tipo de *advertising* não conectável que tenha sido enviado.

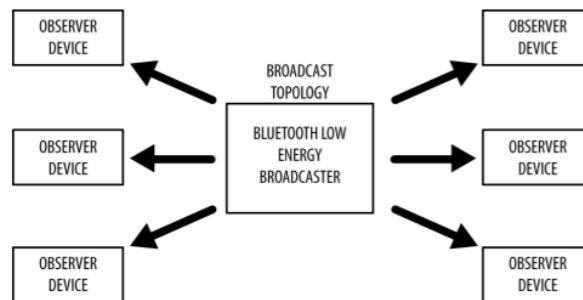


Figura 9 - Topologia Broadcast [17].

Este método é importante porque é a única forma de um dispositivo transmitir dados para vários outros dispositivos ao mesmo tempo. Uma das principais limitações é que não há segurança, pois qualquer dispositivo é capaz de receber os dados que estão a ser transmitidos.

- *Connections*: Este método é utilizado para transmitir dados de forma bidirecional, sendo um mecanismo permanente que permite o intercambio de dados periodicamente entre dois dispositivos [17]. Na Figura 10 é exemplificado este tipo de topologia.

Existem dois perfis fundamentais neste tipo de método:

- *Central (Master)*: Este dispositivo verifica repetidamente as frequências predefinidas em busca de pacotes de *advertising* para se conectar. A partir do momento em que a conexão é estabelecida o *master* gere todos os tempos e inicia o intercambio de dados periodicamente [17].

- *Peripheral (Slave)*: É um dispositivo que envia pacotes de *advertising* periodicamente e aceita estabelecer conexões. Uma vez que a conexão esteja ativa, o *slave* segue os parâmetros impostos pelo *master* e intercambia dados regularmente com ele [17].

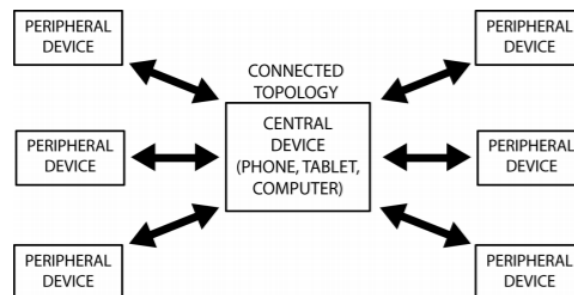


Figura 10 - Topologia Conectada[17].

É de salientar que um dispositivo pode:

- Atuar como dispositivo central e como periférico ao mesmo tempo;
- Um dispositivo central pode estar conectado a vários periféricos;
- Um periférico poder ser conectado a várias centrais.

Para se estabelecer uma conexão, o processo é feito de forma assimétrica, sendo que um anunciante (*slave*) comunica através dos canais de *advertising* indicando que é um dispositivo conectável. Um dispositivo (*Master*) que esteja à escuta nos canais de *advertising* ao encontrar um anunciante, pode transmitir uma mensagem a solicitar uma conexão para o anunciante, o que cria essa conexão ponto-a-ponto entre os dois dispositivos. Ambos os dispositivos podem, em seguida, comunicar usando os canais de dados [17].

A partir do momento que a conexão entre um mestre e um escravo é estabelecida, o canal físico (*Physical Channel*) é dividido em unidades de tempo não sobrepostas chamadas eventos de conexão (*Connection Event*). Dentro de um evento de conexão, todos os pacotes são transmitidos utilizando o mesmo canal de dados. Cada evento de conexão inicia-se com a transmissão de um pacote pelo mestre [17]. Quando o escravo recebe um pacote deve enviar um pacote para o mestre em resposta, como se pode observar na Figura 11.

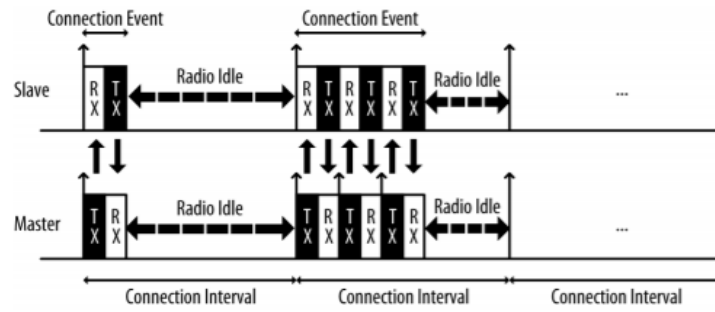


Figura 11 - Eventos de conexão [17].

Enquanto o mestre e o escravo continuam a trocar pacotes, o evento de conexão é considerado ativo. Pacotes enviados através de canais de dados incluem um bit chamado *More Data* (MD) que sinaliza se o remetente tem mais informações a transmitir. Se nenhum dos dispositivos tem mais dados para transmitir, o evento de conexão acabará e o escravo deixa de estar à escuta até ao início do próximo evento de conexão [17].

Para um novo evento de conexão, o mestre e o escravo usam o esquema de salto em frequência de modo a garantir que os dois troquem informação num outro canal num certo instante de tempo, que é calculado através do algoritmo de *frequency hopping* [17].

Esta topologia permite usar menos energia do que o modo *Broadcast*, porque o facto de ambos os dispositivos terem conhecimento de quando é que os eventos de ligação vão ocorrer no futuro, permite que o rádio do periférico seja desligado por mais tempo (*idle mode*), economizando a energia da bateria.

2.4.1.3.2. Vantagens e desvantagens.

O *Bluetooth Low Energy* apresenta várias vantagens uma vez que pode ser usado em aplicações que precisem de enviar pouca quantidade de dados, de forma periódica, com um baixo consumo de energia. É um protocolo bastante conhecido e utilizado, o que faz com que existam muitas soluções e dispositivos disponíveis no mercado, facilitando a integração do protocolo numa casa inteligente, e que permite que os equipamentos sejam mais económicos, comparativamente ao Bluetooth Clássico.

Como desvantagens é importante referir que não é permitida a transferência de grandes quantidades de dados, como por exemplo fotografias ou *streaming* de vídeo, o que limita a sua utilização.

2.4.1.4. Z-Wave

O Z-Wave é um protocolo para automação doméstica de baixo consumo, totalmente sem fios, e que pode ser implementado em qualquer estrutura, nova ou antiga, sem requisitos ao nível de infraestrutura prévia [18]. O utilizador pode começar um sistema com apenas dois dispositivos e aumentar à medida das suas necessidades.

Este protocolo destaca-se pelo seu baixo consumo, fiabilidade, rapidez e alcance. O facto de atuar numa banda estreita, fora dos 2,4 GHz, faz com que não existam interferências de redes sem fios, redes Bluetooth, Wi-Fi ou de aparelhos com transmissão dentro desta banda de frequências. O Z-Wave utiliza frequências livres, mas regulamentadas por órgãos governamentais, na região dos 868,42 MHz na Europa, e dos 908,42 MHz na América do Norte, e possui outras frequências em outros países, em conformidade com as suas regulamentações [18].

O Z-Wave possui uma arquitetura de rede em malha com capacidade para a mensagem saltar até quatro vezes entre nós, o que oferece uma cobertura considerável para a maioria das casas inteligentes, visto que o alcance de comunicação teórica entre dois nós é de 30 metros. Uma rede Z-Wave permite até 232 dispositivos, sendo que é possível interconectar redes se forem necessários mais dispositivos.

Neste protocolo existem dois tipos de nós na rede: os controladores e os dispositivos escravos. Os dispositivos controladores são responsáveis por manter uma tabela de *routing* com as rotas para se atingir cada nó escravo e enviam comandos. Os nós escravos representam os aparelhos controlados, tal como é representado na Figura 12.

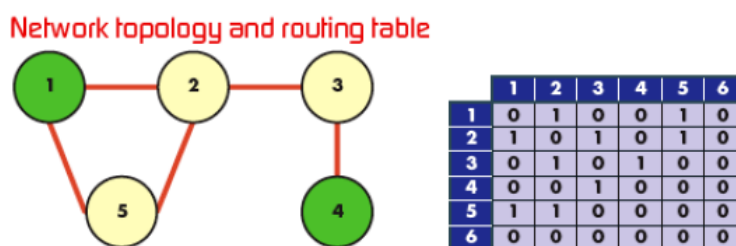


Figura 12 - Exemplo de uma tabela de roteamento Z-Wave [19].

2.4.1.4.1. Vantagens e desvantagens

A principal vantagem relativamente às outras soluções *wireless* apresentadas anteriormente é que este protocolo utiliza uma frequência de transmissão mais baixa, pelo que a degradação do sinal é menor, o que incrementa o alcance da comunicação sempre que seja garantida a mesma potência de transmissão e os mesmos ganhos das antenas, comparativamente aos protocolos mencionados anteriormente. É também um protocolo

simples e flexível, sendo que podem ser adicionados novos dispositivos em qualquer altura, sendo capaz de controlar um único equipamento, uma divisão, um piso ou o edifício inteiro, de acordo com as necessidades.

A principal desvantagem do Z-Wave radica na velocidade de transmissão dos dados, que é baixa, o que impossibilita a transmissão de imagem, som e outros dados.

2.4.1.5. MQTT

O MQTT (*Message Queuing Telemetry Transport*) foi criado e desenvolvido inicialmente pela IBM no final dos anos 90. Como o próprio nome sugere, é um protocolo de mensagens com suporte para comunicação assíncrona [20], que desacopla o emissor e o recetor da mensagem tanto no espaço como no tempo e, portanto, é escalável em ambientes de rede que não são fidedignos. Apesar do nome, este protocolo não está relacionado com filas de mensagens, uma vez que na verdade, é utilizado um modelo de publicação e assinatura.

O protocolo MQTT define dois tipos de entidades na rede: um *broker* e os diferentes clientes.

Um *broker* é um servidor que recebe todas as mensagens dos clientes e, em seguida, encaminha essas mensagens para os clientes de destino relevantes. O *broker* é o “coração” de uma rede MQTT, sendo responsável também por autenticar, criar e manter sessões dos diferentes clientes envolvidos [20].

Um cliente é qualquer dispositivo que possa interagir com o *broker*. Um exemplo de um cliente poderá ser um sensor de *IoT* em campo ou uma aplicação num *data center* que processa dados.

O MQTT pode ser usado através de diferentes meios de transporte, como pode ser o Wi-Fi, sendo que ele opera no nível de aplicação do modelo de referência TCP/IP e utiliza o protocolo de transporte TCP (*Transmission Control Protocol*) para estabelecer uma conexão confiável entre o cliente MQTT e o servidor MQTT.

Os passos para se criar um uma ligação entre um cliente e o *broker* são apresentados de seguida de maneira simplificada e o esquema de ligação está representado na Figura 13:

- O cliente conecta-se ao *broker*. Ele pode assinar qualquer "tópico" de mensagem no *broker*. Essa conexão pode ser uma conexão TCP/IP simples ou uma conexão TLS criptografada para mensagens sensíveis;
- O cliente publica as mensagens no tópico, enviando a mensagem e o tópico ao *broker*;
- Em seguida, o *broker* encaminha a mensagem a todos os clientes que assinam esse tópico.

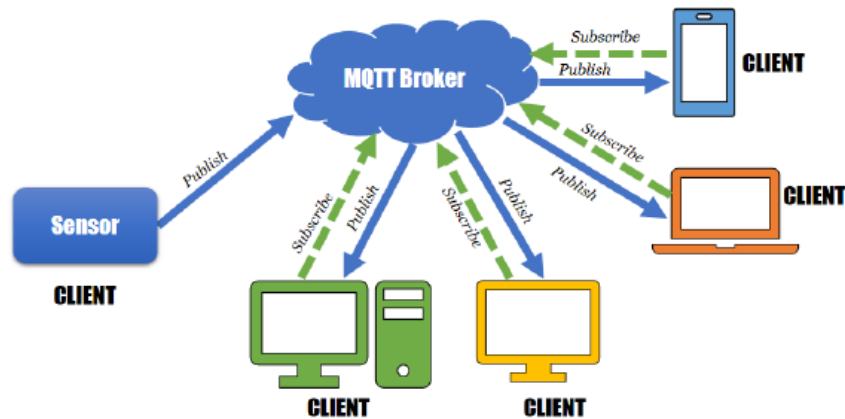


Figura 13 - Esquema MQTT [21].

Os tópicos são uma parte integrante da comunicação entre clientes, pois a comunicação do editor (*publisher*) é transmitida para o *broker* com um determinado tópico, e é de acordo com esse tópico que os clientes subscritores recebem a informação. Observa-se na Figura 14 um exemplo da utilização dos tópicos e da forma como são subdivididos.

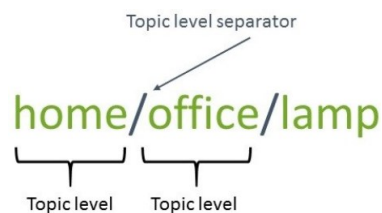


Figura 14 - Separação dos diferentes tópicos [22].

A barra (/) é usada para separar cada nível numa estrutura hierárquica. A utilização de níveis permite organizar a distribuição de mensagens.

Com uma estrutura assim, a subscrição dos tópicos pode ser feita para qualquer um dos níveis. O uso do “cardinal” (#) permite subscrever todos os tópicos a seguir ao separador de nível. A utilização do “mais” (+) no início de uma palavra permite subscrever todos os tópicos com essa palavra [23]. Assumindo o tópico referenciado na figura 14, os modelos abaixo mostram a utilização do “#” e do “+”:

home/# -> subscreve todos os tópicos a seguir ao *home*;

+/lamp -> subscreve todos os tópicos que tenham o nome *lamp* (neste caso seria o tópico *home/office*, se existisse o tópico *home/wc/lamp*, o *home/wc* também seria subscrito).

A nível de QoS (*Quality of Service*) existem três níveis que podem ser definidos pelos clientes, conforme a situação:

- QoS 0: É o nível mais simples e não requer nenhum cuidado especial, sendo que a confiabilidade de entrega da mensagem depende do protocolo. Este modo de transferência é o mais rápido, porém o menos seguro, já que a mensagem pode ser perdida caso o envio falhe. A Figura 15 é um exemplo deste tipo transmissão.



Figura 15 - QoS 0 no protocolo MQTT [24].

- QoS 1: Garante que a mensagem é pelo menos entregue uma vez, pois aguarda que o *broker* confirme a receção da informação enviando uma mensagem PUBACK. Não recebendo o PUBACK, a mensagem continuará a ser enviada até que exista o *feedback* [24].

Neste nível de QoS podem existir mensagens entregues várias vezes, pois em redes de alta latência ou com problemas de conectividade podem acontecer falhas de pacotes. Este é o caso mais usual, mas é importante que os sistemas que usem este nível de QoS saibam gerir mensagens duplicadas. Na Figura 16 representa-se a comunicação neste nível.

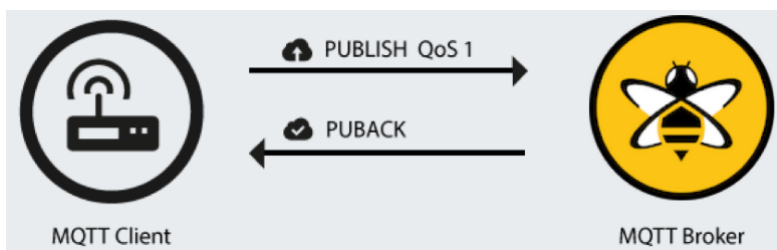


Figura 16 - QoS 1 no protocolo MQTT [24].

- QoS 2: Garante que a mensagem só é entregue mesmo uma vez, o que leva a que a mensagem seja armazenada localmente no emissor e no recetor até que seja processada. Devido à complexidade de configuração este QoS é o menos usado.

Para garantir a segurança deste nível de QoS é necessário o envio de 2 pares de *request response* (um processo chamado de *four-part handshake*), que começa com o envio da mensagem (PUBLISH), seguido da resposta de receção (PUBREC), do aviso do recebimento do PUBREC (PUBREL) e finalmente da confirmação de que o processo foi concluído e de que pode ser feita a exclusão (PUBCOMP) [24]. A Figura 17 ilustra este processo para este nível de QoS.

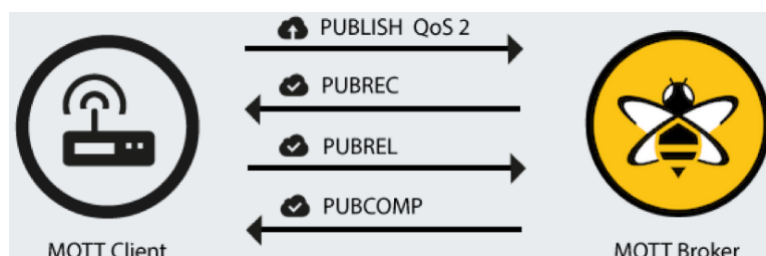


Figura 17 - QoS 2 no protocolo MQTT [24].

2.4.1.5.1. Vantagens e desvantagens

O MQTT tem como vantagem o baixo consumo de recursos, nomeadamente de memória, processador e largura de banda para o envio de mensagens. É igualmente de salientar a pequena quantidade de energia utilizada pelo MQTT.

Como contrapartida a taxa de transmissão é limitada, ocorrem colisões em situações de muita carga e os ciclos de transmissão são mais lentos em comparação com outros protocolos. Além disso, deve ser referida a complexidade na criação de uma rede MQTT globalmente escalável e o facto deste protocolo não possuir encriptação.

2.4.1.6. Comparação entre tecnologias

Nesta secção, serão comparadas de forma sucinta todas as tecnologias referidas anteriormente. Esta comparação pode ser visualizada na tabela 1.

Tabela 1 – Tabela comparativa entre cada tecnologia apresentada.

Protocolo	Frequência	Alcance	Taxa de transmissão	Consumo de energia	Topologia	Segurança
Wi-Fi	2.4 GHz, 5 GHz, 6 GHz (Wi-Fi 6E)	Até 100 metros (interior), até 300 metros (exterior)	Até 9.6 Gbps (Wi-Fi 6)	Alto	Estrela, Malha (Com Wi-Fi Mesh)	WPA3
ZigBee	2.4 GHz (global), 868 MHz (Europa), 915 MHz (EUA)	10-100 metros	Até 250 kbps	Baixo	Estrela, Arvore, Malha	AES-128
BLE	2.4 GHz	Até 50 metros em linha de vista	Até 1 Mbps	Muito baixo	<i>Broadcasting and Observing, Connections</i>	AES-128
Z-Wave	868 MHz (Europa), 908 MHz (EUA)	Até 30 metros	Até 100 kbps	Baixo	<i>Mesh</i>	AES-128
MQTT	Baseado em TCP/IP, normalmente sobre Wi-Fi ou Ethernet	Dependente da rede subjacente	Dependente da rede subjacente	Médio	Cliente-servidor	TLS

3. Equipamentos e software utilizados

Neste capítulo serão apresentados em detalhe os equipamentos e plataformas utilizadas na elaboração deste projeto, com o objetivo de serem utilizados para criar os diferentes cenários domóticos.

3.1. Equipamentos

O propósito desta secção é fornecer uma apresentação concisa, dos equipamentos usados no desenvolvimento deste trabalho.

3.1.1. Raspberry Pi

Um Raspberry Pi é um computador de placa única (*Single Board Computer - SBC*) de arquitetura ARM, de reduzido tamanho e de baixo custo, o que faz com que seja um dos SBC multi-propósitos mais conhecidos e com mais sucesso do mundo [25].

Já foram desenvolvidas mais de 12 versões do Raspberry Pi, tendo sido utilizado neste projeto o modelo 3B+, cujas principais características são as seguintes

- Processador Broadcom BCM2837 a 1,2 GHz 64 bits, Quad Core;
- 1 GB SDRAM @ 400 MHz;
- Wi-Fi 802.11n e *Bluetooth* 4.1 integrados;
- Adaptador micro SD;
- 40 pinos GPIO;
- GPU Videocore IV.

Na Figura 18 são apresentados os pinos de ligação *General Purpose Input/Output* (GPIO) e todos os componentes que compõem o Raspberry Pi 3B+.

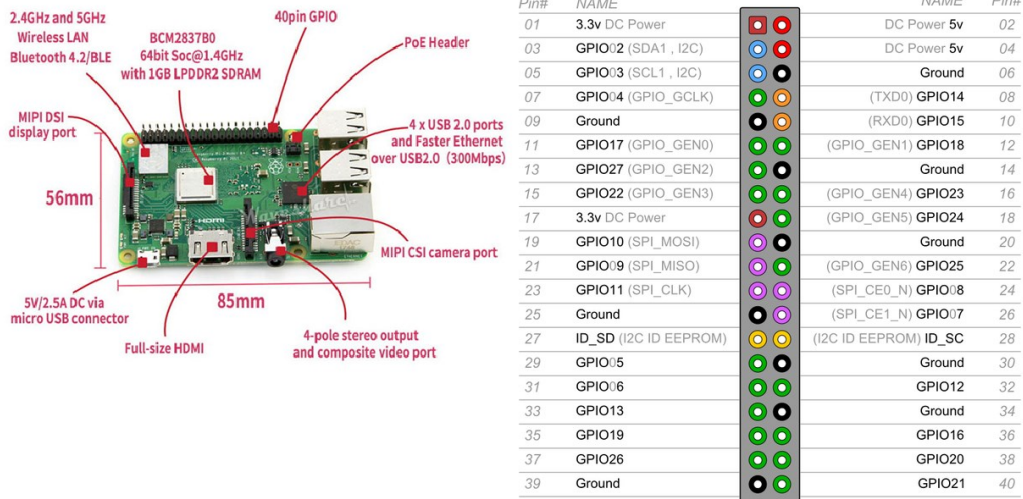


Figura 18 - Raspberry Pi modelo 3B+ e layout dos pinos GPIO [26], [27].

3.1.2. ESP8266

O ESP8266 é um microcontrolador de baixo custo e baixo consumo de energia, projetado para aplicações de Internet das Coisas (IoT) [28]. Desenvolvido pela empresa chinesa Espressif Systems, o ESP8266 integra um processador de 32 bits, capacidade de ligação Wi-Fi e uma variedade de interfaces periféricas, tudo num único *chip* compacto, tal como é possível verificar pelas especificações apresentadas de seguida [28]:

- CPU 32-bit RISC: Tensilica Xtensa LX106 a 80 MHz;
- 64 kB de memória RAM de instruções, 96 kB de memória de dados;
- Memória Flash QSPI externa de 512 kB a 4 MB;
- Wi-Fi 802.11b/g/n que suporta WPA e WPA2.

Na Figura 19 pode ser observada uma placa de desenvolvimento que utiliza como microcontrolador o ESP8266.

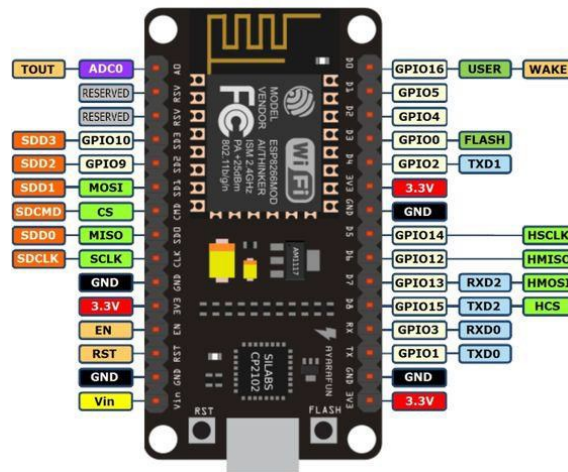


Figura 19 - NodeMCU ESP8266 com layout dos pinos [29].

3.1.3. ESP32

Um ESP32 é um microcontrolador que se destaca por ser uma evolução do ESP8266, incorporando recursos adicionais e várias melhorias. Uma das características distintivas do ESP32 é a presença de dois núcleos de processamento Xtensa 32-bit, proporcionando maior poder de processamento e eficiência energética. Além disso, o ESP32 inclui conectividade Wi-Fi e *Bluetooth* integradas, ampliando as opções de comunicação sem fio o que facilita a comunicação com uma gateway e/ou com uma plataforma de IoT [30]. A Figura 20 é uma representação de uma placa de desenvolvimento (ESP-WROOM-32) que utiliza o ESP32 como microcontrolador. Nessa imagem encontra-se informação relativa a cada um dos pinos disponíveis. Algumas das características do ESP32 são [31]:

- Tensão de operação 3,0 ~3,6 V;
- Temperatura de operação -40 ~ 85°C;
- Corrente de operação (média) 80 mA;
- CPU 32-bit: Tensilica Xtensa 32-bit LX6 microprocessador dual-core a 240 MHz;
- 512 kB de memória SRAM;
- 4 MB de memória Flash QSPI Externa;
- Wi-Fi 802.11b/g/n a 2,4 GHz até 150 Mbit/s;
- *Bluetooth* v4.2 Low Energy.

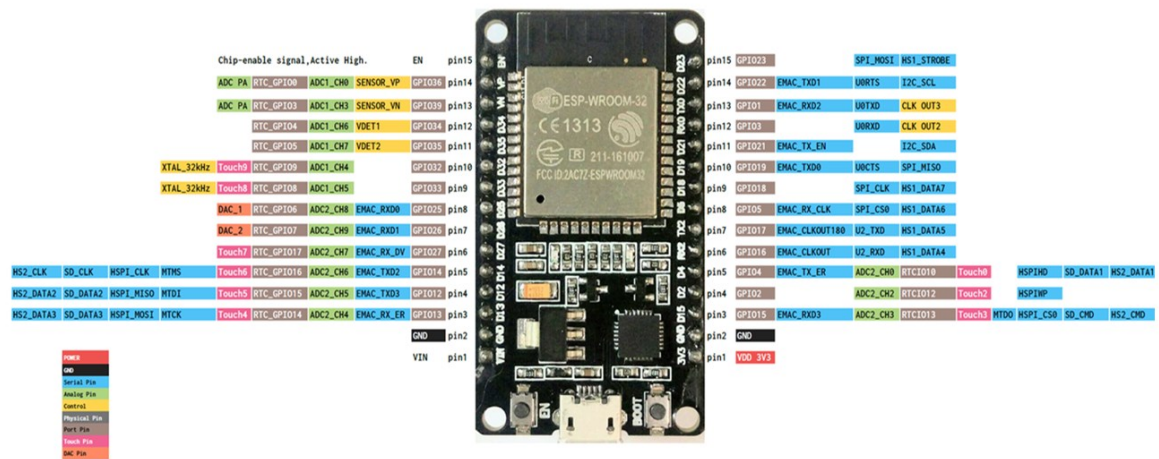


Figura 20 - ESP32 com layout dos pinos [31].

3.1.4. CC2531

O CC2531 é um dispositivo USB que permite implementar soluções baseadas no protocolo *ZigBee* (IEEE 802.15.4) [32]. Este *dongle* USB pode ser ligado diretamente a um PC ou a um Raspberry Pi, e o dispositivo pode ser utilizado como um *sniffer* de pacotes *ZigBee* através de um *firmware* especial, que é programado no dispositivo com um emulador próprio para o efeito, possibilitando a criação de uma rede *ZigBee*. De seguida, são listadas as principais características do dispositivo [32]:

- Microcontrolador de alto desempenho e baixo consumo de energia;
- Memória Flash programável de 256 kB ou 128 kB;
- 8 kB de RAM.
- Suporte para *Carrier-sense multiple access with collision avoidance* (CSMA/CA)

Na Figura 21, é possível visualizar o CC2531 com o respetivo *layout* dos pinos, e na Figura 22 é apresentado o kit completo, requerido para programar o *firmware* no dispositivo.

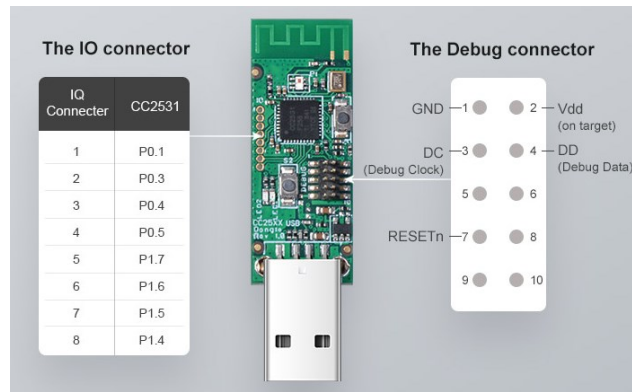


Figura 21 - CC2531 com layout dos pinos [33].



Figura 22 - Kit completo para programar o CC2531 [34].

3.1.5. Arduino UNO

O Arduino é uma plataforma de prototipagem eletrônica de hardware livre e de placa única, projetada, inicialmente, com microcontroladores da família Atmel AVR, com suporte de entrada/saída embutido, e uma plataforma de código aberto usada para a construção de projetos eletrônicos que consiste num software conhecido como IDE (*Integrated Development Environment*), utilizado para escrever e realizar o *upload* do código desde o computador para o microcontrolador [35]. Na Figura 23 pode-se observar um Arduino UNO (Rev. 3) com o respetivo *layout* das diferentes portas. As principais características do Arduino UNO são as seguintes:

- Tensão de operação: 5 V;
- Tensão de entrada recomendada: 7~12 V;

- Tensão de entrada (limite): 6~20 V;
- Número de pinos I/O digitais: 14;
- Número de pinos analógicos: 6;
- Corrente DC máxima por pino I/O: 40 mA;
- Corrente DC máxima no pino 3,3V: 50 mA;
- Memória Flash com 32 kB (ATmega328);
- Memória SRAM: 2 kB (ATmega328);
- Memória EEPROM: 1 kB (ATmega328);
- Velocidade de relógio: 16 MHz.

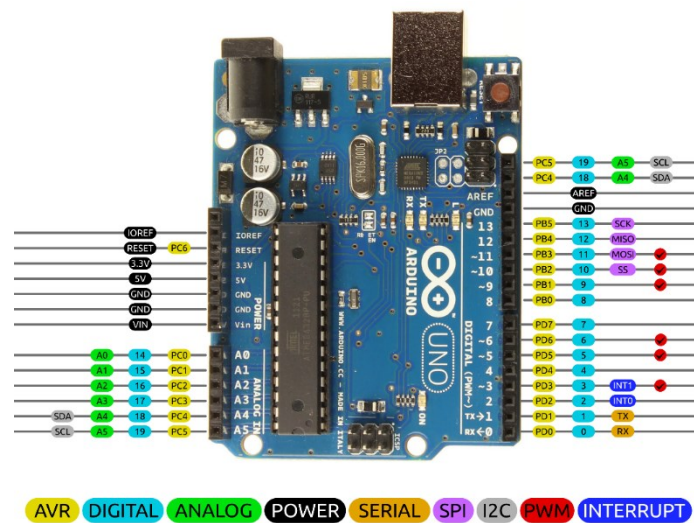


Figura 23 - Arduino UNO com layout das portas analógicas e digitais [36].

3.1.6. Hubitat

O Hubitat Elevation é um *smart hub* que permite a integração de diferentes dispositivos *ZigBee* e *Z-Wave* de forma a criar uma rede de automação residencial [37]. O Hubitat, ao contrário de outros *hubs* que possuem o mesmo objetivo, tem como principal vantagem que tudo o processamento de automação é realizado de forma local, isto quer dizer, que não é necessário manter uma ligação à internet para que o sistema domótico continue a funcionar de forma correta, diminuindo a latência, permitindo uma maior autonomia na execução de automações e garantindo maior privacidade e segurança. Na Figura 24 é apresentado o Hubitat Elevation e na Figura 25 observa-se a vantagem mencionada comparativamente a outras soluções.



Figura 24 - Smart hub da Hubitat Elevation [38].

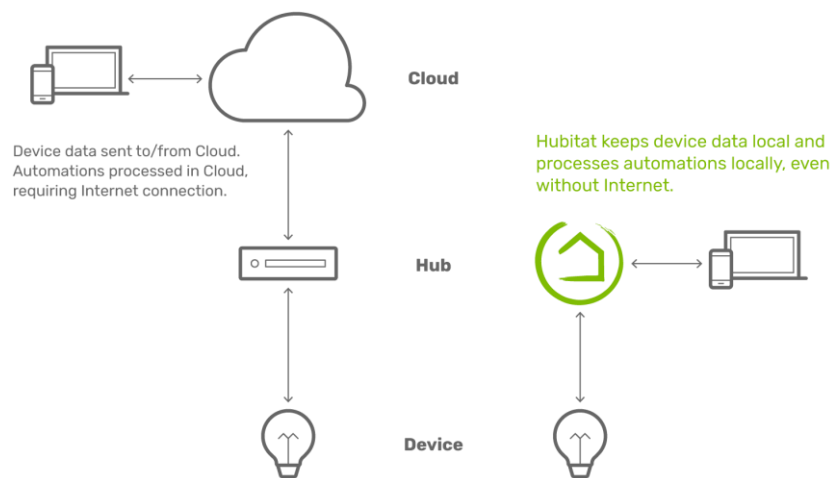


Figura 25 - Automação na nuvem vs automação local com o Hubitat [39].

O Hubitat Elevation possui uma grande variedade de aplicações integradas e tem uma comunidade de utilizadores bastante ativa, para compartilhar ideias, dúvidas e soluções. No entanto, uma das maiores desvantagens é que não está capacitado para suportar Wi-Fi, o que é uma limitação importante a ter em consideração.

3.1.7. Gateway Silvercrest

A Gateway Silvercrest, é uma central domótica pertencente à linha Lidl Smart Home, e destaca-se como uma plataforma que oferece a capacidade de controlar e gerenciar diversos dispositivos inteligentes da marca, utilizando o protocolo ZigBee [40]. Além disso, possibilita a integração com o Google Home, permitindo a utilização dos dispositivos integrados diretamente na plataforma da Google e o controle por meio de comandos de voz com o Google Assistant. Nesse contexto, é importante destacar que esta

central não se diferencia significativamente de outras soluções disponíveis no mercado, trata-se essencialmente, de mais uma opção básica e económica para atender às necessidades dos consumidores.

Uma desvantagem evidente desta central é sua compatibilidade limitada com dispositivos de outras marcas, o que a torna restrita em termos de escolha e implementação. No entanto, sua acessibilidade devido ao baixo preço pode torná-la uma opção atraente para utilizadores iniciantes, que pretendam realizar alguma automatização simples ou controlar alguma tomada ou lâmpada à distância. Adicionalmente, esta *gateway* pode ser utilizada como uma ponte (*bridge*), se for aproveitada a possível integração com o Google Home. Na Figura 26 são apresentadas as principais características da *gateway* Silvercrest.

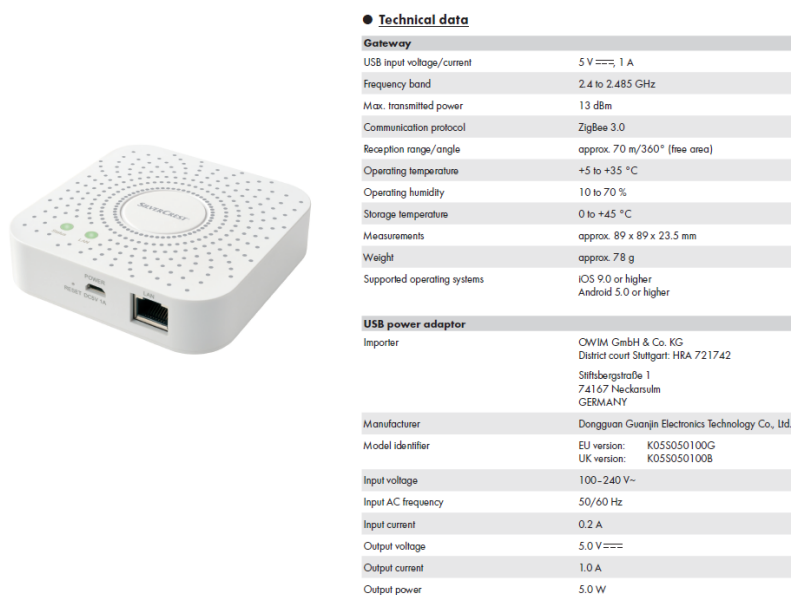


Figura 26 - Central domótica Silvercrest da marca Lidl com as suas características [41].

3.2. Sensores e atuadores

Todos os sensores e atuadores utilizados para a elaboração do trabalho são apresentados de seguida.

3.2.1. AM2302

O dispositivo AM2302 é um sensor de temperatura e humidade digital de baixo custo [42]. A versão utilizada neste trabalho foi um módulo com 3 Pinos (VDD+, DATA

e GROUND) que já inclui um condensador de 100 nF e uma resistência de 5 k Ω , necessária entre o VDD+ e DATA para o seu correto funcionamento (*pull-up*). Na Figura 27 é possível observar o sensor AM2302.

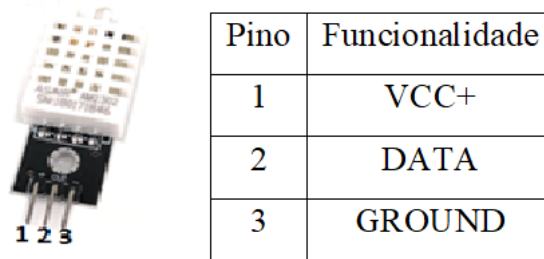


Figura 27 - Sensor de temperatura e humidade AM2302.

As principais características deste sensor são as seguintes:

- Tensão de operação: 3 V – 5 V;
- Resolução: 0,1 °C / 0,1% RH;
- Faixa de medição de humidade: 0-100% RH;
- Faixa de medição de temperatura: -40 ~ 80 °C;
- Precisão na medição de humidade: \pm 2% RH.

3.2.2. PIR

Um sensor PIR (*Passive Infrared Sensor*) é um sensor de movimento que capta radiação infravermelha e deteta alterações no ambiente circundante, resultantes do movimento de pessoas ou de objetos que irradiem calor.

O componente central de um PIR é o sensor piroelétrico, que está fisicamente dividido em duas partes. Em condições de ausência de movimento, ambas as metades recebem uma quantidade equivalente de radiação infravermelha do ambiente circundante. No entanto, quando um objeto em movimento atravessa o campo de visão do sensor, a quantidade de radiação infravermelha recebida numa metade torna-se maior do que na outra, fazendo com que essa discrepância seja interpretada como uma deteção de movimento [43]. Na Figura 28 é possível observar uma explicação do princípio de funcionamento da deteção.

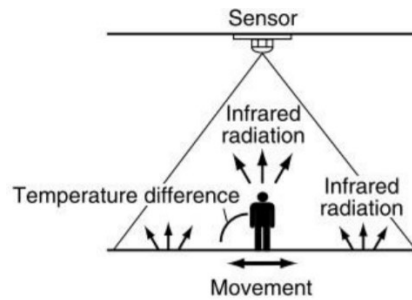


Figura 28 - Princípio de funcionamento do sensor de movimento por infravermelho [44].

O sensor *Grove - Adjustable PIR Motion Sensor* utilizado neste trabalho tem uma saída digital, o que significa que caso seja detetada uma variação na radiação infravermelha o sensor produz uma tensão HIGH (um “1” lógico), caso contrário produziria uma tensão LOW (um “0” lógico). Nas Figura 29 e Figura 30 são apresentados o diagrama de blocos do sensor PIR com saída digital e o sensor utilizado, respetivamente.

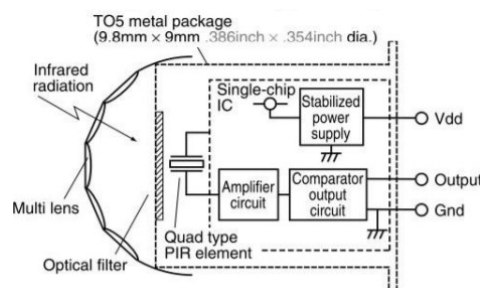


Figura 29 - Diagrama de blocos do PIR [45].

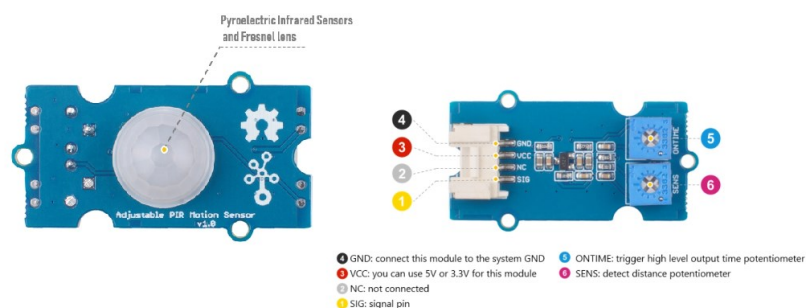


Figura 30 - Sensor utilizado. *Grove - Adjustable PIR Motion Sensor* [46], [47].

As principais características deste sensor são as seguintes [47]:

- Tensão de alimentação: 3 V – 5 V;
- Dimensões: 2,0 x 4,0 cm;
- Angulo de deteção: 120°;

- Distância máxima: 6 metros;
- Corrente de operação (3 V): 100 μ A;
- Corrente de operação (5 V): 150 μ A;
- Temperatura de funcionamento: -20 ~ 70 °C;
- As gamas de velocidade de detecção e de resposta podem ser ajustadas através de 2 potenciômetros soldados na placa.

3.2.3. Relé

Um relé é um interruptor eletromecânico, projetado por Michael Faraday na década de 1830, com inúmeras aplicações possíveis na comutação de contatos elétricos, com o objetivo de ligar ou desligar dispositivos [48].

A movimentação física desse interruptor ocorre quando a corrente elétrica percorre as espiras da bobina do relé, criando assim um campo magnético, que por sua vez atrai a alavanca responsável pela mudança do estado dos contatos. O relé utilizado neste trabalho, um DFR0017 V3.1 é apresentado na Figura 31.



Figura 31 - Relé DFR0017 V3.1 [49].

As principais características deste sensor são as seguintes [49]:

- Corrente de passagem nominal: 10 A (NO) e 5 A (NC);
- Interface: Digital;
- Máxima tensão de comutação 250 VAC / 30 VDC;
- Máxima corrente de comutação 15 A;
- Máxima comutação de energia 250 VA – 240 W.

3.2.4. KY-021

O sensor KY-021, também conhecido como módulo magnético, é um dispositivo eletrônico utilizado para detetar a presença de um campo magnético. Ele consiste num interruptor *Reed*, que é um contato elétrico encapsulado num tubo de vidro hermético, juntamente com uma resistência de *pull-up*.

A operação do sensor KY-021 é relativamente simples. Quando exposto a um campo magnético próximo, o interruptor *Reed* dentro do sensor fecha o circuito, permitindo que a corrente flua através dele, baixando a tensão de saída para o nível lógico baixo (“0”). Por padrão, a resistência de *pull-up* conectada ao interruptor mantém o sinal de saída do sensor no nível lógico alto (“1”) quando o campo magnético não está presente. Dessa forma, o sensor KY-021 indica a chegada de um campo magnético quando o sinal de saída é alterado de alto para baixo.

Uma aplicação comum do sensor KY-021 passa pela deteção da abertura ou fecho de portas ou janelas. O sensor é montado na estrutura fixa, enquanto um ímã é instalado na parte móvel (*e.g.* na porta). Quando a porta é aberta ou fechada, a proximidade do ímã faz com que o interruptor *Reed* no sensor seja ativado ou desativado, respetivamente, alterando assim o sinal de saída do sensor.

É importante ressaltar que o sensor KY-021 é alimentado por uma fonte de energia externa, geralmente de 3,3 V ou 5 V. Além disso, ele pode ser conectado a um microcontrolador, como Arduino, ESP ou Raspberry Pi, para processar o sinal de saída e realizar ações específicas com base na deteção do campo magnético. O sensor KY-021 é apresentado na Figura 32.

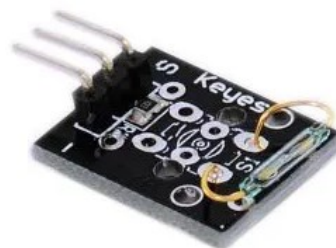


Figura 32 - Sensor KY-021 [50].

As principais características deste sensor são as seguintes:

- Tensão de Operação: 3,3 ~ 5 V;

- Tipo de saída: Digital;
- Resistência: 10 k Ω ;
- Pinos: 3 (V, G, S).

3.2.5. IKEA série TRÅDFRI

O kit IKEA TRÅDFRI, com a referência 403.647.48, ilustrado na Figura 33, é uma tomada inteligente que permite conectar qualquer lâmpada, cafeteira ou outro eletrodoméstico à rede elétrica, sendo possível controlá-la com um comando remoto (botão liga / desliga) [51]. Este dispositivo funciona por meio do protocolo ZigBee.



Figura 33 - Tomada ZigBee da Ikea com comando próprio [51].

3.3. Plataformas

De seguida, serão apresentadas as diversas plataformas de software utilizadas para a elaboração do trabalho.

3.3.1. Home Assistant

O Home Assistant é uma plataforma de automação residencial gratuita e de código aberto, projetada para ser facilmente usada na maioria das máquinas que podem executar o Python 3 [52]. Esta plataforma integra-se com uma grande quantidade de soluções, permitindo ao utilizador utilizar vários protocolos de automação doméstica, como ZigBee, Z-Wave, MQTT, controlar sistemas *wireless* de luz como o Philips Hue e utilizar assistentes virtuais como a Amazon Alexa ou o Google Assistant, entre muitas outras funções. Na Figura 34 são apresentadas algumas das inúmeras soluções disponíveis no Home Assistant.

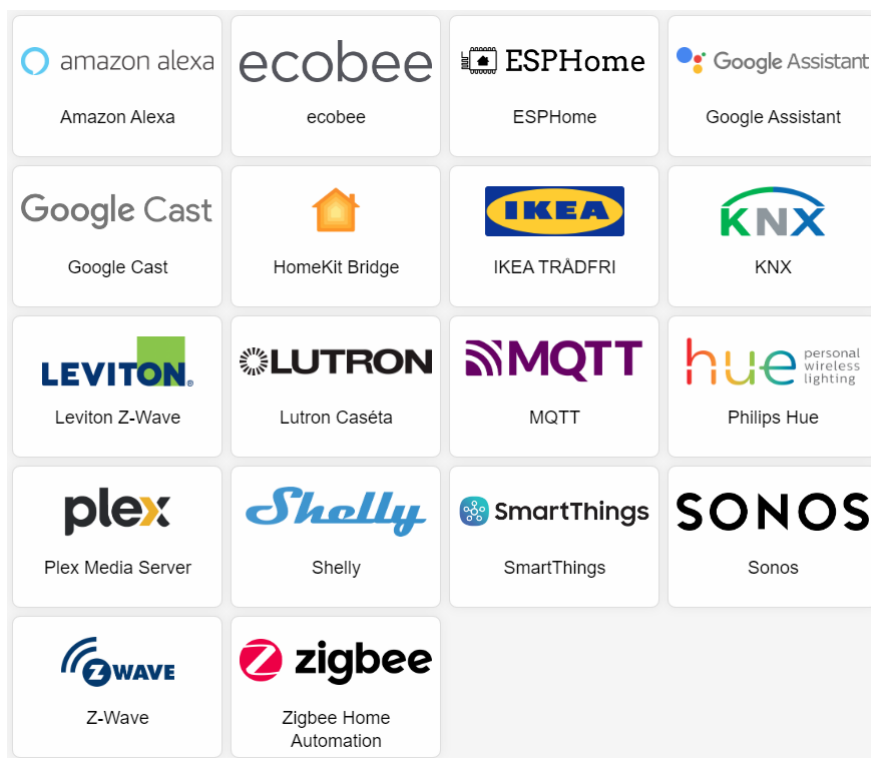


Figura 34 - Soluções integradas no Home Assistant [53].

O Home Assistant é uma alternativa local, ou seja, não depende da *Cloud* para realizar as tarefas, o que permite um tempo de resposta mais rápido, oferece maior segurança, confiabilidade e flexibilidade do que os dispositivos com soluções baseadas na nuvem, que dependem de servidores remotos ou de ligações à internet, como por exemplo o SmartThings da Samsung. É de salientar que ser uma solução local não significa que não seja possível controlar os dispositivos e criar automações fora da rede local da casa.

Uma limitação à utilização do Home Assistant resulta do facto de que o Home Assistant precisa de dispositivos adicionais para poder integrar algumas funcionalidades disponíveis nos “*Add-ons*”, como por exemplo um USB Zigbee Dongle - Sonoff CC2531 para integrar o protocolo ZigBee ou um Z-Stick Gen5 da Aeotec para a integração do protocolo Z-Wave.

De referir que, com o Home Assistant, é possível não só controlar os dispositivos de forma manual, mas também criar automações personalizadas.

3.3.2. Hubitat Elevation

O Hubitat Elevation é um *hub* doméstico de automação residencial com soluções rápidas, privadas, locais e confiáveis [54]. Este dispositivo dispõe de antenas integradas que permitem comunicar com equipamentos Z-Wave e ZigBee, dois dos protocolos de baixo consumo energético mais importantes no que se refere a casas inteligentes. O Hubitat também é compatível com vários aplicativos terceiros, e ao ser uma solução local, dispõe de todas as vantagens mencionadas anteriormente.

O Hubitat tem um software proprietário e não é propriamente a melhor opção para um utilizador menos avançado, que apenas pretenda controlar algumas lâmpadas ou tomadas inteligentes, devido ao seu preço elevado (149,95 USD) e à complexidade na integração de dispositivos e programação de automações. O Hubitat está mais direcionado a utilizadores que pretendam controlar e automatizar grandes quantidades de dispositivos inteligentes espalhados pela casa, e para utilizadores um pouco mais experientes na área da domótica ou com certa sensibilidade na utilização das novas tecnologias, e que poderão aproveitar todas as customizações disponibilizadas.

Comparativamente com o Home Assistant, o Hubitat é muito mais simples e fornece uma solução integral que permite integrar os protocolos Z-Wave e ZigBee sem necessidade de se utilizarem dispositivos externos adicionais, o que é uma vantagem. No entanto, existe uma limitação importante, uma vez que não disponibiliza Wi-Fi, protocolo bastante usado nas casas inteligentes, e é por esta mesma razão que surge outra desvantagem: o Hubitat não pode ser instalado em qualquer lugar da casa devido à necessidade de ter uma ligação direta ao *router* com um cabo Ethernet RJ45 para poder funcionar.

De referir, no entanto, que as duas soluções possuem grandes comunidades de utilizadores, que estão constantemente a procurar tirar o melhor desempenho dos dispositivos e aplicações disponíveis.

3.3.3. Lidl Home

A Lidl Home é uma aplicação desenvolvida para Lidl que permite controlar, automatizar e monitorizar os dispositivos inteligentes da marca de forma simples numa única aplicação [55].

Para funcionar, a *app* Lidl Home requer uma *gateway* (Silvercrest), que é um dispositivo que se liga à sua rede local por meio de um cabo Ethernet RJ45 e permite

comunicar com os dispositivos inteligentes através do protocolo ZigBee. Na Figura 35 é apresentada a *gateway* Silvercrest.



Figura 35 - Gateway Silvercrest [40].

Uma vez configurada a *gateway*, é possível começar a adicionar dispositivos inteligentes à rede ZigBee. A app Lidl Home oferece funcionalidades para controlar os dispositivos inteligente, criar cenários e automatizar tarefas.

Como vantagens desta app pode-se referir que é muito fácil de configurar e de usar, bem como o facto de ser uma aplicação gratuita. Como tudo, também possui desvantagens, sendo muito pouco flexível porque está desenhada para funcionar só com os produtos da Lidl, é muito limitada em termos de tecnologias e de customização, e a *app* não está disponível em todos os países.

4. Instalação e configuração dos sistemas utilizados

Nesta secção será abordado em detalhe todo o processo de instalação e configuração dos dispositivos utilizados, como *gateways* multiprotocolo, nomeadamente o hub da Hubitat, uma *gateway* da Lidl e um Raspberry Pi a executar o Home Assistant.

4.1. Hubitat

Para começar com a instalação do Hubitat Elevation ligou-se o dispositivo ao *router* através de um cabo Ethernet RJ45 e com o cabo de alimentação fornecido pelo fabricante, conectou-se à alimentação por meio de uma tomada elétrica.

Posteriormente, visitou-se o site [56] disponibilizado pela empresa que permite descobrir na rede local o *hub* de forma a poder iniciar todo o processo de configuração do mesmo, para posteriormente passar-se à sua utilização. Para isso, carregou-se na opção “*Find Hubs*” e seleccionou-se o Hubitat de interesse na lista apresentada, para entrar na *Web Interface* do *hub*, a qual pode ser visualizada na Figura 36.

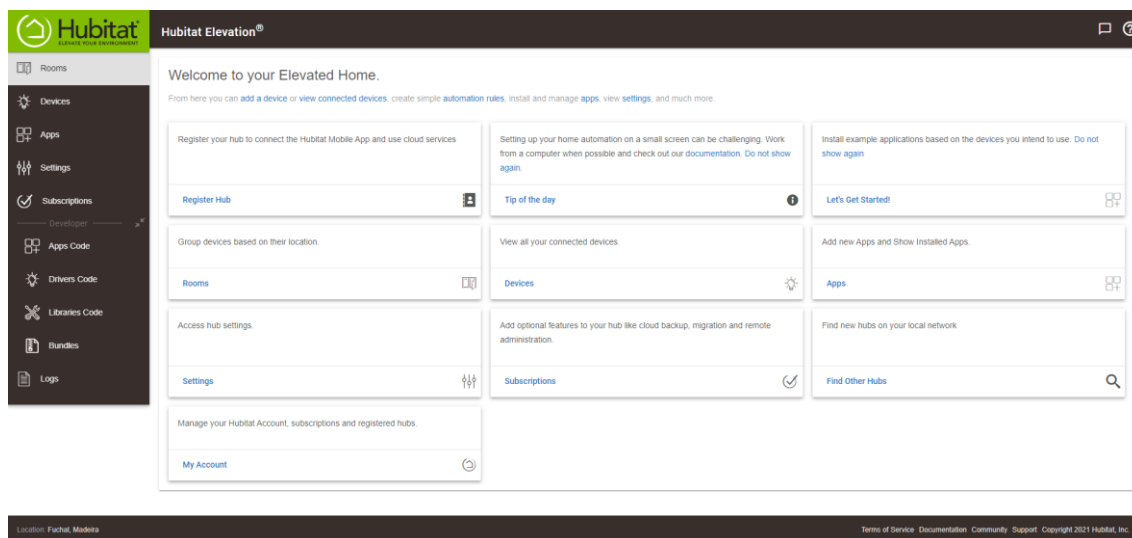


Figura 36 - Web Interface do Hubitat.

A partir desta interface é possível adicionar dispositivos, instalar aplicações, criar grupos de dispositivos baseados na sua localização dentro da casa, regras para automatizações, personalizar a interface do utilizador, entre outras funcionalidades.

4.1.1. Dispositivos

Nesta secção são apresentados os passos para a configuração dos diferentes dispositivos que vão fazer parte de uma *Smart Home* construída com recurso ao Hubitat Elevation. Como dispositivo utilizou-se a tomada apresentada anteriormente da marca IKEA, de forma a trabalhar com o protocolo ZigBee.

4.1.1.1. Tomada ZigBee

Para a integração de uma tomada ZigBee, a abordagem seguida consistiu em colocar a tomada em modo de emparelhamento, para poder ser reconhecida pelo *hub* no momento do *scan* e, posteriormente, acedeu-se à secção “*Devices*” disponível na interface do utilizador do Hubitat. Em seguida seleccionou-se a função “*Add Devices*” e optou-se por escolher a opção “*Start Zigbee Pairing*” no separador “*Add Device Manually*” para iniciar o processo de emparelhamento entre o *hub* e o dispositivo. Depois de ser detetado pelo Hubitat, este permitiu atribuir-lhe um nome e ser gravado, tal como é apresentado na Figura 37.

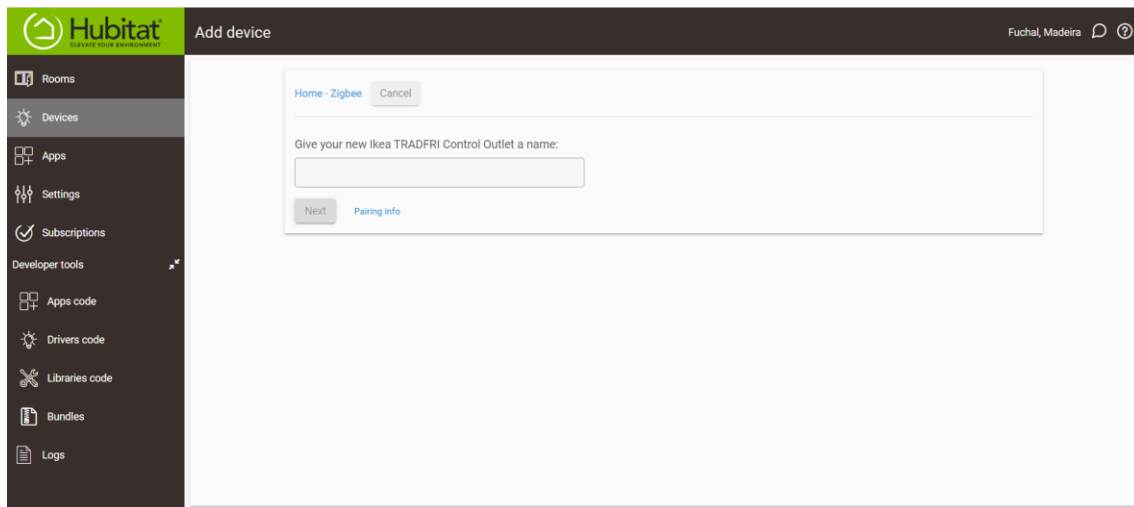


Figura 37 - Integração do atuador IKEA TRÅDFRI no Hubitat.

Para testar se a tomada ficou bem integrada, seleccionou-se a opção “*View device details*” e foi possível encontrar as preferências, eventos, *logs* e diferentes comandos, entre eles o “*On*” e o “*Off*” que permitiam interagir com esse atuador. Confirmou-se que, de facto, o atuador estava a funcionar como expectável. Esse painel pode ser visualizado na Figura 38 onde também aparece o estado do atuador depois de se carregar no comando “*On*”, confirmando que este estava a responder corretamente.

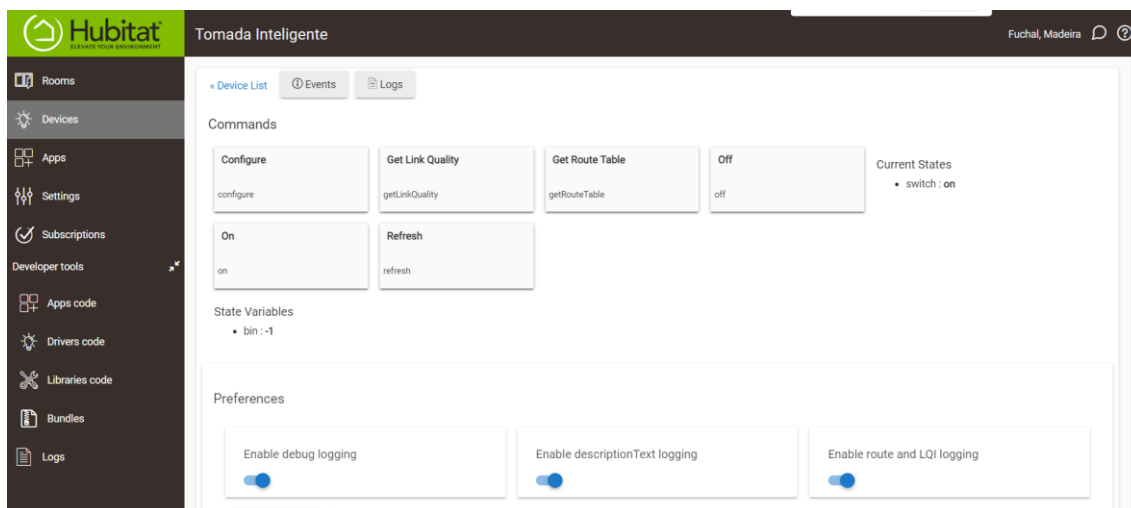


Figura 38 - Comandos e preferências da tomada inteligente disponíveis no Hubitat.

Uma vez integrado o dispositivo, para se poder visualizar e controlar todos os dispositivos disponibilizados na nossa casa inteligente, foi necessária a criação de um *dashboard*.

Para a criação do *dashboard*, selecionou-se “Add Built-In App” na opção “Apps” e, posteriormente, carregou-se em “Hubitat Dashboard”, atribuiu-se um nome e escolheram-se os dispositivos de interesse que integrariam esse *dashboard*, neste caso em particular, só a tomada da IKEA.

A partir desse momento, um painel de controlo foi criado, permitindo a utilização de forma simples da tomada inteligente, e de todos os outros dispositivos que sejam, posteriormente, integrados nesse painel de controlo. Na Figura 39 observa-se o *dashboard* com a tomada integrada.

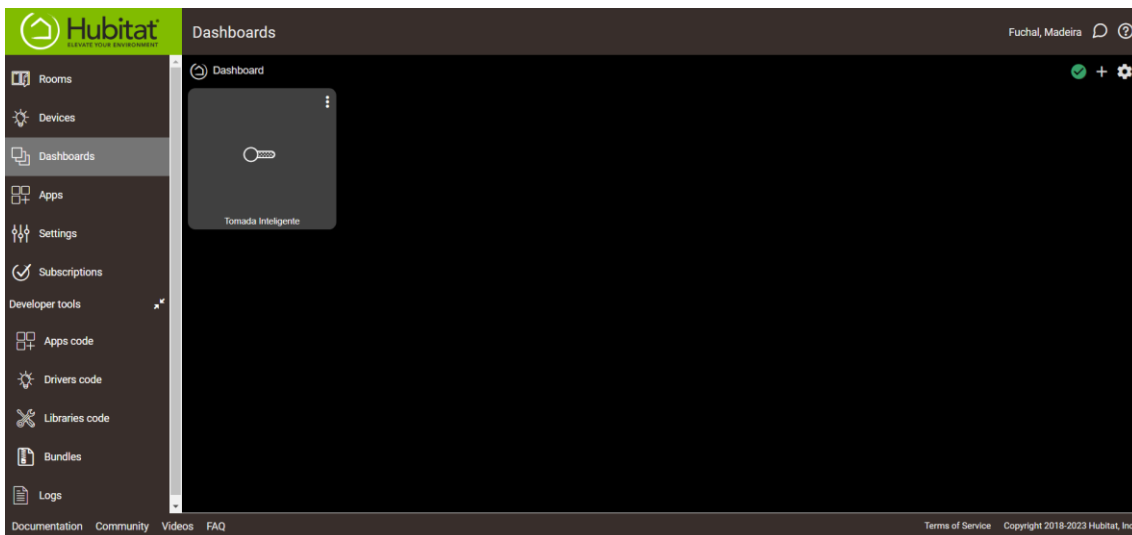


Figura 39 - Dashboard do Hubitat.

4.1.2. Automatizações

Dada a limitação no número de dispositivos disponíveis para o desenvolvimento do projeto nesta plataforma, optou-se por implementar automatizações horaria considerando o momento do nascer e o por do sol, a fim de validar o funcionamento com os dispositivos integrados.

4.1.2.1. Automatização horária

A automatização horária pode ser aplicada, por exemplo, na iluminação externa, com uma lâmpada que é programada para acender num horário específico, geralmente à noite, quando a iluminação é necessária. Da mesma forma, é possível criar uma automatização adicional para desligar a lâmpada num horário em que a luz solar seja suficiente para eliminar a necessidade de iluminação artificial. Além disso, este tipo de automatização pode ser estendido para controlar dispositivos como ventoinhas ou carregadores de dispositivos eletrônicos ou, por exemplo, para ligar as luzes da árvore de Natal todos os dias a partir das 19h, proporcionando maior controlo e eficiência no uso de energia. A flexibilidade desse tipo de automação permite adaptá-la a diversas necessidades e cenários dentro de um ambiente doméstico ou comercial.

Para demonstrar este tipo de automação foi utilizado o atuador da IKEA, previamente integrado no Hubitat, em conjunto com os eventos de *trigger* disponíveis

para criar um interruptor inteligente que levam em consideração os períodos do dia, nomeadamente o pôr do sol e o nascer do sol.

Para iniciar o processo de criação da automatização, foi necessário aceder ao separador “Apps” e selecionar “Rule Machine”. Em seguida, criou-se uma nova regra (“Create New Rule”) denominada “Luzes Exteriores”. O evento de *trigger* escolhido para ativar esta automatização foi programado para ocorrer 10 minutos antes do pôr do sol e as ações correspondentes foram configuradas para ligar a tomada da IKEA imediatamente após o *trigger* acontecer. Posteriormente, foi definida uma segunda ação que exigia esperar por um novo evento para mudar o estado da tomada, o qual foi programado para 10 minutos depois do nascer do sol, desligando as luzes ao colocar a tomada em modo “Off”. Esse ciclo foi programado para se repetir diariamente, garantindo uma iluminação externa adequada ao longo do dia e adaptada às diferentes estações do ano. Na Figura 40 é possível observar a configuração desta automatização com todos os eventos e ações descritos anteriormente.

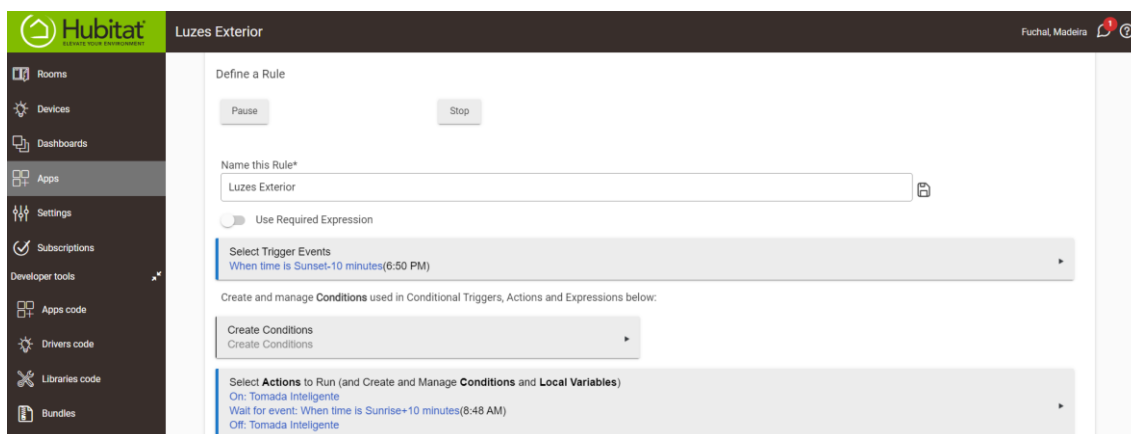


Figura 40 - Automatização de uma tomada no Hubitat.

4.2. Silvercrest

Iniciou-se o procedimento conectando o cabo de rede à porta LAN do dispositivo, estabelecendo assim a conexão com o *router*. Em seguida, procedeu-se à alimentação do dispositivo utilizando o cabo fornecido pelo fabricante.

Como já foi referido anteriormente, esta *gateway* funciona com a *app* Lidl Home, pelo que é necessário descarregar a aplicação desde a *App Store* (se o dispositivo for

Apple) ou da “Google Play Store” (se o dispositivo for Android). Uma vez instalada a aplicação, foi necessário realizar o registo de uma conta com email e palavra-passe, para poder começar-se a adicionar dispositivos e criar cenários de automatização.

Após concluir o processo de registo da conta, procedeu-se à adição da central domótica Silvercrest. Isso foi realizado acedendo à opção “Añadir Dispositivo”, seleccionando a categoria “Gateway”. Como a central já estava previamente conectada ao router, ela foi identificada pelo dispositivo móvel em que o aplicação Lidl Home estava a ser executado, visto que ambos estavam conectados na mesma rede. A conexão foi estabelecida com sucesso. Na Figura 41 é possível observar o procedimento desde o registo de conta até ao estabelecimento da conexão entre o módulo Silvercrest e a aplicação Lidl Home.

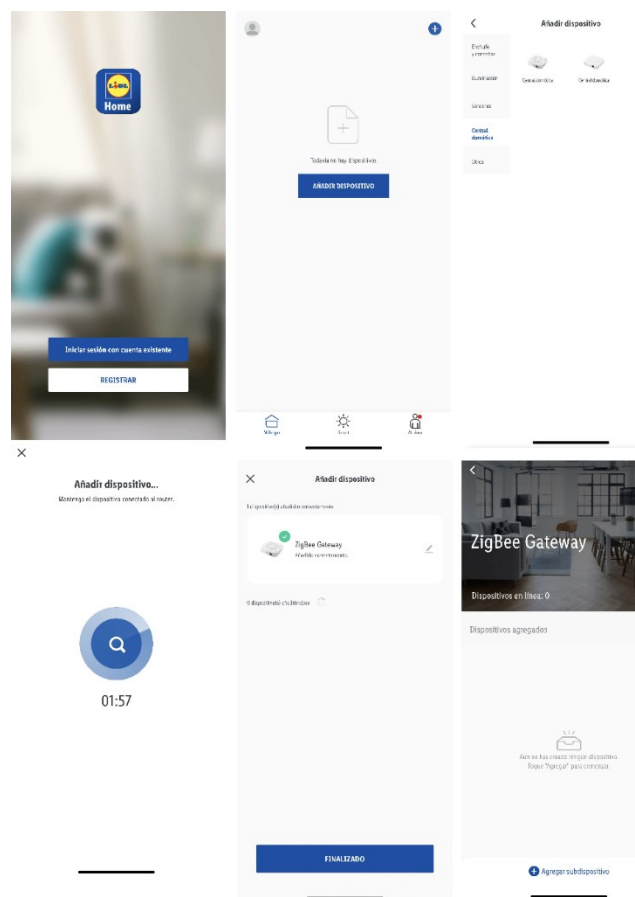


Figura 41 - Integração do módulo Silvercrest na aplicação Lidl Home.

4.2.1. Dispositivos

Nesta secção serão delineados os passos necessários para configurar os dispositivos que foram integrados na central domótica Silvercrest por meio da aplicação Lidl Home. Como exemplo de um dispositivo, utilizou-se a tomada da marca IKEA, mencionada anteriormente.

4.2.1.1. Tomada IKEA

Para integrar uma tomada ZigBee, o procedimento adotado envolveu colocar a tomada no modo de emparelhamento, permitindo que fosse reconhecida pela *gateway* Silvercrest. Em seguida, na interface do utilizador, acedeu-se à secção “Adicionar Subdispositivos” e selecionou-se o dispositivo assim que este foi identificado pela *gateway*, iniciando assim o processo de emparelhamento. Na Figura 42 podem ser observados os passos para integrar a tomada.

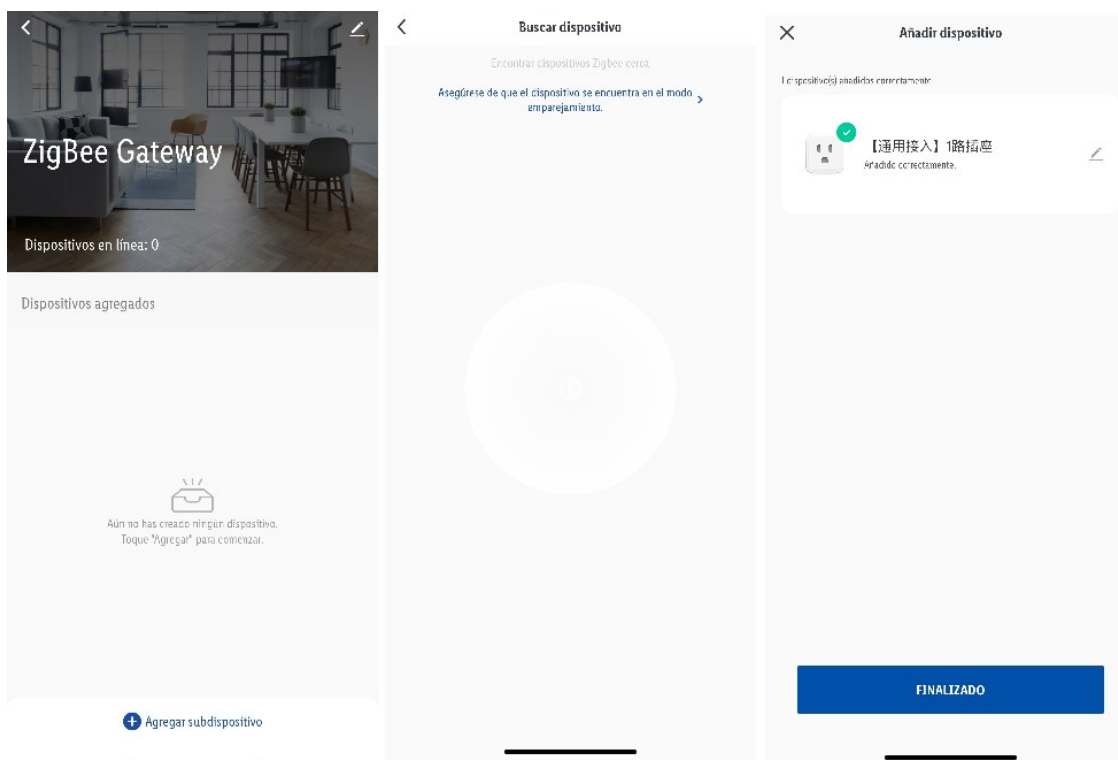


Figura 42 - Emparelhamento da tomada IKEA no Silvercrest.

Uma vez finalizado este processo, foi possível modificar o nome da tomada e atribuir o seu local específico dentro da casa. A tomada estava agora pronta para ser utilizada à distância, de forma manual ou em cenários de automação.

4.2.2. Automatizações

Devido ao reduzido número de dispositivos disponíveis, tomou-se a decisão de implementar automatizações com o atuador integrado no Silvercrest (tomada TRÅDFRI da IKEA), levando em consideração parâmetros como a temperatura, humidade, nascer/pôr do sol, localização do dispositivo móvel, faixas horárias e dias da semana, com a finalidade de avaliar a possibilidade de criar diferentes cenários de automação.

4.2.2.1. Automatização por geolocalização

A automação por geolocalização no âmbito de casas inteligentes é empregue com o propósito de aprimorar o conforto, eficiência e segurança do lar, ajustando automaticamente diversos dispositivos e sistemas com base na localização física dos residentes. A seguir, são apresentados alguns casos e exemplos de como a automação por geolocalização é aplicada em ambientes de casas inteligentes.

- **Controlo de iluminação:** Quando os residentes abandonam a área de geolocalização da residência, o sistema pode desligar automaticamente as luzes para poupar energia. Ao regressarem, as luzes podem ser reativadas.
- **Ajuste de temperatura:** O sistema de climatização pode ser ajustado consoante a localização dos residentes. Se a casa estiver vazia, o sistema pode reduzir o aquecimento ou o ar condicionado para poupar energia. Ao aproximarem-se de casa, a temperatura pode ser novamente ajustada para proporcionar conforto aquando da chegada.
- **Segurança e sistemas de alarme:** Quando todos os residentes saem do limite estabelecido, o sistema pode ativar automaticamente os dispositivos de segurança, como câmaras e alarmes. Ao regressar, o sistema desativa estas medidas de segurança.
- **Abertura e fecho de portas:** Fechaduras inteligentes podem ser configuradas para desbloquear automaticamente quando os residentes se aproximam da casa e bloquear quando se afastam, proporcionando um nível adicional de segurança.

Para esta automatização, aproveitou-se a geolocalização do dispositivo móvel para criar um cenário em que, quando o utilizador sair da casa com seu dispositivo, a *app* Lidl Home, utilizando o GPS do telemóvel, reconhece que o dispositivo se encontra fora da

localização pre-configurada para a *gateway* e, em seguida, a Silvercrest envia, através do protocolo ZigBee, o comando para desligar a tomada da IKEA.

Para configurar essa automação, começou-se por abrir o Lidl Home, carregou-se em “*Smart*” e em “*Crear escena*”. Posteriormente, selecionou-se a opção “*Si se modifica la ubicación*” e definiu-se que esta automatização seria despoletada quando o telemóvel com a *App* se afastasse mais de 100 metros da localização definida. Em seguida, selecionou-se o dispositivo responsável por executar a ação e o estado que assumiria após o acionamento da automação que, neste caso, seria “*OFF*”. Todos os passos da configuração desta automatização são apresentados de seguida, na Figura 43.

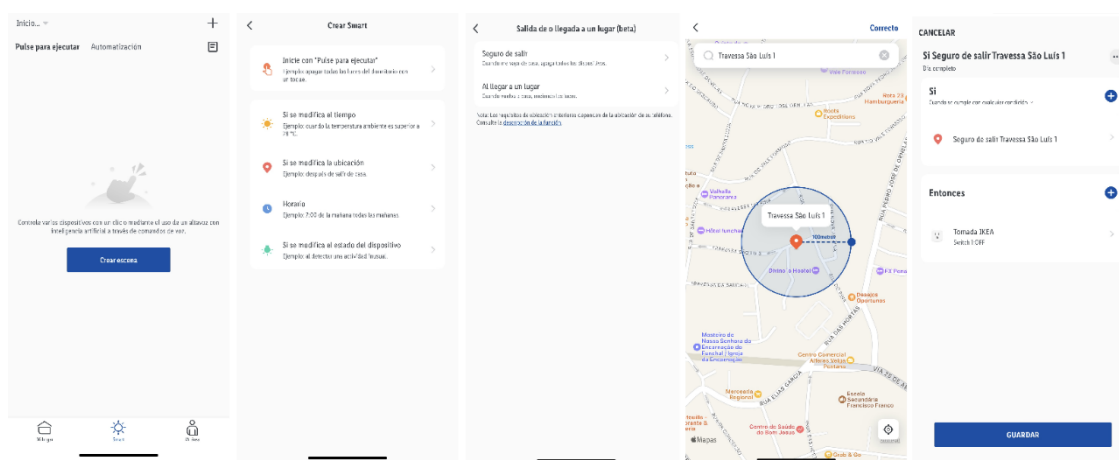


Figura 43 - Criação de uma automatização por geolocalização com a central Silvercrest e a app Lidl Home.

4.2.2.2. Automatização horária

Para demonstrar este cenário criou-se um interruptor inteligente para o qual o utilizador estabeleceu um horário específico para que o estado do interruptor fosse ajustado conforme desejado. Essa configuração é útil em situações em que o utilizador deseja ligar ou desligar um dispositivo num horário específico.

Para começar, tal como na automatização criada anteriormente, foi necessário ir a “*Smart*” e selecionar “*Crear escena*”, posteriormente, escolheu-se a opção “*Horario*” e estabeleceu-se a hora e os dias da semana nos quais esta automatização seria despoletada. Neste caso optou-se por todos os dias às 19h. Uma vez definido o *trigger*, a ação designada foi definir o estado da tomada da IKEA como “*ON*”. Na Figura 44 estão apresentados todos os passos para a configuração do cenário doméstico.

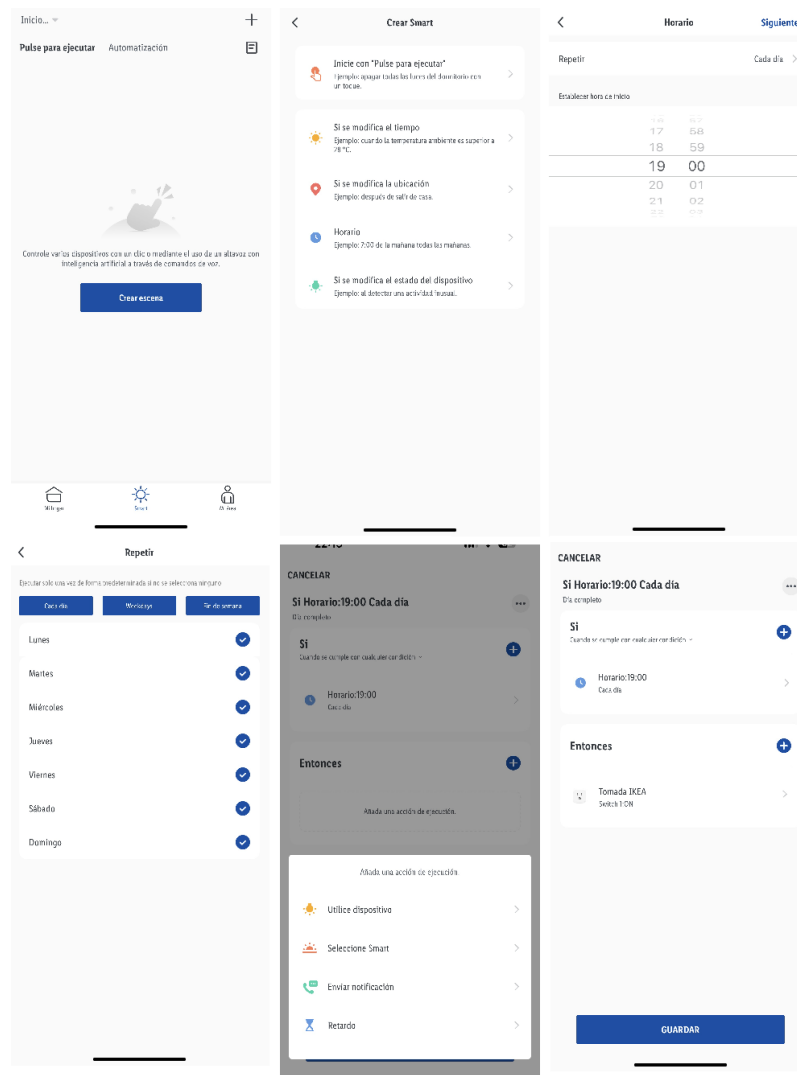


Figura 44 - Criação de uma automatização por dia e hora na Silvercrest e na app Lidl Home.

4.3. Home Assistant

A instalação do Home Assistant no Raspberry Pi é relativamente simples, sendo de seguida, indicados os passos a seguir para realizar de forma correta essa instalação [57]:

- Escrever a imagem no cartão de memória, para tal deverá:
 - Descarregar a imagem do site;
 - Gravar a imagem para o cartão de memória através da aplicação Balena Etcher;
 - Selecionar “Flash from file”;
 - Selecionar o cartão SD que será utilizado pelo Raspberry Pi;
 - Carregar em “Flash!”.

- Inicializar o Raspberry Pi, para tal deverá:
 - Inserir o cartão SD com a imagem descarregada previamente;
 - Conectar um cabo de rede Ethernet na porta correspondente do Raspberry Pi;
 - Caso seja necessário que o Raspberry Pi se ligue à rede local por WiFi, ou com um IP estático, é necessário:
 - Inserir uma PEN USB no Raspberry com o nome CONFIG;
 - Criar uma pasta dentro da PEN com o nome *network*;
 - No interior dessa pasta criar um ficheiro *my-network* sem qualquer extensão;
 - Seguir um dos exemplos apresentados em [58] para ligar a uma rede, ajustando o ficheiro consoante as configurações de rede e segurança.
 - Alimentar o Raspberry Pi com o respetivo cabo e esperar até à inicialização do sistema;
 - O servidor irá arrancar com o nome já definido, podendo ser acedido através de <http://homeassistant.local:8123> ou só <http://homeassistant:8123>;
 - O utilizador deverá definir o login (*username* e *password*) e a morada (de forma manual ou automática).

A página inicial do Home Assistant designa-se por Lovelace. Essa visualização permite utilizar vários tipos de “cartões” com diferentes visualizações, de modo a ser possível adequar as informações dos sensores ao que é pretendido. A Figura 45 apresenta a *dashboard* do Home Assistant sem qualquer tipo de cartões ou dispositivos integrados.

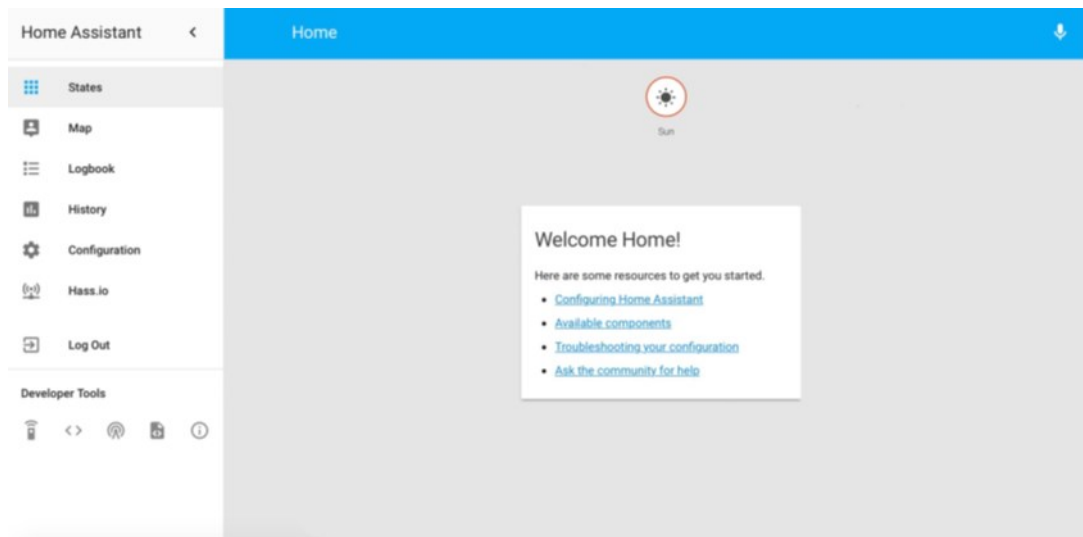


Figura 45 - Página inicial do Home Assistant.

4.3.1. Add-ons

A configuração propriamente dita do Home Assistant deverá agora ser iniciada através da instalação de módulos. Recomenda-se no entanto, começar por uma série de *plugins* que podem ser instalados e que facilitarão todo o processo de programação e configuração do sistema.

Os “*Add-ons*” são pacotes já pré-configurados e otimizados que podem ser instalados facilmente, com o objetivo de permitir ao utilizador estender as funcionalidades do Home Assistant. Para isso, basta recorrer ao menu inicial da interface e encontrar a opção “Supervisor”, de seguida, carregar na opção “Loja de *Add-ons*” e procurar o *Add-on* de interesse.

No caso deste trabalho, foram usados 4 *Add-ons*, apresentados na Figura 46. O primeiro a ser instalado foi o “*File Editor*” que é uma pequena aplicação que permite editar os ficheiros YAML utilizados pelo Home Assistant, através do *frontend*, facilitando assim a sua configuração.

Para a integração do protocolo MQTT foi instalado o *Add-on Mosquitto broker*, que é um *broker* de código aberto que implementa o protocolo MQTT. O *Mosquitto* é leve, o que permite que seja usado em qualquer dispositivo, desde computadores de placa única de baixa potência até grandes servidores. Com configurações simples de autenticação este *Add-on* fica pronto para utilização.

Outro dos *Add-on* que foram utilizados foi o ESPHome. Este *Add-on* permite omitir toda a complexidade no desenvolvimento do software necessário para gerir e programar microcontroladores baseados nos chips ESP8266 e ESP32. Tudo o que é necessário fazer é escrever ficheiros de configuração YAML, com instruções simples, sendo que todo o restante processo (conversão do código, validação, compilação, instalação, atualizações *over-the-air*, etc.) é feito pelo ESPHome.

Outra das integrações utilizadas foi o ZHA (*ZigBee Home Automation*) que permite ligar vários dispositivos ZigBee ao Home Assistant, utilizando um dos vários coordenadores ZigBee compatíveis que se encontram disponíveis. O ZHA usa uma biblioteca Python de código aberto, implementando uma pilha de hardware independente chamada zigpy, o que significa que pode suportar qualquer um dos módulos de rádio ZigBee disponíveis no mercado, sem importar o fabricante, pois desde que o módulo seja compatível com zigpy os dispositivos podem ser facilmente integrados.

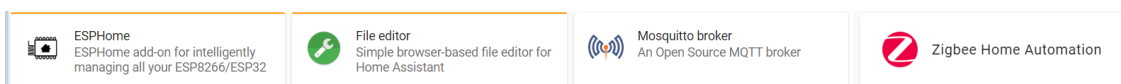


Figura 46 - Add-ons utilizados.

4.3.2. Sensores MQTT

Para realizar a demonstração do funcionamento do protocolo MQTT no Home Assistant optou-se por implementar um sensor de movimento e uma tomada inteligente. O sensor e o atuador foram ligados a uma placa ESP8266 e os métodos de instalação, configuração e programação serão descritos nas secções seguintes.

4.3.2.1. Sensor PIR

Para a elaboração deste trabalho, foi utilizado um sensor de movimento Grove - *Adjustable PIR Motion* Sensor apresentado anteriormente. Para isso, alimentou-se o sensor com uma tensão de 3 V, fornecida pela placa NodeMCU V3 ESP8266, ligou-se a referência do sensor ao *ground* da ESP e, posteriormente, a saída digital do sensor foi ligada na porta D1 da placa, tal como é apresentado na Figura 47.

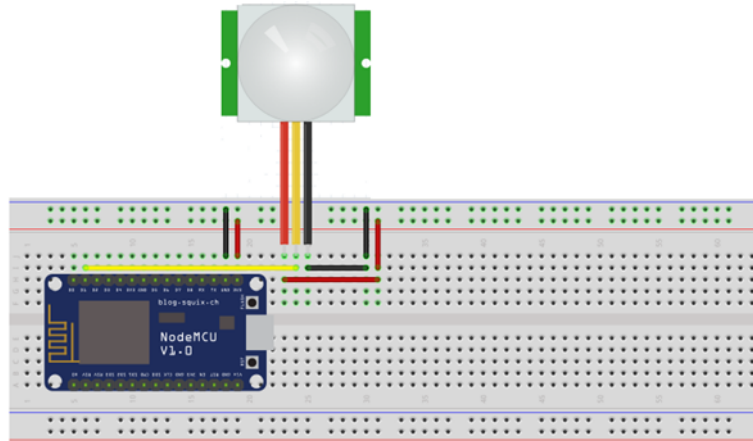


Figura 47 - Esquemático da montagem do circuito.

A placa ESP8266 foi configurada no Arduino IDE V1.8.13 para Windows. Os passos efetuados para a configuração e programação da placa foram:

- Instalação da aplicação Arduino IDE através do site [59] do Arduino;
- Instalação do gestor de placas esp8266 by ESP8266 *Community* versão 3.0.0 tal como é apresentado na Figura 48;

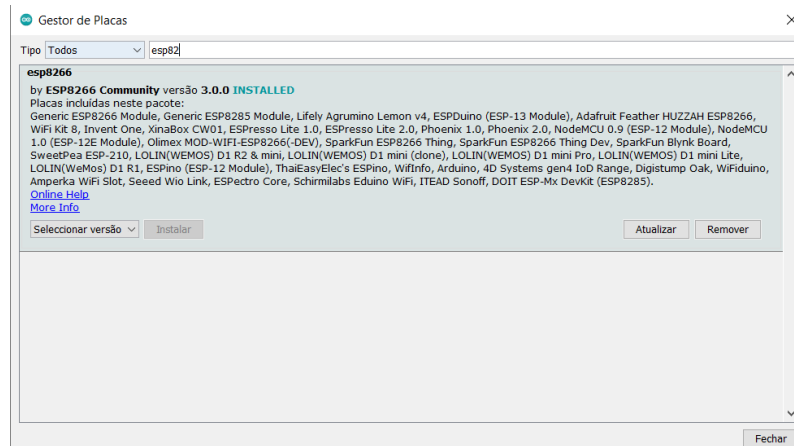


Figura 48 - Instalação do gestor de placas esp8266.

- Selecionar a placa NodeMCU 1.0 (ESP-12E Module) na lista de placas do ESP8266 *Boards* (3.0.0) e configurar a mesma com os parâmetros apresentados na Figura 49;

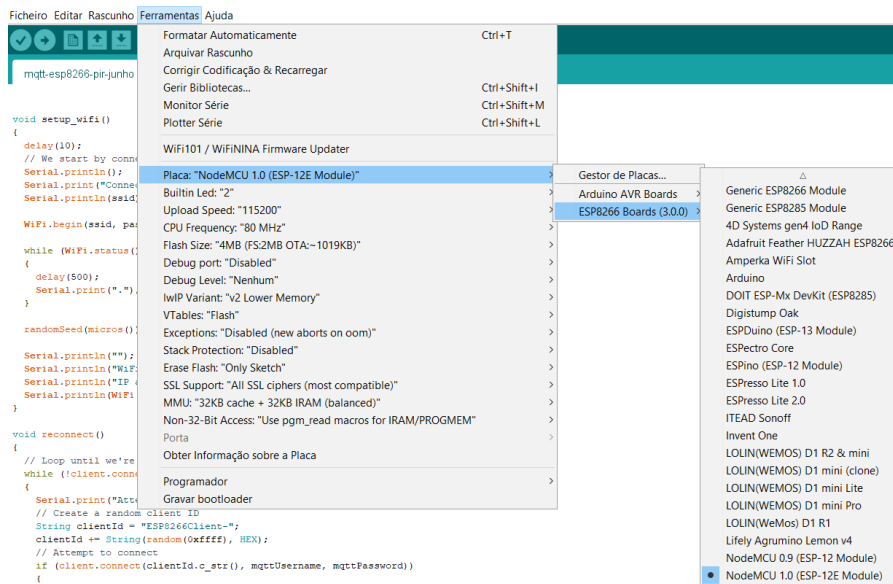


Figura 49 - Seleção da placa NodeMCU.

- Na opção Gestor de Bibliotecas foi necessário acrescentar as seguintes bibliotecas:
 - PubSubClient: Livraria para envio/recepção de mensagens MQTT;
 - Arduino Uno WiFi Dev Ed Library: Livraria que garante as ligações Wi-Fi e MQTT para as placas ESP32 e ESP8266, assim como exibido na Figura 50.

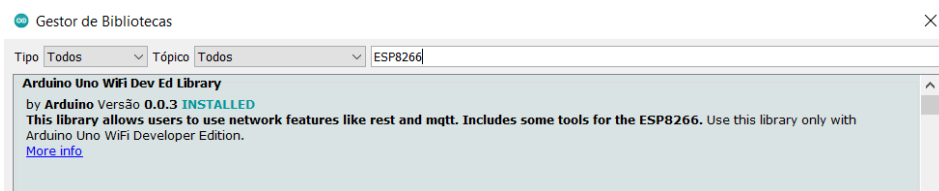


Figura 50 - Instalação da biblioteca necessária para utilizar Wi-Fi nas placas de desenvolvimento ESP32 e ESP8266.

- Definir a porta série de comunicação;
- Programar o código correspondente para a realização da tarefa pretendida;
- Verificar o código e, por fim, efetuar o *upload* para a placa.

Como este sensor possui uma saída digital, é possível diferenciar se o mesmo deteta movimento verificando os impulsos do sinal (0 ou 1) através da leitura do pino digital da placa NodeMCU V3 ESP8266. Essa leitura foi realizada a cada segundo, assegurando uma deteção confiável, uma vez que é improvável que pessoas ou animais

passem pelo sensor em intervalos inferiores a 1 segundo. Os dados deste sensor eram publicados por MQTT no tópic "sensor/pir" apenas quando ocorria uma alteração de estado. Essa abordagem visa preservar a bateria no dispositivo e evitar envios recorrentes que poderiam sobrecarregar o *broker* MQTT, resultando num aumento geral na eficiência do sistema. Na Figura 51 apresenta-se o fluxograma do código implementado, sendo que a instalação do circuito com o sensor está presente na Figura 52. O código completo de programação pode ser consultado no Anexo I.

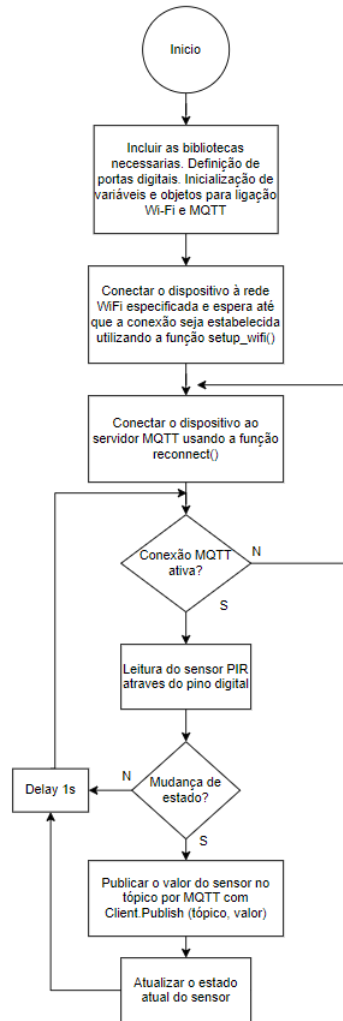


Figura 51 - Fluxograma do código para o sensor PIR com MQTT.

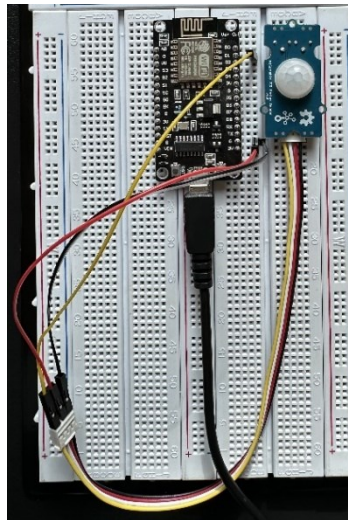


Figura 52- Circuito montado para o PIR.

4.3.2.2. Tomada inteligente

Para a construção de uma tomada inteligente, foi utilizado um relé uDFR0017 V3.1, anteriormente apresentado, e ligou-se a ESP8266 conforme indicado no esquema apresentado na Figura 53, elaborado com o auxílio do software *Fritzing*.

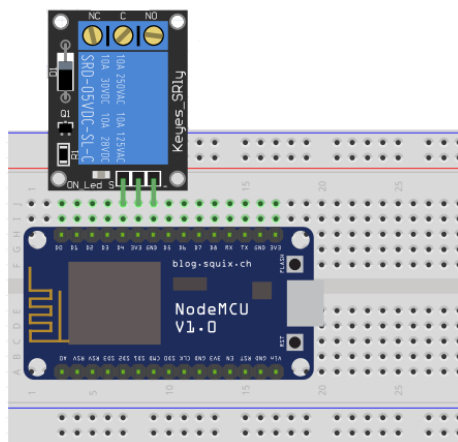


Figura 53 - Esquemático do circuito utilizado.

O relé era alimentado com 3 V provenientes do pino “3V” da ESP8266 e o *ground* “G” da placa foi ligado no terminal negativo do relé. Para a comutação do relé, era necessário o envio de impulsos, que eram fornecidos através de um dos pinos GPIO da ESP8266, tendo-se optado por configurar o pino D4 como pino de saída.

Os passos para a configuração deste relé foram os mesmos que os apresentados para o sensor de movimento, no entanto, o código programado foi naturalmente diferente.

O relé ficou subscrito ao mesmo tópico onde o sensor de movimento estava a publicar os dados e, caso existisse movimento (ou seja, se fosse publicado no tópico o valor esperado pelo relé), este seria ativado ou desativado consoante a situação. Na Figura 54 apresenta-se o fluxograma do código que foi implementado, na Figura 55 encontra-se o circuito montado na placa de desenvolvimento, o código de programação completo pode ser consultado no Anexo II.

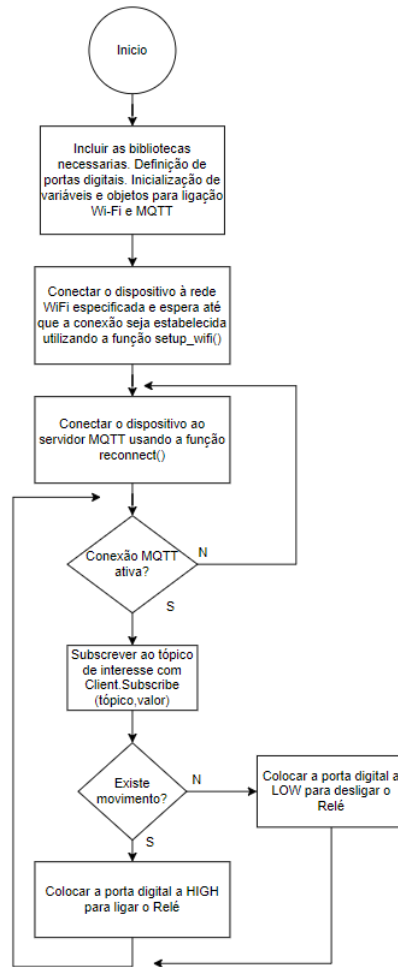


Figura 54 - Fluxograma do código para o relé com MQTT.

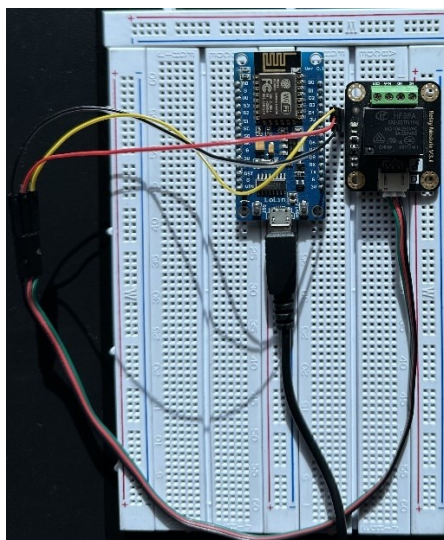


Figura 55 - Circuito montado para o Relé.

4.3.2.3. Integração dos sensores MQTT no Home Assistant

Como já foi referido anteriormente, a integração dos sensores com o protocolo MQTT foi efetuada inicialmente através da instalação e configuração do *Add-on Mosquitto Broker*.

Para tal, começou-se por configurar o *Add-on* indicando o IP do *Broker* que, neste caso, correspondia ao IP do Home Assistant (192.168.1.107), a porta (1883), o nome do utilizador e a palavra-passe, conforme apresentado na Figura 56. A partir desse momento, o Home Assistant ficou configurado como um cliente do *Broker*, mas ainda sem qualquer informação sobre os dados que podia receber pois não estava subscrito em nenhum tópico. Refira-se que são essas credencias que deverão de ser utilizadas na programação dos dispositivos, para se poder realizar a comunicação com o *Broker*, tal como apresentado no Anexo III.

Opções do broker ×

Por favor, insira os detalhes de ligação ao seu broker MQTT.

Broker*
core-mosquitto

Porta*
1883

Nome de Utilizador
homeassistant

Palavra-passe
cohb6ea3hain7shik8aeshai0thaMie1cieb7as1oKeigl 🔒

Figura 56 - Configuração do broker MQTT.

O passo seguinte foi a configuração manual do ficheiro `configuration.yaml`, com a informação dos nomes dos sensores e do tópico onde o sensor PIR estava a publicar os seus dados e onde o relé estava subscrito. A configuração efetuada pode ser consultada na Figura 57.

```
/config/configuration.yaml
1 |
2 | # Loads default set of integrations. Do not remove.
3 | default_config:
4 |
5 | # Text to speech
6 | tts:
7 |   - platform: google_translate
8 |
9 | automation: !include automations.yaml
10 | script: !include scripts.yaml
11 | scene: !include scenes.yaml
12 |
13 | mqtt:
14 |   sensor:
15 |     - name: "Motion Detetion Sensor"
16 |       state_topic: "sensor/pir"
17 |
18 |     - name: "Switch Relay"
19 |       state_topic: "sensor/pir"
```

Figura 57 - Configuração do ficheiro `.yaml` no file editor.

Concluída esta configuração, foi necessário adicionar os respetivos cartões ao painel principal do Home Assistant para assim ser possível visualizar os dados do sensor, e para posteriormente se poder interagir com os atuadores através do software. Para tal, selecionou-se no menu “Visão Geral” a opção “Configurar Painel” e, de seguida,

“Adicionar Cartão”. Posteriormente, escolheu-se como entidade o sensor pretendido e identificou-se o mesmo com um nome perante o utilizador na interface gráfica. A Figura 58 permite visualizar a realização dessa configuração.

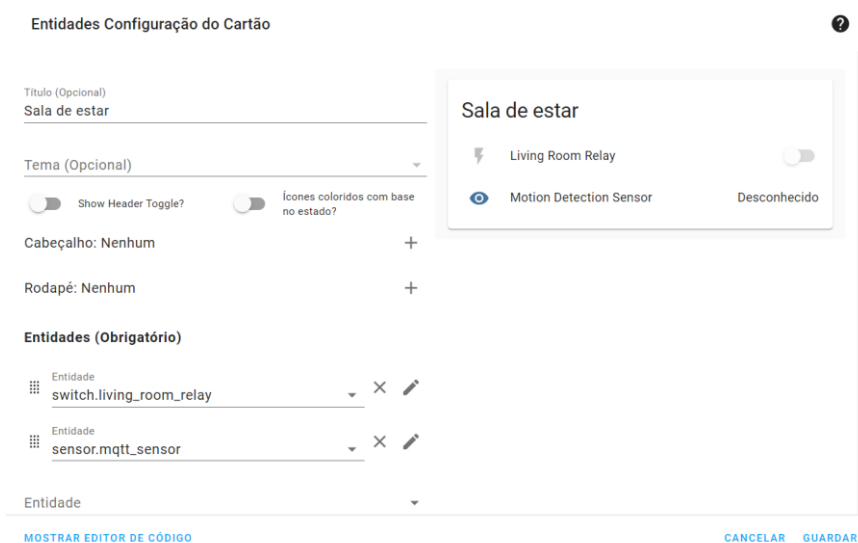


Figura 58 - Configuração dos cartões para a visualização das diferentes entidades na plataforma Home Assistant.

Como é possível observar no lado direito da Figura 58, existem duas entidades na Sala de estar, o *Living Room Relay* que corresponde à tomada inteligente e o *Motion Detection Sensor* que representa o sensor de movimento detetado pelo PIR. Uma vez concluídos estes passos de configuração, era possível visualizar o estado do sensor e interagir com o atuador através do Home Assistant.

4.3.3. Sensores/Atuadores ZigBee

Para se poderem utilizar dispositivos com o protocolo de comunicação ZigBee é necessário um coordenador, que tal como foi mencionado no capítulo 2.4.1.2, é responsável pela formação da rede, seleciona o canal, o PAN ID (*Personal Area Networks ID*), e define as políticas de segurança.

Nesta secção serão abordados todos os passos para a configuração de um coordenador ZigBee que, juntamente com o *Add-on Zigbee Home Automation (ZHA)*, facilitam a integração de dispositivos ZigBee, como lâmpadas, tomadas inteligentes, sensores de temperatura e humidade, sensores de movimento, câmaras de segurança, entre outros.

Como coordenador foi utilizado o *sniffer* CC2531 e como atuador ZigBee foi utilizada a tomada inteligente IKEA TRÅDFRI, apresentada no capítulo 3.10.

4.3.3.1. Coordenador ZigBee

Para programar o *sniffer* CC2531 como coordenador ZigBee, utilizando um computador com o sistema operativo Windows, é necessário seguir os seguintes passos.

1. Instalar o *SmartRF Flash programmer*, que é um software gratuito, mas requer a criação de uma conta no site [60] da Texas Instrumentes;
2. Instalar os drivers do *CC debugger* no computador e verificar que são reconhecidos com sucesso, caso contrário, será necessário realizar uma instalação manual.
3. Ligar o *CC debugger* à PCB verde e, de seguida, ligar esta PCB ao USB *sniffer* CC2531, tal como é apresentado na Figura 59.



Figura 59 - CC Debugger para programar o FW no dongle CC2531.

4. Ligar ambos os dispositivos (*CC debugger* e o CC2531) ao computador usando as respetivas portas USB.
5. Se o LED no *CC debugger* ficar vermelho é necessário pressionar o botão de *reset*, e o LED deverá agora ficar verde.
6. Descarregar o *firmware* CC2531_DEFAULT_20201127.zip

7. Inicializar o *SmartRF Flash Programmer* e utilizar os parâmetros apresentados na Figura 60. Selecionar o ficheiro .HEX e, para finalizar carregar na opção *Perform actions*.

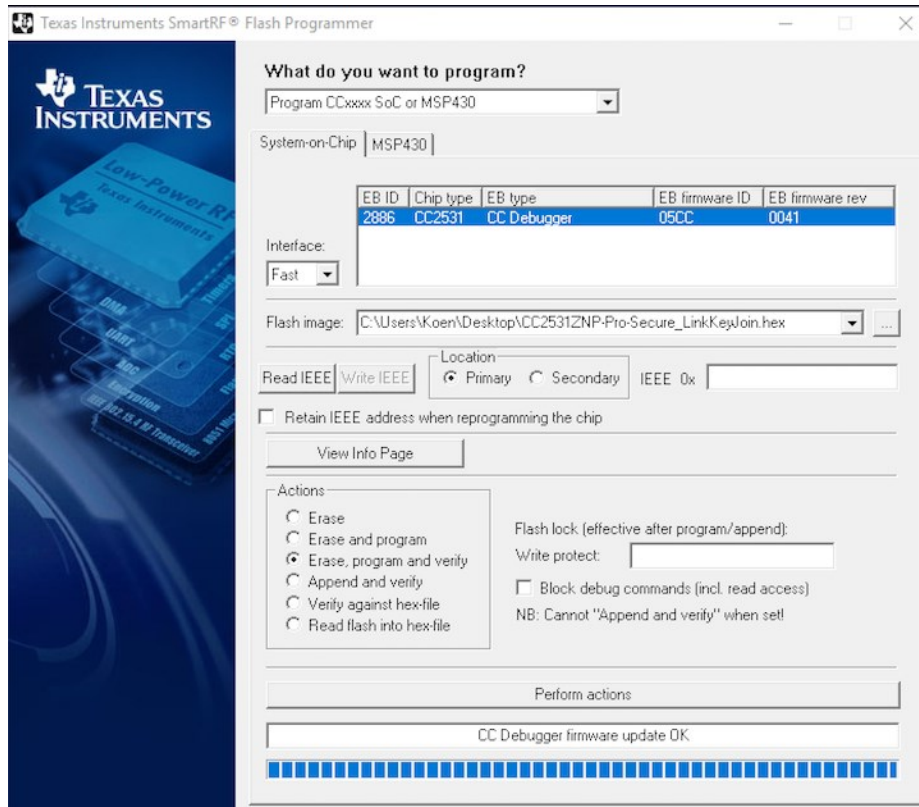


Figura 60 - Flash Programmer com os parâmetros necessários para programar o CC2531.

Para integrar o coordenador no *Add-on Zigbee Home Automation*, seleciona-se o caminho da porta série da Raspberry Pi onde o *sniffer* CC2531 está ligado, escolhe-se a área da casa onde será colocado esse dispositivo e, finalmente, carrega-se na opção de finalizar a configuração. O *Add-on* realizará toda a configuração necessária para integrar de forma correta o coordenador no Home Assistant. Na Figura 61 observa-se o endereço do CC2531.

```
Subsystem: usb
Device path: /dev/bus/usb/001/004
Attributes:
```

```
BUSNUM: '001'
DEVNAME: /dev/bus/usb/001/004
DEVNUM: '004'
DEVPATH: /devices/platform/soc/3f980000.usb/usb1/1-1/1-1.2
DEVTYPE: usb_device
DRIVER: usb
ID_BUS: usb
ID_MODEL: TI_CC2531_USB_CDC
ID_MODEL_ENC: TI\x20CC2531\x20USB\x20CDC
ID_MODEL_ID: 16a8
ID_PATH: platform-3f980000.usb-usb-0:1.2
ID_PATH_TAG: platform-3f980000_usb-usb-0_1_2
ID_REVISION: '0009'
ID_SERIAL: Texas_Instruments_TI_CC2531_USB_CDC___0X00124B001C
ID_SERIAL_SHORT: __0X00124B001CCE070F
ID_USB_INTERFACES: ':020201:0a0000:'
ID_VENDOR: Texas_Instruments
ID_VENDOR_ENC: Texas\x20Instruments
ID_VENDOR_ID: '0451'
MAJOR: '189'
MINOR: '3'
PRODUCT: 451/16a8/9
SUBSYSTEM: usb
TYPE: 2/0/0
USEC_INITIALIZED: '4506311'
```

Figura 61 - Endereço do dispositivo na Raspberry Pi.

4.3.3.2. Integração de sensores/atuadores ZigBee

Para demonstrar a integração de dispositivos ZigBee e verificar o seu funcionamento por meio deste protocolo dentro do ambiente do Home Assistant, utilizou-se a tomada inteligente IKEA TRÅDFRI.

Para começar, ligou-se o dispositivo à corrente elétrica da casa, carregou-se no botão de emparelhamento integrado na tomada durante 5 segundos e, posteriormente, no *Add-on* carregou-se na opção “Adicionar dispositivos” e esperou-se até que o dispositivo de interesse fosse encontrado. Para finalizar a integração, selecionou-se a área onde o dispositivo ficaria ligado.

De referir que esta tomada da IKEA inclui um botão que pode ser utilizado para ligar e desligar a tomada à distância, sem a necessidade de se utilizar a interface gráfica do Home Assistant ou a aplicação da IKEA. Este botão pode ser facilmente integrado seguindo os mesmos passos utilizados para a tomada.

Para interatuar com os dispositivos por meio do Home Assistant, foi necessário adicionar no painel *Lovelace* as respectivas entidades (sensores/atuadores), tal como foi

explicado com os sensores e atuadores MQTT. Na Figura 62 podem-se observar as entidades adicionadas no cartão.

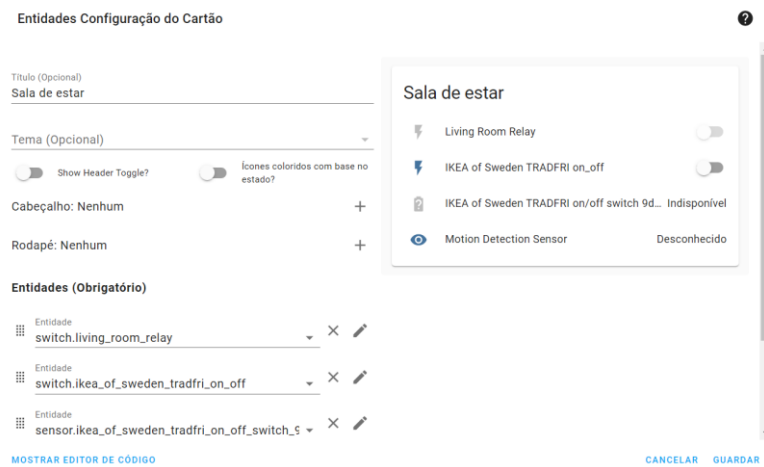


Figura 62 - Configuração das diferentes entidades no cartão do painel principal.

Assim, o atuador integrado ficou disponível para o utilizador na interface gráfica do Home Assistant, sendo possível a sua utilização à distância através da utilização do protocolo ZigBee.

4.3.4. Sensores Wi-Fi

Para demonstrar a integração de dispositivos DIY (*Do-It-Yourself*) via Wi-Fi utilizaram-se dois sensores ligados a duas ESP e ao *Add-on ESPHome*. Os sensores utilizados para este trabalho foram o sensor de temperatura e humidade AM2302 e o módulo magnético KY-021.

4.3.4.1. Sensor AM2302

Começou-se por alimentar o sensor AM2302 com 3 V provenientes da ESP8266, utilizando os pinos “3V” e “G”. Para realizar as leituras dos valores de temperatura e humidade, utilizou-se o pino digital D6, uma vez que o sensor fornece valores digitais na sua saída. Na Figura 63 encontra-se uma representação do circuito em questão e na Figura 64 pode ser visto o circuito com os componentes na *breadboard*.

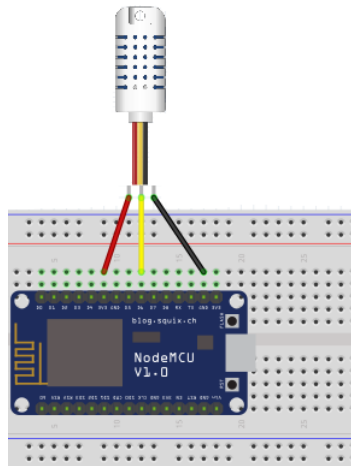


Figura 63 - Esquema do circuito.

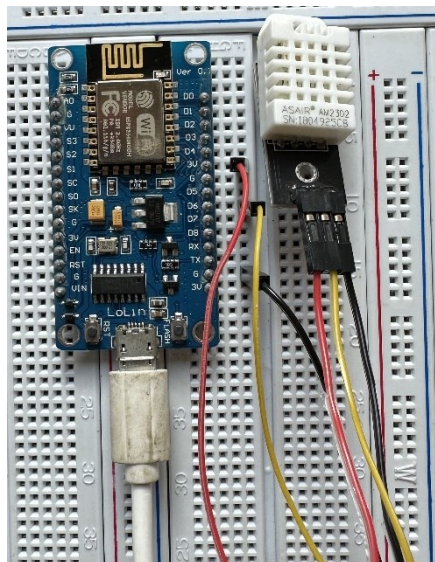


Figura 64 - Circuito montado para o sensor de temperatura e umidade.

Posteriormente, realizou-se a configuração do dispositivo no *Add-on ESPHome*. Para tal, atribuiu-se um nome ao dispositivo, selecionou-se a *board* disponível que, neste caso, foi uma Node MCU e, de seguida, foram inseridas as credenciais da rede Wi-Fi à qual a ESP8266 permaneceria ligada. É de salientar que a rede deveria de ser a mesma que a do Home Assistant. E, por último, introduziu-se uma palavra-passe para o serviço de *Over The Air (OTA) download*, funcionalidade que permite atualizar o *Firmware* (FW) do dispositivo por meio do Wi-Fi, fazendo com que seja muito mais fácil e prática a reprogramação da ESP para novas funcionalidades ou mudanças no comportamento do

microcontrolador. Este dispositivo, depois de ser integrado deverá aparecer no *Add-on* como é apresentado na Figura 65.

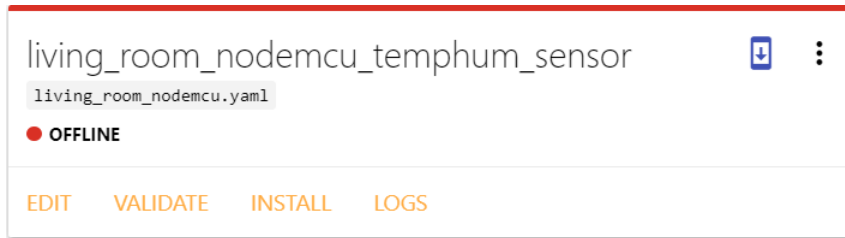


Figura 65 - Visualização do dispositivo no ESPHome.

Para efetuar a programação, carregou-se em “EDIT” e começou-se a programar o ficheiro em linguagem yaml. Inicialmente, verificaram-se os dados previamente inseridos, como o nome do dispositivo, plataforma, *board*, nome da rede wifi (SSID - *Service Set Identifier*) e palavra-passe e, logo depois, definiu-se o sensor. Escolheu-se como plataforma a “dht”, que permite obter os valores atuais de temperatura e de humidade de um dispositivo DHT11, DHT22 ou AM2302. Finalmente, definiu-se o pino digital da ESP encarregado de realizar a leituras dos valores, os nomes utilizados para diferenciar cada grandeza, a periodicidade da leitura e o modelo do sensor. Na Figura 66 observa-se o ficheiro editado.



Figura 66 - Código inserido no ESPHome para o sensor de temperatura e humidade.

Uma vez concluída a programação do ficheiro, guardou-se e compilou-se. No final, descarregou-se o ficheiro binário gerado pelo *Add-on ESPHome* e com a ESP8266 ligada ao PC mediante a porta série e o programa ESPHome Flasher, programou-se o dispositivo. Depois desta primeira programação, era possível atualizar o FW *Over The Air* sempre que fosse necessário.

Para anexar o dispositivo ao *dashboard* do Home Assistant foram repetidos os passos referidos nas secções anteriores, tal como é possível visualizar na Figura 67.

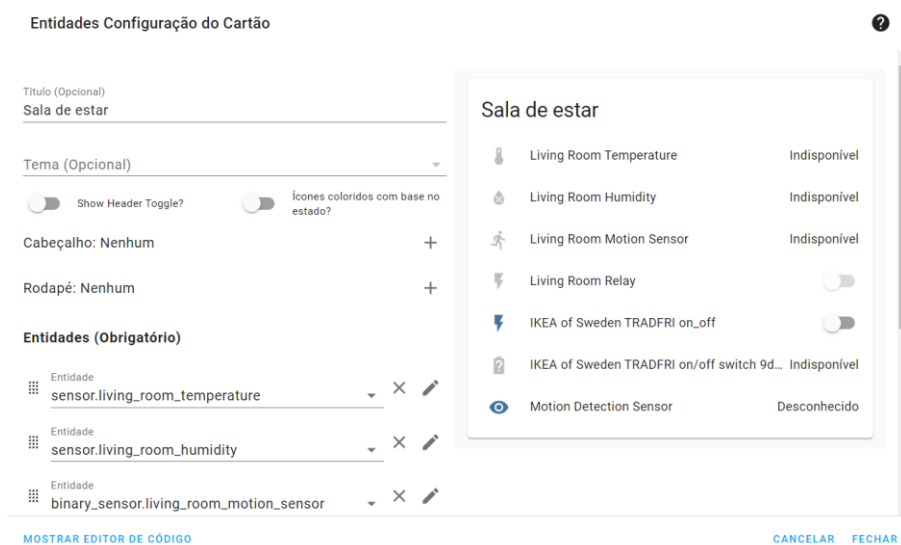


Figura 67 - Aditamento ao cartão do painel principal das novas entidades de interesse.

4.3.4.2. Sensor de portas e janelas

Através da utilização do sensor magnético KY-021, foi desenvolvido um dispositivo com a capacidade de detetar a abertura e fecho de portas e janelas. Para viabilizar a integração do sensor com o ESP8266 e o Home Assistant, optou-se pelo uso do *Add-on ESPHome* durante o processo de desenvolvimento.

Começou-se por alimentar o sensor KY-021 com uma tensão de 3 V, proveniente do pino "3V" da ESP8266, enquanto o pino "G" foi utilizado como referência de terra. Os dados de retorno do sensor foram recebidos através do pino digital D4 da ESP, porque como já é sabido pelas especificações do sensor KY-021, a sua saída é digital. Na Figura 68, está representado o circuito utilizado e na Figura 69 o circuito montado na placa de ensaio.

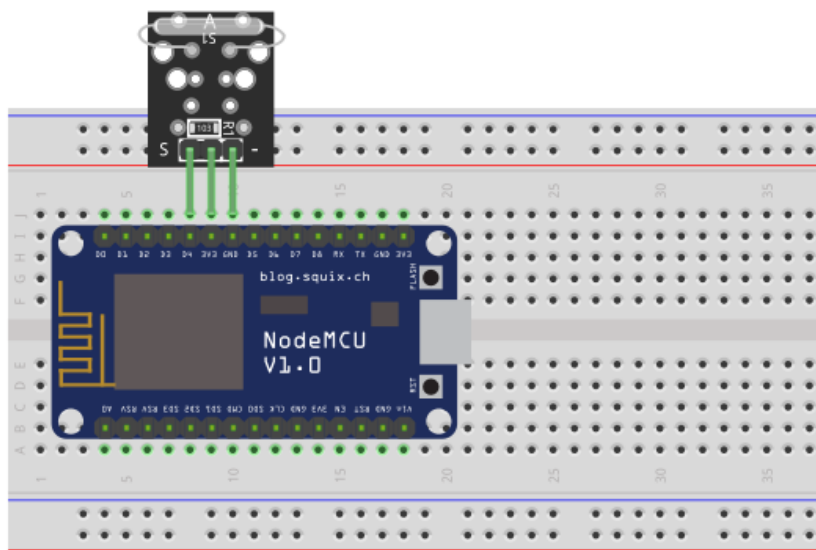


Figura 68 - Esquema de ligação do sensor KY-021 à ESP 8266.

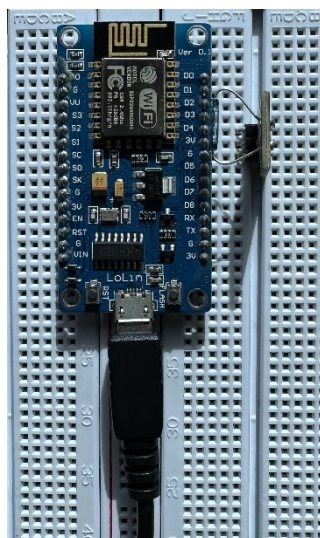


Figura 69 - Circuito montado na protoboard para o KY-021.

A fim de integrar o sensor KY-021 com o ESPHome, foram adotados procedimentos semelhantes aos utilizados anteriormente para a inclusão do sensor de temperatura e humidade. Após a criação do sensor no *Add-on*, deu-se início à programação do mesmo. Para isso, definiu-se o sensor como um sensor binário e foi especificada a plataforma na qual estaria conectado, neste caso, um GPIO da ESP.

Além disso, definiu-se o nome do sensor e o pino da ESP8266 no qual o sinal do sensor seria recebido. O modo do pino foi configurado como *pull-up*, a fim de assegurar que a leitura digital realizada estivesse dentro dos níveis lógicos esperados. Isso foi feito para evitar situações de flutuação, nas quais não seria possível determinar corretamente se o nível de tensão correspondia a um valor lógico 0 ou 1. Essa configuração também permitiu evitar a necessidade de acrescentar uma resistência ao circuito, pois foi utilizada a resistência de *pull-up* interna do pino.

Adicionalmente, foi definida a classe do dispositivo, com o intuito de exibir um ícone na interface gráfica. Neste caso, optou-se por colocar “*door*” de forma a simular uma porta de garagem. Além disso, foi inserido um atraso de 500 ms para evitar instabilidades na leitura. Na Figura 70 pode-se observar o código implementado.

```
garage-door.yaml
1 # Home:
2 name: nodemcu1
3 platform: ESP8266
4 board: nodemcu2
5
6 - wifi:
7   ssid: "ME0-738073"
8   password: "60C480EBC1"
9
10 # Enable logging
11 logger:
12
13 # Example configuration entry
14 - sensor:
15   - platform: wifi_signal
16     name: "WiFi Signal Sensor"
17     update_interval: 60s
18
19 # Enable Home Assistant API
20 api:
21   password: "xxx"
22
23 ota:
24   password: "xxx"
25
26 - binary_sensor:
27   - platform: gpio
28     name: "Garage Door"
29     pin:
30       number: D4
31       mode: INPUT_PULLUP
32     device_class: door
33     filters:
34       - delayed_on_off: 500ms
35
```

Figura 70 - Código utilizado para programar a ESP8266 para usar o sensor KY-021 como sensor de uma porta de garagem.

Após se concluir a programação, o arquivo foi salvo e compilado. Ao finalizar esse processo, descarregou-se o arquivo binário gerado pelo *Add-on ESPHome*. Tal como foi feito com o sensor de temperatura e humidade, o dispositivo foi programado utilizando o *ESPHome Flasher*.

Para finalizar a integração, o dispositivo foi adicionado ao *dashboard* do *Home Assistant*, seguindo os mesmos passos referido em secções anteriores.

4.3.5. Automatizações

Nesta secção foram explorados os sensores previamente desenvolvidos a fim de se estabelecerem processos de automação. Esses processos são fundamentados em regras e eventos definidos pelo utilizador no Home Assistant. Com o auxílio da própria plataforma, foi possível despoletar ações específicas em resposta a esses eventos ou estados do sistema, permitindo a criação de cenários complexos e personalizados para atender às necessidades do utilizador.

Existem duas formas de criar automações: utilizando o editor de automações do Home Assistant ou editando diretamente o arquivo `.yaml`.

As automações desenvolvidas nesta secção do trabalho visavam integrar sensores que utilizam diferentes protocolos de comunicação e funcionalidades integradas no próprio Home Assistant. Pretendia-se, desta forma, demonstrar que o utilizador não precisa de se preocupar com a compatibilidade entre os sensores ou dispositivos, pois o software fornece uma camada de abstração que permite controlar e integrar dispositivos de diferentes protocolos numa única interface, testando desta forma uma das principais vantagens do Home Assistant: a sua flexibilidade.

4.3.5.1. Desenvolvimento de um sensor de garagem com notificação de porta aberta através de dispositivo móvel

Neste ponto optou-se por desenvolver um sensor de garagem com notificação de porta aberta, permitindo que o utilizador fosse informado quando a porta da garagem não estivesse corretamente fechada ou estivesse aberta, sendo enviada uma notificação através do Home Assistant para o dispositivo móvel do utilizador, garantindo uma maior segurança.

Para a criação dessa automação, carregou-se na opção “Automatizações & Cenários” no painel de configurações do Home Assistant e, em seguida, selecionou-se “Criar Automatização”. Posteriormente, indicaram-se as condições que desencadeariam as ações, no caso em questão, se o estado do sensor (ou entidade) “*Garage Door*” que representa o estado do sensor construído com o sensor KY-021 (descrito no ponto 4.3.4.2) permanecesse como “Aberto” por um período de 3 minutos, uma mensagem seria enviada para o dispositivo móvel do utilizador por meio do serviço *notify* que é uma funcionalidade integrada no próprio Home Assistant. Referira-se que esse dispositivo

móvel também deve ser registado previamente no Home Assistant. Nas Figura 71 e Figura 72, encontram-se apresentadas as configurações para esta automação.

Garage Door

Use as automatizações para dar vida à sua casa.

Nome
Garage Door

[Adicionar descrição](#)

O modo controla o que acontece quando a automatização é despoletada enquanto as acções ainda estão em execução devido a uma chamada anterior. Consulte [documentação sobre automatizações](#) para mais informações.

Modo
Único (pré-definido)

Ativar/desativar automação [MOSTRAR REGISTO](#) [DESPOLETAR](#)

Gatilhos

Os gatilhos são o que inicia o processamento de uma regra de automatização. É possível especificar vários gatilhos para a mesma regra. Uma vez iniciado um gatilho, o Home Assistant validará as condições, se existirem, e chamará a acção.

[Saiba mais sobre gatilhos](#)

Tipo de gatilho
Dispositivo

Dispositivo
nodemcu1

Gatilho
Garage Door aberto

Duração (opcional)

hh	mm	ss
0	: 03	: 00

Figura 71 - Configuração dos gatilhos da automação.

Ações

As acções são o que o Home Assistant executará quando a automatização for despoletada.

[Saiba mais sobre acções](#)

Tipo de acção
Chamar serviço

Serviço
Notificar: Send a notification via mobile_app_iphone_de_johevan

Sends a notification message using the mobile_app_iphone_de_johevan integration.

Message
Message body of the notification.

Title
Title for your notification.

Target
An array of targets to send the notification to. Optional depending on the platform.

Data
Extended information for notification. Optional depending on the platform.

[ADICIONAR AÇÃO](#)

Figura 72 - Ações despoletadas na automação pelos gatilhos.

Com esta abordagem, o utilizador não precisa de se preocupar com a forma como o Home Assistant processa e recebe os dados do sensor, nem com os detalhes sobre como enviar a notificação. Tudo isso é gerido pela plataforma, simplificando o processo de integração entre diferentes sensores.

Na Figura 73 pode ser observado o esquema utilizado no desenvolvimento do sensor da porta da garagem.

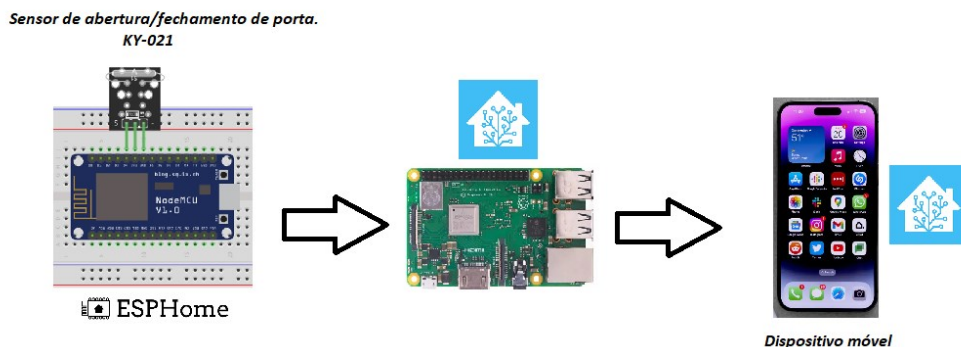


Figura 73 - Esquema do sensor da porta da garagem.

4.3.5.2. Desenvolvimento de um sistema de alarme inteligente

Aproveitaram-se os sensores previamente configurados para implementar um sistema de alarme automatizado, capaz de detetar intrusos na garagem e alertar o utilizador.

Como parte essencial do sistema de alarme, foi necessária a criação de um botão ou interruptor para armar e desarmar o alarme. Esse interruptor serviu como uma interface de interação com o sistema de segurança, oferecendo ao utilizador a capacidade de controlar o funcionamento do alarme, de acordo com as suas necessidades, através do *dashboard* do Home Assistant

Para esta implementação, começou-se por configurar um interruptor na interface do utilizador do Home Assistant, sendo que para tal carregou-se na opção “Criar Auxiliar” na janela “Auxiliares”, localizada dentro do menu “Dispositivos & Serviços” no painel de configurações. Em seguida, adicionou-se o interruptor no *dashboard* do Home Assistant.

Pretendia-se que, caso o alarme estivesse armado, se o sensor da porta da garagem detetasse que a porta foi aberta e o sensor de movimento detetasse um intruso, fosse dado

o *trigger* para ativar o alarme. O Home Assistant era responsável por monitorar estas condições e, quando elas ocorriam, enviava uma notificação para o dispositivo móvel associado à conta do Home Assistant, através do serviço de envio de notificações *notify*. Além disso, uma mensagem era publicada, por MQTT, no tópico “Alarme” com o *payload* “Detecção de Intruso!!”. Esta mensagem que era publicada neste tópico, pode ser utilizada para fazer atuar um *buzzer*, proporcionando uma simulação mais realista de um alarme. Na Figura 74 podem ser visualizadas as condições e o *trigger* utilizado, sendo que, na Figura 75, estão presentes as ações que eram realizadas. Naturalmente, todas estas configurações foram utilizadas para a criação da automação.

Alarme

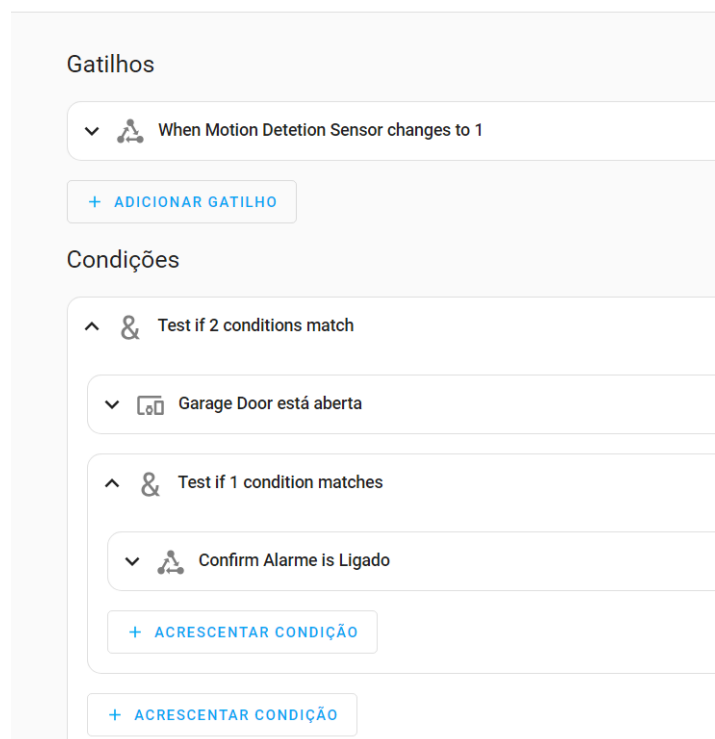


Figura 74 - Trigger e condições utilizadas para despoletar as ações dentro da automação.

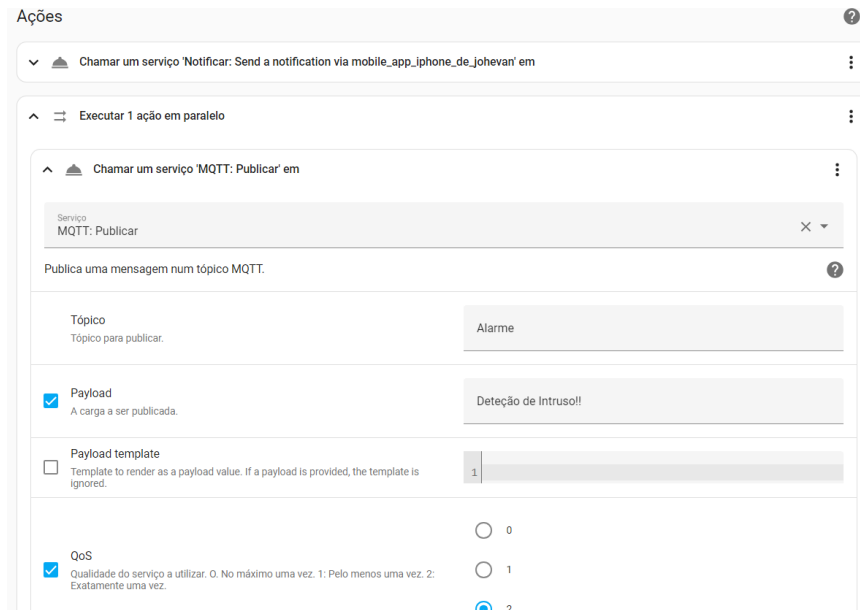


Figura 75 - Ações desencadeadas.

O esquema da automação é apresentado na Figura 76. Neste esquema, é possível observar os diferentes dispositivos utilizados para este cenário doméstico, como o sensor de deteção de campo magnético que funcionará para detetar abertura/fecho de uma porta ou janela, o PIR para a deteção de presença e um *smartphone* com Home Assistant para receber a mensagem de alerta de intrusão. Para o sensor magnético a tecnologia utilizada para o envio da informação ao Home Assistant foi o Wi-Fi através do ESPHome e para o PIR o MQTT. Nesta figura, foi apresentado um *buzzer*, com o intuito de criar um cenário mais real, mas este atuador na prática não foi utilizado pois era necessário adquirir, no entanto, foi publicada uma mensagem por MQTT no tópico de interesse de forma que qualquer dispositivo subscrito a esse tópico pudesse atuar.

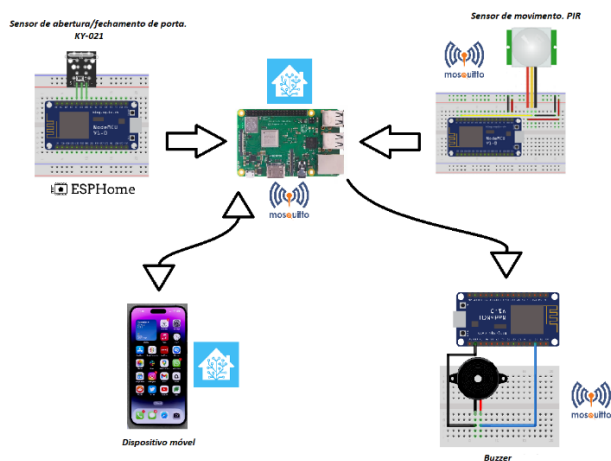


Figura 76 - Esquema do sistema de alarme.

4.3.5.3. Desenvolvimento de um termostato inteligente

Em ambientes domésticos, o dispositivo que regula a temperatura e aciona os sistemas de aquecimento ou refrigeração é chamado de “termostato”. Esses termostatos domésticos são dispositivos comuns em casas e apartamentos para controlar o sistema de aquecimento, ventilação e ar condicionado (HVAC - *Heating, Ventilation, and Air Conditioning*). A sua função principal é manter a temperatura interna da casa dentro de um intervalo desejado, o que permite manter um ambiente interno confortável e economizar energia, uma vez que o sistema não precisa de funcionar continuamente ou ser desligado manualmente.

Para o desenvolvimento desta automação, utilizou-se o sensor de temperatura e humidade previamente apresentado, e criou-se uma automação através do Home Assistant. Esta automação foi projetada para entrar em ação sempre que o valor da temperatura atingisse ou excedesse um limiar específico. Nesse cenário, o atuador inteligente IKEA TRÅDFRI seria acionado por meio da tecnologia ZigBee. Da mesma forma, quando o valor da temperatura atingisse outro limite predefinido, a tomada seria desligada.

O processo de criação dessa automação respeitou os mesmos procedimentos delineados para as automações anteriores, sendo que foi necessário definir os parâmetros para o gatilho e as ações a serem despoletadas. Na Figura 77 visualizam-se as condições e entidades envolvidas na construção desta automação, nomeadamente para a situação em que a temperatura ultrapassava os 29°C e a tomada era ligada.

Gatilhos

Temperatura de Living Room Temperature

Dispositivo: Living-Room-Temphum-Sensor

Gatilho: temperatura de Living Room Temperature

Acima de: 29 °C

Abaixo de:

Duração (opcional): 0 : 00 : 00

Condições

+ ACRESCENTAR CONDIÇÃO

Ações

Ligar IKEA of Sweden TRADFRI ON_OFF

Dispositivo: IKEA of Sweden TRADFRI control outlet

Ação: Ligar IKEA of Sweden TRADFRI ON_OFF

+ ADICIONAR AÇÃO

Figura 77 - Parâmetros utilizados para a constituição da automação.

Realizou-se o mesmo para o caso em que o valor da temperatura atingia outro limite predefinido e a tomada era desligada, tal como pode ser visualizado na Figura 78.

← Controle de Temperatura 2

RASTREIOS DE EXECUÇÃO

Gatilhos

Temperatura de Living Room Temperature

Dispositivo: Living-Room-Temphum-Sensor

Gatilho: temperatura de Living Room Temperature

Acima de:

Abaixo de: 28 °C

Duração (opcional): 0 : 00 : 00

+ ADICIONAR GATILHO

Condições

+ ACRESCENTAR CONDIÇÃO

Ações

Desligar IKEA of Sweden TRADFRI ON_OFF

Dispositivo: IKEA of Sweden TRADFRI control outlet

Ação: Desligar IKEA of Sweden TRADFRI ON_OFF

+ ADICIONAR AÇÃO

Figura 78 - Parâmetros utilizados para a constituição da automação 2.

Os valores definidos para estas duas automações foram selecionados com base na temperatura ambiente e com a intenção de facilitar a realização dos testes para a validação da automação, permitindo verificar de forma mais fácil e conveniente os valores registados pelo sensor durante o teste, para que pudessem ser despoletadas as ações programadas na automação.

O esquema de ligação é apresentado na Figura 79.

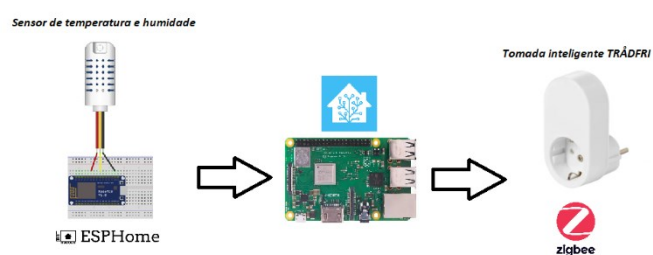


Figura 79 - Esquema do termostato inteligente.

5. Resultados

Neste capítulo apresentam-se os resultados obtidos para cada uma das *gateways* testadas e para cada um dos sensores construídos, de forma a verificar o seu funcionamento de forma individual, assim como também testar as automações que foram propostas e detalhadas em secções anteriores.

5.1. Hubitat Elevation

Nesta secção serão apresentados os resultados conseguidos uma vez integrado o atuador ZigBee da IKEA na plataforma Hubitat. Os resultados serão divididos em duas subsecções, uma primeira para avaliar o controlo do dispositivo por meio do *dashboard*, sem nenhuma automação relacionada, e uma segunda onde se testou se a automação era desencadeada e se tinha o comportamento esperado.

5.1.1. Controlo do atuador através do *dashboard*

Tendo o atuador sido integrado com o Hubitat e sendo apresentado no *dashboard*, era possível interagir com a tomada e ativá-la/desativá-la conforme desejado pressionando o ícone correspondente ao dispositivo. Dessa forma, foram realizados os testes para verificar a resposta do dispositivo, pressionando varias vezes o ícone e observando se o estado da tomada acompanhava o que era apresentado no *dashboard* do Hubitat. Na Figura 80 observa-se a situação em que a tomada estava no estado desligada e foi pressionado o ícone do dispositivo para mudar o seu estado para “ON”.

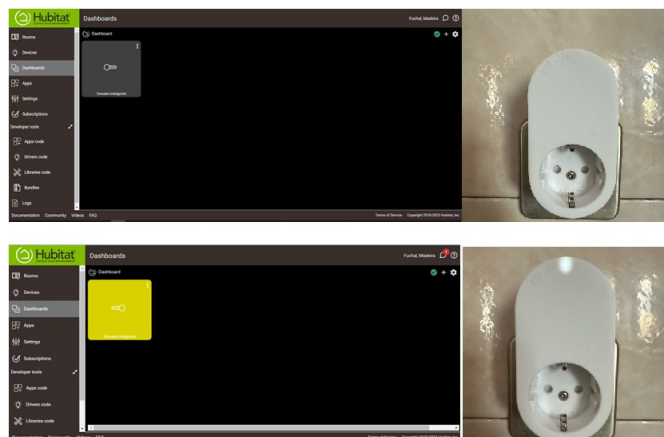


Figura 80 - Controlo do atuador TRÅDFRI IKEA através do Hubitat.

Confirmou-se que, de facto, a tomada respondia de forma rápida e de acordo com os estados apresentados no *dashboard*. Assim, pode-se considerar a integração do dispositivo com a plataforma como validada.

5.1.2. Automação horária

Para validar essa automação, inicialmente, colocou-se o estado da tomada em “*OFF*”, visto que a primeira ação da automação acionaria a mudança para “*ON*” e a observação da transição de estados permitia validar se a primeira ação do cenário doméstico era bem sucedida. Considerando que este cenário estava associado ao horário do pôr do sol, foi preciso aguardar até 10 minutos antes desse momento, tal como programado, para observar a correta execução da primeira fase.

Uma vez atingido o momento aguardado, verificou-se que o gatilho para a primeira ação foi acionado conforme previsto, e o Hubitat, por meio da comunicação ZigBee, ativou a tomada da IKEA de acordo com o estabelecido na automação.

Para uma avaliação completa da implementação, foi necessário aguardar até o dia seguinte, 10 minutos depois do nascer do sol. Nesse instante, a segunda ação deveria ser desencadeada, desligando a tomada. A confirmação de que essa ação foi realizada com êxito validou o cenário doméstico e reforçou a eficácia do uso de regras e atuadores no Hubitat para automações.

5.2. Silvercrest e Lidl Home

Esta parte do trabalho exhibe os resultados obtidos para os diferentes cenários de automação expostos no capítulo anterior com a *gateway* Silvercrest e a aplicação Lidl Home, assim como os resultados do teste do controle de forma manual do próprio atuador.

5.2.1. Controle manual da tomada

Como seria de esperar, após integrar um atuador numa central doméstica, o objetivo é poder controlá-lo remotamente. Para isso, utilizou-se a aplicação Lidl Home e na secção “*Mi Hogar*”, onde todos os dispositivos estão listados, selecionou-se a tomada da IKEA para interagir com ela.

Após a seleção da tomada a aplicação apresenta uma interface de utilizador que simula um interruptor, exibindo o estado atual da tomada. Nessa interface, é possível interagir com o dispositivo para alterar seu estado, seja clicando diretamente na tomada ou na opção “Power”. A sequência de passos descrita anteriormente é ilustrada na Figura 81, onde inicialmente a tomada estava no estado “OFF” e, após o clique, passou para “ON”.

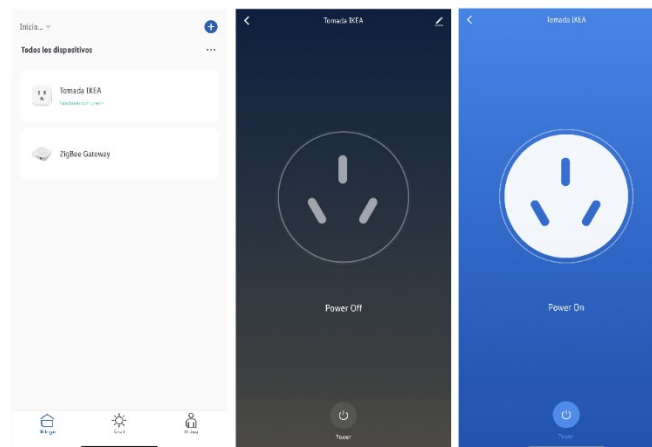


Figura 81 - Controle manual da tomada IKEA através do Silvercrest e da app Lidl Home.

Foi constatado que a tomada respondia conforme o esperado e refletia os estados apresentados pela *app*. Dessa forma, foi possível controlar remotamente esse dispositivo tanto dentro da mesma rede, quanto fora de casa, conforme desejado.

5.2.2. Automatização por geolocalização

Para realizar o teste deste cenário domótico e verificar se de facto a tomada era desligada, inicialmente foi necessário colocar o estado da tomada em “ON”, de forma a poder visualizar se, uma vez despoletada a automação, o estado do atuador em questão mudava conforme o desejado (para “OFF”). Posteriormente, afastou-se o telemóvel associado ao aplicação do raio de distância estabelecido na criação da automação de forma a acionar a automação, que neste caso foi de 100 metros.

Confirmou-se que a automação foi despoletada e que a tomada passou a estar desligada, pelo que se pôde considerar como válido o cenário domótico proposto.

Uma vez confirmada que esta central e a aplicação possibilitam a criação de cenários de automação com base na localização do utilizador, fica claro que é possível

gerar automações utilizando os exemplos descritos no capítulo 4.2.2.1, integrando outros sensores e atuadores conforme necessário. Este facto abre oportunidades para desenvolver automações mais avançadas e personalizadas, adaptadas às necessidades específicas do ambiente doméstico e às preferências do utilizador.

5.2.3. Automatização horária

Para testar esta automação, semelhante à anterior, começou-se por colocar a tomada no estado oposto ao qual ela passaria a estar após a automação ser acionada, a fim de validar o correto funcionamento do sistema.

De acordo com o cenário domótico definido, que consistia em ligar o atuador da IKEA todos os dias às 19h, a tomada foi configurada inicialmente no estado desligado e aguardou-se até chegar a essa hora para confirmar se o atuador se comportava como era esperado, e mudava para o estado “ON”.

Confirmou-se que a automação funcionou corretamente, demonstrando que é possível utilizar este tipo de automatização juntamente com outros sensores e atuadores para criar novos cenários domóticos.

5.3. Home Assistant

Nesta secção serão revelados todos os resultados referentes aos sensores e atuadores integrados para funcionar com o Home Assistant, com as diferentes tecnologias e *Add-ons*, juntamente com as várias automatizações configuradas, as quais foram apresentadas no capítulo anterior.

5.3.1. Sensor e atuador MQTT

Entrando no tópico dos resultados, esta secção analisará os dados do sensor e atuador transmitidos através de MQTT para determinar se estavam a funcionar como era suposto.

5.3.1.1. Sensor PIR

Este sensor, tal como foi explicado na secção 4.3.2.1, foi desenvolvido para enviar por MQTT os dados da variação térmica, que eram interpretados como deteção de

movimento. Se o sensor não detetar nenhuma variação no ambiente (movimento) envia um “0”, caso contrário envia um “1”.

Para realizar o teste colocou-se o dispositivo virado para uma área onde não estivesse ninguém a passar e, posteriormente, passou-se a mão em frente do sensor e visualizou-se em tempo real o envio dos dados para o Home Assistant através do *dashboard*, configurado previamente para a entidade “*Motion Detetion Sensor*”. Na Figura 82(a) é possível observar a transmissão do estado do sensor quando ele não detetava qualquer movimento, situação indicada como “0”. Por outro lado, na Figura 82(b), encontra-se a situação oposta, em que o sensor detetou movimento.

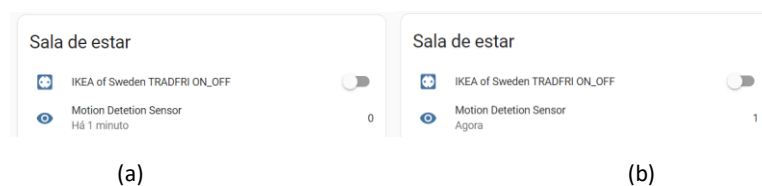


Figura 82 - Dados enviados pelo dispositivo construído com o sensor PIR: (a) sem movimento; (b) com movimento.

O envio dos dados também pôde ser consultado através do software MQTT Explorer [61] colocando o IP do *Broker* utilizado no Home Assistant com as credenciais e subscrevendo o tópico utilizado pelo sensor para publicar os dados. Na Figura 83 é possível visualizar o envio dos dados para as duas situações, com deteção de movimento e sem deteção de movimento.

```
Mensagem 4 recebida em sensor/pir às 16:53 :  
0  
QoS: 0 - Retain: false  
-----  
Mensagem 3 recebida em sensor/pir às 16:53 :  
1  
QoS: 0 - Retain: false  
-----  
Mensagem 2 recebida em sensor/pir às 16:51 :  
0  
QoS: 0 - Retain: false  
-----  
Mensagem 1 recebida em sensor/pir às 16:51 :  
1  
QoS: 0 - Retain: false
```

Figura 83 - Visualização dos dados enviados pelo sensor de movimento PIR, através do software MQTT Explorer.

Confirmou-se, desta forma, que o dispositivo para a deteção de movimento funcionava tal como estava previsto.

5.3.1.2. Relé

Para testar o funcionamento desse dispositivo atuador, era necessário que o sensor PIR estivesse em pleno funcionamento. Caso contrário, era preciso publicar manualmente uma mensagem no tópico ao qual o relé estava subscrito, para que ele pudesse ser acionado.

Como já se tinha verificado o funcionamento do sensor PIR, optou-se por fazer com que este fosse o responsável por detetar movimento e publicar automaticamente os dados no tópico apropriado. Esperava-se assim que o relé reagisse à mensagem recebida, realizando a ação pretendida conforme a configuração previamente definida. Na Figura 84(a) e Figura 84(b) são expostas as duas situações, quando o PIR não detetou movimento, e o relé permaneceu desativado, ou seja, no estado normalmente aberto, e quando foi detetado movimento e o relé foi ativado.

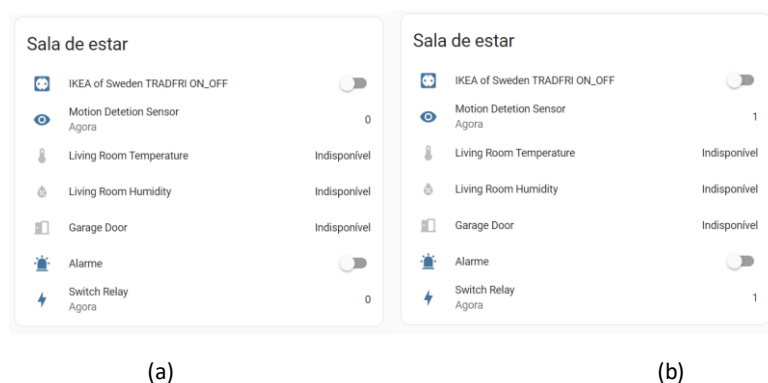


Figura 84 – Comportamento do relé: (a) sem movimento; (b) com movimento.

Confirmou-se, assim, que o sistema funcionava corretamente.

Este tipo de comportamento pode ser utilizado para a criação de sistemas de iluminação automatizada em casas de banho, corredores, garagens, etc. Os sistemas de iluminação automatizada permitem uma economia no consumo, porque a iluminação é ativada apenas quando há presença de pessoas no ambiente, evitando assim o desperdício de energia. Outra das vantagens é o conforto oferecido pela automação pois dispensa a intervenção manual para acionar interruptores. Outro ponto positivo é que com esta gestão inteligente da iluminação é possível prolongar a vida útil das lâmpadas, o que contribui para a sustentabilidade e redução dos custos operacionais.

5.3.2. ZigBee

Nesta subsecção serão apresentados os resultados obtidos com os dispositivos ZigBee integrados no Home Assistant.

5.3.2.1. Tomada ZigBee

Para testar este atuador, foi necessário simplesmente ligar e desligar o interruptor na *user interface* do Home Assistant e confirmar que o dispositivo respondia ao pedido. Na Figura 85(a) e Figura 85(b) são demonstrados os dois cenários, quando o interruptor estava desligado e quando estava ligado.

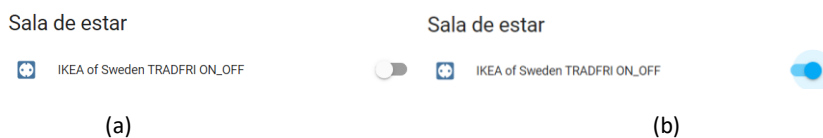


Figura 85 - Interruptor TRADFRI: (a) desativado; (b) ativado.

De igual forma, foi possível verificar o histórico do atuador, que é um registo onde fica disponível toda a informação sobre o estado da tomada inteligente, como por exemplo as horas do dia em que o atuador esteve desligado e ligado, o tempo que permaneceu nesse estado, etc. A Figura 86 apresenta um recorte de ecrã do histórico da tomada inteligente.

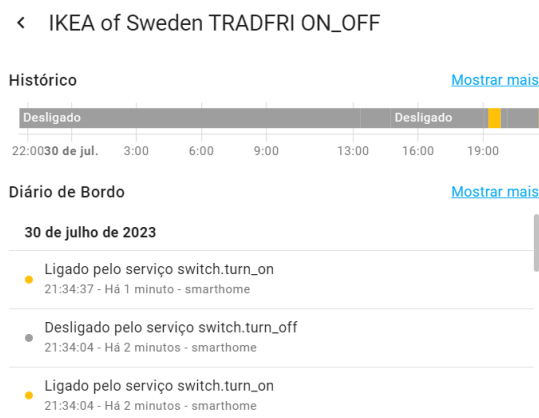


Figura 86 - Histórico do estado do atuador.

Como pode ser verificado na figura anterior, a tomada TRÅDFRI da IKEA respondeu corretamente aos pedidos do Home Assistant através do ZigBee com o *Add-*

on Zigbee Home Automation, verificando-se assim a compatibilidade e flexibilidade da plataforma com diferentes protocolos e dispositivos existentes no mercado.

5.3.3. Sensores Wi-Fi

Nesta secção apresenta-se a avaliação e teste dos sensores desenvolvidos e integrados com o ESPHome. Através deste *Add-on* foram desenvolvidos e integrados dois sensores, tal como foi explicado no capítulo anterior. Esses sensores abrangiam tanto a medição de temperatura e humidade como a deteção de abertura e fecho de portas e janelas.

5.3.3.1. Sensor de temperatura e humidade AM2302

Para verificar o correto funcionamento deste dispositivo bastou aceder ao painel principal e observar os valores exibidos na entidade correspondente. Uma leitura obtida a partir do sensor pode ser observada na Figura 87 e as leituras realizadas num intervalo específico (neste caso desde as 18h até às 21h) podem ser observadas na Figura 88.

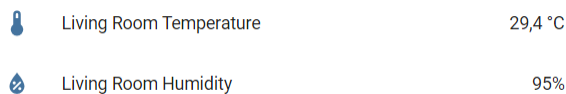


Figura 87 - Valores instantâneos de temperatura e humidade medidos com o AM2302.

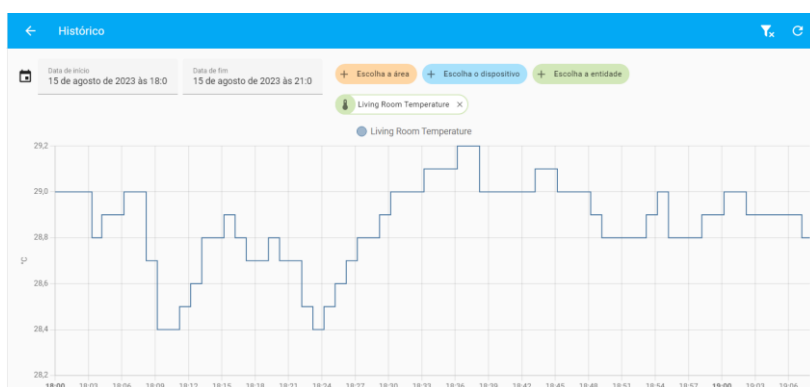


Figura 88 - Histórico do sensor de temperatura e humidade.

Tal como se estava à espera, o sensor de temperatura e humidade operou corretamente. Para validar a precisão das leituras do dispositivo, realizou-se uma

comparação com os dados de temperatura do site “*The Weather Channel*”, patrocinado pela IBM, apresentada na Figura 89.

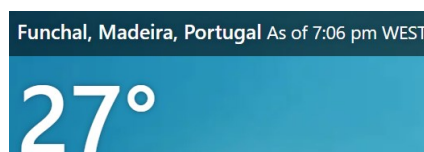


Figura 89 - Valor de temperatura obtido no “*The Weather Channel*” [62].

Verificou-se que a leitura obtida pelo sensor às 19:06 era bastante similar ao valor fornecido pelo *site*, com uma diferença de 1,8°C. É importante salientar que essa diferença observada pode ser devida à possível divergência na localização utilizada pelo *site* para a medição em relação à temperatura interna da residência onde a experiência com o sensor AM2302 foi realizada, pelo que se podem dar como válidos os valores medidos.

5.3.3.2. Sensor de portas e janelas

A fim de testar esse dispositivo foi necessária a utilização de um ímã, conforme foi mencionado na secção 3.2.4, onde o sensor magnético KY-021 foi apresentado.

Para tal foram estudadas duas situações distintas, a primeira quando o ímã estava em contacto com o KY-021 e o interruptor *Reed* dentro do sensor fechava o circuito, permitindo a circulação de corrente e fazendo com que no pino D4 fosse detetado um sinal que era interpretado como indicando que a porta ou janela estava fechada. Na segunda situação, o ímã foi afastado, simulando a abertura da porta ou janela. Estas duas situações podem ser verificadas nas Figura 90(a) e Figura 90(b).



Figura 90 – Resultados do sensor KY-021: (a) com campo magnético; (b) sem campo magnético.

Foi constatado que, através da combinação do sensor KY-021 e do *Add-on* ESPHome, foi possível criar um dispositivo capaz de monitorar em tempo real o estado de uma porta ou janela, conforme havia sido planeado inicialmente.

5.3.4. Automações

Até ao momento, os sensores e atuadores desenvolvidos foram submetidos a avaliações individuais, a fim de comprovar a transmissão de dados e a receção de comandos. Contudo, o que torna uma casa realmente inteligente são as automações que podem ser criadas com esses dispositivos para responder às necessidades do utilizador. Um dos principais focos de estudo neste trabalho residiu na interoperabilidade entre dispositivos que operam sob distintos protocolos de comunicação e são produzidos por diferentes marcas. Geralmente, a interligação desses dispositivos representa um desafio, que poderia ser dado como ultrapassado se os resultados dos testes fossem satisfatórios.

5.3.4.1. Sensor de porta de garagem com notificação através do dispositivo móvel

Para testar esta automação procedeu-se à utilização do dispositivo concebido com o sensor KY-021 e afastou-se o íman permitindo que o Home Assistant identificasse a abertura da porta da garagem. Foi então aguardado um período de 3 minutos, que corresponde ao *timeout* definido na automação para despoletar a notificação, e verificou-se que esta era efetivamente recebida pelo dispositivo móvel. Esta é uma automação que utiliza um sensor que comunica através de Wi-Fi e um serviço interno do Home Assistant.

A Figura 91 apresenta um recorte do ecrã do Home Assistant onde se visualiza a data e a hora em que a abertura da porta foi registada.



Figura 91 - Abertura da porta da garagem para despoletar a notificação.

Uma vez passados os 3 minutos, surgiu de facto o alerta no dispositivo móvel, tal como estava previsto e pode ser confirmado na Figura 92, validando assim a solução implementada.

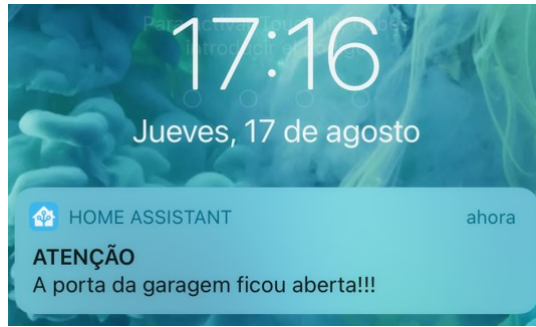


Figura 92 - Notificação despoletada pela automação passados os 3 minutos.

5.3.4.2. Sistema de alarme

Para verificar esta automação foi necessário simular uma situação de intrusão dentro da casa. Para isso, inicialmente, deslizou-se o interruptor de alarme disponível no *dashboard* do Home Assistant com a finalidade de armar o alarme e tornar o sistema sensível a qualquer alteração nas condições que poderiam disparar o alerta. Em seguida, simulou-se a abertura da porta da garagem (ou seja, o íman foi afastado do sensor) e passou-se a mão em frente ao sensor de presença (PIR). Dessa forma, todas as condições previamente definidas para acionar o alerta foram cumpridas pelo que, segundo o que foi estabelecido na automação, um alerta deveria ser enviado através do serviço *notify* ao telemóvel do utilizador, sendo também publicada uma mensagem por MQTT no tópico “Alarme”, que poderia ser utilizada para despoletar qualquer ação em outro dispositivo subscrito a esse tópico e que estivesse a aguardar especificamente por essa mensagem. Esse dispositivo poderia ser uma câmara para tentar capturar uma imagem do intruso, ou uma sirene para emitir um alerta (*buzzer*).

Na Figura 93 observa-se um recorte do ecrã com as condições cumpridas para despoletar o alerta.

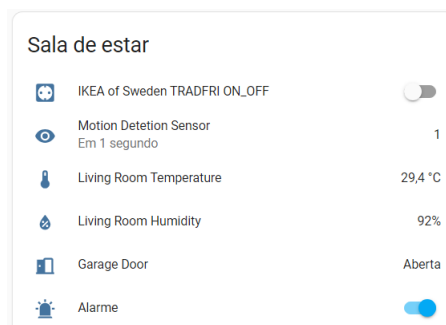


Figura 93 - Recorte de ecrã com as condições cumpridas para despoletar o alerta do alarme.

A Figura 94 demonstra que, depois de se cumprirem as condições acima mencionadas, foi efetivamente enviado o alerta ao dispositivo móvel do utilizador.

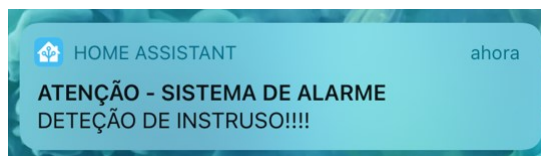


Figura 94 - Alerta de deteção de intruso enviado ao dispositivo móvel do utilizador.

A mensagem publicada no tópico “Alarme” é apresentada na Figura 95.

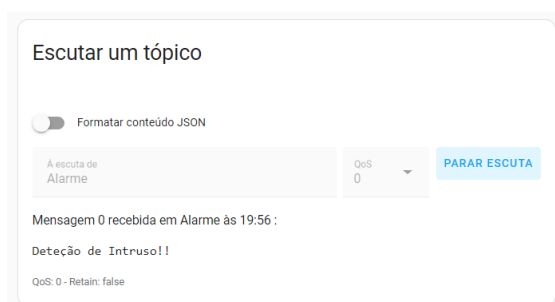


Figura 95 - Mensagem publicado no tópico “Alarme” após ser detetado um intruso.

Estas imagens demonstram que a implementação desta automação, que combina sensores Wi-Fi com MQTT, foi testada e validada. É de salientar que se alguma das condições não fosse cumprida o alerta não era despoletado. Naturalmente, podem ser acrescentadas novas condições e/ou pode ser reconfigurada a automação segundo as exigências do utilizador.

5.3.4.3. Termostato inteligente

Para validar esta implementação foi necessário recorrer à utilização de um aquecedor, com o objetivo de elevar a temperatura e assim controlar o valor detetado pelo sensor, para verificar se o atuador TRÁDFRI era ativado quando a temperatura medida pelo sensor AM2302 superava o limite predefinido na automação. Utilizou-se igualmente uma cubete contendo gelo para reduzir a temperatura, a fim de testar se o atuador era desativado quando a temperatura medida caía abaixo do outro valor definido.

Inicialmente, procedeu-se à avaliação da condição em que o dispositivo atuador deveria ser ativado, ao ser detetada uma temperatura superior a 29°C. Como a temperatura

ambiente era de 28,7°C, utilizou-se um isqueiro para induzir maior calor e dessa forma aumentar a temperatura registada pelo sensor até se atingir o limiar necessário para desencadear o gatilho que ativaria, por meio da tecnologia ZigBee, a tomada da IKEA. Na Figura 96 apresenta-se o gráfico dos valores de temperatura registados pelo sensor AM2302 e na Figura 97 o estado do atuador, ou seja, da tomada TRÅDFRI.

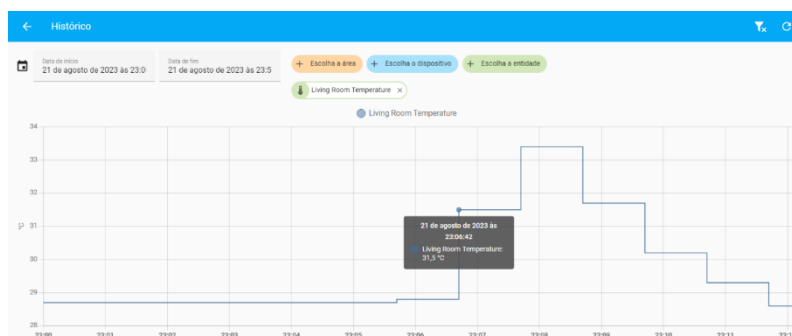


Figura 96 - Valores de temperatura registados pelo sensor AM2302.

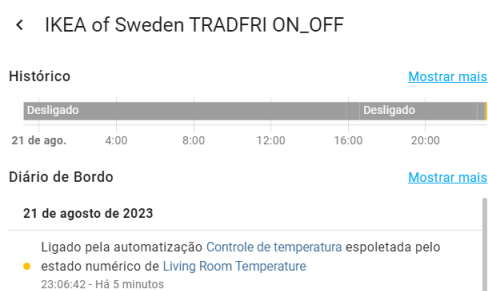


Figura 97 - Estado da tomada TRÅDFRI.

O primeiro valor acima dos 29°C foi registado pelo sensor de temperatura às 23:06:42, quando foi atingido um valor de 31,5°C, o suficiente para despoletar a automação e ligar a tomada através do ZigBee, tal como efetivamente aconteceu.

Quando a temperatura ambiente era inferior a 28°C era dado o *trigger* para que a tomada fosse desligada. Para simular essa situação utilizou-se uma cuvette de gelo para “forçar” esse valor, e desta forma verificar o comportamento da automação. Na Figura 98 e na Figura 99 podem ser observados mais uma vez os gráficos de temperatura obtidos pelo sensor e o comportamento da tomada respetivamente.

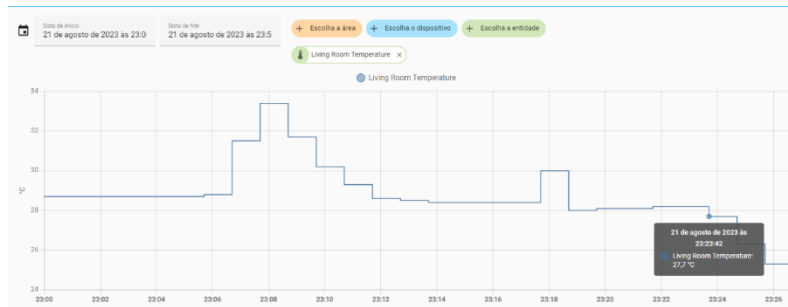


Figura 98 - Valores de temperatura registrados.

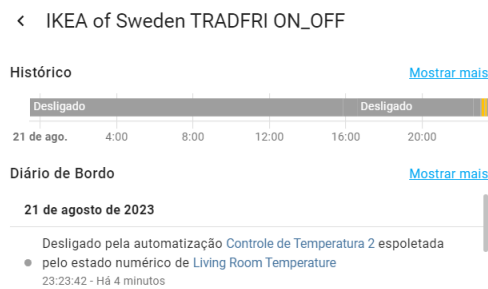


Figura 99 - Estado do atuador IKEA TRÅDFRI.

O primeiro valor de temperatura abaixo dos 28°C foi registrado pelo sensor as 23:23:42, quando foi atingido um valor de 27,7°C, o suficiente para despoletar a automação e desligar a tomada, tal como se verificou.

Assim sendo, pode-se dar como válida a solução proposta, pelo que neste momento passou a dispor-se de um termostato inteligente dentro da casa, que permitiria manter uma temperatura dentro dos valores desejados de forma automática, integrando um sensor e um atuador com diferentes tecnologias de comunicação, demonstrando assim a flexibilidade que o Home Assistant permite.

6. Conclusões

Após a conclusão deste projeto, é possível extrair algumas conclusões sobre o trabalho desenvolvido.

A domótica, que consiste na integração de tecnologia em residências, edifícios ou indústrias para automatizar e controlar diversas funções e sistemas, surgiu com o propósito de simplificar o cotidiano, aprimorar a eficiência, segurança, conforto e gestão dos locais onde é implementada. A implementação da domótica envolve a utilização de dispositivos como sensores e atuadores, juntamente com aplicações desenvolvidas para o controle e configuração do sistema. Através dessas aplicações, é possível realizar o controle e monitorização do espaço em que o sistema está instalado, além da criação de diferentes cenários de automação. As topologias mais comuns são a centralizada, a descentralizada e a distribuída.

A Internet das Coisas (IoT) e a domótica estão inter-relacionadas, sendo a IoT um conceito mais abrangente que engloba a interconexão em diversos setores, enquanto a domótica está mais focada na automação residencial. O objetivo principal da IoT é criar um ecossistema no qual dispositivos possam coletar, transmitir e receber dados, permitindo a automação de processos e tomada de decisões inteligentes na domótica.

As tecnologias e protocolos de comunicação mais utilizados nos sistemas IoT incluem Wi-Fi, MQTT, ZigBee, Z-Wave e Bluetooth.

Para o desenvolvimento do projeto de automação em questão, foram utilizadas três plataformas distintas. Uma delas, de código aberto, instalada num Raspberry Pi, capaz de integrar uma grande variedade de dispositivos e serviços de diferentes fabricantes. Essa solução permite a utilização de protocolos de comunicação sem fio, como ZigBee e Z-Wave, enquanto aproveita as capacidades do Raspberry Pi para comunicação através de Wi-Fi, MQTT e Bluetooth, constituindo uma solução abrangente.

Contudo, essa abordagem apresenta desvantagens, como uma curva de aprendizagem íngreme, especialmente para iniciantes, devido à complexidade das configurações e automações. Algumas configurações podem exigir intervenção manual, o que pode ser intimidante para utilizadores menos familiarizados com a edição de arquivos de configuração.

Outra plataforma utilizada foi o Hubitat Elevation, que oferece uma solução menos flexível em comparação com o Home Assistant, mas ainda assim permite a integração de dispositivos de diferentes marcas. O Hubitat opera apenas com os protocolos ZigBee e Z-Wave, mas sua vantagem mais significativa é a capacidade de automação independente da nuvem, executando a lógica localmente no próprio dispositivo. Isso garante tempos de resposta mais rápidos e maior privacidade, ao evitar a dependência de servidores externos.

A terceira *gateway* empregada foi a Silvercrest da Lidl, uma solução acessível em termos de custo para uma central domótica. No entanto, por ser uma marca mais comercial, os produtos são projetados para facilitar a instalação e configuração, mas apresentam menos recursos, menos flexibilidade e poucas opções de personalização em comparação com plataformas mais robustas, como o Home Assistant ou Hubitat Elevation. Essa plataforma é compatível apenas com dispositivos ZigBee 3.0, existindo até a possibilidade de não reconhecer todos os dispositivos que operam com esse protocolo, o que constitui uma limitação adicional.

Resumidamente, o Home Assistant oferece uma abordagem mais complexa e altamente personalizável, enquanto o Hubitat busca equilibrar a personalização com uma interface mais amigável. A Silvercrest, por ser uma solução mais simples, pode ser mais adequada para utilizadores menos técnicos.

Em termos de suporte a dispositivos e protocolos, o Home Assistant geralmente oferece uma ampla gama, o Hubitat está em expansão, mas pode não ser tão abrangente, e a Silvercrest, como depende de uma marca específica, possui suporte limitado.

A escolha entre o Home Assistant, Hubitat Elevation e Silvercrest dependerá, em última instância, de suas preferências pessoais, habilidades técnicas, orçamento e requisitos específicos de automação residencial.

6.1. Trabalhos futuros

Apesar deste trabalho ter obtido resultados positivos e atingido os objetivos estabelecidos, é de reconhecer que ainda existe muito por explorar e aprofundar quando se fala de domótica, até porque é uma área que está em constante desenvolvimento. Este trabalho oferece uma base para trabalhos futuros, no entanto, existem algumas áreas de

pesquisa promissoras que podem servir como pontos de partida para investigações subsequentes, nomeadamente:

- Utilizar mais sensores e atuadores compatíveis com as diferentes plataformas para criar mais cenários domóticos e customizar os *dashboards*.
- Explorar outros protocolos de comunicação, como o Thread [63] e o Matter [64], que podem oferecer uma ampla gama de oportunidades para aprimorar ainda mais a conectividade e a interoperabilidade dos dispositivos domóticos.
- Investigar a área da segurança dos dados, que com o aumento contínuo da integração de dispositivos e de dados pessoais nas casas inteligentes, se torna uma área de extrema importância, e que exige uma atenção especial.
- Integrar algoritmos de inteligência artificial que têm a capacidade de prever ações ou eventos futuros e, ao mesmo tempo, adaptar-se às preferências do utilizador em tempo real, com o objetivo de otimizar o conforto e a eficiência energética das residências inteligentes.
- Explorar as capacidades oferecidas pelos assistentes virtuais populares, como a Alexa, a Siri e o Google Assistant, o que pode aprimorar a experiência dos utilizadores das casas inteligentes.
- Aprofundar a análise dos padrões de consumo de energia em residências inteligentes permitirá avaliar o impacto da automação residencial na eficiência energética.

7. Referências

- [1] R. Nunes, “Análise Comparativa de Tecnologias para Domótica,” Instituto Superior Técnico/INESC-ID, Lisboa, 2002.
- [2] P. J. S. Monteiro, “Aplicação Android para sistema de Domótica,” masterThesis, Instituto Politécnico de Viseu. Escola Superior de Tecnologia e Gestão de Viseu, 2015. Accessed: Dec. 30, 2021. [Online]. Available: <https://repositorio.ipv.pt/handle/10400.19/3018>
- [3] L. A. C. Sousa, “Sistema modular de comunicação e controlo de dispositivos sensores/atuadores: Um ensaio na NextToYou - Network Solutions, Lda.,” Faculdade de Engenharia da Universidade do Porto, 2012.
- [4] A. Barros, “Edifícios Inteligentes e a Domótica Proposta de um Projecto de Automação Residencial utilizando o protocolo X-10,” Universidade Jean Piaget de Cabo Verde, Cabo Verde, 2010.
- [5] A. Rodríguez, “Diseño de un sistema domótico centralizado,” Universidad de Valladolid, Valladolid, 2014.
- [6] J. A. O. Ferreira, “Interface homem-máquina para domótica baseado em tecnologias Web,” Faculdade de Engenharia da Universidade do Porto, Porto, 2008.
- [7] P. Andrade, “Domótica en la vivienda,” Monografias.com. Accessed: Dec. 15, 2020. [Online]. Available: <https://www.monografias.com/trabajos93/domotica-vivienda/domotica-vivienda>
- [8] V. Williams, S. Terence J., and J. Immaculate, “Survey on Internet of Things based Smart Home,” in *2019 International Conference on Intelligent Sustainable Systems (ICISS)*, Feb. 2019, pp. 460–464. doi: 10.1109/ISS1.2019.8908112.
- [9] P. P. Gaikwad, J. P. Gabhane, and S. S. Golait, “A survey based on Smart Homes system using Internet-of-Things,” in *2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)*, Apr. 2015, pp. 0330–0335. doi: 10.1109/ICCPEIC.2015.7259486.
- [10] K. Rose, S. Eldridge, and L. Chapin, “The Internet of Things: An Overview”.
- [11] Y. Cheng, H. Zhang, and Y. Huang, “Overview of Communication Protocols in Internet of Things: Architecture, Development and Future Trends,” in *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, Santiago: IEEE, Dec. 2018, pp. 627–630. doi: 10.1109/WI.2018.00-25.
- [12] M. V. Diwakar and D. Pathiramanna, “A survey on 802.11n over 802.11g”, Accessed: Feb. 22, 2021. [Online]. Available: https://uh.edu/tech/isgrin/_documents/Divya_Madhu_report.pdf
- [13] P. Gupta, G. S. Tripathi, and S. Verma, “A compact MIMO antenna of meander-line structure for Wi-Fi application,” in *2016 International Conference on Emerging Trends in Electrical Electronics & Sustainable Energy Systems (ICETEESES)*, Sultanpur, India: IEEE, Mar. 2016, pp. 95–98. doi: 10.1109/ICETEESES.2016.7581359.
- [14] C. Gomez, “Diseño, desarrollo y prototipado de un Gateway M2M multiprotocolo para aplicaciones IoT.”
- [15] M. Saleiro and E. Ey, “ZigBee – uma abordagem prática,” Escola Superior de Tecnologia - EEE.
- [16] D. Gislason, *Zigbee wireless networking*. Oxford New York: Elsevier, Newnes, 2008.
- [17] K. Townsend, C. Cufí, Akiba, and R. Davidson, *Getting started with Bluetooth Low Energy*, 1st edition. Beijing Cambridge Farnham Köln Sebastopol Tokyo: O’Reilly, 2014.

- [18] P. M. Linh An and T. Kim, “A Study of the Z-Wave Protocol: Implementing Your Own Smart Home Gateway,” in *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*, Nagoya: IEEE, Apr. 2018, pp. 411–415. doi: 10.1109/CCOMS.2018.8463281.
- [19] E. Staff, “Catching the Z-Wave,” *Embedded.com*. Accessed: Jul. 14, 2021. [Online]. Available: <https://www.embedded.com/catching-the-z-wave/>
- [20] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, “Internet of Things: Survey and open issues of MQTT protocol,” in *2017 International Conference on Engineering & MIS (ICEMIS)*, Monastir: IEEE, May 2017, pp. 1–6. doi: 10.1109/ICEMIS.2017.8273112.
- [21] “What is Mosquitto MQTT?” Accessed: Jun. 01, 2021. [Online]. Available: <https://www.eginnovations.com/documentation/Mosquitto-MQTT/What-is-Mosquitto-MQTT.htm>
- [22] admin, “What is MQTT and How It Works,” *keepuptechnology*. Accessed: May 28, 2023. [Online]. Available: <https://keepuptechnology.com/what-is-mqtt-and-how-it-works/>
- [23] “MQTT V3.1 Protocol Specification.” Accessed: Jun. 03, 2021. [Online]. Available: <https://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html>
- [24] “Protocolo MQTT - Redes 1.” Accessed: Jul. 01, 2021. [Online]. Available: <https://www.gta.ufrj.br/ensino/eel878/redes1-2019-1/vf/mqtt/>
- [25] B. Nuttall, “What is a Raspberry Pi? | Opensource.com.” Accessed: Jul. 08, 2021. [Online]. Available: <https://opensource.com/resources/raspberry-pi>
- [26] “Raspberry gPIo - SparkFun Learn.” Accessed: Feb. 02, 2024. [Online]. Available: <https://learn.sparkfun.com/tutorials/raspberry-gpio/gpio-pinout>
- [27] “RASPBerry PI 3 B+ KIT STEM ELEMENT14 COMPLETO 32GB,” *Starware*. Accessed: Feb. 02, 2024. [Online]. Available: <https://tienda.starware.com.ar/producto/raspberry-pi-3-b-kit-stem-element14-completo-32gb/>
- [28] I. Hübschmann, “ESP8266 for IoT: A Complete Guide,” *Nabto*. Accessed: Jul. 03, 2021. [Online]. Available: <https://www.nabto.com/esp8266-for-iot-complete-guide/>
- [29] D. Madeira, “GUIA RÁPIDO #9 — Realizando a leitura de variáveis analógicas com o ESP8266 NodeMCU,” *Medium*. Accessed: Jul. 28, 2021. [Online]. Available: <https://medium.com/@automacaoem5minutos/guia-r%C3%A1pido-9-realizando-a-leitura-de-vari%C3%A1veis-anal%C3%B3gicas-com-o-esp8266-nodemcu-b6c9c4fd245e>
- [30] D. Says, “What is ESP32, how it works and what you can do with ESP32?,” *Circuit Schools*. Accessed: Dec. 02, 2022. [Online]. Available: <https://www.circuitschools.com/what-is-esp32-how-it-works-and-what-you-can-do-with-esp32/>
- [31] ESPRESSIF, “ESP32-WROOM-32 Datasheet.” Feb. 13, 2023. Accessed: Oct. 08, 2023. [Online]. Available: <https://www.espressif.com/en/support/documents/technical-documents?keys=ESP32-WROOM-32>
- [32] Texas Instruments, “A USB-Enabled System-On-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee Applications.” Jun. 2010. [Online]. Available: <https://www.ti.com/product/CC2531>
- [33] “CC2531 SONOFF Adaptador USB ZigBee.” Accessed: Dec. 02, 2023. [Online]. Available: https://mauser.pt/catalog/product_info.php?products_id=096-9491

- [34] “RF System Programmer, CC Debugger 250Kbaud Easy Operation 2.405-2.485GHz USB For Debugging CC2540,” Walmart.com. Accessed: Dec. 02, 2023. [Online]. Available: <https://www.walmart.com/ip/RF-System-Programmer-CC-Debugger-250Kbaud-Easy-Operation-2-405-2-485GHz-USB-For-Debugging-CC2540/1718455302>
- [35] “What is Arduino?” Accessed: Dec. 02, 2023. [Online]. Available: <https://www.arduino.cc/en/Guide/Introduction>
- [36] “Arduino Uno,” Components101. Accessed: Dec. 02, 2023. [Online]. Available: <https://components101.com/microcontrollers/arduino-uno>
- [37] “Hubitat Modelo de Elevação C-7 HUB – Vesternet.” Accessed: Feb. 02, 2024. [Online]. Available: <https://www.vesternet.com/pt-eu/products/hubitat-elevation-model-c-8-hub>
- [38] “Hubitat Elevation Home Automation Hub (Model C-7) Compatible with Alexa, HomeKit, Google Home, Zigbee, Z-Wave, Lutron - Amazon.com,” Amazon.com. Accessed: Dec. 18, 2022. [Online]. Available: <https://www.amazon.com/Hubitat-Elevation-Home-Automation-Hub/dp/B07D19VVTX>
- [39] “Home Automation Features and Benefits | Hubitat Elevation Local Home Automation,” HUBITAT. Accessed: Dec. 02, 2023. [Online]. Available: <https://hubitat.com/pages/home-automation-features>
- [40] “Silvercrest Gateway Zigbee Smart Home,” European House Hold. Accessed: Feb. 03, 2024. [Online]. Available: <https://europeanhousehold.com/products/silvercrest-gateway-zigbee-smart-home>
- [41] “GATEWAY SGWZ 1 A1.” Silvercrest.
- [42] A. Industries, “AM2302 (wired DHT22) temperature-humidity sensor.” Accessed: Feb. 03, 2024. [Online]. Available: <https://www.adafruit.com/product/393>
- [43] K. C. Sahoo and U. C. Pati, “IoT based intrusion detection system using PIR sensor,” in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore: IEEE, May 2017, pp. 1641–1645. doi: 10.1109/RTEICT.2017.8256877.
- [44] A. Engelniederhammer, G. Papastefanou, and L. Xiang, “Crowding density in urban environment and its effects on emotional responding of pedestrians: Using wearable device technology with sensors capturing proximity and psychophysiological emotion responses while walking in the street,” *J. Hum. Behav. Soc. Environ.*, vol. 29, no. 5, pp. 630–646, Jul. 2019, doi: 10.1080/10911359.2019.1579149.
- [45] “Smart Lighting Enters the Workplace,” DigiKey. Accessed: Dec. 27, 2022. [Online]. Available: <https://www.digikey.be/nl/articles/smart-lighting-enters-the-workplace>
- [46] “Grove - Adjustable PIR Motion Sensor | Seeed Studio Wiki.” Accessed: Feb. 03, 2024. [Online]. Available: https://wiki.seeedstudio.com/Grove-Adjustable_PIR_Motion_Sensor/
- [47] “Grove - PIR Motion Sensor,” Arduino Official Store. Accessed: Dec. 02, 2023. [Online]. Available: <https://store.arduino.cc/products/grove-pir-motion-sensor>
- [48] “What is an Electrical Relay? OMRON Device & Module Solutions - Europe.” Accessed: Feb. 03, 2024. [Online]. Available: <https://components.omron.com/eu-en/products/basic-knowledge/relays/basics>
- [49] “Gravity: Digital 5A Relay Module for Arduino.” Accessed: Dec. 02, 2023. [Online]. Available: <https://www.dfrobot.com/product-64.html>

- [50] ArduinoModules, “KY-021 Mini Magnetic Reed Switch Module,” ArduinoModulesInfo. Accessed: Dec. 02, 2023. [Online]. Available: <https://arduinomodules.info/ky-021-mini-magnetic-reed-switch-module/>
- [51] “TRÅDFRI Cj tomada c/comando,” IKEA. Accessed: Dec. 05, 2023. [Online]. Available: <https://www.ikea.com/pt/pt/p/tradfri-cj-tomada-c-comando-smart-40364748/>
- [52] H. Assistant, “Getting Started,” Home Assistant. Accessed: Feb. 03, 2024. [Online]. Available: <https://www.home-assistant.io/getting-started/>
- [53] H. Assistant, “Integrations,” Home Assistant. Accessed: Dec. 05, 2023. [Online]. Available: <https://www.home-assistant.io/integrations/>
- [54] “Hubitat Elevation® | Local, Reliable, Fast and Private Home Automation,” HUBITAT. Accessed: Feb. 03, 2024. [Online]. Available: <https://hubitat.com/>
- [55] “Lidl Home,” App Store. Accessed: Apr. 14, 2024. [Online]. Available: <https://apps.apple.com/pt/app/lidl-home/id1504839856>
- [56] “My Hubitat.” Accessed: Aug. 10, 2022. [Online]. Available: <http://findmyhub.hubitat.com/>
- [57] H. Assistant, “Raspberry Pi,” Home Assistant. Accessed: Dec. 08, 2021. [Online]. Available: <https://www.home-assistant.io/installation/raspberrypi/>
- [58] “operating-system/Documentation/network.md at dev · home-assistant/operating-system,” GitHub. Accessed: Dec. 08, 2021. [Online]. Available: <https://github.com/home-assistant/operating-system/blob/dev/Documentation/network.md>
- [59] “Software.” Accessed: Dec. 08, 2021. [Online]. Available: <https://www.arduino.cc/en/software>
- [60] “FLASH-PROGRAMMER Software programming tool | TI.com.” Accessed: Dec. 14, 2021. [Online]. Available: <https://www.ti.com/tool/FLASH-PROGRAMMER>
- [61] T. Nordquist, “MQTT Explorer,” MQTT Explorer. Accessed: Feb. 06, 2024. [Online]. Available: <http://mqtt-explorer.com/>
- [62] “Previsão e condições meteorológicas para Socorro, Lisboa - The Weather Channel | Weather.com,” The Weather Channel. Accessed: Nov. 16, 2023. [Online]. Available: <https://weather.com/pt-PT/clima/hoje/l/São+Cristóvão+e+São+Lourenço+Lisboa?canonicalCityId=df480f6b855b81f2b888b29a1a3c15a2ecf42747c8f36839acbb6f22e4f4f7aa>
- [63] “Thread Benefits.” Accessed: Feb. 17, 2024. [Online]. Available: <https://www.threadgroup.org/What-is-Thread/Thread-Benefits>
- [64] “Build With Matter | Smart Home Device Solution,” CSA-IOT. Accessed: Feb. 17, 2024. [Online]. Available: <https://csa-iot.org/all-solutions/matter/>

Anexo I

Neste anexo apresenta-se o código utilizado para programar o microcontrolador ESP8266 para funcionar com o sensor PIR.

```
#include "credentials.h"
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <Bounce2.h>
#define PIR D1 //Define o pino D1 como o pino ao qual o sensor PIR está conectado

//Define as credenciais de rede e MQTT através do ficheiro "credentials.h" para
manter informações sensíveis fora do código principal.
const char* ssid = networkSSID;
const char* password = networkPASSWORD;
const char* mqttServer = mqttSERVER;
const char* mqttUsername = mqttUSERNAME;
const char* mqttPassword = mqttPASSWORD;
int lastPirState = LOW; //Inicializa a variável para o estado anterior do sensor
PIR.
char pubTopic[] = "sensor/pir"; //Define o tópico MQTT para o qual os dados do
sensor serão enviados.

Bounce debouncer = Bounce(); //Inicializa o objeto debouncer para eliminar o
efeito de rebote do sensor PIR.

//Inicializa os objetos necessários para a conexão WiFi e MQTT.
WiFiClient espClient;
PubSubClient client(espClient);

void setup_wifi() { //Configura a conexão WiFi chamando esta função no setup
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password); //Inicia a conexão WiFi com as credenciais
fornecidas

  while (WiFi.status() != WL_CONNECTED) { //Aguarda até que a conexão WiFi
seja estabelecida.
    delay(500);
    Serial.print(".");
  }

  randomSeed(micros());

  Serial.println("");
```

```

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

void reconnect() { //Tenta reconectar ao servidor MQTT em caso de desconexão.
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        //Gera um ID de cliente único para a conexão MQTT.
        String clientId = "ESP8266Client-";
        clientId += String(random(0xffff), HEX);
        //Tenta se conectar ao servidor MQTT com o ID do cliente e as credenciais.
        if (client.connect(clientId.c_str(), mqttUsername, mqttPassword)) {
            Serial.println("connected");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            //Aguarda 5 segundos antes de tentar a conexão novamente em caso de falha.
            delay(5000);
        }
    }
}

void setup() {
    Serial.begin(115200); //Inicia a comunicação serial
    pinMode(PIR, INPUT); //Configura o pino do sensor PIR como entrada.
    //Configura o debouncer para o sensor PIR.
    debouncer.attach(PIR);
    debouncer.interval(15);
    setup_wifi(); //Chama a função para configurar a conexão WiFi.
    client.setServer(mqttServer, 1883); //Configura o servidor MQTT e a porta à
    qual se conectará.
}

void loop() { //Verifica se a conexão MQTT está ativa; se não estiver, tenta
reconectar.
    if (!client.connected()) {
        reconnect();
    }
    client.loop(); //Mantém a conexão MQTT ativa.
    int pirState = digitalRead(PIR); //Lê o estado atual do sensor PIR
    //Converte o estado do sensor para uma string.
    char payload[1];
    itoa(pirState, payload, 10);

    if (pirState != lastPirState) { //Compara o estado atual do sensor com o
estado anterior.
        if (pirState == HIGH) {

```

```
        Serial.println("Movimiento detetado");
        client.publish(pubTopic, payLoad); //Publica o estado do sensor no tópico
MQTT correspondente
    } else {
        Serial.println("No hay movimiento");
        client.publish(pubTopic, payLoad); //Publica o estado do sensor no tópico
MQTT correspondente
    }
}
lastPirState = pirState; //Atualiza o estado anterior do sensor
delay(1000); //Aguarda 1 segundo antes de realizar a próxima leitura do sensor
}
```


Anexo II

Neste anexo apresenta-se o código utilizado para programar o microcontrolador ESP8266 para funcionar com o relé.

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include "credentials.h"
#include <Bounce2.h>
#define RELAY D4 //Define o pino D4 para controlar o relé

//Define as credenciais de rede e MQTT através do ficheiro "credentials.h" para
manter informações sensíveis fora do código principal.
const char* ssid = networkSSID;
const char* password = networkPASSWORD;
const char* mqttServer = mqttSERVER;
const char* mqttUsername = mqttUSERNAME;
const char* mqttPassword = mqttPASSWORD;
WiFiClient espClient;
PubSubClient client(espClient);
unsigned long lastMsg = 0;
#define MSG_BUFFER_SIZE (50)
char msg[MSG_BUFFER_SIZE];
int value = 0;

void setup_wifi() { //Configura a conexão WiFi

    delay(10);

    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    randomSeed(micros());

    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
```

```

    Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length) {
    Serial.println("Mensaje recibido desde el t3pico: " + String(topic));

    // Converte o payload para String para comparaci3n
    String payloadStr = "";
    for (int i = 0; i < length; i++) {
        payloadStr += (char)payload[i];
    }

    // Verifica o conteudo da mensagem
    if (payloadStr == "1") {
        digitalWrite(RELAY, HIGH); // Activa o rel3
        Serial.println("Relay activado");
    } else if (payloadStr == "0") {
        digitalWrite(RELAY, LOW); // Desactiva o rel3
        Serial.println("Relay desactivado");
    }
}

void reconnect() { //Tenta reconectar ao servidor MQTT em caso de desconex3o.
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        // Gera um ID de cliente 3nico para a conex3o MQTT.
        String clientId = "ESP8266Client-";
        clientId += String(random(0xffff), HEX);
        //Tenta se conectar ao servidor MQTT com o ID do cliente e as credenciais.
        if (client.connect(clientId.c_str(), mqttUsername, mqttPassword)) {
            Serial.println("connected");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            //Aguarda 5 segundos antes de tentar a conex3o novamente em caso de falha.
            delay(5000);
        }
    }
}

void setup() {
    pinMode(RELAY, OUTPUT); // Initialize the BUILTIN_LED pin as an
output
    Serial.begin(115200);
    setup_wifi();
    client.setServer(mqttServer, 1883);
    client.setCallback(callback);
}

```

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


Anexo III

Neste anexo apresenta-se o código utilizado para programar os dispositivos com as credenciais necessárias para utilizar o Broker.

```
char networkSSID[] = "SSID";  
char networkPASSWORD[] = "XXXXXXXX";  
  
char mqttSERVER[] = "192.168.1.107";  
char mqttUSERNAME[] = "homeassistant";  
char mqttPASSWORD[] = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx";
```