



Sistema Multiagente baseado numa Rede Mesh

DAVID JOSÉ BRANDÃO SILVA

novembro de 2022

POLITÉCNICO DO PORTO
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Sistema Multiagente baseado numa Rede Mesh

David José Brandão Silva

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



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA
Instituto Superior de Engenharia do Porto

Novembro, 2022

Esta dissertação satisfaz, parcialmente, os requisitos que constam da Ficha de Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Electrotécnica e de Computadores, Área de Especialização em Automação e Sistemas.

Candidato: David José Brandão Silva, N.º 1141294, 1141294@isep.ipp.pt

Orientação Científica: Luís Gomes, lfg@isep.ipp.pt

Coorientação Científica: Zita Vale, zav@isep.ipp.pt



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA
Instituto Superior de Engenharia do Porto
Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto

Novembro, 2022

Agradecimentos

À minha família e amigos pelo apoio durante os anos de mestrado.

Aos meus orientadores pelo uso da grande experiência e capacidade para ajudar e apoiar a dissertação da melhor forma.

Resumo

A Internet das Coisas é um conceito cada vez mais presente nas nossas vidas, tendo em conta o aumento acentuado de obtenção de objetos inteligentes para uso profissional e doméstico, com capacidade para entreter e informar com base na interação com o mundo real, através de ligações à internet.

A Inteligência Artificial tem aplicação em objetos e dispositivos móveis, com vista a aumentar o seu poder de computação e implementar novas funcionalidades nas suas aplicações, como o reconhecimento facial. Uma das áreas de inteligência artificial em expansão é Inteligência Artificial distribuída, dedicada à investigação e desenvolvimento de soluções distribuídas e na qual os Sistemas Multiagente são uma das categorias.

No âmbito desta dissertação foi feito um estudo do estado da arte de sistemas multiagente, internet das coisas e edifícios inteligentes, e sobre tecnologias e ferramentas que possam ser utilizadas para implementar um sistema multiagente em *hardware*. Com base nesse estudo, apresenta-se uma solução de sistema multiagente, concebido para um ambiente de edifícios inteligentes. Os agentes pertencentes ao sistema, utilizam Bluetooth Mesh para a comunicação entre si e são implementados nos microcontroladores ESP32.

Para analisar o desempenho da solução, foram definidos testes específicos que permitem avaliar o comportamento da comunicação entre os agentes, num ambiente interior de um edifício. Esses testes comprovaram a fiabilidade da comunicação entre os agentes dentro da mesma divisão e em divisões diferentes, bem como o comportamento da rede no controlo e monitorização do ambiente envolvente e a respetiva velocidade.

Os resultados dos testes demonstraram que a rede de Bluetooth Mesh consegue entregar as mensagens transmitidas, sem perdas nos casos em que os nós estão dentro da mesma divisão, e quando existe no meio obstáculos e um nó retransmissor. Por outro lado, existiu uma maior perda de mensagens quando as distâncias entre os nós eram maiores e/ou existiam paredes no meio, baixando a fiabilidade desta tecnologia de comunicação.

Palavras-Chave: Agentes, Bluetooth Mesh, ESP32, ESP-IDF, Inteligência Artificial, Sistemas Multiagente

Abstract

The Internet of Things is a concept increasingly present in our lives, given the sharp increase in obtaining smart objects for professional and domestic use, with the ability to entertain and inform based on interaction with the real world, through internet connections.

Artificial Intelligence has applications in mobile objects and devices, in order to increase their computing power and implement new functionalities in their applications, such as facial recognition. One of the expanding areas of artificial intelligence is distributed Artificial Intelligence, dedicated to the research and development of distributed solutions and in which Multi-agent Systems are one of the categories.

As part of this dissertation, a study of the state of the art of multi-agent systems, internet of things and smart buildings, and about technologies and tools that can be used to implement a multi-agent system on hardware was done. Based on this study, a multi-agent system solution, designed for a smart building environment, is presented. The agents belonging to the system, use Bluetooth Mesh for communication with each other and are based on ESP32 microcontrollers.

To analyze the performance of the solution, specific tests were defined to evaluate the behavior of the communication between the agents, in an indoor environment of a building. These tests proved the reliability of the communication between the agents within the same room and in different rooms, as well as the behavior of the network in controlling and monitoring the surrounding environment and its speed. The test results showed that the Bluetooth Mesh network is able to deliver the transmitted messages without losses in cases where the nodes are within the same room, and when there are obstacles and a relay node in between. On the other hand, there was a greater loss of messages when the distances between the nodes were greater and/or there were walls in the middle, lowering the reliability of this communication technology.

Keywords: Agents, Artificial Intelligence, Bluetooth Mesh, ESP32, ESP-IDF, Multi-Agent Systems

Índice

Lista de Figuras	ix
Lista de Tabelas	xi
Lista de Acrónimos	xiii
1 Introdução	1
1.1 Contextualização	1
1.2 Definição do Problema	2
1.2.1 Resultados esperados	2
1.2.2 Objetivos	3
1.3 Organização da Dissertação	3
2 Estado da Arte	5
2.1 Internet das Coisas	5
2.1.1 Arquiteturas aplicadas no âmbito dos sistemas de Internet das Coisas	6
2.1.2 Aplicações	12
2.2 Agentes e Sistemas Multiagente	14
2.2.1 Noção de Agente	15
2.2.2 Conceito de Sistemas Multiagente	18
2.2.3 Comunicação	19
2.2.4 Coordenação	21
2.2.5 Negociação	22
2.2.6 Organização	23
2.3 Edifícios Inteligentes	28
2.3.1 Sistemas de Automação e Gestão de Edifícios	28
2.3.2 Internet das Coisas em Edifícios Inteligentes	29
2.3.3 Inteligência Artificial em Edifícios Inteligentes	33
3 Tecnologias e Ferramentas	37
3.1 Bluetooth	38
3.2 Bluetooth Low Energy (BLE)	38
3.2.1 Conceito	39
3.2.2 Arquitetura	39

3.2.3	Áreas de aplicação	41
3.3	Bluetooth Mesh	42
3.3.1	Conceitos e terminologia	43
3.3.2	Modelos de um nó de rede	45
3.3.3	Topologia <i>mesh</i>	46
3.3.4	Funcionalidades	47
	Funcionalidade de retransmissão	47
	Funcionalidade de proxy	48
	Funcionalidade de amizade	49
3.3.5	Arquitetura	50
3.3.6	Segurança	52
3.4	Componentes de <i>hardware</i>	54
4	Proposta de Solução	57
4.1	Requisitos do sistema	57
4.1.1	Requisitos dos agentes	57
4.1.2	Requisitos da rede de comunicação	58
4.1.3	Requisitos de <i>software</i>	58
4.2	Conceito da solução	58
4.3	Definição de um nó	59
4.4	Rede de nós	60
4.5	Arquitetura da solução	60
4.6	Nó <i>gateway</i>	62
5	Sistema Multiagente Baseado numa Rede Mesh	63
5.1	Estruturas de dados do Bluetooth Mesh no ESP-IDF	64
5.2	Inicialização do módulo de Bluetooth Mesh	65
5.3	Integração de dispositivos na rede	66
5.4	Modelos de Bluetooth Mesh	67
5.5	Configuração do nó	68
5.6	Funções de notificação e gestão de eventos	68
5.7	Envio de mensagens entre modelos	69
5.8	Receção de mensagens	71
5.9	Aplicação sobre Bluetooth Mesh	71
6	Testes e Análise de Resultados	75
6.1	Funcionamento de On/Off em rede mesh	76
6.1.1	Casos de testes	76
6.1.2	Resultados dos testes	78
6.2	Controlo e monitorização de temperatura	78
6.2.1	Casos de testes	79

6.2.2	Resultados dos testes	80
6.3	Controlo de presença	82
6.3.1	Casos de testes	83
6.3.2	Resultados dos testes	83
7	Conclusões	85
7.1	Conclusões e Discussão	85
7.2	Fatores Limitadores	86
7.3	Trabalho Futuro	87
	Referências	88

Lista de Figuras

2.1	Áreas de Aplicação para a Internet das Coisas [Fadda et al., 2019]	6
2.2	Arquitetura de Três Camadas [Omoniwa et al., 2019]	8
2.3	Arquitetura de Quatro Camadas [Omoniwa et al., 2019]	9
2.4	Arquitetura de Cinco Camadas [Omoniwa et al., 2019]	9
2.5	Níveis de Computação utilizados na Internet das Coisas [Omoniwa et al., 2019]	11
2.6	Composição de uma Cidade Inteligente [Silva et al., 2018a]	14
2.7	Estrutura de um Sistema Multiagente [Reis and Oliveira, 2003]	19
2.8	Disposição de uma Hierarquia [Horling and Lesser, 2004]	23
2.9	Disposição de uma Holarquia [Horling and Lesser, 2004]	24
2.10	Disposição de uma Aliança [Horling and Lesser, 2004]	24
2.11	Disposição de uma Equipa [Horling and Lesser, 2004]	24
2.12	Disposição de uma Congregação [Horling and Lesser, 2004]	25
2.13	Disposição de uma Sociedade [Horling and Lesser, 2004]	25
2.14	Disposição de uma Federação [Horling and Lesser, 2004]	25
2.15	Disposição de uma Mercado [Horling and Lesser, 2004]	26
2.16	Disposição de uma organização em Matriz [Horling and Lesser, 2004]	26
2.17	Disposição de uma organização Composta [Horling and Lesser, 2004]	27
2.18	Sumário de Aplicações de Edifícios com recurso a IoT [Jia et al., 2019]	33
3.1	Arquitetura de Bluetooth Low Energy [Bluetooth Special Interest Group (SIG), 2022b]	40
3.2	Estrutura Hierárquica dos Servidores, Características e Descritores [Bluetooth Special Interest Group (SIG), 2022b]	41
3.3	Conceito de Publicação/Subscrição [Bluetooth Special Interest Group (SIG), 2020]	42
3.4	Rede com topologia <i>mesh</i> [Rondon et al., 2020]	47
3.5	Exemplo da funcionalidade de Retransmissão [Bluetooth Special Interest Group (SIG), 2020]	48
3.6	Exemplo da funcionalidade de Proxy [Bluetooth Special Interest Group (SIG), 2020]	49
3.7	Exemplo da funcionalidade de Amizade [Bluetooth Special Interest Group (SIG), 2020]	50
3.8	Arquitetura do Bluetooth Mesh [Bluetooth Special Interest Group (SIG), 2020]	52

3.9	ESP32-WROOM-32D	54
4.1	Arquitetura geral da solução	61
5.1	Comunicação entre os modelos <i>Generic OnOff</i>	67
5.2	Comunicação entre modelos <i>Generic OnOff</i> com retransmissão	70
5.3	Componentes de aplicação	71
6.1	Casos de testes de desempenho de rede de Bluetooth Mesh	77
6.2	Ligações entre os nós para controlo e monitorização de temperatura	79
6.3	Resultados de Testes de Controlo e Monitorização de Temperatura com a evolução da Temperatura (a) e do Estado de Ventilador (b) ao longo do tempo	81
6.4	Resultados de Testes de Controlo e Monitorização de Temperatura com a evolução da Temperatura (a) e do Estado de Aquecedor (b) ao longo do tempo	82
6.5	Ligações entre os nós para o controlo de presença	83
6.6	Resultados de Testes de Controlo de Presença com a evolução do Estado do Sensor PIR (a) e do Estado da Luz(b) ao longo do tempo	84

Lista de Tabelas

2.1	Componentes necessárias no comportamento de um agente	15
2.2	Principais objetivos funcionais de edifícios inteligentes	32
3.1	Comparação de funcionalidades de plataformas de Hardware	55
6.1	Critérios de testes de desempenho de rede de Bluetooth Mesh	78
6.2	Resultados de testes de desempenho de rede de Bluetooth Mesh	78

Lista de Acrónimos

AGMO	<i>Algoritmos Genéticos Multi-objetivo</i>
AGMOH	<i>Algoritmos Genéticos Multi-objetivo Híbridos</i>
AMQP	<i>Advanced Message Queing Protocol</i>
API	<i>Application Programming Interface</i>
AppKey	<i>Application Key</i>
ATT	<i>Attribute Profile</i>
AVC	<i>Sistema de Aquecimento, Ventilação e Climatização</i>
BAS	<i>Building Automation System</i>
BLE	<i>Bluetooth Low Energy</i>
Bluetooth SIG	<i>Bluetooth Special Interest Group</i>
CPU	<i>Central Processor Unit</i>
DevKey	<i>Device Key</i>
DNN	<i>Deep Neural Networks</i>
DL	<i>Deep Learning</i>
DR	<i>Demand Response</i>
DSM	<i>Demand Side Management</i>
EI	<i>Edifício Inteligente</i>
FHSS	<i>Frequency Hopping Spread Spectrum</i>
GPIO	<i>General-Purpose Input/Output</i>
GATT	<i>Generic Attribute Profile</i>
EI	<i>Edifício Inteligente</i>
IA	<i>Inteligência Artificial</i>
IAD	<i>Inteligência Artificial Distribuída</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IIoT	<i>Industrial Internet of Things</i>
IoMT	<i>Internet of Medical Things</i>
IoT	<i>Internet of Things</i>
ISM	<i>Industrial Scientific Medical</i>
LTE	<i>Long Term Evolution</i>
ML	<i>Machine Learning</i>
MQTT	<i>Message Queue Telemetry Transport</i>
MTC	<i>Machine Type Communication</i>
NetKey	<i>Network key</i>
OSI	<i>Open Systems Interconnection</i>

PIR	<i>Passive Infrared</i>
RAM	<i>Random Access Memory</i>
RDP	<i>Resolução Distribuída de Problemas</i>
RL	<i>Reinforcement Learning</i>
RFID	<i>Radio Frequency Identification</i>
RTOS	<i>Real Time Operating System</i>
SAE	<i>Sistema de Automação de Edifícios</i>
SAGE	<i>Sistema de Automação e Gestão de Edifícios</i>
SGEE	<i>Sistema de Gestão de Energia do Edifício</i>
SGE	<i>Sistema de Gestão de Edifícios</i>
SMA	<i>Sistema Multi-Agente</i>
SoA	<i>Service oriented Approach</i>
SoC	<i>System-on-Chip</i>
STI	<i>Sistemas de Transportes Inteligentes</i>
TDMA	<i>Time Division Multiple Access</i>
TTL	<i>Time-to-Live</i>
UTBS	<i>United Technology Building Systems</i>
V2I	<i>Vehicle to Infrastructure</i>
V2V	<i>Vehicle to Vehicle</i>
Wi-Fi	<i>Wireless Fidelity</i>

Capítulo 1

Introdução

O primeiro capítulo desta dissertação tem como objetivo apresentar a introdução, começando por contextualizar e introduzir conceitos chave, seguido da descrição do problema da falta de soluções de computação distribuída para edifícios inteligentes, a enumeração dos objetivos a alcançar com base no objetivo principal, os resultados esperados do sistema multiagente a definir e termina com uma descrição da estrutura do documento.

1.1 Contextualização

A Internet das Coisas (IoT) tem revelado um crescimento acentuado de aplicação nas mais diversas áreas, para uso profissional e uso doméstico, com capacidade de entreter, informar e atuar, com base na interação com o mundo real e na recolha de dados automática, através de ligações à Internet [Bembe et al., 2019]. Estes objetos, como relógios, televisões, sistemas de climatização e frigoríficos são cada vez mais procurados tendo em conta a nossa procura constante por conforto, segurança e eficiência, os quais estes objetos aumentam, através da perceção do ambiente envolvente e a reação perante esse ambiente.

Ao nível da Inteligência Artificial (IA), este conceito tem-se focado na aplicação em objetos ou dispositivos móveis, de forma a melhorar a qualidade de computação e criar novos tipos de aplicação, como reconhecimento facial, reconhecimento de voz e realidade virtual [Chen et al., 2020]. Uma das áreas de inteligência artificial em expansão é a área de inteligência artificial distribuída, dedicada à pesquisa e desenvolvimento de soluções distribuídas.

Os Sistemas Multiagente são uma das categorias de algoritmos de Inteligência Artificial Distribuída (IAD), para além da Inteligência Artificial Paralela e Resolução Distribuída de Problemas (RDP) [Dorri et al., 2018]. Os Sistemas Multiagente (SMA) são um conceito que possibilita a divisão de uma tarefa complexa em tarefas mais pequenas, pelas quais os agentes, entidades autónomas e interligadas entre si, são responsáveis [Chira and Dumitrescu, 2006].

Cada agente de um SMA é autônomo para tomar decisões sobre a ação a tomar para resolver as suas tarefas. No entanto, estas entidades também têm capacidade de trabalhar em conjunto com os seus pares, para colaborar na execução de tarefas e cumprir um conjunto de objetivos. Esta interação e trabalho em conjunto dá origem aos sistemas multiagente [Reis and Oliveira, 2003].

Os SMAs podem ser utilizados como uma abordagem para melhorar as capacidades e eficiência em sistema para aplicações de edifícios inteligentes (EI), nomeadamente na lógica de controlo [Dragomir, 2014]. Os EI são um conceito que permite aos utilizadores novas possibilidades de conveniência, conforto e eficiência para os seus ocupantes através de meios tecnológicos [Qolomany et al., 2019].

1.2 Definição do Problema

A massificação de conceitos como IoT e IA veio tornar possível a aplicação de sistemas inteligentes distribuídos que suportem o utilizador no seu dia-a-dia. Porém, não existe uma solução que nos permita tirar proveito, em simultâneo, da computação distribuída, *hardware* de baixo custo, comunicação *mesh* e colaboração entre dispositivos sem o recurso a um sistema centralizador (por exemplo, o uso de uma *cloud*). Tal sistema iria permitir ao utilizador a programação de vários dispositivos colaborativos para a resolução de problemas singulares de cada utilizador.

1.2.1 Resultados esperados

Usando vários cenários definidos, o meio utilizado para a comunicação entre os diferentes agentes, transmite as respetivas mensagens com baixa latência e perda de pacotes. Nos cenários de controlo e monitorização, é esperada a leitura e ação corretas sobre o ambiente. Para testar a performance da solução, irão ser utilizados três cenários de aplicação: funcionamento de on/off em rede *mesh*, controlo e monitorização de temperatura e controlo de presença.

O funcionamento de on/off em rede *mesh*, visa nomeadamente demonstrar o potencial e escalabilidade de uma rede *mesh* de um sistema multiagente. O cenário de controlo e monitorização de temperatura é uma aplicação que visa o controlo e a monitorização de temperatura, através de agentes de *hardware* com aquecedores ou ventiladores ligados, os quais irão receber dados de temperatura e pedidos de ligação de outro agente de *hardware* com sensores de temperatura ligados. O controlo de presença é uma aplicação de controlo de iluminação com base na deteção de movimento, isto é, ligar e desligar luzes conectadas a um agente de *hardware*, com base em mensagens de pedidos enviadas por outro agente de *hardware*, que por conseguinte está ligado um sensor de movimento.

1.2.2 Objetivos

O objetivo principal desta dissertação visa a conceção, implementação e teste de uma solução de um sistema multiagente composto por sensores e atuadores distribuídos para implementação em edifícios inteligentes.

Para atingir o objetivo principal, foram traçados os seguintes objetivos intermédios:

- O_1 Análise do estado da arte referente a sistemas multiagente, internet das coisas e edifícios inteligentes;
- O_2 Estudo das metodologias, formas de organização e de comunicação em sistemas multiagente;
- O_3 Análise das tecnologias e ferramentas associadas a sistemas multiagente e a internet das coisas, nomeadamente tecnologias de comunicação;
- O_4 Planeamento de uma solução de um sistema multiagente capaz de controlar vários parâmetros (temperatura, iluminação, etc.) num contexto de edifícios inteligentes;
- O_5 Implementação do protótipo desenhado no planeamento;
- O_6 Testes da solução e recolha dos respetivos resultados.

1.3 Organização da Dissertação

Esta dissertação é constituída por sete capítulos, cada um estruturado com secções e subsecções. A seguir ao presente capítulo de Introdução, encontra-se o capítulo 2 onde é apresentado o estudo do estado da arte sobre a internet das coisas, sistema multiagente e edifícios inteligentes.

O capítulo 3 apresenta as tecnologias, ferramentas e componentes de *hardware* que irão ser aplicadas na solução definida nesta dissertação.

O capítulo 4 apresenta uma visão do sistema com a apresentação do seu conceito, arquitetura da solução, a sua composição e os requisitos de *software* e *hardware*.

O capítulo 5 apresenta uma descrição do desenvolvimento da solução, nomeadamente no *software*.

O capítulo 6 apresenta as definições dos cenários e dos testes de desempenho e os respetivos resultados.

Por último, o capítulo 7 apresenta as principais conclusões, restrições e sugestões de possíveis melhorias no futuro.

Capítulo 2

Estado da Arte

No presente capítulo é apresentado o estado da arte de temas relevantes para a definição da proposta de solução. A apresentação do estado da arte é dividido por três secções: Internet das Coisas, Sistemas Multiagente e Edifícios Inteligentes.

Na secção de internet das coisas, são apresentados os requisitos e os principais tipos de arquitetura atualmente aplicados em sistemas de Internet das Coisas, e as principais áreas de aplicação do paradigma.

Para endereçar o estado da arte de sistemas multiagente, apresenta-se na primeira subsecção a noção de agente, enquanto unidade essencial constituinte de um SMA, seguindo para a apresentação do conceito de Sistema Multiagente, modos de comunicação, formas de coordenação e negociação bem como tipos de organização de agentes, dentro de um SMA.

No que toca a edifícios inteligentes, apresenta-se a definição e exemplos de aplicação de sistemas de automação e gestão de edifícios, a forma como a internet das coisas cria impacto num sistema de edifícios inteligentes e termina com uma enumeração de aplicações de algoritmos de inteligência artificial nos mesmos.

2.1 Internet das Coisas

Internet das Coisas (IoT) é um conceito que possui diferentes definições na literatura científica, dependendo da sua perspetiva de aplicação. O primeiro uso da designação "Internet das Coisas" foi identificado numa apresentação sobre *Radio Frequency Identification* (RFID), para a empresa Proctor & Gamble [Chin et al., 2019].

Segundo [Lin et al., 2017], a "Internet das Coisas pode ligar dispositivos e instalações omnipresentes com várias redes para fornecer serviços eficientes e seguros para todas as aplicações a qualquer hora e em qualquer lugar". Em [Dorsemaine et al., 2016], o conceito de IoT é descrito como "um grupo de infraestruturas que interconectam objetos e permitem a sua gestão, mineração de dados e o respetivo acesso aos

dados gerados". Existe ainda a definição de internet das coisas de [Whitmore et al., 2015] como um paradigma onde objetos do quotidiano podem ser equipados com capacidades de identificação, sensorização, ligação em rede e de processamento que vão permitir que estes comuniquem entre si e com outros dispositivos e serviços através da Internet para atingir algum objetivo.

Como mostra a Figura 2.2, a IoT pode ter diversas áreas de aplicação, tais como cuidados de saúde, agricultura, gestão de recursos, edifícios inteligentes e transportes inteligentes [Chettri and Bera, 2020]. Em todas estas áreas de aplicação, existem várias tecnologias ou protocolos em comum como Bluetooth Low Energy (BLE), RFID, *Wireless Fidelity* (Wi-Fi) ou *Message Queue Telemetry Transport* (MQTT). Todas estas tecnologias são utilizadas em diferentes níveis de arquitetura de IoT, que por sua vez pode variar, dependendo da aplicação para a qual está implementada.

Esta secção do documento, é dividida por várias secções relacionadas com IoT, a começar por falar sobre vários tipos de arquitetura de IoT retratados na literatura científica, onde são enumerados os respetivos níveis de arquitetura e a função de cada um deles na hierarquia, bem como uma subsecção sobre áreas de aplicação onde IoT pode ser implementada.

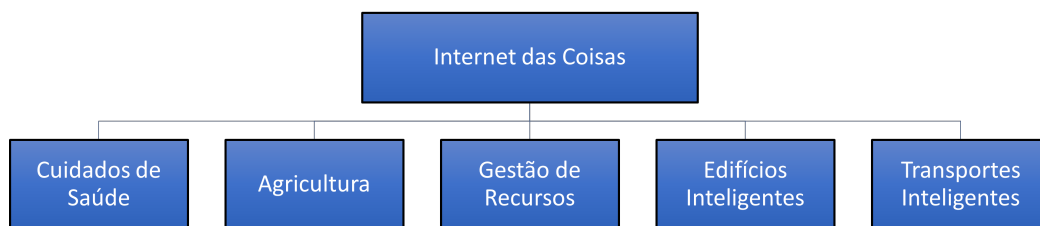


Figura 2.1: Áreas de Aplicação para a Internet das Coisas [Fadda et al., 2019]

2.1.1 Arquiteturas aplicadas no âmbito dos sistemas de Internet das Coisas

Ligar objetos à internet implica a cooperação de várias etapas, que vão desde a aquisição de dados no objeto/dispositivo à aplicação específica de modelos de automação e/ou otimização. Em [Jia et al., 2019], é considerado que, na perspetiva dos utilizadores, um sistema de IoT normalmente consiste em seis componentes essenciais:

- Dispositivos ou Sensores – objetos ou os meios para obter dados dos objetos;
- Redes – infraestrutura de comunicações;

- Computação, armazenamento e processamento – repositório e infraestrutura de processamento de dados;
- Análise – análise computacional ou para algoritmos de mineração de novos dados;
- Atuadores – meios para efetuar manutenção dos dados no controlo dos dispositivos;
- Interfaces de utilizadores – meios de interação do utilizador com o sistema da Internet das Coisas.

O projeto de uma arquitetura de internet das coisas tem de permitir a interconexão e interoperabilidade destas componentes heterogéneas a qualquer hora e em qualquer lugar através da internet, a funcionalidade chave da IoT [Jia et al., 2019]. Um modelo de arquitetura de um sistema IoT é normalmente dividido em camadas, e existem vários modelos de arquitetura propostos por investigadores na literatura científica, de acordo com as suas necessidades. Os modelos mais conhecidos são a arquitetura de três camadas, de quatro camadas ou orientada a serviços (SoA-based - *Service Oriented Approach*) e a arquitetura de cinco camadas.

A arquitetura de três níveis é considerada a arquitetura básica da internet das coisas, utilizada num grande número de sistemas. Esta arquitetura é constituída por três camadas [Jia et al., 2019]:

- Camada de perceção – camada mais inferior que interage com os dispositivos e componentes físicos através de dispositivos inteligentes, tais como os atuadores, sensores e componentes de RFID. O seu principal objetivo é medir, adquirir e processar a informação dos estados das "coisas" através dos dispositivos inteligentes, bem como transmitir a informação processada para camadas superiores.
- Camada de rede – camada intermédia responsável por receber a informação das camadas de perceção de vários dispositivos inteligentes das diversas redes interconectadas, usando *gateways*, e também por reencaminhar a transmissão de dados e informação para dispositivos e aplicações, através das múltiplas redes integradas. Esta camada é considerada a mais importante na arquitetura.

- Camada de aplicação – camada superior com a função de usar os dados recebidos das camadas inferiores, para fornecer serviços requisitados ou operações de armazenamento, processamento e gestão de dados, para análise e produção de resultados de tomada de decisão. Esta camada, para este tipo de arquitetura, tem uma grande variedade de serviços para dispositivos e utilizadores como também serviços de dados.

A simplicidade inerente à arquitetura de três níveis, representada na Figura 2.2, permite a execução de funções e operações complexas em apenas três camadas [Jia et al., 2019]. No entanto, em outros modelos essas mesmas funções e operações podem passar para novas camadas. Por exemplo, serviços de dados que eram prestados nas camadas de rede e de aplicação na arquitetura de três camadas, passaram para a camada de serviços da arquitetura de quatro camadas.

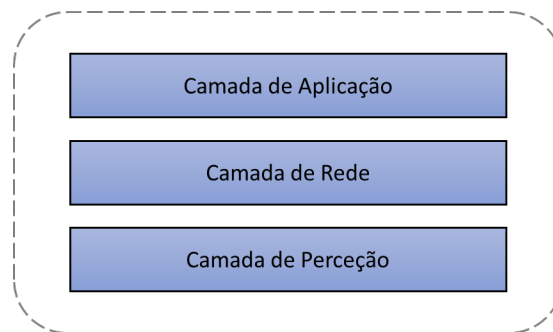


Figura 2.2: Arquitetura de Três Camadas [Omoniwa et al., 2019]

A arquitetura de quatro camadas ou SoA, representada na Figura 2.3, é uma arquitetura que permite às várias partes interessadas construir, instalar e integrar serviços, solicitados por utilizadores, de forma independente dos sistemas de tecnologia em que correm [Omoniwa et al., 2019].

Sendo que as restantes três camadas tem as mesmas funções que a arquitetura de três camadas, a camada de serviços cria, gere e fornece uma variedade de serviços para suportar a camada de aplicação [Lin et al., 2017]. Esta camada pode ser constituída por várias componentes, tais como a descoberta, composição, gestão, e interfaces, de serviços. A SoA é constituída por quatro camadas: Percepção, Rede, Serviços e Aplicações.

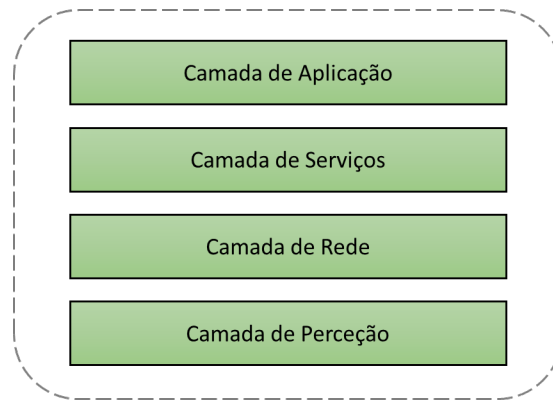


Figura 2.3: Arquitetura de Quatro Camadas [Omoniwa et al., 2019]

A arquitetura de cinco camadas é um modelo de arquitetura ainda mais complexo e com uma perspectiva de negócio, representado na Figura 2.4, na qual é adicionado um novo nível, a camada de negócios [Sethi and Sarangi, 2017]. Esta camada tem como foco fornecer serviços mais complexos, nomeadamente gestão, registo e análise de todas as operações de IoT que ocorreram nos vários sistemas heterogêneos. As análises e registos nesta camada são operações de *big data*, cujo volume de dados estão na escala dos *petabytes* [Omoniwa et al., 2019].

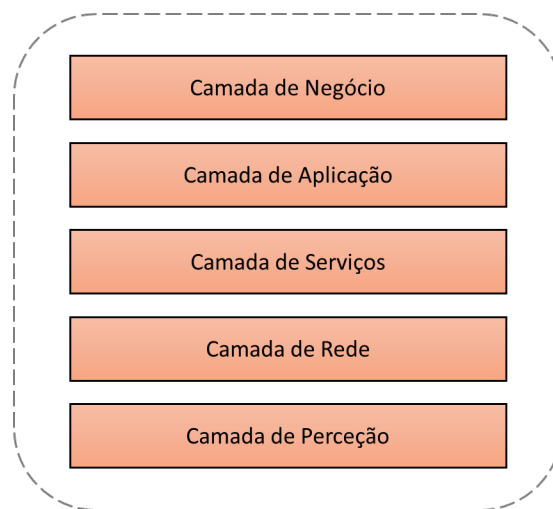


Figura 2.4: Arquitetura de Cinco Camadas [Omoniwa et al., 2019]

Nas arquiteturas descritas nos parágrafos anteriores, e representadas na Figura 2.5, podem ser utilizados os três níveis de computação [Omoniwa et al., 2019]: *cloud*, *fog* e *edge*. Na década de 2010, existiu uma tendência crescente de deslocar o armazenamento, gestão de redes e computação para máquinas remotas (centros de

dados centralizados, redes de espinha dorsal IP e redes centrais), ou a nuvem (*cloud*). No entanto, a tendência de ligar "coisas" à Internet, gerou no meio o trânsito de volumes gigantes de dados e, conseqüentemente, o aumento da latência. As soluções de processamento e computação centralizadas em nuvem para aplicação de sistemas de IoT tornam-se, assim, menos viáveis.

Estima-se que em 2030 se atinja os 24.1 mil milhões de dispositivos de IoT ativos em todo o mundo [TransformaInsights, 2020]. Portanto, a Internet das Coisas requer uma arquitetura robusta e resiliente que possibilite o processamento rápido de dados e o armazenamento. Esses avanços podem ser atingidos com a utilização dos três níveis de computação referidos no parágrafo anterior. Melhora o fornecimento de serviços no caminho entre dispositivos IoT e a nuvem através da aproximação do poder de computação, processamento e armazenamento do local de produção dos dados [Omoniwa et al., 2019]. Nos três parágrafos seguintes, são apresentadas as características, vantagens e desvantagens dos modelos de computação em nuvem, *fog* e *edge*.

A computação em nuvem é um modelo que oferece análise de *big data*¹, tomada de decisão e computação, de forma centralizada, em centros de dados, oferecendo os benefícios de flexibilidade, eficiência e a capacidade para armazenar e usar dados [Omoniwa et al., 2019]. No entanto, existem grandes desvantagens do uso de computação em Nuvem em aplicações de IoT. De acordo com [Chiang and Zhang, 2016], os serviços de nuvem têm uma maior dificuldade em fornecer serviços a aplicações de tempo crítico, as quais precisam de ser atualizadas remotamente ao longo do tempo, e não podem ser deslocadas para a nuvem devido aos atrasos, largura de banda, entre outros.

A comunicação entre dispositivos IoT, a chamada Comunicação Tipo Máquina (MTC - *Machine Type Communication*), aumenta, produzindo uma quantidade massiva de dados a transitar dentro de redes de IoT [Omoniwa et al., 2019]. Nestas condições, a congestão e o tráfego são difíceis de serem geridos dentro da rede, usando o modelo de computação de Nuvem.

¹*Big data* – Refere-se a grandes conjuntos de dados em crescimento que incluem formatos heterogêneos

A computação *fog* é uma plataforma virtualizada que permite estender os benefícios de computação em nuvem para as proximidades das "coisas" que a rede suporta [Bonomi et al., 2012]. A computação *fog* tem como principal foco fornecer dados, armazenamento e serviços para os utilizadores finais na proximidade dos dispositivos de ponta. Este paradigma de computação atua, nomeadamente, sob o nível de rede de área local na movimentação de dados e informação, e utiliza *gateways* de IoT para processar. No contexto de IoT, retiram-se os benefícios de redução de movimentação de dados na rede, da congestão da rede, latência ponta-a-ponta, afunilamento e de aumento da segurança, privacidade e de escalabilidade [Chiang and Zhang, 2016].

A computação *edge*, à semelhança de *fog*, aproxima os meios de computação, processamento e armazenamento à ponta do sistema [Lin et al., 2017]. Este modelo tem como características uma natureza distributiva de computação e uma redução de movimento de dados entre os dispositivos de ponta e a nuvem. Desta forma, existe uma minimização de carga de computação nos centros de dados, situados a uma longa distância na nuvem, uma melhoria da resposta em tempo real e uma redução da latência.

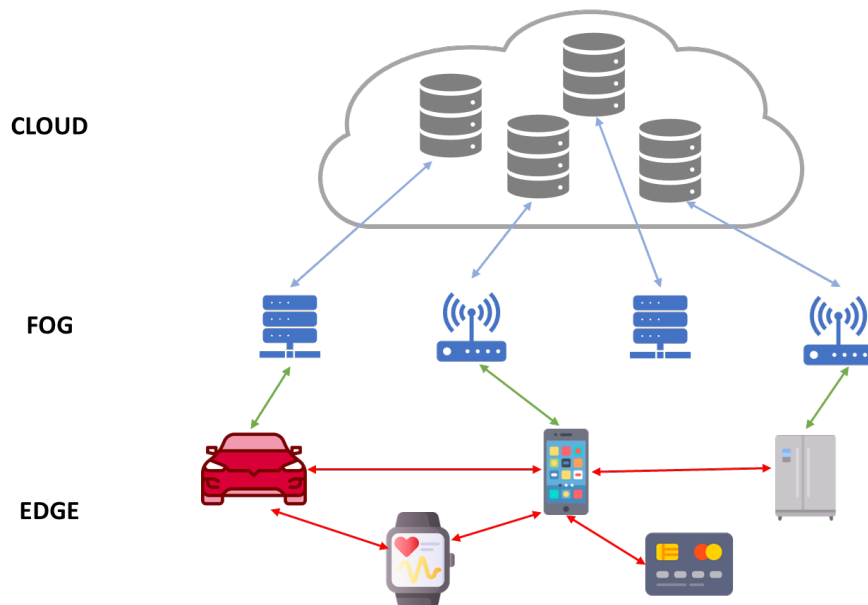


Figura 2.5: Níveis de Computação utilizados na Internet das Coisas [Omoniwa et al., 2019]

Assim, ambos os modelos *fog* e *edge* possuem características comuns [Lin et al., 2017, Fadda et al., 2019, Chiang and Zhang, 2016]:

- Oferta de funcionalidades da nuvem numa localização mais próxima dos utilizadores finais e dos dispositivos;

- Possibilidade de colocação em pontos de acesso, *routers*, estação de base, computadores e *gateways*;
- Instalação sem fios, na maioria das vezes, e fornecimento com baixa latência;
- Fornecimento de serviços de computação em localizações geográficas distribuídas, permitindo uma redução da carga na nuvem;
- Reforça a segurança;
- Melhor capacidade de escalabilidade.

Apesar da capacidade de computação ser trazida para mais perto dos utilizadores finais e dispositivos de ponta nos dois modelos, não quer dizer que ambos tenham o mesmo grau de proximidade no mais baixo nível [Chiang and Zhang, 2016, Fadda et al., 2019]. Neste aspeto de distância, o poder de computação *edge* tem a possibilidade de ser integrado nos dispositivos IoT. O modelo de computação *fog* é residente ao nível de rede local, estando localizado na rede à distância de pelo menos um nível da extremidade do sistema.

Em termos de capacidade de computação, o modelo *edge* é implementado em dispositivos com restrições em energia e armazenamento, sendo, neste aspeto, um modelo mais restrito do que o *fog*, devido à proximidade com os dispositivos de IoT [Omoniwa et al., 2019]. Considera-se que a computação *edge* foca-se mais no domínio das coisas, enquanto que o *fog* se orienta mais para o domínio da infraestrutura. Tendo em conta as semelhanças e diferenças aqui expostas, ambos os modelos de computação *fog* e *edge* podem coexistir dentro da mesma rede. Para cada aplicação é definido onde, como e que tarefas vão para a nuvem.

2.1.2 Aplicações

O IoT é um paradigma com um crescimento muito rápido a nível de quantidade de aplicações e de penetração das indústrias existentes atualmente [Atzori et al., 2010, Lin et al., 2017, Shafique et al., 2020]. Esse crescimento é notado nomeadamente nos domínios de cuidados de saúde, cidades inteligentes, indústria inteligente e edifícios inteligentes.

Em termos de cuidados de saúde, a disciplina *Internet of Medical Things* (IoMT) permite soluções de arquitetura de rede que permitem a conexão entre paciente e estruturas de saúde [Rodrigues et al., 2018]. Nesta área de IoT existem cenários de aplicação tais como monitorização de pacientes a partir de localizações remotas, rastreio de receitas de medicação, sistemas de colocação junto ao corpo para transmitir informação de saúde para os respetivos profissionais de saúde e sistemas de informação [Atzori et al., 2010, Wei et al., 2020].

Para transportes, a utilização de IoT tem como objetivo de tornar os sistemas de transporte mais seguros, fiáveis e eficientes [Atzori et al., 2010]. Os chamados Sistemas de Transportes Inteligentes (STI) são os meios para atingir esses objetivos. Nos STI ocorre uma integração de sistema de gestão de transporte, sistema de controlo e de infraestrutura de comunicações e técnicas de computação. Desta forma, todos os tipos de meios de transporte cobertos por este tipo de sistemas, estarão interconectados entre si. Essa interconexão pode ocorrer entre veículos (V2V) e entre veículos e infraestruturas em localização física (V2I) [Mohanty et al., 2016].

A rede elétrica inteligente emprega serviços e produtos juntamente com o controlo inteligente, comunicação, monitorização e regeneração [Atzori et al., 2010, Morais et al., 2014, Gomes et al., 2020]. Esta área de aplicação faz fornecimento de energia elétrica, bem como a respetiva informação, permitindo aos consumidores que façam parte da otimização da operação do sistema. Existe uma interoperabilidade de geradores de diferentes tipos de tecnologia e tamanhos, e alberga dispositivos de armazenamento e produção intermitente. Permite também operações *'Plug-and-Play'* de microgeradores (carros elétricos), melhorando a flexibilidade da rede elétrica [Fadda et al., 2019, Sousa et al., 2012].

As cidades inteligentes são uma área de aplicação onde se pretende fazer uma melhor utilização dos serviços públicos urbanos, aumentando a qualidade desses serviços para os cidadãos, e por conseguinte a qualidade de vida, enquanto se reduzem os custos operacionais da administração pública [Zanella et al., 2014]. Neste aspeto, o IoT é utilizado para instalar uma infraestrutura de comunicação que fornece acesso a uma diversidade de serviços públicos urbanos de forma integrada, simples e económica, podendo melhorar a cooperação entre serviços públicos e aumentar a transparência para os cidadãos. Neste paradigma, existe uma composição de várias componentes, como a integração de rede de transportes inteligentes, redes elétricas inteligentes, edifícios inteligentes, gestão de tráfego inteligente, indústria inteligente ou cuidados de saúde inteligentes [Silva et al., 2018b].

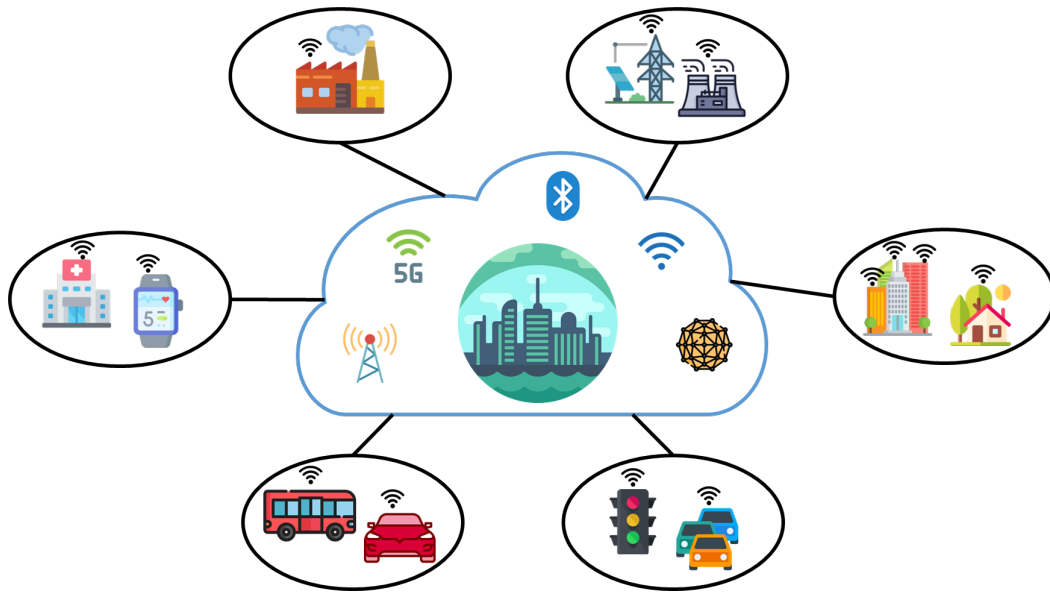


Figura 2.6: Composição de uma Cidade Inteligente [Silva et al., 2018a]

Os parágrafos anteriores apresentaram alguns exemplos de áreas de aplicação de IoT que atualmente estão numa fase de crescimento. A área de edifícios inteligentes é aquela pela qual esta dissertação terá maior foco e, como tal, na secção 2.3 será exposta com maior grau de especificação, bem como exemplos de aplicação.

2.2 Agentes e Sistemas Multiagente

Os Sistemas Multiagente (SMA) são um meio que permitem dividir um problema em múltiplas tarefas mais pequenas, sendo estas atribuídas a entidades autónomas interligadas entre si, designadas de agentes. Os SMA são uma das categorias de algoritmos de Inteligência Artificial Distribuída (IAD), sendo as restantes Inteligência Artificial (IA) paralela e Resolução Distribuída de Problemas (RDP) [Dorri et al., 2018].

Cada agente é autónomo para decidir uma ação apropriada para resolver a tarefa usando múltiplas entradas e o seu respetivo objetivo [Dorri et al., 2018, Cai et al., 2016]. Os agentes têm capacidade de aprender e, com isso, adquirem competências para tomarem decisões autónomas [Gomes et al., 2020]. A possibilidade de interação com os agentes vizinhos ou com o ambiente, permite a aprendizagem de novos contextos e ações. Esses novos conhecimentos ajudam os agentes sobre como tomar decisões e como atuar no ambiente com vista a resolver a respetiva tarefa atribuída ao agente. Para a interação com outros agentes utiliza protocolos baseados em processos de interação social humana, onde se inclui coordenação, cooperação, competição e negociação [Horling and Lesser, 2004, Chira and Dumitrescu, 2006].

Nesta subsecção apresentam-se mais aspetos sobre os agentes dos SMA, a coordenação entre estes, tipos de organização entre agentes e a comunicação entre estes.

2.2.1 Noção de Agente

O conceito de agente é bastante aberto e existe um grande número de aplicações de agentes. Portanto, não existe um grande consenso entre os autores da área sobre o conceito de Agente e, por conseguinte, na literatura científica existem várias definições. Por exemplo, o agente pode ser definido como “uma entidade autónoma e flexível capaz de perceber o ambiente através de sensores conectados a este” em [Russel and Norvig, 2012]. Considera-se também uma “entidade que é colocada num ambiente e deteta diferentes parâmetros que são usados para tomar uma decisão baseada no objetivo da entidade, sendo que esta desempenha as ações necessárias no ambiente baseada nessa decisão”, em [Dorri et al., 2018].

Estes dois exemplos de definições e de outras que se encontram na literatura apresentam termos em comum, nomeadamente “entidade”, “ambiente” e “ação”, como resume a tabela 2.1 [Dorri et al., 2018]. O termo “entidade” é usado para referir o tipo de agente, podendo este ser uma componente de *software* ou de *hardware*.

O termo “ambiente” refere-se ao local onde o agente é instalado podendo este ser um objeto físico no mundo real, um ambiente simulado ou um software [Dorri et al., 2018]. Os agentes utilizam as informações retiradas do ambiente envolvente para a tomada de decisão e, por sua vez, agir sobre ele com atuadores, se necessário.

“Ação” refere-se ao facto de cada agente desempenhar uma ação que desencadeia alterações no ambiente envolvente. Um agente pode desempenhar ações contínuas ou discretas [Dorri et al., 2018]. No caso contínuo, o agente está programado para proceder a um conjunto ilimitado de ações. No caso discreto, o agente só tem de atuar pontualmente, para um conjunto finito de ações.

Tabela 2.1: Componentes necessárias no comportamento de um agente

Termos	Significado
"Entidade"	Tipo de Agente, pode ser uma componente de <i>software</i> e <i>hardware</i>
"Ambiente"	Local onde o Agente é instalado (local físico, ambiente simulado ou um <i>software</i>)
"Ação"	Alterações no meio ambiente desencadeadas com ações dos agentes

O objetivo de cada agente é resolver a tarefa a ele atribuída respeitando possíveis restrições, como prazos [Wooldridge and Jennings, 1995, Dorri et al., 2018]. Na resolução da tarefa existe a detecção e obtenção de informação. Esta informação permite constituir conhecimentos do agente sobre o meio envolvente. O agente também tem a opção de solicitar conhecimentos dos agentes vizinhos. Estes conhecimentos, em conjunto com a aprendizagem do agente com as várias ações passadas e os seus objetivos, são usados para as próximas decisões para ações apropriadas a serem tomadas pelo agente.

Os autores Wooldridge e Jennings desenvolveram uma das abordagens com mais destaque na comunidade científica, na qual o agente é definido como uma peça de *hardware* ou um computador com *software* com as propriedades de Autonomia, Reatividade, Proatividade e Habilidade Social [Wooldridge and Jennings, 1995, Huerta et al., 2020].

- Autonomia – o agente pode executar o processo de tomada de decisão e proceder à ação de forma independente de outros sistemas e de outros seres humanos.
- Reatividade – resposta atempada dos agentes a alterações do ambiente envolvente.
- Proatividade – os agentes agem e têm o comportamento necessário, no sentido de atingir os seus objetivos.
- Habilidade social – através de uma linguagem de comunicação entre agentes, existe a capacidade de interagir com outros agentes e com humanos. Nesta propriedade estão implícitas as capacidades de coordenar e negociar dinamicamente.

Os agentes podem ter outras propriedades mais específicas e com uma visão antropomórfica, isto é, um agente é visto com uma entidade cognitiva e consciente com a capacidade de efetuar processos de interação social e de exibir sentimentos, emoções e perceções [Chira and Dumitrescu, 2006, Reis and Oliveira, 2003]. Trata-se das propriedades pertencentes à noção forte de agente, definida pelos investigadores Wooldridge e Jennings:

- Mobilidade – capacidade de mobilidade de um agente de um lugar para outro (lugar físico ou virtual). Esta propriedade está nomeadamente associada ao contexto de *software* e onde existe movimentação de agentes dentro de uma rede de computadores;

- Veracidade – dever dos agentes em comunicar sempre informação verdadeira e nunca informação falsa;
- Benevolência – os agentes devem sempre tentar fazer as tarefas que lhes são solicitadas e nunca serem contraproducentes, isto é, não terem objetivos em conflito;
- Personalidade – os agentes possuem um carácter e um estado emocional credíveis;
- Intenções e obrigações – as intenções são objetivos de longo prazo. As obrigações referem-se aos compromissos assumidos anteriormente pelo agente;
- Racionalidade - os agentes agem sempre no sentido de atingir os seus objetivos e não devem impedir, de nenhuma maneira, alcançar esses objetivos. A cada momento, o agente deve utilizar os seus conhecimentos e capacidades para executar a melhor ação com vista a atingir os objetivos estabelecidos.

Complementando as propriedades de noção fraca e noção forte de Wooldridge e Jennings, existe outras muito comuns aos agentes [Chira and Dumitrescu, 2006, Reis and Oliveira, 2003]:

- Inteligência – o conhecimento constitui o estado de um agente (objetivos, crenças, planos) e comunica-se com outros agentes usando uma linguagem simbólica;
- Continuidade temporal – os agentes são processos executados continuamente ao longo do tempo, isto é, as ações levadas a cabo pelos agentes estão em execução continuamente;
- Carácter – o agente tem uma personalidade fiável e ainda um "estado emocional";
- Capacidade de aprendizagem - aptidão para adquirir novo conhecimento por parte do agente e alteração do seu comportamento com base na sua experiência prévia.

Utilizando as propriedades enumeradas anteriormente, os agentes funcionam de forma autónoma, nomeadamente na tomada de decisão e na atuação [Reis and Oliveira, 2003]. Por outro lado, todo o benefício do funcionamento de agentes pode ser atingido se estes trabalharem de forma colaborativa ou competitiva. Este trabalho conjunto dá origem aos sistemas multiagente.

2.2.2 Conceito de Sistemas Multiagente

A interação e o trabalho em conjunto entre dois ou mais agentes, de modo a executar tarefas e cumprir um conjunto de objetivos, dá origem a um sistema computacional distribuído designado de sistema multiagente [Chira and Dumitrescu, 2006, Deng et al., 2011].

Para a criação de comunidades de agentes, é necessário haver coordenação, isto é, gerir as interações e as dependências das atividades dos diferentes agentes que constituem o sistema [Reis and Oliveira, 2003]. Esta coordenação é uma característica crítica para este tipo de sistemas, uma vez que os SMA são sistemas distribuídos. Os agentes podem ser coordenados usando dois tipos de metodologias principais: o domínio de competição, no qual agentes estão preocupados com o seu bem próprio, e o domínio de cooperação, onde os agentes usam uma ideia de bem comum.

Num sistema multiagente, os vários agentes existentes interagem e trabalham em conjunto [Reis and Oliveira, 2003]. Na estrutura de SMA, o agente é o elemento com capacidade autónoma de resolver problemas com base na perceção do ambiente circundante e nas informações trocadas na comunicação com os restantes agentes do sistema. Assim, será necessário a existência de uma infraestrutura entre os agentes pertencentes ao sistema, tal como aquela que está ilustrada na Figura 2.7.

Estes agentes do sistema têm diferentes capacidades de perceção e de ação no ambiente em redor [Reis and Oliveira, 2003]. Cada agente terá um determinado alcance, isto é, uma esfera de influência. Estas esferas de influência podem ser comuns a vários agentes, conforme as respetivas tarefas, objetivos e a relação entre os diferentes agentes.

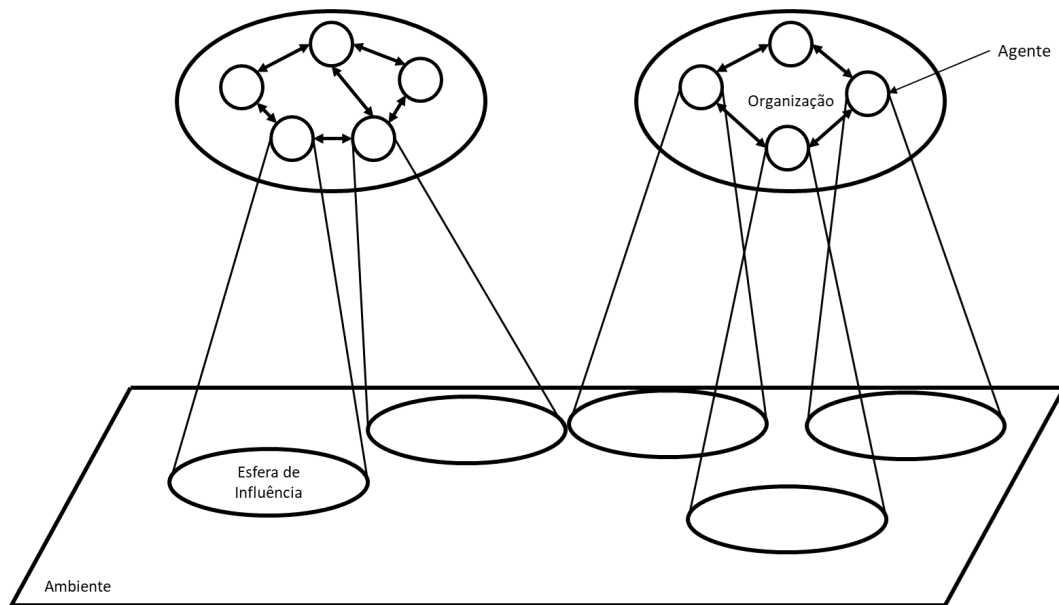


Figura 2.7: Estrutura de um Sistema Multiagente [Reis and Oliveira, 2003]

2.2.3 Comunicação

Juntamente com as capacidades de percepção, processamento e atuação, os agentes devem possuir a importante capacidade de comunicação e habilidade social, isto é, a aptidão de interação com outros agentes e humanos posicionados no ambiente [Chira and Dumitrescu, 2006]. A habilitação de comunicação nos agentes permite a troca de informação e conhecimentos entre si e a requisição de execução de tarefas, que tenham de ser efetuadas fora do alcance do agente no ambiente.

Segundo os investigadores Terán, Aguilar e Mariela, a caracterização das interações entre os agentes, os chamados "atos de fala", permite obter uma abordagem generalizada do ponto de vista da comunicação [Terán et al., 2014]. As conversas, e possivelmente as sub-conversas, podem ser caracterizadas por padrões particulares de interações:

- Atribuição – Um agente atribui tarefas a outro(s) agente(s);
- Consulta – Um agente procura qualquer tipo de informação em locais como bases de dados, repositórios e internet;
- Informação – Um agente informa outros sobre a ocorrência de um determinado evento, dados que podem estar a ser processados ou outro tipo de informação;
- Pedido – O agente remetente solicita ao destinatário para executar alguma ação ou serviço.

Para haver um processo de comunicação eficaz entre os agentes, é necessário, por natureza, uma linguagem de comunicação e um entendimento comum do vocabulário trocado entre os agentes, isto é, uma definição clara dos conceitos incluídos no vocabulário [Chira and Dumitrescu, 2006]. Existem casos de diferentes palavras com o mesmo significado, ou palavras iguais para diferentes significados. Este problema pode ser resolvido com Ontologias.

As ontologias são representações formais explícitas de conceitos partilhados entre duas ou mais entidades que concordaram com a uma ontologia em comum [Reis and Oliveira, 2003]. Por outras palavras, faz a representação de um certo domínio, facilitando a interoperabilidade entre os diversos agentes.

Dentro de uma comunidade de IA, ontologias descrevem entidades e as suas propriedades, relações, restrições e comportamento para que as máquinas tenham a mesma compreensão [Ma et al., 2019]. Em IA, as ontologias têm as seguintes funções:

- Comunicação – as ontologias fornecem glossários para comunicação entre diferentes indivíduos;
- Interoperabilidade – as ontologias têm a capacidade de interpretar e mapear entre vários métodos de modelação, linguagens e ferramentas de *software*.
- Reutilização – as ontologias podem ser reutilizadas, de forma a prevenir análises de conhecimento repetidas. Para isso, as análises das ontologias filtram e refinam a estrutura de conhecimento de uma área que trataram.
- Aquisição e partilha de conhecimentos – as ontologias podem ser utilizadas para supervisionar a aquisição de conhecimento a qual vai ser feita a uma maior velocidade e legibilidade. A partir deste conhecimento, tem a capacidade de gerar novo conhecimento por inferência.

O conhecimento de uma ontologia pode ser definido por humanos ou outras máquinas e depois ser disponibilizado a essa ontologia. Assim, possibilita a sua partilha entre várias máquinas e a sua reutilização [Reis and Oliveira, 2003]. As ontologias são distintas dos esquemas tradicionais de bases de dados nos seguintes pontos de vista:

- São sintática e semanticamente mais ricas do que as bases de dados comuns;
- A informação das ontologias contém texto em linguagem natural semiestruturada;
- Devem ser partilhadas e aceites pela comunidade de agentes que irá utilizá-la;

- Fornecem teoria de domínio.

As ontologias permitem comunicar aos vários agentes presentes num sistema, o vocabulário disponível sobre uma certa área de conhecimento importante para o seu funcionamento [Chira and Dumitrescu, 2006]. Todos os termos desse vocabulário são acompanhados com os respetivos significados no contexto desse sistema.

Os processos de desenvolvimento de ontologias são uma área de estudo relativamente nova, incluindo métodos, ciclos de vida e metodologias para a construção de ontologias, mas ainda não existe processos padrão [Pakdeetrakulwong et al., 2016, Ma et al., 2019]. No entanto, já existem vários modelos de código aberto que podem ser usados ou estendidos.

2.2.4 Coordenação

A coordenação é um conceito que pode estar por trás de uma boa organização de vários tipos de ações do nosso dia-a-dia, tais como o início de consultas médicas à hora marcada, chegada de uma encomenda no dia e na hora previamente delineados ou evento ocorrer como o planeado.

Num contexto dos SMA, a coordenação pode ser definida como "o ato de trabalhar em grupo de forma harmoniosa" [Malone and Crowston, 2003]. À semelhança do conceito de Agente, existe uma maior discordância entre os investigadores sobre como definir coordenação. Neste documento irá ser usada a definição de Jennings, o qual considera a coordenação como o "processo pelo qual um agente raciocina acerca das suas ações locais e das ações (antecipadas) dos outros para tentar assegurar que a comunidade atue de forma coerente"[Jennings, 1996].

A coordenação é o meio usado pelos agentes para definir uma estrutura de organização num grupo de agentes e alocar os recursos e tarefas [Chira and Dumitrescu, 2006]. A comunicação entre agentes é crítica para permitir coordenação de agentes dentro do SMA. O facto de os vários agentes terem capacidades diferentes e limitadas e também diferentes níveis de conhecimentos é considerada outra razão para existência de coordenação. Por fim, a coordenação é uma forma de prevenir caos e anarquia na ocorrência de conflitos entre agentes.

De acordo com os investigadores Lesser e Corkil, os objetivos de coordenação são assegurar que [Durfée et al., 1991]:

- Todas as partes constituintes do problema são resolvidas por pelo menos um agente ou que todas as partes da tarefa cooperativa são executadas;
- Todos os agentes comuniquem com vista a permitir que as suas atividades

sejam realizadas como porção da execução da tarefa cooperativa e assim incluídas na solução global;

- Os agentes de uma equipa agem no sentido de atingir os objetivos globais e de forma consistente;
- A obtenção dos três objetivos dos pontos anteriores dentro da capacidade computacional e dos recursos disponíveis.

A cooperação pode ser um requisito para a existência de coordenação, apesar de isso poder ser atingido por métodos não cooperativos [Reis and Oliveira, 2003]. A competição ou negociação são meios de coordenação não cooperativa entre agentes. Assim, é possível distinguir duas abordagens de SMA: Sistemas formados por agentes cooperativos e Sistemas formados por agentes competitivos.

Nos SMA cooperativos, os agentes comportam-se no sentido de servir globalmente o sistema ao invés de preferir a sua utilidade pessoal [Reis and Oliveira, 2003]. Nestes sistemas existe um maior foco pelo bem comum, dentro do sistema, deixando de lado os próprios interesses.

Nos SMA competitivos, os agentes focam-se nos seus objetivos e tarefas, deixando para segundo plano o interesse pelo bem da comunidade [Reis and Oliveira, 2003]. Estas entidades dão maior prioridade à própria satisfação do que à cooperação com outros agentes com necessidades de assistência. Esta competição ocorre, nomeadamente, entre agentes com conflitos de objetivos e recursos.

2.2.5 Negociação

Um dos ramos de investigação realizada em coordenação de SMA estuda a coordenação de agentes não cooperativos ou egoístas e tem a designação de Negociação [Reis and Oliveira, 2003]. Assim, a negociação é um meio que permite a dois ou mais participantes a aquisição de algo necessário a cada um deles e que o oponente possui, após um período de discussão que poderá permitir, no final, um acordo satisfatório entre as várias partes envolvidas.

As negociações resultam, portanto, da existência de um conflito entre os participantes envolvidos. Desse conflito resulta uma tentativa de resolução através do processo de negociação multiagente [Reis and Oliveira, 2003]. A utilização de negociação, apesar de ser importante, pode não conseguir resolver todos os tipos de problemas, porque este processo não permite a alteração de objetivos e crenças dos agentes envolvidos no conflito.

2.2.6 Organização

A organização de um SMA é um conjunto de papéis, relações e estruturas de autoridade que dirigem o comportamento do sistema [Horling and Lesser, 2004]. As organizações de agentes indicam como os membros da sociedade comunicam entre si, à semelhança de uma organização humana. Estas orientações podem afetar relações de autoridade, fluxo de dados, alocações de recursos, padrões de coordenação ou outro tipo de características do sistema [Reis and Oliveira, 2003].

A orientação da organização permite a grupos de agentes simples apresentarem comportamentos complexos e, em simultâneo, simplificar o raciocínio de agentes mais complexos [Horling and Lesser, 2004]. No conceito de organização em SMA, a forma, o tamanho e as características de estrutura organizacional podem influenciar o comportamento do sistema.

Segundo os autores Horling e Lesser, os agentes podem ser organizados nos seguintes tipos [Horling and Lesser, 2004]:

- Hierarquia – os agentes são estruturados em forma de árvore, como ilustra a Figura 2.8, em que os agentes nas posições superiores têm uma visão global do sistema ao contrário dos que estão nas posições inferiores.

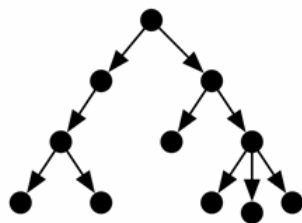


Figura 2.8: Disposição de uma Hierarquia [Horling and Lesser, 2004]

- Holarquia – os agentes são agrupados, e cada agrupamento possui propriedades distintas e derivadas das restantes entidades que são membros do grupo. Em simultâneo, esse grupo em questão oferece as suas propriedades aos grupos em níveis hierárquicos superiores. A Figura 2.9 demonstra uma ilustração de uma Holarquia.

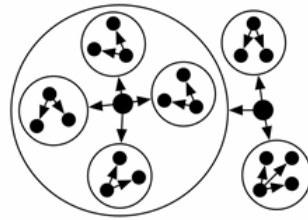


Figura 2.9: Disposição de uma Holarquia [Horling and Lesser, 2004]

- Aliança – grupos de agentes formados com orientação para objetivos de curto prazo, sendo que estes grupos se dissolvem assim que esses objetivos da aliança são cumpridos. Dentro destes grupos provisórios, existe necessidade de coordenação entre os agentes, no sentido de completar as necessidades da aliança. O mesmo não acontece com agentes de outras alianças, exceto se houver alguma relação de dependência entre os grupos, para cumprirem as suas necessidades. A Figura 2.10 ilustra uma organização do tipo Aliança.

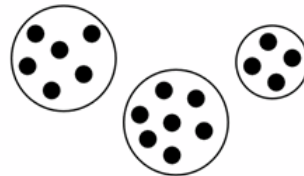


Figura 2.10: Disposição de uma Aliança [Horling and Lesser, 2004]

- Equipe – conjunto de agentes que concordaram em trabalhar juntos no sentido de atingir um objetivo comum. Comparativamente com alianças, as equipas tentam maximizar o potencial da equipa em geral, e não de membros individuais. A Figura 2.11 representa uma possível estrutura do tipo equipa.

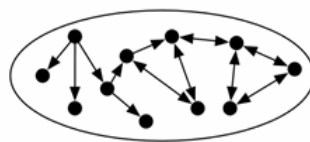


Figura 2.11: Disposição de uma Equipe [Horling and Lesser, 2004]

- Congregação – conjuntos de agentes que se unem numa organização tipicamente plana para obter benefícios adicionais. Os vários grupos de agentes possuem diferentes objetivos e tarefas. Ao contrário das Equipas e das Alianças, este tipo de organização é concebido para longa duração e não para objetivos únicos. A Figura 2.12 mostra uma possível disposição de uma congregação,

em que as diferentes cores representam diferentes propósitos dos grupos de agentes.

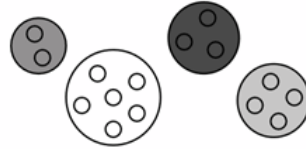


Figura 2.12: Disposição de uma Congregação [Horling and Lesser, 2004]

- Sociedade – os agentes têm diferentes objetivos, vários níveis de racionalidade e capacidade heterogêneas. A sociedade é um paradigma que oferece um domínio comum através do qual os agentes podem agir e comunicar. Tal como uma sociedade humana, esta organização impõe ordem e estrutura, mas as circunstâncias da interação entre agentes são bastante abrangentes. Dentro da sociedade, os agentes podem ser organizados em outros paradigmas ou serem completamente não relacionados. A Figura 2.13 demonstra uma ideia de sociedade.

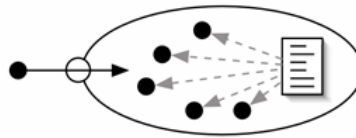


Figura 2.13: Disposição de uma Sociedade [Horling and Lesser, 2004]

- Federação – os agentes são agrupados em federações, onde alguma autonomia é cedida a um único agente, designado de agente delegado, que representa o grupo nas interações entre a Federação e as entidades exteriores. Uma ilustração de Federação está representada na Figura 2.14.

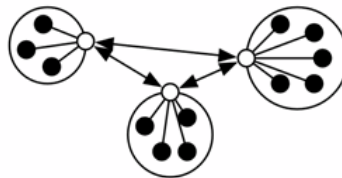


Figura 2.14: Disposição de uma Federação [Horling and Lesser, 2004]

- Mercados – semelhante a uma Federação, e que permite aos agentes solicitar, adquirir ou trocar um conjunto comum de recursos, tarefas, serviços ou bens partilhados, como ilustra a Figura 2.15. Nesta topologia, os agentes podem

disponibilizar itens no mercado para utilização comum. Os vendedores (pontos pretos) ou outros terceiros designados, também chamados de leiloeiros, processam licitações sobre os itens e determinam os vencedores. Esta disposição permite recriar sistemas produtor-consumidor para modelar aproximadamente o funcionamento real de economias de mercado.

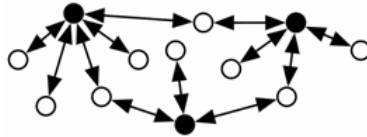


Figura 2.15: Disposição de uma Mercado [Horling and Lesser, 2004]

- Matricial – permite que vários agentes assumam o papel de "gestor", dentro da organização, e assim influenciar as atividades de outros pares. A topologia de matriz introduz, assim, um padrão semelhante ao das relações humanas, nas quais as pessoas também estão sujeitas à orientação e pressões de colegas de trabalho, patrões ou familiares. Como mostra a Figura 2.16, o termo de matriz tem origem na colocação em grelha dos participantes. Nesta disposição, os agentes gestores (a preto) são colocados à volta dos agentes trabalhadores (a branco) e as setas indicam o sentido de autoridade.

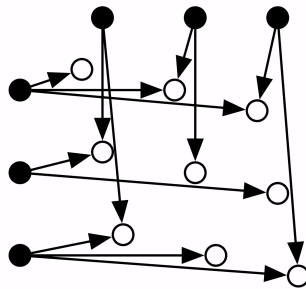


Figura 2.16: Disposição de uma organização em Matriz [Horling and Lesser, 2004]

- Composta – em certas aplicações de SMA, as estruturas organizacionais não se encaixam completamente nas topologias apresentadas nos pontos anteriores, e podem incluir características de várias estruturas. Um sistema pode ser composto por várias topologias, em que cada uma tem uma função específica, como controlo, perceção, fluxo de dados, etc. As organizações compostas podem ser sobrepostas, havendo uma operação no mesmo nível ou podem ser colocadas umas dentro das outras, em que alguns subconjuntos de agentes

estão estruturados de forma diferente do nível superior. A figura 2.17 ilustra um exemplo da disposição dos agentes.

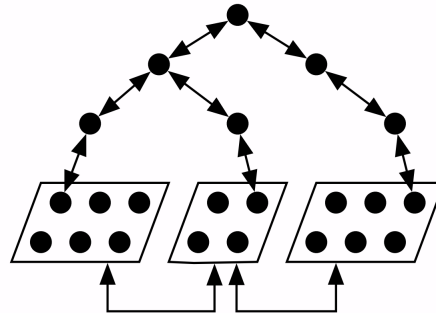


Figura 2.17: Disposição de uma organização Composta [Horling and Lesser, 2004]

Dos tipos de organização enumerados nos pontos anteriores, apenas se destacarão nesta dissertação: as equipes e federações. Uma federação é o tipo de organização que se apresenta sob diversas formas, sendo a única característica comum a todos eles, a existência de um agente a assumir um papel de mediador que representa o grupo, o agente delegado [Horling and Lesser, 2004]. Com este papel, um agente representa o grupo de pares, tendo uma maior autonomia, cedida pelos restantes elementos. Os membros do grupo interagem apenas com o agente delegado, o qual vai intermediar a comunicação entre o grupo e o mundo externo. Normalmente, o agente delegado recebe dos elementos do seu grupo mensagens com descrições de competências e necessidades dos agentes, sendo estes dados usados pelo intermediário para encontrar correspondências com pedidos de agentes delegados de outros grupos.

Uma equipe de agentes consiste num conjunto de agentes cooperativos que concordam em trabalhar em conjunto para atingir um objetivo comum. Cada equipe prioriza maximizar a utilidade da equipe em si, ao invés de maximizar a utilidade individual dos seus membros [Horling and Lesser, 2004]. Para que as ações individuais dos membros da equipe sejam consistentes e sejam solidárias com o esforço de atingir o objetivo comum, é necessário haver a coordenação entre os agentes. Dentro da equipe, cada agente tem um ou vários papéis necessários para tratar das diferentes subtarefas necessárias para atingir o objetivo da equipe. A comunicação entre os membros da equipe não tem nenhuma restrição ou regra específica, podendo ser arbitrária. Esses papéis podem mudar ao longo do tempo para responder em conformidade a eventos planejados e não planejados, enquanto que o objeto global previamente definido mantém-se relativamente consistente. As equipes de agentes distinguem-se das restantes estruturas, através da existência de representações de objetivos partilhados, crenças comuns e planos ao nível da equipe. Com estas

representações, existe flexibilidade e robustez na equipa, ao promover o raciocínio explícito por parte dos agentes sobre os seus comportamentos ao nível da equipa.

2.3 Edifícios Inteligentes

Edifícios Inteligentes (EI) é um conceito que permite aos utilizadores novas possibilidades de conveniência, conforto e eficiência para os seus ocupantes, através de meios tecnológicos [Qolomany et al., 2019, Panchalingam and Chan, 2019]. O desenvolvimento de *Machine Learning* (ML), *Deep Learning* (DL), análise de *Big Data*, tecnologias de sensores e da IoT, tem permitido a transformação de edifícios convencionais em EIs de forma económica e com modificações mínimas de infra-estrutura. A área de edifícios inteligentes abrange escritórios inteligentes, bibliotecas inteligentes ou casas inteligentes, que oferecem serviços automatizados com valor acrescentado, tais como redução de perdas de energia.

O termo edifícios inteligentes surgiu pela primeira vez em 1981, pela United Technology Building Systems (UTBS) [Qolomany et al., 2019]. Desde os anos 80, quando os investigadores começaram a endereçar este termo, não existe um consenso sobre o que é a inteligência de um edifício e o que realça essa inteligência. O termo é ainda bastante abstrato, do qual existem múltiplas interpretações, definições e nomes [Froufe et al., 2020].

A presente secção começa com a apresentação sobre Sistemas de Automação e Gestão de Edifícios (SAGE), sistemas usados como uma componente em edifícios inteligentes, é feita uma revisão sobre aplicações de IoT num ambiente de edifícios inteligentes, e por último, apresenta aplicações de inteligência artificial em edifícios inteligentes.

2.3.1 Sistemas de Automação e Gestão de Edifícios

Os Sistemas de Automação de Edifícios (SAE), também conhecidos como Sistemas de Gestão de Edifícios (SGE), têm como função fornecer controlo e monitorização das condições de ambiente de espaços interiores através de sistemas de aquecimento, ventilação e climatização (AVC), iluminação, segurança contra incêndios, segurança e sistemas de energia [Kim and Ahn, 2013, Pradeep et al., 2019]. As tecnologias de SAE podem ser adotadas em vários tipos de edifícios, como estruturas comerciais e industriais e atualmente em residências domésticas. As estruturas comerciais estão a ter uma maior necessidade de funcionalidades de automação e fluxos integrados de informação em toda a organização, de forma a reduzir os custos de operação. Estas necessidades comerciais geram procura de aplicações de SAE.

Um sistema SAE é geralmente um computador onde está baseado o sistema de controlo, instalado nos edifícios, que controla e monitoriza os sistemas mecânicos e eletrónicos integrados no SAE [Pradeep et al., 2019]. Os sistemas de controlo e gestão de edifícios são mais frequentemente implementados em projetos de grande dimensão, os quais possuem amplos sistemas mecânicos, AVC e elétricos. Os sistemas SAE também podem ser ligados a sistemas de segurança, como controlo de acessos ou sistemas de videovigilância.

2.3.2 Internet das Coisas em Edifícios Inteligentes

Para além das aplicações enumeradas em 2.1.2, a Internet das Coisas tem um impacto significativo em edifícios inteligentes, nomeadamente na construção, operação e gestão de edifícios ao fornecer serviços e funcionalidades eficientes, rumo a objetivos de desenvolvimento [Jia et al., 2019, Sovacool and Furszyfer Del Rio, 2020]. Nesta área de aplicação, pretende-se usar sistemas de IoT como um meio para fornecer serviços fiáveis de eficiência energética, sem comprometer o conforto e nível de satisfação das pessoas naquele contexto controlado pelo sistema.

Uma parte do conceito de edifícios inteligentes abrange o conceito de IoT, isto é, integra uma rede de comunicações que interliga os elementos dos edifícios de forma que estes possam ser manipulados ou monitorizados remotamente [Jia et al., 2019, Sovacool and Furszyfer Del Rio, 2020]. Para isso, certos sensores e atuadores acoplados a esses elementos estão interconectados com sistemas de controlo. Sistemas de IoT podem ser utilizados em edifícios inteligentes para diversos objetivos funcionais, e como mostra a Tabela 2.2, destacam-se os seguintes:

(a) Localização de pessoas e bens

Identifica localizações e movimentos de ocupantes do edifício ou recursos, para melhoria da conveniência de serviços no edifício. A localização interior é uma funcionalidade que adiciona valor ao desempenho do edifício. A localização de ocupantes permite a pessoas não familiarizadas com um edifício o fornecimento de navegação para o seu destino ou para compreender os comportamentos dos ocupantes e prever eventos únicos. A localização de bens pode ser utilizada por gestores para localizar equipamentos ou instalações que precisam de manutenção ou reparações para aumentar a sua produtividade de trabalho [Jia et al., 2019].

(b) Gestão de energia

Otimiza o uso da energia do edifício, idealmente em edifícios com consumo líquido de energia nulo, simultaneamente que mantém um elevado nível de

serviço. A eficiência energética dos edifícios é um dos tópicos mais pesquisados na área de edifícios inteligentes, uma vez que é estimado que os edifícios tenham uma quota de 39% do consumo de energia total no mundo e 40% do total de emissões de dióxido de carbono em áreas urbanas [Farzaneh et al., 2021]. Para manter o nível de serviço para os ocupantes do edifício ao mesmo tempo que o consumo energético é reduzido, são necessárias soluções para satisfazer ambos os requisitos [Jia et al., 2019].

Atualmente, existem no mercado sistemas designados de Sistemas de Gestão de Energia do Edifício (SGEE), para ajudar a controlar, monitorizar e otimizar o uso atual de energia [Airedale, 2022]. Este tipo de sistemas usa medidores não intrusivos em circuitos elétricos para a recolha de dados de uso energético. No entanto, muito do potencial de edifícios inteligentes não é coberto pelos SGEE.

Um sistema de edifícios inteligentes precisa de um nível alto de perceção do contexto. Isto significa que é necessário conhecer os estados do ambiente e dos ocupantes, chaves no modo de operação de edifícios inteligentes. Por exemplo, os sistemas de AVC precisam de saber quantas pessoas estão na sala, para ajustar a temperatura, e os sistemas de iluminação necessitam de monitorizar o nível de iluminação no exterior do edifício e ajustar a iluminação no interior, em conformidade com esses dados. A utilização de IoT permite alcançar esses mecanismos, como foi demonstrado por vários investigadores [Pan et al., 2015, Wei and Li, 2011, Viswanath et al., 2016].

A eficiência energética de edifícios é um tópico com um destaque crescente a nível mundial, tendo em conta as metas globais para travar as alterações climáticas. Vários países criaram objetivos para promover essa redução, como os Estados Unidos, através da sua agência United States Environmental Protection Agency (USEPA), definiram como meta uma redução de 20% de energia consumida em edifícios comerciais, entre 2020 e 2030 [Jia et al., 2019]. Essas metas podem ser atingidas pela indústria com o suporte de IoT, e beneficiar a poupança de recursos e a luta contra as alterações climáticas.

(c) Gestão de instalações

Operação e controlo organizados e manutenção preventiva das instalações e equipamentos do edifício, para reduzir tempo e custos de operações e de manutenção. A gestão de instalações é outro alvo de edifícios inteligentes e faz a integração de atividades de organização para manter serviços eficientes e eficazes.

A gestão de instalações de um edifício requer periodicamente manutenções preventivas e deteção de falhas do equipamento do edifício para garantir o funcionamento pleno das instalações. Em atividades de gestão de instalações tradicionais, havia dificuldades na obtenção de dados de boa qualidade, maior tempo de notificação e atrasos na execução de operações e manutenção [Jia et al., 2019].

A adição de IoT em atividades de gestão de instalações permite um acesso em tempo real e adaptável a instalações do edifício, para o pessoal competente. Neste aspeto de edifícios inteligentes, o aumento de eficiência resulta num potencial de melhoria de saúde e conforto dos ocupantes, aumento de qualidade dos serviços em geral nas instalações, redução dos custos de reparação e uso de energia do edifício [i-SCOOP, 2017]. Na literatura científica, foram descritos vários casos de uso de IoT para gestão de instalações [D’Elia et al., 2010, Srinivasan et al., 2017, i-SCOOP, 2017, Fearn, 2017, Mannino et al., 2021].

(d) Melhoria das condições do interior

Otimiza as condições ambientais da área de intervenção, de acordo com as preferências dos ocupantes do edifício para melhorar a saúde e os custos. Os sistemas de edifícios inteligentes também visam no requisito básico de manter condições ambientais confortáveis para os ocupantes ou utilizadores do edifício. Estima-se que as pessoas passam 80% do seu tempo de vida em edifícios, sendo que a existência de um ambiente interior saudável e confortável é essencial para o bem-estar e produtividade dos ocupantes [Yang and Wang, 2012]. No contexto de garantia de conforto dos ocupantes, é necessária uma robustez na monitorização e controlo do ambiente, realizados em tempo real [Jia et al., 2019].

A preservação do conforto dos ocupantes e a vertente de poupança de energia nos edifícios inteligentes são oponentes, e, assim, é importante serem ambos considerados em conjunto na fase de projeto das soluções inteligentes para o edifício. Com o uso de sistemas de IoT é possível atingir o balanço entre o conforto e a poupança de energia [Jia et al., 2019].

Os domicílios são normalmente o tipo de edifícios nos quais existe um maior interesse no conforto dos ocupantes. Diferentes autores implementaram diferentes abordagens de aplicação de IoT, com o foco no conforto dos ocupantes [Kelly et al., 2013, Bashir and Gill, 2017].

Tabela 2.2: Principais objetivos funcionais de edifícios inteligentes

Objetivos	Descrições
Localização de Pessoas e Bens	Identifica localizações e movimentos de ocupantes do edifício ou recursos, para melhoria da conveniência de serviços no edifício.
Gestão de Energia	Otimiza o uso da energia do edifício, idealmente em edifícios com consumo líquido de energia nulo, simultaneamente que mantém um elevado nível de serviço.
Gestão de Instalações	Operação e controlo organizados e manutenção preventiva das instalações e equipamentos do edifício, para reduzir tempo e custos de operações e de manutenção.
Melhoria das Condições do Interior	Otimiza as condições ambientais da área de intervenção, de acordo com as preferências dos ocupantes do edifício para melhorar a saúde e os custos.

Um edifício inteligente pode incluir muitos outros aspetos para além dos principais referidos anteriormente. Estas funcionalidades inteligentes são implementadas para certos cenários de aplicação ou para algum tipo de edifício específico [Jia et al., 2019]:

- Proteção dos ocupantes e segurança de saúde;
- Gestão eficiente de recursos para conveniência;
- Controlo de saúde do edifício.

Tendo em conta os diferentes casos de uso revistos nos pontos anteriores, a Figura 2.18 enumera as principais aplicações de edifícios inteligentes, bem como os sistemas de edifícios relacionados.

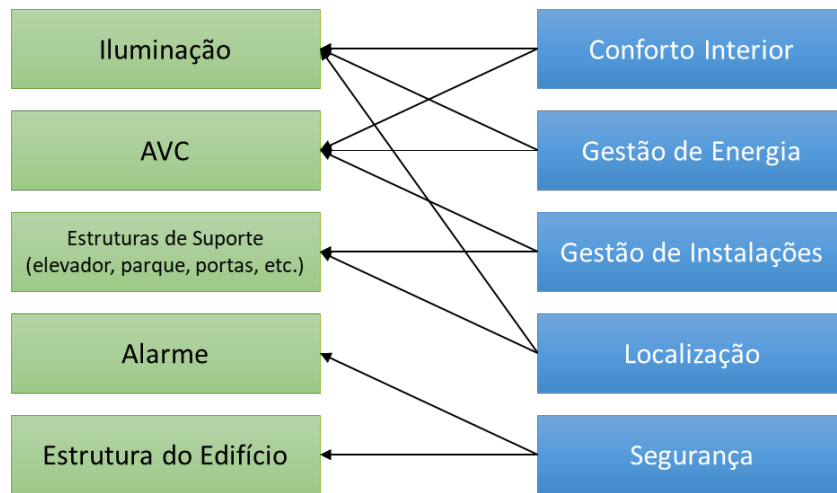


Figura 2.18: Sumário de Aplicações de Edifícios com recurso a IoT [Jia et al., 2019]

2.3.3 Inteligência Artificial em Edifícios Inteligentes

As tecnologias de IA têm cada vez mais uso em edifícios inteligentes, dado que as suas aplicações para o aumento de eficiência da operação e gestão do edifício, resultam em benefícios em larga escala para os ocupantes. Esse aumento de eficiência passa por reduzir os custos de manutenção e de operação, redução do consumo de energia, e, em simultâneo, melhora o conforto e segurança dos ocupantes do edifício.

As tecnologias de IA podem ser utilizadas nas várias fases do ciclo de vida de um edifício, desde logo no seu projeto, permitindo uma melhor qualidade de construção e sistemas de controlo [Panchalingam and Chan, 2019]. Como os edifícios possuem divisões com ambientes e aspetos semelhantes, a automatização destas divisões terá custos e tempo de execução mais reduzidos. A IA também demonstra ser um bom aliado na projeção de características que visam uma melhor propagação de sinal e uma redução de conflitos na interoperabilidade entre dispositivos.

Nos sistemas energéticos dos EIs, a IA utiliza os inúmeros sensores, subsistemas e atuadores, para efetuar monitorização e controlo automatizado com vista a gerar uma poupança de energia [Farzaneh et al., 2021]. Essas reduções de consumo energético são acompanhadas pelas reduções de emissões de gás de efeito de estufa. Diferentes autores demonstram que o potencial impacto para os EIs está nomeadamente nas áreas urbanas, rumo à proteção do ambiente, redução de custos de operação dos edifícios e poupança de energia.

Os SMAs são uma das abordagens de IA para melhorar as capacidades e eficiência dos sistemas de controlo dos EI. Alguns artigos científicos descrevem diferentes casos de aplicação de SMA em edifícios inteligentes que demonstram melhorias na lógica de controlo convencional [Dragomir, 2014, Dounis, 2010, Wang et al., 2010]. No que toca à área de EIs, os SMAs demonstram melhorias significativas no desempenho de lógica de controlo, em relação a sistemas convencionais. Os SMAs visam a gestão de serviços os quais em alguns casos são apenas modelados por equações não-lineares, através de arquiteturas abertas onde os agentes podem ser configurados de forma fácil e dinâmica. Em [Shaikh et al., 2016], é apresentado um sistema de controlo multiagente, o qual foi desenvolvido em combinação com otimização inteligente estocástica. Nos algoritmos de otimização foram usados Algoritmos Genéticos Multi-Objetivo (AGMO) e Algoritmos Genéticos Multi-Objetivo Híbridos (AGMOH). O caso de estudo proposto alcançou melhorias significativas e conseguiu obter um balanço entre consumo de energia e conforto de ambiente interior. Com o uso do AGMOH, atingiu uma melhoria de eficiência energética geral por volta dos 32% em relação a um sistema sem controlo multiagente, e uma melhoria de eficiência em torno dos 13% em relação à implementação com o AGMO. O índice de Conforto aumentou em 72% sobre um sistema normal, e aumentou em 8% no índice sobre um simples AGMO.

As tecnologias *Machine Learning* (ML) e *Deep Learning* (DL) podem ser utilizadas em EIs para abordar os desafios provocados pelo grande volume, alta velocidade e larga variedade de dados gerados pelos dispositivos de IoT, nomeadamente os sensores, câmeras e dispositivos móveis, utilizados para a operação de EIs [Panchalingam and Chan, 2019]. A literatura demonstra que na área de EIs, ML e DL são usados em primeira instância para preservar e prever o consumo de energia dos edifícios e para gerir os recursos do edifício de forma mais eficaz. Os algoritmos de ML e DL demonstram serem eficazes a cumprirem esses objetivos, quando existe uma grande quantidade de dados disponível. A nível geral, verifica-se que os modelos baseados em DL entregam melhores resultados do que os modelos de ML.

Em [Santur et al., 2017], é apresentado um caso com uma abordagem de prospecção de dados, baseada em DL, a partir de grande volumes de dados obtidos através de diferentes sensores de cidades inteligentes, de forma a melhorar os processos de gestão de dados. Os sensores foram usados para recolher dados como humidade, calor, taxa de dióxido de carbono e velocidade do ar. Neste caso de estudo, é feito o armazenamento de dados brutos recebidos de sensores de IoT numa arquitetura descentralizada. Os conjuntos de grandes volumes de dados estão a ser alvo de processos *MapReduce* que obtém um conjunto de dados consistente e bem concebido, para ser usado no processo de treino de modelo. Por último, foi executado um processo baseado em Rede Neural Convolucional. A abordagem apresentada demonstra como grandes conjuntos de dados podem ser usados como mecanismo de tomada de decisão.

Em [Mocanu et al., 2016], é apresentada uma combinação de DL e *Reinforcement Learning* (RL) para montar uma otimização online de agendas para sistemas de gestão de energia do edifício. Para a execução de diferentes ações em simultâneo, foram usados os métodos de aprendizagem *Deep Reinforcement Learning* e *Deep Policy Gradient*. Neste caso, RL foi combinado com *Deep Neural Networks* (DNN) para fornecer uma melhor precisão do que o RL definido pela norma, ao extrair automaticamente padrões, como o consumo de eletricidade. A DNN foi treinado com uma variante do algoritmo de *Deep Q-Learning* e *Deep Policy Gradient*. Este artigo fornece uma comparação entre os dois métodos de aprendizagem, na qual se concluiu que *Deep Policy Gradient* é mais adequado do que *Deep Q-Learning* para desempenhar o agendamento *online* de recursos de energia do edifício.

A IA pode aplicar uma monitorização complexa de vários fatores ambientais em simultâneo, baseada em múltiplos sensores, tendo em conta as preferências dos vários ocupantes [Panchalingam and Chan, 2019]. Este facto torna um sistema de gestão de edifícios mais superior do que um sistema de gestão tradicional, o qual se baseia nas preferências de um único ocupante e não são concebidos para uma natureza dinâmica de um edifício, no qual o número de ocupantes e de divisões muda constantemente.

Capítulo 3

Tecnologias e Ferramentas

No desenvolvimento *full stack* de um sistema de IoT, estão envolvidos múltiplos protocolos e tecnologias. No presente capítulo apresentam-se as principais ferramentas e tecnologias envolvidas na implementação da solução. Nas quatro secções, apresentam-se o modo de operação do Bluetooth, BLE, Bluetooth Mesh e a especificação das componentes de *Hardware*.

A começar pelo Bluetooth, expõem-se as aplicações e a evolução do seu modo de funcionamento, no que toca aos débitos de dados e topologias de rede.

Em relação ao BLE, a secção é composta pela apresentação do conceito do seu modo de operação, a especificação das camadas da arquitetura de pilha de BLE e as respetivas funções e termina com a enumeração das principais áreas de aplicação.

No que toca ao Bluetooth Mesh, a secção começa por dar a conhecer os conceitos e terminologia chave relacionada com a tecnologia, seguindo para a classificação e explanação de uso dos principais tipos de modelos usados na tecnologia, modo de funcionamento da topologia *mesh*, enumeração das principais funcionalidades e a especificação das camadas constituintes da arquitetura da pilha de Bluetooth Mesh e as respetivas funções. Por último apresenta-se o modo de funcionamento das diferentes funcionalidades de segurança da comunicação de Bluetooth Mesh.

O capítulo termina com a especificação das componentes de *hardware* a utilizar na implementação da solução. Nesta secção apresenta-se uma comparação de dois microcontroladores para desempenhar o papel de agente de *hardware* na solução a desenvolver, sendo eles o ESP32 e o Raspberry Pi Zero 2 W. Vários aspetos são comparados, incluindo a Unidade de Processamento Central (CPU), a Memória de Acesso Aleatório (RAM) e a existência de módulos de comunicação Bluetooth, BLE, Bluetooth Mesh e Wi-Fi.

3.1 Bluetooth

Bluetooth é uma tecnologia de comunicação sem fios de curto alcance, desenvolvida pela empresa sueca de telecomunicações Sueca Ericsson, em 1994 [Zeadally et al., 2019]. Foi inicialmente concebida para permitir a comunicação de dados sem fios entre dois dispositivos, sem a necessidade de equipamento de rede intermediário, e que rapidamente adquiriu um papel importante em produtos como ratos sem fios para computador, *kits* de mãos livres para carros ou colunas de som portáteis. O Bluetooth, à semelhança de outras normas de comunicação, opera na gama de frequências aberta Industrial Scientific Medical (ISM) 2.4 GHz. Ao longo dos anos, a tecnologia tem evoluído em diferentes aspetos, nomeadamente no débito de dados, nas funcionalidades e modo de operação.

A primeira versão de Bluetooth, da especificação v1.1, Bluetooth BR (Basic Rate) ou Bluetooth clássico, oferecia um débito de dados em torno de 1 milhão de bits por segundo (1 Mb/s), e permitia apenas comunicação ponto-a-ponto para transmissão de áudio e transferência de dados [Bluetooth Special Interest Group (SIG), 2022b]. Do Bluetooth BR, a norma evoluiu para o Bluetooth Enhanced Data Rate (EDR), o qual oferecia um débito de dados em torno dos 2 Mb/s, mas manteve apenas a topologia ponto-a-ponto.

Na versão 4.0 da especificação Bluetooth Core, é definida uma nova versão da tecnologia Bluetooth, Bluetooth Low Energy (BLE), com o intuito de ser uma alternativa aos seus antecessores, com novas capacidades e qualidades para responder a novos requisitos técnicos e funcionais, tornando-se mais adequado para as novas gerações de produtos [Zeadally et al., 2019].

3.2 Bluetooth Low Energy (BLE)

O BLE é uma versão de Bluetooth concebida para uma operação com maior eficiência energética. Definido pela primeira vez pelo Bluetooth Special Interest Group (SIG) em 2010, o BLE, tal como o Bluetooth clássico, opera na gama de frequências ISM 2.4 GHz [Baert et al., 2018, Bluetooth Special Interest Group (SIG), 2022b]. Mais especificamente, opera entre os 2402 MHz e 2480 MHz. Neste intervalo de frequências, o espetro é dividido em vários canais, nos quais o BLE usa um espaçamento de 2 MHz, resultando em 40 canais diferentes. Destes 40 canais, três deles são os canais principais de divulgação (37, 38 e 39) que evitam os canais principais de Wi-Fi, e os restantes 37 canais são orientados para conexões.

Em termos funcionais, esta tecnologia não tem requisitos de *hardware* e *software* importantes, dado que em grande medida é compatível com o circuito do Bluetooth

clássico [Bluetooth Special Interest Group (SIG), 2020, Lonzetta et al., 2018]. Esse fator ajudou o BLE a ganhar uma posição dominante em *smartphones*, auriculares, colunas de som, impressoras, teclados, automóveis e dispositivos médicos, e por sua vez, permitiu a comunicação de baixo consumo de energia entre si .

3.2.1 Conceito

Todo o conceito do BLE está focado nas interações entre dispositivos móveis, sendo o telemóvel o centro da rede, permitindo a esta tecnologia uma posição predominante nas conexões de IoT [Lonzetta et al., 2018]. No entanto, esta abordagem deixou de parte a capacidade de rede *mesh*, utilizados por concorrentes de redes de baixo consumo energético, como Zigbee e Thread. A capacidade de rede *mesh* utiliza a topologia "muitos para muitos", na qual cada nó da rede possa comunicar com qualquer outro nó, diretamente ou através múltiplos reencaminhamentos [Baert et al., 2018].

O BLE oferece dois modos de comunicação entre dispositivos: modo de divulgação e o modo orientado a conexões [Lonzetta et al., 2018, Baert et al., 2018]. No modo de divulgação um ou mais dispositivos anunciam a informação nos três canais de divulgação com um certo intervalo de tempo (papel de emissor ou anunciante) enquanto outros dispositivos no alcance desses emissores monitorizam nos canais de divulgação, um de cada vez, com um certo intervalo de tempo e apenas durante uma janela de monitorização, para obter essa informação (papel de monitorizador).

O modo orientado à conexão usa os restantes 37 canais, de uma forma mais sincronizada. Um dispositivo assume o papel de mestre e gere as conexões para os vários escravos [Bluetooth Special Interest Group (SIG), 2021]. A gestão e sincronização é feita através de um esquema *Time Division Multiple Access* (TDMA) e cada conexão usa uma técnica de *Frequency Hopping Spread Spectrum* (FHSS) para garantir robustez.

3.2.2 Arquitetura

Como ilustra a Figura 3.1, a pilha de Bluetooth LE é constituída por diferentes camadas e módulos funcionais, os quais alguns são obrigatórios e outros opcionais [Bluetooth Special Interest Group (SIG), 2021]. Estas diferentes partes do sistema são colocadas em dois blocos principais da arquitetura: o anfitrião (*Host*) e o controlador (*Controller*), existindo uma interface lógica padrão que define um meio de comunicação entre as duas componentes, designada de Interface Anfitrião-Controlador (HCI).

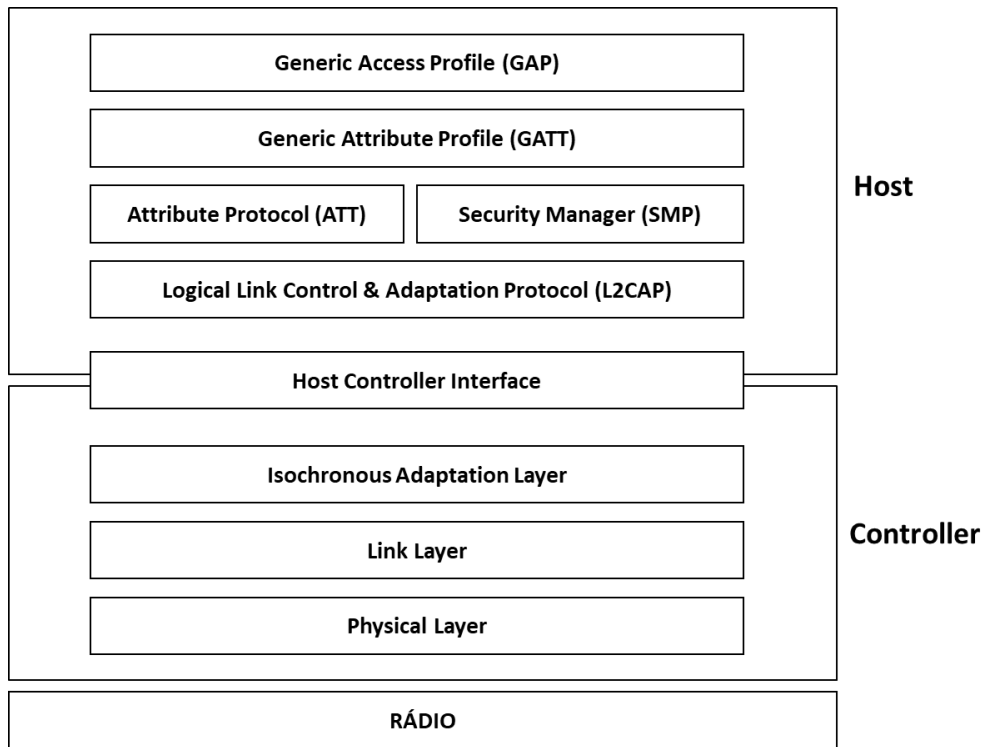


Figura 3.1: Arquitetura de Bluetooth Low Energy [Bluetooth Special Interest Group (SIG), 2022b]

Toda a pilha do BLE expande-se por todas as camadas do modelo de referência Open Systems Interconnection (OSI), ao contrário de outras normas de comunicação sem fios, as quais apenas definem o correspondente a um subconjunto do OSI [Bluetooth Special Interest Group (SIG), 2022b]. Essa característica permite ao BLE estabelecer um sistema de comunicação *full stack* e, assim, ser independente de outras tecnologias externas, para permitir a implementação e a evolução enquanto tecnologia sem quaisquer restrições.

Das camadas referidas na Figura 3.1, destaca-se o nível Perfil de Atributo Genérico (GATT), o qual define os tipos de dados ou atributos de alto nível, conhecidos como Serviços, Características e Descritores [Bluetooth Special Interest Group (SIG), 2021]. Também define um conjunto de procedimentos necessários para o uso destes tipos de dados através da camada de Protocolo de Atributo (ATT).

Os serviços são funções de grupo que fornecem um contexto dentro do qual são utilizadas as características que possuem e as quais têm um tipo de variável definido [Bluetooth Special Interest Group (SIG), 2022b]. Os serviços correspondem a uma funcionalidade ou comportamento do dispositivo.

As características são variáveis de dados de estados individuais, possuindo um certo tipo, um valor associado e um conjunto de propriedades que indicam como os dados podem ser usados em termos dos respetivos procedimentos de GATT, como permissões de escrita e leitura dos nós. Os valores das características podem corresponder, por exemplo, a valores de temperatura do dispositivo par. As características podem ser membros de mais do que um serviço, e tendo por base os respetivos contextos, as regras de utilização podem variar.

Os descritores são um tipo de variável que fornecem metadados sobre as características às quais pertencem ou meios para controlar o comportamento da característica. As características podem ter zero ou mais descritores associados a si, sendo opcionais. Os descritores podem referir-se, por exemplo, aos valores máximo e mínimo de tensão que um sensor suporta. A Figura 3.2 ilustra a estrutura hierárquica dos Servidores, Características e Descritores.

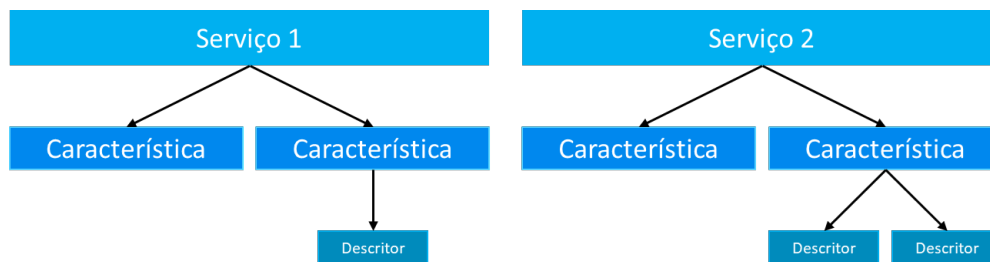


Figura 3.2: Estrutura Hierárquica dos Servidores, Características e Descritores [Bluetooth Special Interest Group (SIG), 2022b]

3.2.3 Áreas de aplicação

Atualmente, a tecnologia BLE fornece suporte *full stack* para as quatro áreas de solução: transmissão de áudio, transferência de dados de baixo consumo de energia, serviços de localização no interior de edifícios e redes de dispositivos de larga escala [Bluetooth Special Interest Group (SIG), 2022a]. Esta última área de solução, também designada de Bluetooth Mesh, é aplicável para a IoT, com características únicas que permitem fazer controlo, monitorização e automação de centenas ou milhares de dispositivos, interligados em rede.

O desenvolvimento do suporte *mesh* para BLE representa uma alteração do paradigma da tecnologia de Bluetooth, que antes suportava apenas a topologia estrela e ponto-a-ponto [Bluetooth Special Interest Group (SIG), 2020]. No Bluetooth Mesh, cada dispositivo está apto para comunicar com todos os outros nós pertencentes à *mesh*. Essa comunicação é alcançada através de mensagens, e cada nó tem capacidade de reencaminhar mensagens para outros nós, permitindo aumentar o alcance de comunicação ponto-a-ponto para além do alcance de rádio de cada um dos dispositivos.

3.3 Bluetooth Mesh

A norma de Bluetooth Mesh é definida como um modelo de *publish/subscribe*, no qual os publicadores podem publicar para um tópico ou grupo específico e os subscritores podem subscrever para um ou para vários tópicos de interesse [Rondon et al., 2020]. Tal como ilustra a Figura 3.3, os interruptores e outros sensores publicam para um certo tópico e os respetivos atuadores (por exemplo, luzes, ventiladores, aquecedores, etc) podem subscrever para um ou vários tópicos.

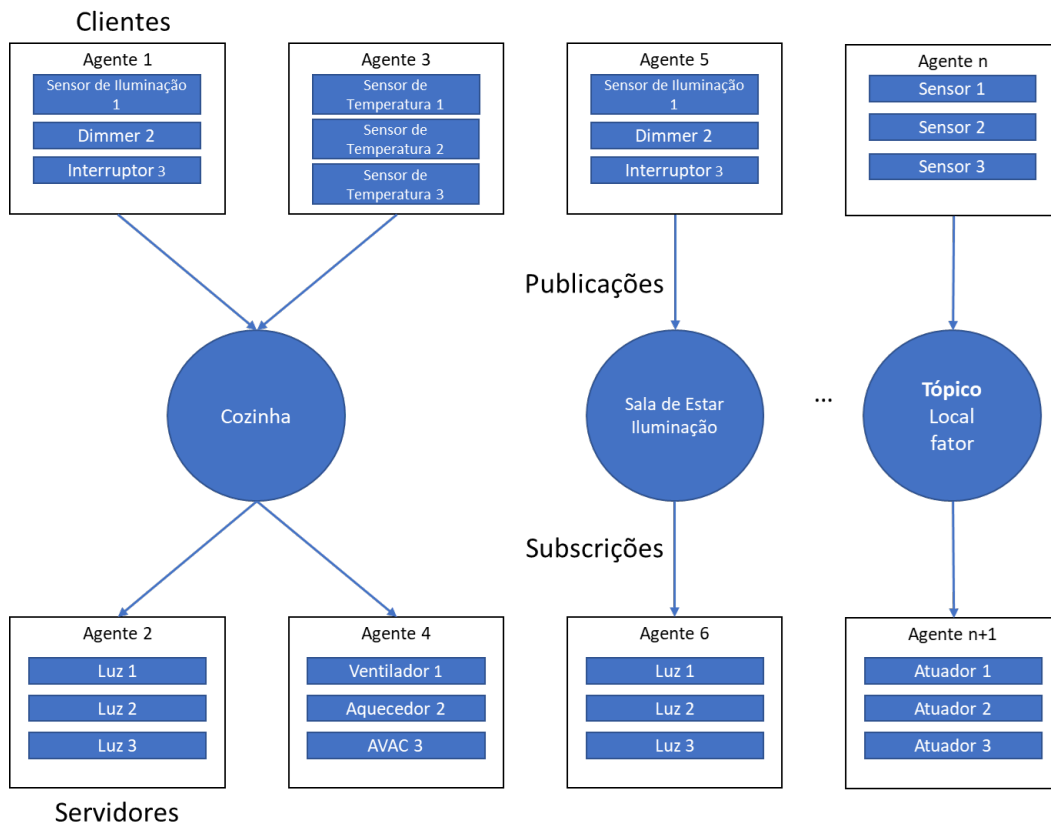


Figura 3.3: Conceito de Publicação/Subscrição [Bluetooth Special Interest Group (SIG), 2020]

Tendo em conta este conceito, um nó da rede de Bluetooth Mesh pode subscrever um ou vários endereços, armazenados na lista de subscritores, e publicar para um endereço específico, armazenado na variável de endereço de publicação [Baert et al., 2018, Bluetooth Special Interest Group (SIG), 2020]. A norma define dois tipos de endereços: *unicast*, endereços de grupo e endereços virtuais. Os endereços de *unicast* são atribuídos a cada um dos nós que integram a rede e identificam unicamente esses nós. O endereço de grupo representa um grupo de nós.

3.3.1 Conceitos e terminologia

Os dispositivos pertencentes à rede *mesh* designam-se de nós e aqueles que não pertencem designam-se de dispositivos *unprovisioned* [Hortelano et al., 2021]. O processo para transformar esses dispositivos em nós designa-se de *provisioning*. O *provisioning* é o procedimento de segurança que resulta nos dispositivos possuírem uma série de chaves de encriptação e serem apresentados ao dispositivo *provisioner*, normalmente um *tablet* ou um *smartphone*. Uma dessas chaves designa-se de chave de rede (NetKey).

Todos os nós da rede *mesh* devem possuir pelo menos uma chave de rede, um parâmetro de segurança designado de Índice IV e um endereço *unicast*, e a posse desses dados é o requisito essencial que torna aquele dispositivo num membro da rede, um nó [Hortelano et al., 2021].

Os nós podem ter várias partes constituintes, nas quais cada uma pode ser controlada de forma independente [Bluetooth Special Interest Group (SIG), 2020]. No contexto de Bluetooth Mesh, essas partes são chamadas elementos. Por exemplo, se o nó estiver ligado a três lâmpadas LED, é possível ligar e desligar as lâmpadas individualmente.

Para um nó determinar ou controlar o estado de outros nós, é necessário o envio de uma mensagem do tipo adequado. Se um nó necessita de reportar o seu estado para outros, também envia uma mensagem [Rondon et al., 2020]. Toda a comunicação dos nós da rede é orientada a mensagens e vários tipos de mensagens são definidos, aos quais são atribuídos números de identificação únicos, os códigos de operação.

As mensagens são divididas em duas categorias: mensagens confirmadas e mensagens não confirmadas [Bluetooth Special Interest Group (SIG), 2020]. As mensagens confirmadas requerem uma resposta dos nós que a recebem. A mensagem de resposta contém a confirmação propriamente dita da receção da mensagem e devolve dados relacionados com o recetor da mensagem ao respetivo remetente.

Caso não receba a mensagem de resposta esperada, o remetente pode reenviar a mensagem [Bluetooth Special Interest Group (SIG), 2020]. Nestes casos, as mensagens confirmadas devem ser idempotentes, ou seja, o efeito da recepção de uma dada mensagem múltiplas vezes, deve ser o mesmo que a recepção da mensagem uma única vez. As mensagens sem confirmação não requerem resposta.

O envio e a recepção de uma mensagem requer a existência dos endereços do nó remetente e do nó destinatário. Dentro de uma rede de Bluetooth Mesh, existem três tipos de endereços [Bluetooth Special Interest Group (SIG), 2020, Rondon et al., 2020]:

- Endereço de *unicast* – após a atribuição aos dispositivos durante o processo de *provisioning*, o endereço identifica um único elemento;
- Endereço de grupo – endereço *multicast* que representam um ou mais elementos. Podem ser definidos pelo Bluetooth SIG e são conhecidos como Endereços de Grupo Fixos do SIG ou dinamicamente atribuídos. Existem quatro endereços de grupo fixos definidos: um para todos os *proxies*, todos os amigos, todos os retransmissores e para todos os nós. Os endereços dinâmicos vão ser estabelecidos pelo utilizador através de uma aplicação de configuração.
- Endereço virtual – endereço que pode ser atribuído a um ou a vários elementos, abrangendo um ou vários nós. Pode ser pré-configurado durante o fabrico do dispositivo e ser usado para cenários.

As mensagens são geralmente endereçadas para endereços de grupo ou virtuais [Bluetooth Special Interest Group (SIG), 2020]. Os nomes dos endereços virtuais ou de grupo terão um significado prontamente entendido pelo utilizador final, tornando-os fáceis e intuitivos na respetiva utilização.

A utilização de endereços de grupo ou virtuais no modelo de comunicação publicar/subscrever tem benefícios extra os quais na remoção, substituição ou adição de novos nós não requerem a reconfiguração de outros nós [Bluetooth Special Interest Group (SIG), 2020]. Assim, ao adicionar um novo nó à rede, através do processo de *provision* e estiver configurado para subscrever o respetivo endereço de grupo, os restantes nós da rede e do grupo não serão afetados por esta alteração na rede, e o mesmo acontece com a sua remoção.

Os elementos da rede de Bluetooth podem estar em várias condições, as quais são representadas através de valores de estado [Bluetooth Special Interest Group (SIG), 2020]. Neste caso, um estado é um valor de um certo tipo associado a um elemento. Os estados têm comportamentos intrínsecos e não podem ser utilizados noutros

contextos. Como exemplo, num nó de Bluetooth Mesh uma luz que pode ter os valores de estado ligado e desligado, sendo esses definidos num estado designado de *Generic OnOff*. Assim, este estado irá ter o valor "ligado" para refletir que a luz está de facto ligada, e o valor "desligado" quando a luz não está a iluminar.

As operações na rede *mesh* são invocadas através de mensagens [Rondon et al., 2020]. Um certo tipo de mensagem representa uma operação num estado ou num conjunto de estados. Todas as mensagens têm um dos três tipos que refletem os tipos de operação que o Bluetooth Mesh suporta [Bluetooth Special Interest Group (SIG), 2019]:

- Get – pedidos de valores de um certo estado de um ou vários nós.
- Set – pedidos de alteração de valores de um dado estado.
- Status – mensagens enviadas em resposta a mensagens *get*, *set* com confirmação ou de forma independente de outras mensagens, se, por exemplo, existir uma configuração de envio destas mensagens de forma periódica, com base num valor de intervalo de tempo definido.

Os estados específicos das operações referidos por mensagens são inferidos através do *opcode* da mensagem [Bluetooth Special Interest Group (SIG), 2019].

3.3.2 Modelos de um nó de rede

Os modelos juntam os conceitos referidos na subsecção 3.3.1 - mensagens, estados e propriedades - e definem as funcionalidades de um elemento dentro da rede *mesh*. Existem três categorias base de modelos [Bluetooth Special Interest Group (SIG), 2020]:

- Modelo servidor – define um conjunto de estados, transições de estado, ligação de estados e mensagens que o respetivo elemento pode enviar e receber. Também define comportamentos para mensagens, estados e transições de estados.
- Modelo cliente – define as mensagens que pode enviar ou receber, ler, alterar e obter valores dos estados definidos no respetivo modelo servidor.
- Modelo de controlo – constituído por um modelo servidor, para permitir a comunicação com outros modelos cliente, e um modelo cliente, para a comunicação com outros modelos servidor.

Alguns modelos podem ser estendidos, criando modelos personalizados. No entanto,

noutros tal não é possível, sendo designados de modelos raiz [Bluetooth Special Interest Group (SIG), 2020]. O Bluetooth SIG criou um conjunto de estados genéricos que podem ser utilizados por modelos em geral, para aplicações genéricas como ligar e desligar (*on/off*) e ajuste de nível de atuadores (luzes, ventoinhas, etc).

Para esse conjunto de estados genéricos, o Bluetooth SIG definiu igualmente um conjunto de mensagens, para a comunicação das operações desses estados genéricos [Bluetooth Special Interest Group (SIG), 2020]. Estas mensagens e estados genéricos podem ser utilizados por modelos cliente, como o *Generic OnOff Client* e modelos servidor, como o *Generic OnOff Server*.

Os modelos genéricos permitem uma definição e implementação rápidas de comunicação de estados genéricos, dado que suportam a ligação de um vasto número de tipos de dispositivos a uma rede de Bluetooth Mesh, sem a necessidade de criação de novos modelos [Bluetooth Special Interest Group (SIG), 2020]. No entanto, os modelos genéricos podem ser utilizados como base para a criação de novos modelos, através da extensão destes.

A criação de modelos personalizados pode ser feita através da criação de raiz da própria infraestrutura do modelo, nomeadamente a definição dos seus atributos ou estados, operações e estruturas de mensagens. Por outro lado, pode utilizar a infraestrutura do modelos genéricos como base, e efetuar as alterações nas respetivas propriedades [Bluetooth Special Interest Group (SIG), 2020].

3.3.3 Topologia *mesh*

No sentido de interligar esses diferentes nós publicadores e subscritores, é necessária a criação de uma rede com topologia *mesh*, como ilustra a Figura 3.4 [Rondon et al., 2020]. A norma utiliza o bloco de controlador da pilha de BLE (camada de ligação e camada física), enquanto que o bloco de anfitrião é substituído por camadas alternativas, para implementar os protocolos e procedimentos de Bluetooth Mesh. Adicionalmente, as técnicas de divulgação e análise do BLE são também utilizadas como tecnologia de base para implementar a comunicação [Bluetooth Special Interest Group (SIG), 2022b].

Numa rede de Bluetooth Mesh, a comunicação utiliza um mecanismo de *flooding*, o qual assegura que cada nó da rede repete mensagens recebidas, e, se estiver ativada, permite a retransmissão até ao nó de destino [Hortelano et al., 2021]. Ao contrário do funcionamento do mecanismo de divulgação da comunicação normal de BLE, os nós de Bluetooth Mesh não enviam os seus pacotes de acordo com intervalos de tempo estabelecidos para a divulgação, mas sim com valores de intervalo gerados aleatoriamente.

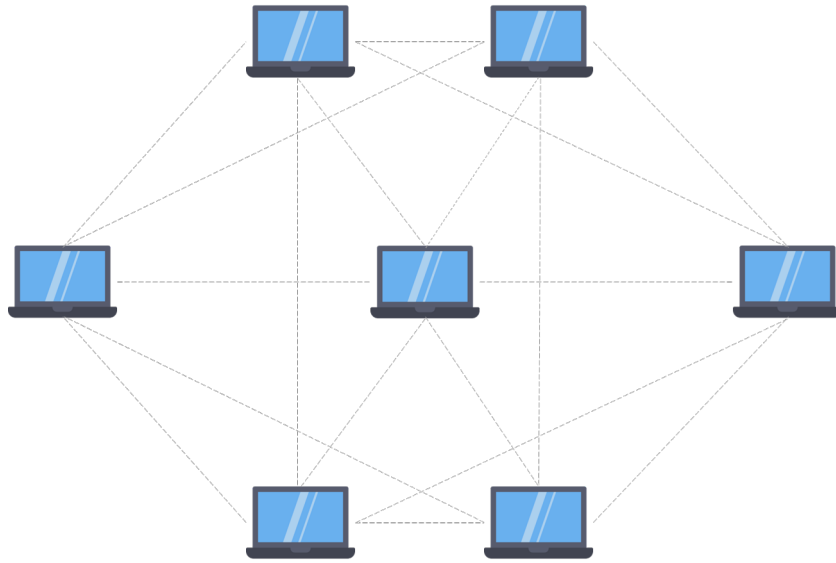


Figura 3.4: Rede com topologia *mesh* [Rondon et al., 2020]

Na monitorização dos canais de divulgação para pacotes recebidos, os nós da rede *mesh* usam o valor de *duty cycle* a 100% [Baert et al., 2018]. Isto significa que os nós da rede estão sempre a monitorizar, exceto quando estão a enviar pacotes. Para a monitorização a janela e o intervalo ainda são utilizados. A janela de monitorização é igual ao intervalo de monitorização, para garantir a continuidade da monitorização. O intervalo de monitorização garante que o nó muda entre canais de divulgação para monitorizar.

3.3.4 Funcionalidades

Além da capacidade de transmissão e receção de mensagens dentro da rede *mesh*, um nó pode implementar algumas funcionalidades opcionais, com o fim de gerir e melhorar a comunicação. Assim, pelo menos uma destas funcionalidades opcionais pode ser ativada a qualquer momento: retransmissão, representação/*proxy*, baixa energia e amizade [Rondon et al., 2020].

Funcionalidade de retransmissão

A utilização de nós com a funcionalidade de retransmissão serve para retransmitir mensagens recebidas, ou seja, a possibilidade de uma mensagem atravessar toda a rede *mesh*, saltando de nó a nó até atingir o seu destino final, como ilustra a Figura 3.5 [Baert et al., 2018, Rondon et al., 2020].

A norma também introduz uma cache de mensagens para assegurar que um nó de retransmissão procede com o reencaminhamento de uma mensagem específica apenas uma vez, bem como a definição de um campo da estrutura da mensagem que estabelece o tempo de vida da mensagem (TTL - *Time-to-Live*) [Bluetooth Special Interest Group (SIG), 2019].

Neste parâmetro da mensagem é colocado um valor inteiro, o qual vai ser usado para limitar o número de saltos que uma mensagem pode fazer ao longo da rede [Baert et al., 2018]. Por exemplo, se o TTL for definido a 4, a mensagem irá ser retransmitida no máximo de quatro saltos a partir do nó remetente. A retransmissão de mensagens só ocorre quando estas não estão na cache e o respetivo valor de TTL é maior que 1.

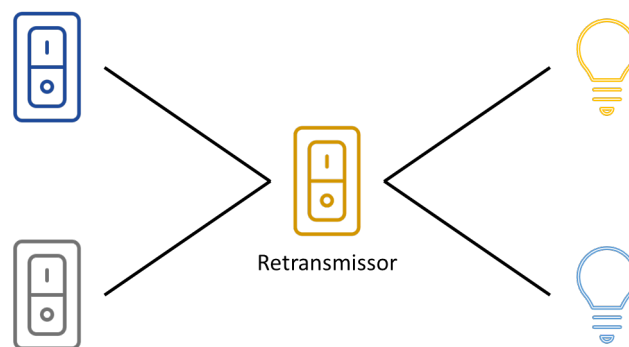


Figura 3.5: Exemplo da funcionalidade de Retransmissão [Bluetooth Special Interest Group (SIG), 2020]

Funcionalidade de proxy

Atualmente existem inúmeros dispositivos no mundo que suportam BLE, nomeadamente *smartphones* e *tablets*, mas, como o Bluetooth Mesh foi adotado posteriormente à entrada no mercado, estes dispositivos não possuem a pilha de Bluetooth Mesh [Bluetooth Special Interest Group (SIG), 2020].

A norma define uma funcionalidade de retrocompatibilidade para dispositivos BLE que não suportam Bluetooth Mesh. É um meio que permite a um dispositivo BLE nativo, como um *smartphone*, conectar-se a uma rede de Bluetooth Mesh [Baert et al., 2018].

Através da pilha de BLE, os dispositivos têm a capacidade de conexão com outros pares através das interfaces de GATT e de divulgação. Os nós *proxy* expõem a interface GATT para permitir a interação de dispositivos com BLE com a rede Mesh. Essa exposição é feita através do protocolo *proxy* [Baert et al., 2018, Rondon et al., 2020].

Os dispositivos de GATT lêem e escrevem os pacotes de dados com as características de GATT implementadas pelo nó *proxy*. Este transforma esses pacotes de dados para pacotes de dados compatíveis com a rede *mesh* [Bluetooth Special Interest Group (SIG), 2019].

Como demonstra a figura 3.6, os dispositivos não compatíveis ligam-se a um intermediário, um nó da rede com a funcionalidade de *proxy*, ou seja, esta ligação usa a funcionalidade de retro compatibilidade, através de ligações de BLE. O nó intermediário, por conseguinte, usa os canais de divulgação, para reencaminhar a mensagem para outro nó. Um nó *proxy* pode ser considerado também como um nó de retransmissão [Baert et al., 2018].

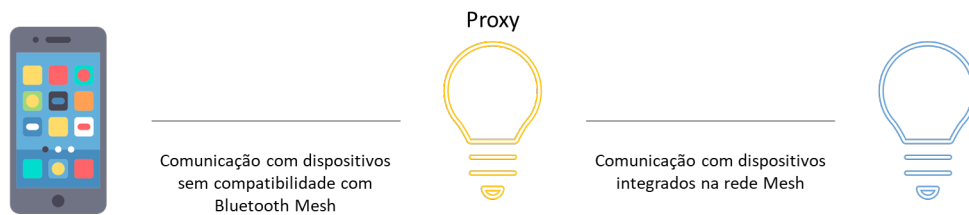


Figura 3.6: Exemplo da funcionalidade de Proxy [Bluetooth Special Interest Group (SIG), 2020]

Funcionalidade de amizade

Por norma, um nó de BLE utiliza um mecanismo de *flooding*, definido pela especificação, no qual utilizam um valor de *duty cycle* de 100% para monitorizar os diferentes canais de divulgação. Esta condição limita a utilidade de aplicação de dispositivos com baixo consumo energético em sistemas com grandes limitações energéticas [Bluetooth Special Interest Group (SIG), 2020].

A limitação de consumo energético é mitigada com a funcionalidade de amizade introduzida na especificação de Bluetooth Mesh. Certos dispositivos necessários numa rede *mesh*, não são aptos para monitorizar com um *duty cycle* de 100%, devido a limitações energéticas [Bluetooth Special Interest Group (SIG), 2019].

Como ilustra o exemplo da Figura 3.7, um nó da rede com maior capacidade de consumo energético, como lâmpadas de iluminação, pode apoiar nós de baixo consumo, para a existência de condições na integração dos mesmos na rede *mesh* [Baert et al., 2018]. Ambos os nós podem estabelecer uma relação de amizade, constituída pelas duas componentes da funcionalidade: os nós de baixo consumo energético e os nós amigos. Neste contexto, um dispositivo com consumo limitado de energia age como um nó de baixo consumo e um nó, como uma lâmpada de iluminação, age como um nó amigo.

Com o papel de nó amigo existem as responsabilidades de armazenamento de mensagens recebidas do nó de baixo consumo energético e a respetiva retransmissão para esse nó, dentro da rede [Bluetooth Special Interest Group (SIG), 2019]. Essas responsabilidades requerem que o nó amigo tenha a funcionalidade de retransmissão. A retransmissão de mensagens para o nó de baixo consumo energético deve ocorrer depois de enviar ao seu nó amigo um pedido periódico de novas mensagens. O envio de pedidos periódicos e a respetiva receção das mensagens retransmitidas não necessita de 100% de *duty cycle*, diminuindo a exigência de consumo energético.



Figura 3.7: Exemplo da funcionalidade de Amizade [Bluetooth Special Interest Group (SIG), 2020]

3.3.5 Arquitetura

Um dispositivo que está numa rede de Bluetooth Mesh, necessita de implementar a pilha de Bluetooth Mesh representada na Figura 3.8. Trata-se de uma arquitetura por camadas, sendo constituída por [Bluetooth Special Interest Group (SIG), 2019, Rondon et al., 2020]:

- Camada de Modelos – Responsável pela implementação de Modelos, isto é, a implementação de propriedades como comportamentos, mensagens, estados ou troca de estados, tal como são definidos nas especificações de um ou vários modelos.

- Camada de Modelos Fundamentais – Encarregues pela implementação de modelos que tratam da configuração e da gestão da rede *mesh*.
- Camada de Acesso – Faz interface entre as camadas mais orientadas para aplicação (aplicação, modelos e modelos Fundamentais) e as camadas mais baixas, isto é, define vários aspetos da ligação entre as aplicações e a camada superior de transporte, tais como o formato dos dados das aplicações e o processo de encriptação e desencriptação (executados na camada superior de transporte).
- Camada Superior de Transporte – Trata da encriptação, desencriptação e autenticação dos dados de aplicação que estão a passar de e para a camada de acesso. Também é competente pelas mensagens de controlo de transporte, geradas e enviadas entre camadas superiores de transporte dos diferentes nós da rede. Tratam-se das mensagens de amizade e *heartbeats*.
- Camada Inferior de Transporte – Leva as mensagens (Unidades de Dados de Protocolo - PDU) da camada superior de transporte para a camada inferior de transporte do nó de destinatário. Se a mensagem/PDU for maior que o limite máximo de um pacote de transporte, executa a fragmentação e a separação em múltiplos pacotes de transporte. A camada inferior de transporte do outro dispositivo, o recetor da mensagem, vai remontar todos os segmentos numa única mensagem e é transferida para as camadas superiores.
- Camada de Rede – Define os vários tipos de endereço de mensagem e o formato da mensagem de rede que permite mensagens/PDUs da camada de transporte serem transportadas para a camada de portador. Suporta múltiplos portadores, nos quais cada um pode suportar múltiplas interfaces de rede, entre as quais a interface local que é usada para a comunicação entre os múltiplos elementos que pertencem ao mesmo nó. Esta camada é também responsável pela determinação das interfaces de rede que devem ser usadas para entregar as mensagens. As funcionalidades de retransmissão e *proxy* são implementadas nesta camada.
- Camada de Portador – Define como as mensagens/PDUs da rede mesh vão ser tratadas num dado sistema de comunicações. Trata-se de uma abstração da especificação de BLE subjacente para as camadas superiores. Essas abstrações representam-se sob a forma de portadores. Até ao momento, foram definidos dois portadores e esses designam-se de portador de divulgação e portador de GATT. O portador de divulgação potencia as funcionalidades de divulgação e rastreio do BLE para a transmissão e receção de mensagens/PDUs. Os

portadores de GATT permitem que um dispositivo sem compatibilidade com um portador de divulgação, possa comunicar indiretamente com os nós da rede *mesh*, através de um protocolo conhecido como o protocolo de *proxy*.

- Especificação do Núcleo de Bluetooth Low Energy – A norma de Bluetooth Mesh é construído sobre a especificação de BLE e usa os mecanismos orientados a divulgação e conexão.

Uma aplicação é implementada sobre a pilha de Bluetooth Mesh [Baert et al., 2018]. A norma também define medidas de segurança, nomeadamente na segurança das camadas de aplicação e de rede, atualização de chaves, etc. e mecanismos de *provisioning* e configuração de um dispositivo para poder fazer parte da rede *mesh*.

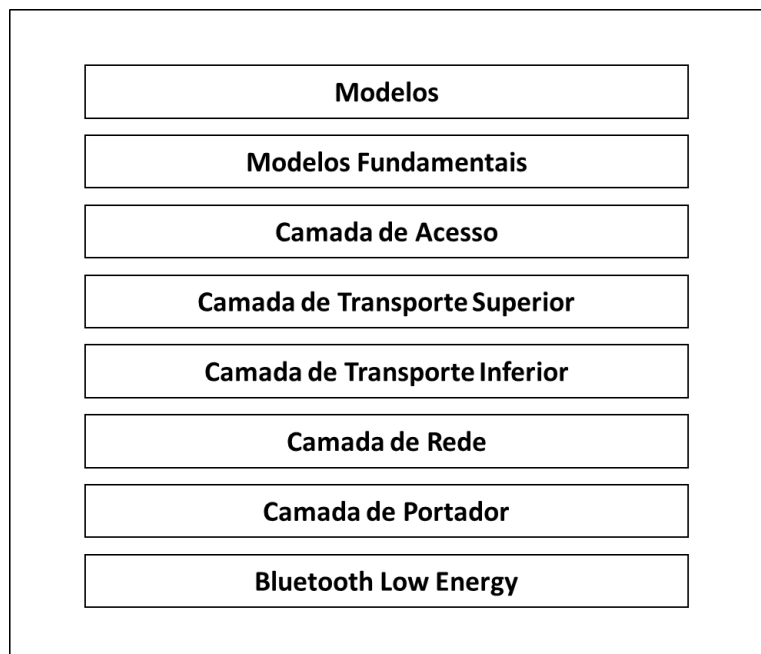


Figura 3.8: Arquitetura do Bluetooth Mesh [Bluetooth Special Interest Group (SIG), 2020]

3.3.6 Segurança

De acordo com as referências [Bluetooth Special Interest Group (SIG), 2020, Bluetooth Special Interest Group (SIG), 2019], a norma de BLE oferece diferentes mecanismos de segurança, desde várias abordagens de emparelhamento até requisitos de segurança específicos para certas características. O *designer* do dispositivo ou fabricante é responsável por analisar as ameaças e determinar os requisitos de segurança e soluções para os seus produtos. No contexto de Bluetooth Mesh, a implementação dos seus mecanismos de segurança é obrigatória, isto é, as aplicações

individuais e dispositivos devem ser todos seguros de acordo com a norma de BLE e Bluetooth Mesh, e não pode ser desligada ou minimizada de nenhuma maneira.

A segurança do Bluetooth Mesh centra-se na utilização de três tipos de chaves de segurança. Essas chaves fornecem segurança para diferentes aspetos da rede *mesh*. Tratam-se da Netkey, Chave de Aplicação (Appkey) e chave de dispositivo (DevKey). Esses diferentes códigos vão servir para separar vários contextos nos diferentes níveis do Bluetooth Mesh, ou seja, prevenir conflitos de interesses.

A separação de contextos pode ser exemplificada por um cenário, no qual um conjunto de luzes ligado em *mesh*, pode agir como retransmissor. Nesta capacidade, os mesmos nós podem ser utilizados para a retransmissão de mensagens de outras aplicações, como sistemas de controlo de acessos e sistemas de climatização. Os nós das luzes não têm a necessidade de acessar ao conteúdo dessas mensagens, limitando-se a retransmiti-las. Para lidar com um possível conflito de interesses, a rede usa as várias chaves de segurança para proteger mensagens na camada de rede de outras com dados relacionados com aplicações específicas tais como iluminação, segurança física ou aquecimento.

Como foi exposto na subsecção 3.3.1, os nós da rede *mesh* devem ter uma NetKey para tornar um dispositivo num membro da rede. A partir da NetKey são criadas duas novas chaves, uma chave de encriptação da rede e chave de privacidade. A posse da NetKey permite ao nó descriptar e autenticar até à camada de rede, de forma a ter acesso a funções de rede, como a retransmissão. Apenas com a NetKey, não é possível descriptar dados de aplicação. A rede *mesh* pode ser dividida em sub-redes, das quais cada uma tem a própria NetKey, a qual irá estar na posse dos nós pertencentes à sub-rede. Essas sub-redes podem ser utilizadas para isolar certas áreas físicas.

A descriptação de dados de uma aplicação específica requer a posse da AppKey por parte dos nós. Nestas unidades da rede, podem existir diferentes AppKeys, das quais cada chave vai ser possuída por um pequeno conjunto de nós, nomeadamente aqueles que participam ativamente numa dada aplicação. Num caso de aplicação de iluminação, um nó com fontes de luz e outro nó com interruptores de luz devem ter a mesma chave de forma a possibilitar a interação entre si, no contexto desta aplicação.

As AppKeys são usadas pela camada de transporte superior para descriptar e autenticar mensagens antes de transferir os dados para a camada de acesso. As AppKeys devem ser associadas com apenas uma NetKey. Esta associação denomina-se de "ligação de chave" e significa que aplicações específicas, com a posse de uma

dada AppKey, podem funcionar apenas numa rede específica, enquanto uma rede pode receber várias aplicações protegidas independentes.

A DevKey é um tipo de chave de aplicação, a qual cada nó possui a sua chave única, sendo conhecida apenas pelo dispositivo *provisioner*. A DevKey é usada no processo de *provisioning* para proteger a comunicação entre o *provisioner* e o nó. Uma chave de privacidade, derivada da NetKey, é usada para esconder os valores de cabeçalho dos pacotes de dados, como o endereço de origem. Essa oclusão garante que escutas passivas não possam ser usadas para rastrear dispositivos e as pessoas que usam os mesmos.

3.4 Componentes de *hardware*

No sentido de usar BLE Mesh e Wi-Fi é necessário um componente de Hardware que tenha o suporte para estas tecnologias. O ESP32, representado na Figura 3.9 cumpre esse requisito, inclui as ferramentas necessárias para implementar uma rede de Bluetooth Mesh e é considerado um dos componentes de desenvolvimento mais baratos disponível no mercado.

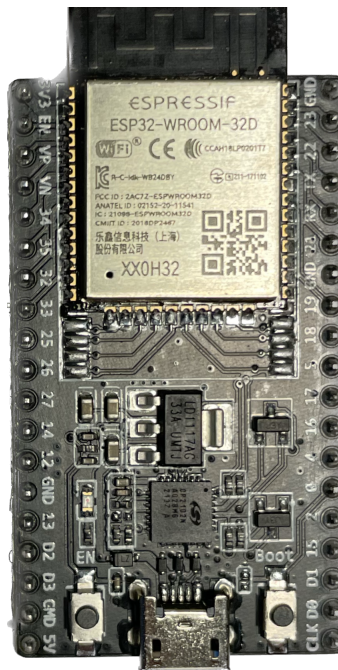


Figura 3.9: ESP32-WROOM-32D

O ESP32 é um conjunto de microcontroladores desenvolvido pela Expressif Systems, com suporte de tecnologias de comunicação sem fios, através da Wi-Fi e Bluetooth, bem como outros periféricos comuns de um microcontrolador (GPIOs, UART, SPI, ADC, etc) [Systems, 2021]. São componentes largamente usados em pequenos projetos devido ao seu baixo custo e a grande quantidade de funcionalidades existentes.

A Expressif Systems fornece uma *framework*, designada Expressif IoT Development Framework (ESP-IDF), na qual os núcleos dos diferentes periféricos, incluindo Wi-Fi e Bluetooth, e as respetivas Interfaces de Programação de Aplicação (API) são implementadas, de modo a facilitar o desenvolvimento, compilação, gravação do programa e *debugging* [Espressif, 2019]. O código compilado sobre ESP-IDF, é executado numa versão modificada do FreeRTOS, um Sistema Operativo de Tempo Real (RTOS) de código aberto para sistemas embebidos.

O desenvolvimento da solução descrita neste documento, irá depender do desenvolvimento de código, compilação, gravação no *hardware* e sessões de *debugging*, com base no ESP-IDF. Esta *framework* inclui a pilha de Bluetooth Mesh, o que não acontece com a programação em Arduino.

Tabela 3.1: Comparação de funcionalidades de plataformas de Hardware

Plataforma	ESP32	Raspberry Pi Zero 2 W
Processador (CPU)	Xtensa dual-core 32-bit 160/240MHz	Arm Cortex-A53 quad-core 64-bit 1GHz
RAM	520 KB	512 MB SDRAM
802.11 (Wi-Fi)	Sim	Sim
BLE	Sim	Sim
Bluetooth Mesh	Sim	Não
GPIO	34 pinos configuráveis	40 pinos configuráveis
Periféricos	I2C, I2S, UART, SPI, IR, PWM de motores e LEDs, sensor de efeito Hall, sensor de toque, Interface de Ethernet MAC, contador, relógio	HDMI, SPI, I2C, UART, PWM, USB, conector para Câmara

O *System on-Chip* (SoC) Raspberry Pi Zero 2 W foi considerado como uma alternativa ao ESP32 para a solução apresentada por esta tese. A Tabela 3.1 apresenta uma perspetiva geral das funcionalidades do ESP32 e Raspberry Pi Zero 2 W

[Espressif, 2019, Pi, 2022]. A segunda plataforma contém maior quantidade de pinos configuráveis de GPIO, capacidade de processamento e uma maior quantidade de memória, tendo um melhor desempenho do que o ESP32. Por outro lado, não suporta diretamente Bluetooth Mesh, sendo necessário determinar uma tecnologia alternativa para ligar os vários nós em rede. O facto de o ESP32 ter um menor custo, suporte para a implementação de redes de Bluetooth Mesh e ter uma larga quantidade de documentação e ferramentas de desenvolvimento, motiva a opção de utilização deste componente de *hardware* nesta dissertação.

Capítulo 4

Proposta de Solução

O Capítulo 4 apresenta a definição dos requisitos, conceito e a arquitetura da solução a implementar, sendo relevantes para a implementação do protótipo funcional. É constituído por seis secções, nas quais são especificados o conceito da solução, os requisitos, a arquitetura, as funcionalidades na rede de nós, a definição de um nó no contexto do sistema e termina com a definição das funções do nó *gateway*.

4.1 Requisitos do sistema

O sistema multiagente deve ser capaz de medir e controlar fatores ambientais, tais como temperatura, luminosidade e presença de pessoas, bem como ter a capacidade de controlar esses fatores. Com base neste objetivo principal, uma lista de requisitos funcionais e não funcionais foi definida. A lista de requisitos apresentada irá servir como base para a proposta de arquitetura. Os requisitos serão utilizados como condições para confirmar a validade da solução apresentada, através da avaliação do desempenho em relação aos requisitos definidos.

4.1.1 Requisitos dos agentes

No sistema proposto, os nós são considerados dispositivos individuais que representarão na rede *mesh* os agentes do sistema multiagente. Para um dispositivo ser convertido num nó, deve cumprir os seguintes requisitos:

- Ter pinos de GPIO e compatibilidade com outros periféricos que permita a extensão de funcionalidades do dispositivo como a ligação de sensores e atuadores;
- Ter a capacidade de processamento e armazenamento, para a execução do controlo e monitorização dos respetivos fatores;

- Ser apto para executar aplicações de edifícios inteligentes;
- Ser apto para a ligação em rede Bluetooth Mesh, ou seja suportar a versão BLE 4.2, ou superior.

4.1.2 Requisitos da rede de comunicação

De forma a permitir a interação entre os agentes, a rede de comunicação do sistema multiagente deve cumprir os seguintes requisitos:

- Suportar pelo menos uma tecnologia de comunicação sem fios;
- Permitir a interligação com uma topologia *mesh*, para o envio de mensagens para outros dispositivos que estão fora do alcance direto do remetente;

4.1.3 Requisitos de *software*

Para a implementação de *software* da rede de comunicação do sistema multiagente e dos agentes, os seguintes requisitos devem ser reunidos:

- Usar bibliotecas ou ferramentas de *software* que implementam o núcleo das tecnologias necessárias para o sistema;
- Usar ferramentas de *software* com APIs para permitir a sua utilização na implementação das aplicações;
- As bibliotecas ou ferramentas de *software* devem incluir documentação a descrever as respetivas funcionalidades e estrutura e como são utilizadas nas aplicações.

4.2 Conceito da solução

Com esta dissertação, pretende-se demonstrar casos de aplicação de um sistema distribuído inteligente baseado em sistemas multiagente e IoT, permitindo retirar os benefícios das duas abordagens para conceber um sistema inovador que permite, de forma distribuída, o controlo inteligente de sistemas. Os casos de aplicação serão implementados num contexto de edifícios inteligentes.

O sistema será constituído por vários elementos aptos a comunicar entre si, com a designação de nós, e estarão dispersos pela área de intervenção. Os elementos foram concebidos para permitir a monitorização e o respetivo controlo de um fator específico.

O facto de o sistema ser distribuído, possibilita uma maior área de cobertura e facilidade na substituição e adição de novos elementos, sem a necessidade de reconfiguração do sistema. Os seus elementos podem comunicar entre si através de vários tipos de mensagens, entre os quais pedidos, envios de estados e respostas.

Como exemplo, com este sistema, caso a temperatura interior de uma divisão do edifício seja inferior ao valor estabelecido, o elemento com o atuador adequado irá ser ativado de forma a colocar a temperatura na gama de valores estabelecida.

4.3 Definição de um nó

No contexto deste sistema, um nó é um dispositivo ligado à rede *mesh*, através da qual recebe e envia informação de/para os restantes nós. Estas unidades têm como fim permitir o funcionamento do sistema distribuído inteligente, através da comunicação entre eles. Nas aplicações desse sistema, será possível controlar os atuadores ligados a um nó, através de um outro nó no qual está ligado um sensor, ou remotamente numa plataforma de monitorização e controlo.

A maior parte dos nós terão a função de monitorização e controlo de algum fator específico do ambiente em redor. Assim, a comunicação entre os vários nós na rede irá ser relacionada com essas funções de monitorização e controlo. Em cada um destes nós, estão ligados pelo menos um elemento, um sensor ou um atuador. Os nós da rede irão desempenhar essencialmente dois papéis diferentes: obtenção das leituras dos sensores, e ajuste das condições do ambiente através dos atuadores.

Um nó com a missão de obtenção dos valores dos sensores, tem pelo menos um sensor ligado para obter os dados do fator que está a monitorizar e controlar. Os nós com este papel verificam se os valores lidos pelo sensor estão a respeitar as condições ambientais do fator que está a monitorizar, e, assim, enviam mensagens de atuação e medição de ligação e valores, para os respetivos nós com a função de atuar no ambiente.

Um nó com o papel de atuador no ambiente em redor, tem pelo menos um atuador ligado para executar ajustes do fator a controlar. Para ajustar o ambiente em conformidade, é necessário receber as mensagens de atuação provenientes dos nós com a função de leitura do ambiente, confirmar novamente se os valores lidos não estão a respeitar os definidos, e ligar os atuadores.

4.4 Rede de nós

Para a implementação do sistema multiagente, é necessário estruturar os nós com uma topologia *mesh*. Essa rede *mesh* irá permitir aos membros, os agentes, comunicarem entre si, essencial para a criação de soluções descentralizadas a implementar em edifícios inteligentes.

A rede de nós irá permitir a retransmissão de mensagens de forma a aumentar o alcance do sistema, e a aplicação do paradigma de SMA, através da distribuição do processamento dos dados. Os nós irão ter o seu respetivo processador e armazenamento, para a comparação de valores obtidos dos sensores, criação das mensagens para enviar para outros nós, de acordo com a respetiva estrutura definida, e a obtenção dos dados das mensagens recebidas dos outros nós.

A comunicação entre os nós é gerada a partir da necessidade da monitorização e controlo de fatores do ambiente, como a luminosidade e a temperatura. Por exemplo, os nós que fazem a leitura da temperatura numa divisão do edifício, enviam pedidos de ajuste para outros nós com aquecedores ou sistemas AVC ligados, caso os valores da temperatura lidos estejam fora da gama de valores de temperatura definida para o controlo.

Integrado nesta rede, um dos nós irá receber e enviar dados de outros nós, como leituras de sensores e estados dos atuadores, e por conseguinte, irá possibilitar o envio de dados e integração com outros sistemas de edifícios inteligentes e interface web. Este nó pode ser considerado uma *gateway*, uma vez que permite a entrada e saída de dados no exterior.

4.5 Arquitetura da solução

Tendo em conta o objetivo principal definido na subsecção 1.2.2, foram analisados os requisitos definidos na secção 4.1, para a definição de um sistema multiagente, composto por sensores e atuadores distribuídos para controlo e monitorização de fatores ambientais. No âmbito desta dissertação, a tecnologia Bluetooth Mesh foi analisada e escolhida para a comunicação distribuída entre os vários agentes constituintes do sistema multiagente.

Neste contexto, na rede de Bluetooth Mesh a implementar, os agentes do sistema multiagente vão assumir o papel de nós. Um dos agentes, o agente delegado, irá assumir o papel de *gateway*, de forma a possibilitar a comunicação do sistema com o mundo exterior. Neste caso, o *gateway* será utilizado para interligar a rede Bluetooth Mesh com a Internet (TCP/IP), através do módulo Wi-Fi do agente delegado. A

4.6 Nó *gateway*

Para a comunicação e interação com o sistema multiagente a partir da internet, é necessário que um dos nós, respetivo ao agente delegado, tenha uma interface com o mundo exterior, capaz de retransmitir mensagens entre os agentes integrados na rede de Bluetooth Mesh e um servidor remoto a correr um *broker* MQTT.

O nó *gateway* irá ser utilizado para receber mensagens com pedidos de leitura e escrita de algum tipo de fator ambiental específico, como temperatura ou luminosidade, e reencaminhar essa mensagem para o grupo de nós da rede de Bluetooth Mesh, o qual está a alojar a equipa de agentes comuns. No sentido inverso, o agente delegado, ou nó *gateway* no contexto da rede de Bluetooth Mesh, irá receber dos agentes comuns as respostas das mensagens previamente recebidas da Internet, e proceder ao seu envio para o servidor através da Internet.

Capítulo 5

Sistema Multiagente Baseado numa Rede Mesh

Este capítulo descreve a implementação do protótipo da solução definida no capítulo 4. Esta será feita com módulos de desenvolvimento ESP32 (ESP32-WROOM-32D), os quais irão desempenhar o papel de nós de uma rede de Bluetooth Mesh.

No âmbito desta dissertação, os nós irão alojar a monitorização e o controlo de *software* de vários fatores ambientais, nomeadamente a temperatura e a iluminação. Por exemplo, no contexto de edifícios inteligentes, mencionado na secção 2.3, no qual se enquadra o protótipo a desenvolver, um sensor seria ligado a um nó, no qual todos os dados de temperatura seriam lidos, e este nó iria enviar mensagens com valores e pedidos de ligação a outro par, no qual estaria ligado um atuador.

Com base nos conceitos de Bluetooth Mesh descritos nas secções 3.2 e 3.3, a *framework* ESP-IDF define uma stack com a implementação dessa comunicação, permitindo a abstração dos seus respetivos conceitos. As componentes de *software* de ESP-IDF são escritas nas linguagens de programação C ou C++. O *software* das aplicações da solução será desenvolvido e compilado com a *framework* ESP-IDF versão 4.4, também mencionado na secção 3.4.

O presente capítulo é constituído por nove secções, as quais irão aprofundar diferentes aspetos da implementação de *software*, desde a inicialização do módulo de Bluetooth Mesh, no ESP-IDF, até ao *software* de aplicação. Por ordem, o capítulo 5 começa por enumerar as estruturas de dados de Bluetooth Mesh no ESP-IDF e os passos requeridos para a inicialização do Bluetooth Mesh. De seguida, existe uma explanação do modo automático de integração de nós na rede (*provisioning*), utilizado na implementação, passando para a especificação dos modelos utilizados na comunicação de Bluetooth Mesh e a enumeração das funções implementadas para notificar e gerir eventos de envio e receção de mensagens. Por último, apresentam-se as principais componentes constituintes do *software* de aplicação implementado.

5.1 Estruturas de dados do Bluetooth Mesh no ESP-IDF

Na pilha de *software* de *Bluetooth Mesh* implementada no ESP-IDF, foram desenvolvidas várias estruturas de dados e APIs de conceitos introduzidos nas secções 3.3, nomeadamente os elementos, modelos, envio e receção, bem como a subscrição e publicação de mensagens, referidos na secção 3.3. Essas componentes vão ser necessárias para possibilitar a integração, configuração e comunicação dos nós na rede. Algumas das estruturas e APIs, incluem tipos de dados e chamadas de funções definidos no núcleo da pilha de *software* de Bluetooth Mesh no ESP-IDF, e que não serão foco da dissertação.

Para a definição de modelos na aplicação, é necessário criar objetos das seguintes estruturas, que tem como base as definições de modelos do Bluetooth SIG, mencionadas na subsecção 3.3.2:

- `esp_ble_mesh_client_op_pair` – abstração que faz a correspondência do código de operação da mensagem de leitura ou escrita do modelo cliente com o respetivo código de operação da mensagem de estado;
- `esp_ble_mesh_model_op_t` – estrutura que descreve o contexto de uma operação do modelo, isto é, a definição do nome da operação e o comprimento mínimo da mensagem;
- `esp_ble_mesh_client_t` – contexto de dados de utilizador do modelo cliente. Estrutura responsável por lidar com publicações de mensagens e respostas de confirmação nos modelos clientes dentro da pilha de Bluetooth Mesh no ESP-IDF;
- `esp_ble_mesh_model_t` – abstração que descreve um objeto do modelo de Bluetooth Mesh;
- `esp_ble_mesh_gen_onoff_srv_t` – dados de utilizador do modelo servidor genérico *OnOff*.

Após a definição de todos os aspetos do modelo, é necessário definir estruturas relativas à integração de nós na rede (*provisioning*) e configuração:

- `esp_ble_mesh_cfg_srv_t` – define o contexto do modelo servidor de configuração, isto é, ativa as funcionalidades suportadas pelo Bluetooth Mesh e define configurações específicas do nó. É estrutura implementada para ser utilizada com a macro `esp_ble_mesh_model_cfg_srv` para inicializar o modelo servidor de configuração;
- `esp_ble_mesh_elem_t` – abstração que descreve um elemento do BLE Mesh. Nas aplicações são definidos *arrays* do `esp_ble_mesh_elem_t` com os vários elementos constituintes do nó, nos quais para cada instância são armazenados os respetivos modelos raiz, mencionados na subsecção 3.3.2, os modelos personalizados do utilizador (*vendor models*) e os respetivos tamanhos;
- `esp_ble_mesh_comp_t` – contexto de dados de composição do nó. Estrutura de dados onde vão ser armazenados os identificadores do nó e o array dos seus elementos;
- `esp_ble_mesh_prov_t` – estrutura que armazena todas as capacidades e propriedades de *provisioning*.

5.2 Inicialização do módulo de Bluetooth Mesh

A implementação de aplicações de Bluetooth Mesh sobre o ESP-IDF requer a inicialização do respetivo sub-módulo e o módulo de Bluetooth. Esse processo do módulo de Bluetooth, necessário para comunicar através dele, envolve vários passos, nomeadamente as inicializações do controlador de Bluetooth do ESP32 e do *host* Bluedroid¹. De seguida, o Bluedroid é ativado, reunindo todas as condições necessárias para proceder com a comunicação via Bluetooth. Como esses passos são feitos internamente na pilha de *software* de Bluetooth Mesh do ESP-IDF, não sendo relevante no âmbito desta dissertação detalhar o processo em profundidade.

A inicialização do Bluetooth Mesh é um processo que compreende os seguintes passos:

- Criação e registo de *callbacks* de funções que irão ser chamadas depois dos seus eventos serem desencadeados pelas mensagens de Bluetooth Mesh;

¹Bluedroid é uma das pilhas da parte *host* de de Bluetooth suportada no ESP-IDF, de forma a permitir o suporte de Bluetooth clássico e BLE

- Inicialização das capacidades de *provisioning* e a informação dos dados de composição;
- Ativação das capacidades de *provisioning*, com a ativação dos portadores de divulgação, para divulgar a disponibilidade para ser integrado numa nova rede.

Todos os passos são essenciais a qualquer aplicação desenvolvida sobre a pilha de *software* de Bluetooth Mesh do ESP-IDF. São executados através de chamadas de API que são implementadas dentro da *framework*, não sendo relevante no âmbito desta dissertação detalhar o processo em profundidade.

5.3 Integração de dispositivos na rede

Para integrar novos dispositivos na rede de forma automática e rápida, foram incluídas chamadas de funções de *Fast Provisioning* no código. Desta forma, o processo de *provisioning*, descrito na subsecção 3.3.1, não requer a utilização de dispositivos externos a servir como *provisioner*, como um *smartphone*, mas sim um dispositivo ESP32 integrado na rede que irá desempenhar esse papel.

Neste caso, quando um novo dispositivo divulga a disponibilidade para ser integrado numa rede de Bluetooth Mesh, o dispositivo *provisioner* recebe esse sinal e automaticamente procede à integração desse dispositivo na rede.

A utilização de *fast provisioning* por parte do dispositivo *provisioner* requer a definição de um modelo personalizado, dado que as mensagens devem ter uma estrutura própria. Assim, os modelos cliente e servidor têm um conjunto de APIs, estruturas, códigos de operação, os seus respetivos comprimentos de mensagem, correspondências entre códigos de operações de leitura e escrita e as respetivas respostas, que não foram definidas pelo grupo de desenvolvimento de Bluetooth (Bluetooth SIG).

O modelo cliente e de servidor do *fast provisioning* foi concebido pelo fabricante do ESP32, dentro da pilha Bluetooth Mesh, não sendo relevante no âmbito desta dissertação detalhar o processo em profundidade.. No contexto deste sistema, o *fast provisioning* foi utilizado para gerir pacotes de divulgação de dispositivos não integrados na rede e enviar pedidos de adição das chaves de aplicação, mencionadas na secção 3.3.6.

5.4 Modelos de Bluetooth Mesh

No contexto de redes BLE Mesh no ESP-IDF, existem as definições dos modelos de raiz, essenciais para implantar a rede de Bluetooth Mesh, e dos modelos genéricos, para a comunicação específica para a aplicação. Essas definições são referidas na subsecção 3.3.2. No contexto da dissertação foram utilizados o modelo de raiz servidor de configuração e os modelos genéricos *OnOff* cliente e *OnOff* servidor.

O modelo de servidor de configuração está presente em todos os nós e implementa todas as definições necessárias para converter um dispositivo num nó de rede (*provisioning*), gere a atribuição de chaves de aplicação para os vários modelos implementados no nó e a configuração de publicação das respetivas mensagens.

O modelo *Generic OnOff* cliente responsável pelas operações de leitura e escrita de estados *on/off*, envio das mensagens para o(s) nó(s) com o modelo *Generic OnOff* servidor, necessárias para essas operações. No caso da aplicação descrita na dissertação, essas mensagens de pedidos de leitura irão ser feitas assim que o nó estiver integrado na rede de Bluetooth Mesh e estiver ligado com a respetiva chave de aplicação, enquanto que as mensagens com pedidos de escrita ocorrem, caso as condições definidas com base nos dados do sensor ou botão estejam reunidas (ex: a temperatura ultrapassar o limite definido). Como um modelo cliente, essas mensagens de leitura e escrita iniciam a comunicação com o modelo *Generic OnOff* servidor.

O *Generic OnOff* servidor é responsável pela receção de mensagens de escrita e leitura de estados por parte do nó com modelo *Generic OnOff* cliente, envio dos estados a escrever para o atuador e o envio da resposta dos pedidos de escrita e leitura para o nó com modelo *Generic OnOff* cliente, com os valores dos estados atuais. Assim, agindo como um modelo servidor nunca envia mensagens de pedidos ao modelo *Generic OnOff* cliente. A Figura 5.1 ilustra a comunicação entre os modelos *Generic OnOff*.

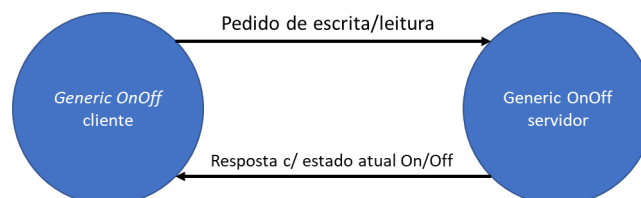


Figura 5.1: Comunicação entre os modelos *Generic OnOff*

5.5 Configuração do nó

A implementação das diferentes funcionalidades e configurações é feita pelo modelo servidor de configuração através da macro `esp_ble_mesh_model_cfg_srv`. Este lida com todas as definições necessárias para converter um dispositivo num nó de rede (*provisioning*), a atribui chaves de aplicação para os vários modelos implementados no nó e configuração a publicação das mensagens.

Esta configuração pode ser definida e controlada através de uma aplicação móvel de criação de redes Bluetooth Mesh (nRF Mesh), quando é utilizado um telemóvel como *provisioner* ou, neste caso, através do modelo cliente de configuração implementado no dispositivo *provisioner*, da rede Bluetooth Mesh concebida neste sistema.

O modelo cliente de configuração, utilizado no dispositivo ESP32 que serve de *provisioner*, adiciona chaves de aplicação, e faz a ligação aos modelos necessários, quer nos nós acabados de integrar, quer no próprio *provisioner*. No caso deste sistema, assim que o nó é inicializado, o modelo cliente de configuração adiciona uma nova chave ao dispositivo, e a seguir faz a sua ligação ao modelo cliente da aplicação.

Após a integração de um novo dispositivo na rede, o *provisioner* irá enviar uma mensagem ao novo nó para pedir a adição da chave nesse componente. Caso o envio desse pedido seja bem sucedido, isto é, o respetivo evento desencadeado, irá ser enviada uma nova mensagem ao nó com o pedido de ligação da chave de aplicação ao modelo servidor utilizado pela aplicação.

5.6 Funções de notificação e gestão de eventos

No contexto de implementação de Bluetooth Mesh no ESP-IDF, são gerados eventos nas camadas superiores da pilha de Bluetooth Mesh no ESP-IDF, no momento da receção e envio de mensagens. Para gerir e notificar os eventos desencadeados em diferentes contextos, devem ser criadas e registadas funções na pilha, de forma a serem chamadas quando estes eventos ocorrem. Para o registo dessas funções, o ESP-IDF tem diferentes *callbacks* para fins distintos. No âmbito da aplicação implementada nesta dissertação, as seguintes *callbacks* foram tratadas:

- *Callback* de *provisioning* – notifica eventos e valores de parâmetros relacionados com a integração do dispositivo na rede. Esta *callback* é apenas utilizada durante o processo de integração, o qual notifica as diferentes fases do processo;
- *Callback* servidor de configuração – notifica eventos e valores de parâmetros associados à implementação de configurações do nó;

- *Callback* cliente de configuração – notifica eventos e valores relacionados com a definição de configurações a implementar no servidor;
- *Callback* servidor de modelo genérico – notifica e gere eventos relacionados com a receção de mensagens com pedidos de leitura e escrita de valores de modelos genéricos;
- *Callback* cliente de modelo genérico – notifica e gere eventos relacionados com o envio de mensagens com pedidos de escrita e receção de mensagens com respostas de pedidos de leitura e escrita, enviados aos nós servidores;
- *Callback* de Modelo Personalizado – notifica e gere todos eventos relacionados com mensagens dos modelos personalizados, incluindo os processos de envio, receção e publicação de mensagens, de clientes ou servidores. No contexto do sistema, é utilizada para notificar e gerir eventos relacionados com o mecanismo de *fast provisioning*, descrito na secção 5.3;

As *callbacks* de *provisioning* e servidor de configuração são críticas para a integração dos nós dentro da rede Bluetooth Mesh e da aplicação para a qual vão ser alocados. A *callback* do modelo personalizado é necessária para a troca de mensagens relacionadas com a automatização do processo de integração de um dispositivo na rede Bluetooth Mesh. As *callbacks* relacionadas com os modelos genéricos são utilizadas para a funcionalidade das aplicações do sistema de edifício inteligente.

Em todas as funções de *callback*, são passados dois argumentos pela pilha de Bluetooth Mesh no ESP-IDF, o tipo de evento e os respetivos valores de parâmetros. Os parâmetros variam com os diferentes eventos, sendo que as funções de *callback* devem utilizar os valores de argumento dos parâmetros para gerir o evento.

Os parâmetros dos eventos geralmente incluem informações sobre a mensagem, como o seu código de operação, o seu comprimento, o endereço do nó remetente, caso seja uma receção de mensagem, e códigos de erro, caso a mensagem tenha falha de envio. Os parâmetros também podem armazenar o modelo usado para enviar ou para o qual é esperado receber uma mensagem, através da utilização de apontadores.

5.7 Envio de mensagens entre modelos

Dentro da pilha de Bluetooth Mesh, existem três funções que permitem o envio e publicação de mensagens, entre modelos clientes e modelos servidores. O envio de mensagens requer a integração dos nós na rede e a ligação do respetivo modelo a uma chave de aplicação. Na implementação do sistema, foram utilizadas apenas APIs de envio de mensagens, deixando de parte a publicação em grupos.

Nas aplicações de Bluetooth Mesh sobre ESP-IDF, o envio de mensagens é um evento que ocorre uma única vez, e caso esteja ativado na configuração dos nós, pode ser alvo de retransmissão por outros nós até atingir o destino, como ilustra a Figura 5.2.

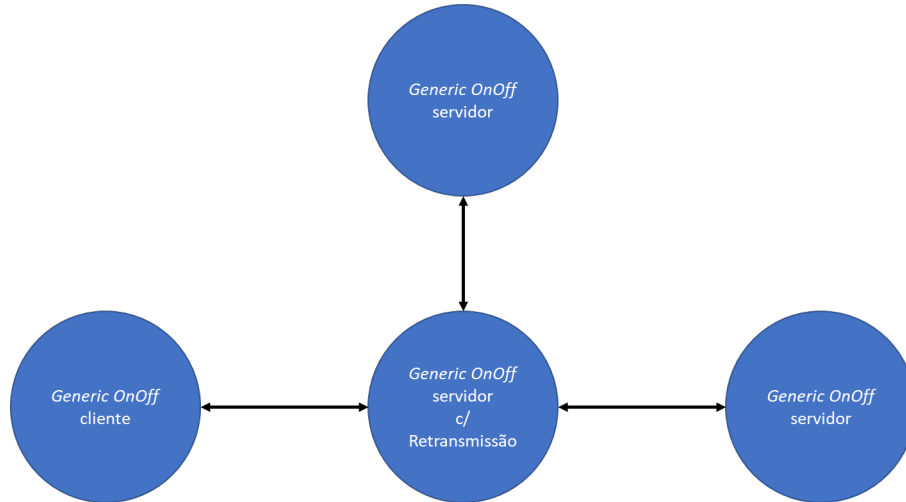


Figura 5.2: Comunicação entre modelos *Generic OnOff* com retransmissão

Nas aplicações do sistema, foi definida a utilização de mensagens das operações do tipo *set*, *get* e *status* na comunicação entre modelos cliente e servidor:

- *Get* – mensagens utilizadas por modelos clientes implementados em nós, acabados de serem integrados na rede, principalmente para obter valores iniciais de estados de modelos servidores.
- *Set* – mensagens utilizadas por modelos clientes, com pedidos de escrita de estados de modelos servidores.
- *Status* – mensagens de resposta enviadas por parte de modelos servidores, para mensagens do tipo *set* e *get* de modelos clientes.

Como foi explicado na subsecção 5.1, um *array* da estrutura de dados de um modelo `esp_ble_mesh_client_op_pair` faz a correspondência de mensagens entre os códigos de operação *set* e *get* e os respectivos códigos de operação de resposta, normalmente, as mensagens *status*.

Esta correspondência é importante para desencadear eventos que notificam o final de pedidos *get* e *set*. Os códigos de operação não só identificam a operação propriamente dita, como também são um critério de verificação da estrutura do conteúdo da mensagem.

5.8 Receção de mensagens

Um nó tem duas formas de receber mensagens: receção de mensagens enviadas diretamente por outros nós, usando o seu endereço *unicast*, ou através de mensagens publicadas em grupos, subscritos pelos modelos inicializados no nó. No caso deste sistema foi utilizado o envio direto para os nós.

Quando um nó recebe uma mensagem, funções do núcleo da pilha do Bluetooth Mesh irão verificar o modelo e o seu código de operação, e encaminhar para a função de *callback* e evento corretos. Dentro da função de *callback*, irão ser executadas as ações de gestão para o evento determinado.

5.9 Aplicação sobre Bluetooth Mesh

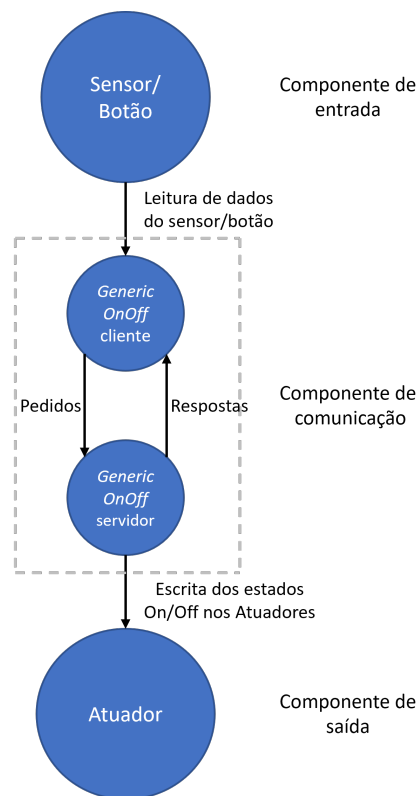


Figura 5.3: Componentes de aplicação

No desenvolvimento da parte de aplicação que irá depender diretamente da comunicação do Bluetooth Mesh, foram definidas três componentes essenciais, ilustradas na Figura 5.3:

- Componente de entrada - leitura dos valores obtidos por sensores ou botões. Componente ligada diretamente ao nó com o modelo de aplicação cliente.

- Componente de comunicação - envio e receção de pedidos e das respetivas respostas através da rede de Bluetooth Mesh.
- Componente de saída - execução dos pedidos recebidos. Componente ligada diretamente ao nó com o modelo de aplicação servidor.

A componente de entrada é a componente responsável pela inicialização dos sensores ou botões no GPIO, implementação de funções de leitura, nomeadamente para a leitura do estado do pino, e, caso necessário, lógica para converter esses estados para a base decimal. A inicialização de sensores e botões no GPIO implica pelo menos definir o pino do ESP32 a que está ligado, como entrada. As funções de leitura de sensores, consistem na obtenção do nível de entrada no respetivo pino do ESP32.

A leitura de certos sensores, como o DHT22, implica uma variação da direção do pino de ESP32 a que está ligado, isto é, antes de obter leituras do sensor, poderá ser necessário definir o pino como saída, para fazer pedidos de dados, com a definição do nível do pino no estado alto por um instante de tempo (normalmente na casa dos μs). Certos sensores, como o DHT22, também exigem a conversão dos dados para a base decimal, uma vez que estes entregam vários bits de dados, respetivos a diferentes parâmetros e que precisam de ser separados.

A componente de comunicação é dividida na parte relativa ao modelo de cliente e na parte referente ao modelo de servidor. Ambas têm em comum a inicialização do módulo de Bluetooth Mesh do ESP-IDF, descrita na subsecção 5.2.

Na parte de modelo cliente, é utilizado o mecanismo de *fast provisioning*, descrito na subsecção 5.3, para permitir escalar a rede, e o modelo *Generic OnOff* cliente, necessário para enviar mensagens com pedidos de escrita e leitura para o nó com o modelo servidor e receber as respetivas respostas.

Nesta parte de modelo cliente, assim que o respetivo nó é inicializado, é feito o envio de pedido de leitura dos estados ao nó com a parte de servidor. De seguida, a tarefa respetiva à leitura do sensor dá início, entrando num ciclo infinito, onde são feitas as chamadas para as funções de leitura pertencentes à componente de entrada. Ainda dentro do ciclo, são verificadas as condições definidas para o envio de mensagens com pedidos de ligação dos atuadores, para a parte de modelo servidor, recebendo uma resposta de confirmação com os valores atualizados.

Por exemplo, no caso da temperatura, é verificado se o seu valor lido dos sensores está fora dos limites mínimo e máximo, e se o aquecedor ou ventilador estão desligados, com base nos valores das variáveis internas, que armazenam os valores dos atuadores.

Se ambas as condições ocorrerem, o modelo cliente irá enviar um pedido de ligação dos atuadores ao nó com o modelo servidor.

O passo final dentro do ciclo será atualizar as variáveis internas com os novos valores de estados dos atuadores, e reiniciar. Se o oposto ocorrer, isto é, se os valores estiverem dentro dos limites estabelecidos e os atuadores ligados, de acordo com as variáveis internas, o modelo cliente envia um pedido para desligar os atuadores, recebendo de volta uma resposta de confirmação com os valores atualizados.

A parte de modelo servidor utiliza o modelo genérico *OnOff* servidor para receber os pedidos de um nó com modelo cliente e envia as respostas com o estado atual dos atuadores que pertencem à componente de saída. Nesta parte foi implementada uma função, para atualizar os valores dos atuadores através da chamada de outra função, responsável pela operação dos atuadores, e também para enviar a mensagem de resposta com os valores atualizados.

Por último, a componente de saída implementa os estados dos atuadores recebidos, através da ligação direta ao nó com o modelo servidor. Nesta componente, o pino do ESP32, no qual se liga o atuador, é inicializado como uma saída. Para operar o atuador, é criada uma função de operação, a qual recebe como argumento os novos valores, para verificar se o valor atual do atuador é diferente dos novos valores recebidos e, se for esse o caso, procede com a atualização do estado no pino a que está ligado.

Capítulo 6

Testes e Análise de Resultados

De forma a testar o desempenho da solução de sistema definida, num ambiente interior de um edifício, foram definidos três cenários de aplicação, no âmbito de monitorização e controlo de fatores ambientais. Esses cenários de aplicação consistem em demonstrar:

- Funcionamento de on/off em rede *mesh* – implementação de uma aplicação on/off para iluminação, através de vários nós ligados a fontes de iluminação e botões, visando demonstrar algumas funcionalidades do Bluetooth Mesh, a tecnologia de rede que é utilizada na comunicação efetuada entre os vários nós do sistema de edifícios inteligentes.
- Controlo e monitorização de temperatura – aplicação que visa o controlo e a monitorização de temperatura, através de nós servidores com aquecedores ou ventiladores ligados, os quais recebem dados de temperatura e pedidos de ligação de nós clientes com sensores de temperatura ligados.
- Controlo de presença – aplicação de controlo de iluminação com base na deteção de movimento, isto é, ligar e desligar luzes conectadas a um nó servidor, com base nas mensagens de pedidos enviadas por nós clientes, e que, por sua vez, está ligado um sensor de movimento Passive Infrared (PIR).

Para cada um destes cenários irão ser igualmente definidos casos de testes, necessários para o registo de diferentes métricas em diferentes condições ambientais, como, por exemplo, a latência para diferentes distâncias entre nós e a existência de obstáculos no meio. Para cada um dos casos de testes definidos, irão ser criados gráficos ou tabelas que demonstrarão a evolução das métricas com a variação das condições.

O presente capítulo é constituído por três secções, uma para cada cenário definido, as quais irão especificar os seus casos de testes e os respetivos resultados, sob a forma de tabelas ou gráficos.

6.1 Funcionamento de On/Off em rede mesh

O sistema multiagente para edifícios inteligentes, concebido nesta dissertação, depende de Bluetooth Mesh, a tecnologia de comunicação desenvolvida para adicionar a topologia *mesh* ao já existente Bluetooth. O cenário de demonstração do funcionamento de Bluetooth Mesh mostra a funcionalidade de retransmissão em ação, através de uma aplicação de on/off para iluminação, como a retransmissão de mensagens.

O cenário visa demonstrar a escalabilidade de uma rede de Bluetooth mesh no que toca ao número de nós, e a rapidez de integração de novos dispositivos na rede *Mesh*, através da automatização do processo de *provisioning*. A aplicação de ligar e desligar as luzes requer a utilização do modelo genérico *OnOff* cliente ou servidor. Todos os nós integrados na rede, que tenham estes modelos, podem ter os mesmos ligados à respetiva chave de aplicação, necessária para garantir a segurança da comunicação relativa à aplicação.

6.1.1 Casos de testes

Este cenário foi utilizado como contexto para testar o comportamento da rede de Bluetooth Mesh no ambiente de um edifício, onde existem vários obstáculos, como paredes. Para isso, foram definidas as métricas:

- Latência entre nós;
- Taxa de Perda de Mensagens.

Essas métricas são medidas com a variação de diferentes condições ambientais, como a distância entre os nós, a existência de nós retransmissores e de obstáculos entre nós, como paredes.

Este cenário é constituído por dois agentes permanentes, um com modelo cliente e outro com modelo servidor, havendo um terceiro utilizado de forma intermitente, para servir como retransmissor. Todos eles são configurados para comunicar na rede de Bluetooth Mesh.

As medições respetivas à latência e taxa de perda de mensagens são feitas a partir do nó com modelo cliente, com base nos registos de ocorrências enviados, via porta série. Neste contexto, a taxa de perda de mensagens é a percentagem de mensagens que o nó com modelo cliente enviou ao nó com modelo servidor, sem receber a respetiva resposta, gerando um *timeout*.

Utilizando diferentes distâncias e o uso intermitente de obstáculos e nós retransmissores são definidos seis testes diferentes, especificados na Tabela 6.1. Estes são

feitos dentro de casa, em que os nós são colocados na mesma divisão ou em divisões diferentes, com paredes no meio. A Figura 6.1 ilustra as seis situações de teste.

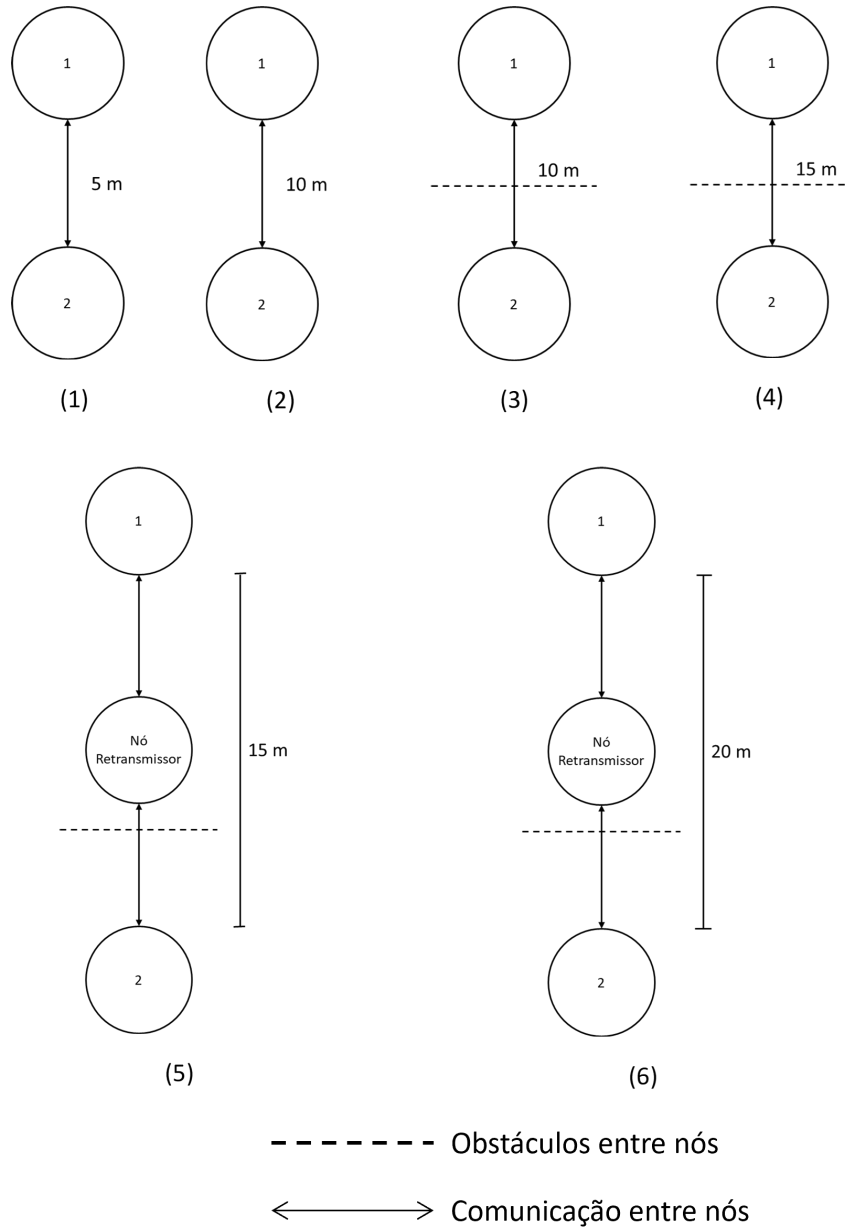


Figura 6.1: Casos de testes de desempenho de rede de Bluetooth Mesh

Tabela 6.1: Critérios de testes de desempenho de rede de Bluetooth Mesh

Número de Teste	Distância entre nós (m)	Existência de Obstáculos	Existência de Nós Retransmissores
1	5	Não	Não
2	10	Não	Não
3	10	Sim	Não
4	15	Sim	Não
5	15	Sim	Sim
6	20	Sim	Sim

6.1.2 Resultados dos testes

Seguindo a definição de testes da subsecção 6.1.1, estes resultaram nos valores apresentados na Tabela 6.2. Com a análise dos resultados, é perceptível que a rede de Bluetooth Mesh começou a falhar em distâncias de comunicação maiores, com maior latência e taxa de perda de mensagens, aos 10 m e 20 m de distância.

Tabela 6.2: Resultados de testes de desempenho de rede de Bluetooth Mesh

Número de Teste	Perdas de Pacotes (%)	Latência Mínima (s)	Latência Máxima (s)	Latência Média (s)
1	0%	0.084	0.114	0.0967
2	0%	0.126	0.128	0.127
3	0%	0.100	0.110	0.105
4	50%	0.1325	0.14	0.13625
5	0%	0.12	0.168	0.140125
6	10%	0.16	0.192	0.176

Por outro lado, com adição de nós de retransmissão, a perda de mensagens foi quase ou mesmo nula, mas a latência aumentou. Apesar disso, a funcionalidade de retransmissão aumentou a garantia de entrega de mensagens enviadas, com uma taxa de perda menor relativamente a uma comunicação direta.

6.2 Controlo e monitorização de temperatura

A temperatura é um dos fatores ambientais incluídos no sistema para controlar e monitorizar, sendo o foco de um dos cenários a implementar e a testar. Tendo em conta uma definição prévia de limites de temperatura, o controlo é feito por dois nós com modelo servidor, aos quais cada um é ligado respetivamente a um ventilador e um aquecedor. O aquecedor é ativado se a temperatura fosse menor que o limite inferior, e o ventilador, se a temperatura fosse maior que o limite superior.

Os dados de temperatura são obtidos por um nó cliente, através de um sensor de temperatura a ele ligado. Caso os valores de temperatura estejam fora dos limites definidos, o nó cliente enviaria um pedido para os nós servidores, para ligar o atuador de controlo necessário, de forma a voltar a colocar a temperatura dentro dos limites definidos. Se a temperatura atingir valores dentro dos limites definidos e os atuadores continuarem ligados, o nó cliente envia uma nova mensagem para os nós servidores, com o pedido de desligar os atuadores, dado que já não é necessário.

6.2.1 Casos de testes

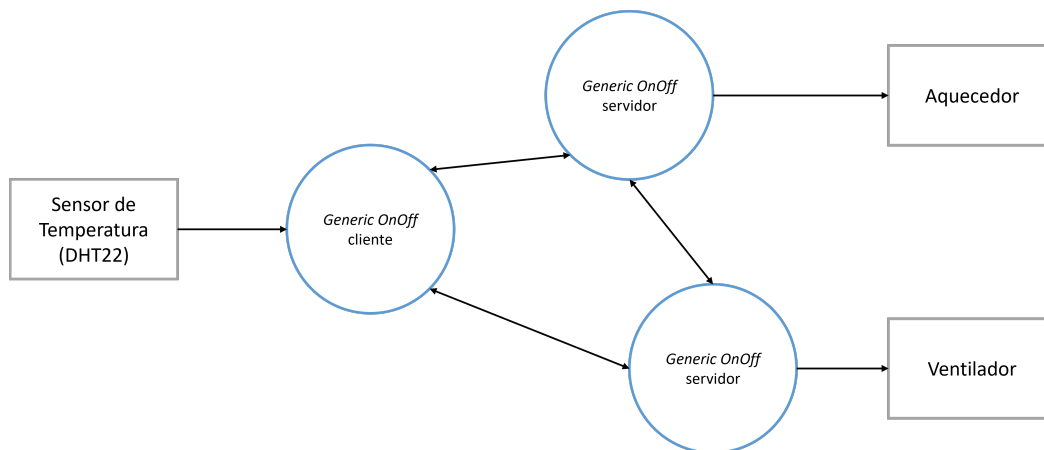


Figura 6.2: Ligações entre os nós para controlo e monitorização de temperatura

Como forma de testar o desempenho em termos de controlo e monitorização de fatores ambientais de um sistema multiagente, com base numa rede de Bluetooth Mesh, é feito um teste a partir da descrição do cenário, nos dois parágrafos anteriores. Assim, neste teste é criada uma rede de Bluetooth Mesh, constituída por três nós, um com o modelo cliente, onde está ligado o sensor e verifica se a temperatura está dentro dos limites, e outros dois nós servidores, aos quais em cada um está ligado respetivamente uma ventoinha e um aquecedor. As ligações entre nós, atuadores e sensores definidas para o teste, estão ilustradas na Figura 6.2.

Para executar o teste, é definido o registo periódico de minuto a minuto da temperatura lida pelo nó cliente através do sensor DHT22, e o respetivo estado do atuador, a testar naquele momento. São feitos dois casos de testes: um orientado para a resposta do ventilador, e outro para a resposta do aquecedor. Para isso, é necessário colocar a temperatura ambiente nas condições ideais para os diferentes atuadores darem resposta.

Para este teste é definido o valor 19 °C como o limite inferior de temperatura e 21 °C como limite superior, tendo em conta que este cenário usado como contexto foi concebido para a área de uma divisão. Para testar a resposta de arrefecimento,

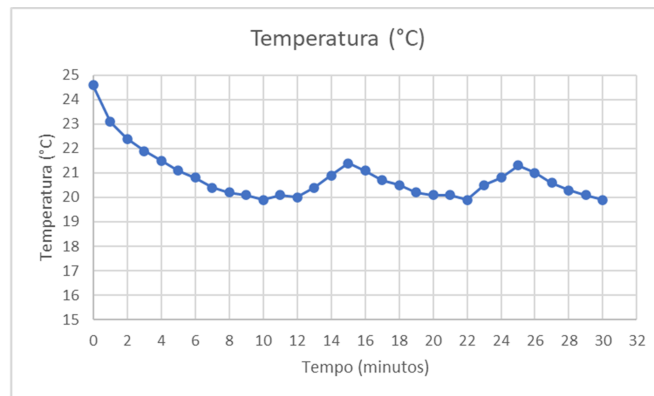
é necessário colocar a divisão numa temperatura superior a 21 °C através de um sistema de climatização externo, e iniciar a aplicação de monitorização e controlo, para proceder ao arrefecimento da divisão. Para testar a resposta de aquecimento, requer climatizar a divisão para uma temperatura inferior a 19 °C, para a aplicação ligar o aquecedor assim que fosse iniciada.

O registo periódico é feito em duas tabelas, uma respetiva ao teste de ventilação e outra ao teste de aquecimento. Cada tabela dá origem a dois gráficos, um que representa a evolução da temperatura ao longo do tempo, e o outro a evolução do estado do atuador correspondente ao longo do tempo.

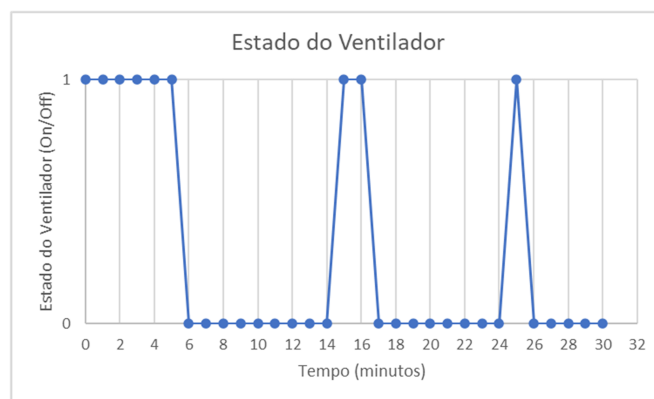
6.2.2 Resultados dos testes

As Figuras 6.3 e 6.4 demonstram o quão rápido o controlo e monitorização de temperatura pode ser numa solução de sistemas Multiagente baseada em Bluetooth Mesh. No caso da Figura 6.3, a aplicação foi iniciada (criação da rede e integração dos respetivos nós) quando a temperatura ambiente da divisão estava próxima dos 25 °C. Portanto, logo que esse valor de temperatura foi lido, foi enviada uma mensagem para o nó cliente com ventoinha, ligar o respetivo atuador.

Nota-se pelo gráfico (b) da Figura 6.3 que, quando os valores de temperatura começaram a ser registados, a ventoinha já estava ligada. Ao longo do teste, foi usado um sistema de climatização externo para aumentar a temperatura artificialmente, que, como esperado, ao ultrapassar os 21 °C, ativava a ventoinha, resultando numa descida da temperatura. Quando a temperatura entrava nos limites, a ventoinha desligava, fruto da mensagem de pedido do nó com modelo cliente, o qual estava a ler os valores de temperatura.



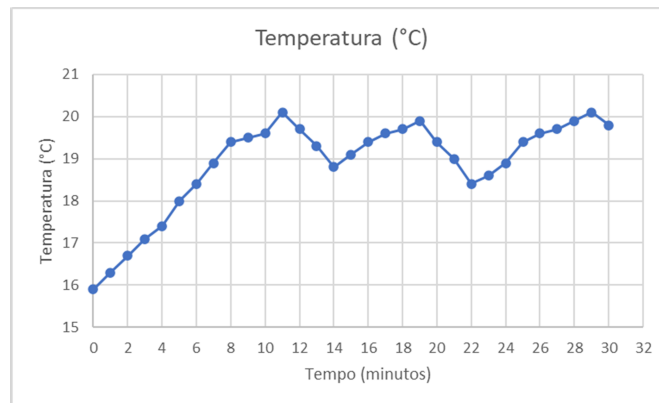
(a)



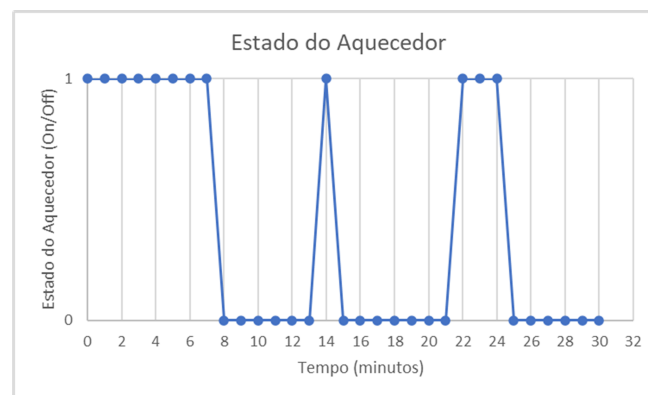
(b)

Figura 6.3: Resultados de Testes de Controlo e Monitorização de Temperatura com a evolução da Temperatura (a) e do Estado de Ventilador (b) ao longo do tempo

Relativamente ao teste de uso do aquecedor, a temperatura da divisão estava a 16°C, quando a aplicação iniciou a tarefa de controlo e monitorização. Neste teste, também é notável nos gráficos da Figura 6.4, que o aquecedor começou a trabalhar logo no momento em que foram lidos valores abaixo do limite, parando de trabalhar assim que a temperatura chegou aos 19 °C. Também durante este teste, a temperatura da divisão foi alterada artificialmente, para registar a forma como a aplicação reagiria às alterações, e aí notou-se igualmente a rapidez da ligação do atuador.



(a)



(b)

Figura 6.4: Resultados de Testes de Controlo e Monitorização de Temperatura com a evolução da Temperatura (a) e do Estado de Aquecedor (b) ao longo do tempo

6.3 Controlo de presença

A iluminação de uma divisão de um edifício pode ser controlada através de sensores de movimento ligados a nós de uma rede de Bluetooth Mesh. Neste cenário a simular a aplicação numa divisão, nós servidores implementam controlo de iluminação de fontes luminosas a eles ligados, sendo esse controlo feito com base na deteção de movimento de um sensor de movimento ligado a um nó cliente.

Caso haja deteção de movimento, o nó cliente envia mensagens aos nós servidores com pedidos de ligação das respetivas fontes luminosas. Caso não haja deteção e as luzes nos nós servidores continuarem ligadas, o nó cliente envia pedidos para desligar as luzes. A Figura 6.5 ilustra as ligações entre os nós, sensor PIR e fontes de luz.

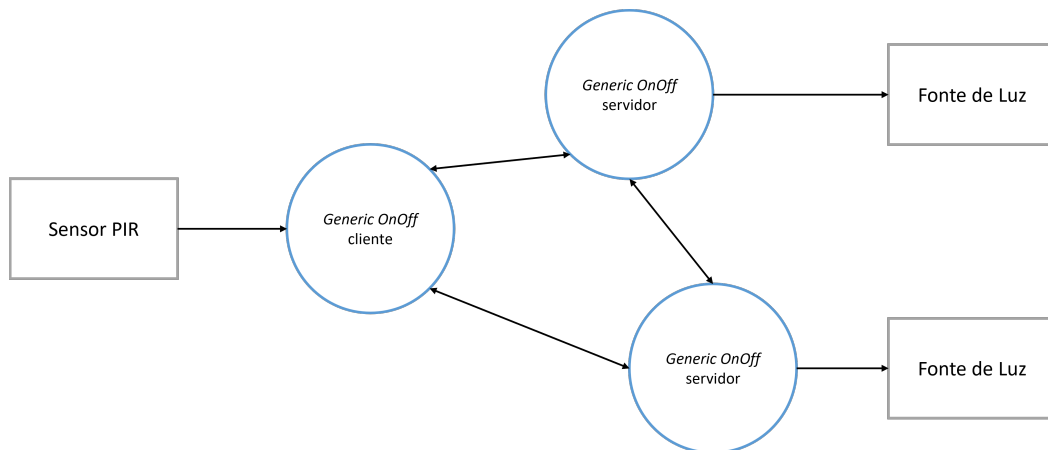


Figura 6.5: Ligações entre os nós para o controlo de presença

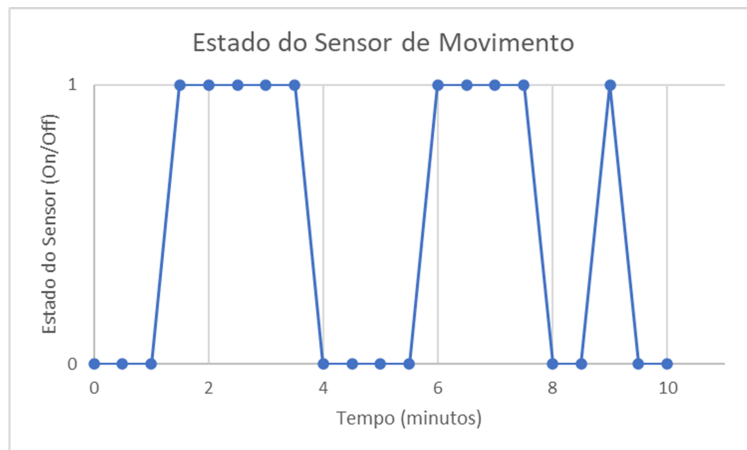
6.3.1 Casos de testes

O controlo de Presença também é utilizado para testar controlo e monitorização de fatores ambientais, com base numa rede de Bluetooth Mesh. Neste teste foi utilizada uma estratégia semelhante à descrita na secção 6.2. Neste caso, o valor de leitura do sensor de movimento PIR ligado ao nó com modelo cliente foi registado numa tabela de 30 em 30 s, juntamente com o estado da luz, ligada ao nó servidor.

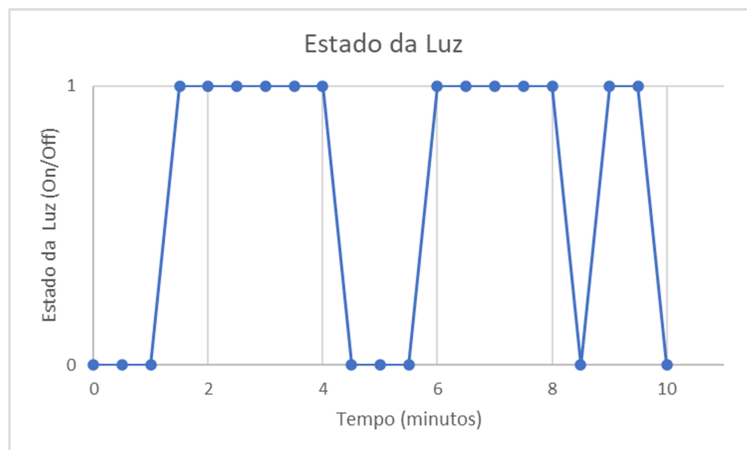
À semelhança do teste com o cenário de Controlo e Monitorização de temperatura, foram criados dois gráficos, que representam a variação dos estados lidos pelo PIR, e a variação do estado da fonte luminosa ao longo do tempo.

6.3.2 Resultados dos testes

Com a análise dos gráficos da Figura 6.6, é possível perceber a velocidade com que as luzes reagiram à deteção de movimento. Tal como foi implementado, estas não desligaram no momento em que deixou de ser detetado movimento, como pode ser visto no minuto 4 do teste, no qual a luz desligou segundos depois das leituras de estado baixo por parte do sensor PIR.



(a)



(b)

Figura 6.6: Resultados de Testes de Controlo de Presença com a evolução do Estado do Sensor PIR (a) e do Estado da Luz(b) ao longo do tempo

Capítulo 7

Conclusões

No desenvolvimento desta dissertação, os diferentes passos tomados permitiram chegar a conclusões concretas sobre o funcionamento de um sistema multiagente. Neste capítulo serão apresentadas as principais conclusões, relativamente ao trabalho efetuado e aos resultados gerados nos testes de desempenho, e serão propostas sugestões de melhoria futura.

7.1 Conclusões e Discussão

Na fase inicial do desenvolvimento desta dissertação foram definidos o objetivo principal e os seus objetivos intermédios, apresentados na subsecção 1.2.2. O trabalho e estudo iniciais focaram-se na análise do estado da arte relativo a sistemas multiagente, internet das coisas e edifícios inteligentes, para atingir o objetivo O_1 . Essa análise permitiu ganhar conhecimentos sobre os últimos desenvolvimentos dos três tópicos. Os resultados da análise do estado da arte estão registados no Capítulo 2. De forma a definir a topologia da comunicação, foi feito um estudo aprofundado das metodologias, formas de organização e de comunicação dos agentes num SMA (O_2), apresentados na secção 2.2.

Para a escolha das tecnologias de comunicação, componentes de *hardware* e ferramentas adequadas para o desenvolvimento da solução, foi feita uma análise das tecnologias e ferramentas associadas a sistemas multiagente e a internet das coisas (O_3). Os resultados da análise forneceram os dados necessários para escolher a tecnologia Bluetooth Mesh para a interação entre os agentes, bem como a opção do ESP32 como agente de *hardware* e a utilização da *framework* do fabricante do ESP32, o ESP-IDF. A explanação das diferentes tecnologias e ferramentas encontram-se no Capítulo 3. Com base nos conhecimentos adquiridos no estado da arte e análise de tecnologias e ferramentas, foi feito um planeamento de uma solução de um sistema multiagente capaz de controlar vários parâmetros num contexto de edifícios inteligentes (O_4), descrito no Capítulo 4. Utilizando a definição da solução feita no planeamento, procedeu-se à implementação do protótipo funcional (O_5), de acordo com a especificação dos diferentes passos no Capítulo 5.

Para testar o desempenho do protótipo funcional, num edifício inteligente, foram definidos três cenários de aplicação, no âmbito de monitorização e controlo de fatores ambientais. Para cada um dos cenários foram definidos casos de teste, necessários para o registo de diferentes métricas em diferentes condições ambientais. Partindo das definições dos cenários de aplicação, os testes foram executados e os seus resultados registados (O_6). O Capítulo 6 contém a descrição dos cenários e casos de testes, o registo dos seus resultados e a discussão dos mesmos. Os valores dos resultados comprovam que a rede de Bluetooth Mesh consegue entregar as mensagens transmitidas, sem perdas nos casos em que os nós estão dentro da mesma divisão e sem a necessidade de retransmissores. Em casos onde existem obstáculos pelo meio ou dispositivos em múltiplas divisões, o uso de retransmissores pode ser necessário, dependendo do tipo de obstáculos e constituição dos edifícios. Relativamente à resposta da rede *mesh* em contexto de controlo e monitorização em tempo real, dentro da mesma divisão, foi considerada adequada, dada a velocidade com que a rede *mesh* reagia à alterações do ambiente. Com esta dissertação foi concebida, implementada e testada uma solução de um sistema multiagente composto por sensores e atuadores distribuídos para a implementação em edifícios inteligentes, de acordo com a definição do objetivo principal na subsecção 1.2.2.

7.2 Fatores Limitadores

Durante a fase de desenvolvimento da aplicação de Bluetooth Mesh, foi evidente a existência de documentação incompleta ou mesmo a sua total inexistência, sobre algumas APIs essenciais para a implementação, causando alguns atrasos na compreensão do seu código fonte.

Na fase de testes, embora a rede *mesh* tenha sido bem sucedida a transmitir as mensagens na maior parte dos casos, existiu uma maior perda de mensagens quando as distâncias entre os nós eram maiores e/ou existia paredes no meio, baixando a fiabilidade desta tecnologia de comunicação. Assim, dentro de divisões o Bluetooth Mesh será adequado para estabelecer redes fiáveis, mas para a comunicação entre divisões será adequado interligar um dos nós com uma segunda tecnologia de comunicação como o Wi-Fi, isto é, no contexto de sistemas multiagente, ter um agente delegado em cada divisão.

7.3 Trabalho Futuro

Apesar da solução apresentar bons resultados, este sistema pode ser escalado no número de aplicações associadas e trabalhar em rede com outras soluções de sistemas de edifícios inteligentes, de forma a tirar melhor partido do seu potencial. Assim sugiram-se algumas ideias de melhorias:

- Integração desta solução com um *Broker MQTT*, para uma utilização mais ativa do nó de Gateway e para a possibilidade de interligação com a internet;
- Adição de uma interface *web*, na qual são feitos cenários e pedidos individuais de leitura e escrita de fatores ambientais a monitorizar e a controlar;
- Otimizar o código de aplicação dos nós de Bluetooth Mesh, nomeadamente na criação de um método geral de envio de mensagens, uma vez que existem vários métodos que fazem o mesmo mas para diferentes códigos de operação;
- Substituir o modelo Generic OnOff por modelos personalizados para o envio de mensagens da aplicação, para que a quantidade de dados enviada numa mensagem, possa ser maior;
- Realização de mais testes com maior número de nós na rede.

Referências

- [Airedale, 2022] Airedale (2022). ACIS™ Building Energy Management System. Disponível em <https://www.airedale.com/products/acis-bms/>. (Last accessed in 26/04/2022).
- [Atzori et al., 2010] Atzori, L., Iera, A., and Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15):2787–2805.
- [Baert et al., 2018] Baert, M., Rossey, J., Shahid, A., and Hoebeke, J. (2018). The bluetooth mesh standard: An overview and experimental evaluation. *Sensors (Switzerland)*, 18(8).
- [Bashir and Gill, 2017] Bashir, M. R. and Gill, A. Q. (2017). Towards an iot big data analytics framework: Smart buildings systems.
- [Bembe et al., 2019] Bembe, M., Abu-Mahfouz, A., Masonta, M., and Ngqondi, T. (2019). A survey on low-power wide area networks for iot applications.
- [Bluetooth Special Interest Group (SIG), 2019] Bluetooth Special Interest Group (SIG) (2019). *Mesh Profile - Bluetooth® Specification*. Kirkland, WA, USA.
- [Bluetooth Special Interest Group (SIG), 2020] Bluetooth Special Interest Group (SIG) (2020). *Bluetooth Mesh Networking*. Kirkland, WA, USA.
- [Bluetooth Special Interest Group (SIG), 2021] Bluetooth Special Interest Group (SIG) (2021). *Bluetooth Core Specification*. Kirkland, WA, USA.
- [Bluetooth Special Interest Group (SIG), 2022a] Bluetooth Special Interest Group (SIG) (2022a). Bluetooth Technology Overview. Available at <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>. (Last accessed in 19/06/2022).
- [Bluetooth Special Interest Group (SIG), 2022b] Bluetooth Special Interest Group (SIG) (2022b). *The Bluetooth® Low Energy Primer*. Kirkland, WA, USA.
- [Bonomi et al., 2012] Bonomi, F., Milito, R., Zhu, J., and Addepalli, S. (2012). Fog computing and its role in the internet of things. In *MCC'12 - Proceedings of the 1st ACM Mobile Cloud Computing Workshop*.

- [Cai et al., 2016] Cai, J., Kim, D., Jaramillo, R., Braun, J. E., and Hu, J. (2016). A general multi-agent control approach for building energy system optimization. *Energy and Buildings*.
- [Chen et al., 2020] Chen, L., Chen, P., and Lin, Z. (2020). Artificial intelligence in education: A review. *IEEE Access*, 8.
- [Chettri and Bera, 2020] Chettri, L. and Bera, R. (2020). A Comprehensive Survey on Internet of Things (IoT) Toward 5G Wireless Systems.
- [Chiang and Zhang, 2016] Chiang, M. and Zhang, T. (2016). Fog and IoT: An Overview of Research Opportunities.
- [Chin et al., 2019] Chin, J., Callaghan, V., and Ben Allouch, S. (2019). The Internet-of-Things: Reflections on the past, present and future from a user-centered and smart environment perspective. *Journal of Ambient Intelligence and Smart Environments*, 11(1).
- [Chira and Dumitrescu, 2006] Chira, C. and Dumitrescu, D. (2006). Multi-Agent Systems for Distributed Collaboration. 3.
- [D’Elia et al., 2010] D’Elia, A., Roffia, L., Zamagni, G., Vergari, F., Toninelli, A., and Bellavista, P. (2010). Smart applications for the maintenance of large buildings: How to achieve ontology-based interoperability at the information level. page 1077–1082.
- [Deng et al., 2011] Deng, H., Zhang, X., and Liu, J. (2011). Intelligence learning model of multi-agent system.
- [Dorri et al., 2018] Dorri, A., Kanhere, S. S., and Jurdak, R. (2018). Multi-Agent Systems: A Survey. *IEEE Access*.
- [Dorsemayne et al., 2016] Dorsemayne, B., Gaulier, J. P., Wary, J. P., Kheir, N., and Urien, P. (2016). Internet of Things: A Definition and Taxonomy. In *Proceedings - NGMAST 2015: The 9th International Conference on Next Generation Mobile Applications, Services and Technologies*.
- [Dounis, 2010] Dounis, A. I. (2010). Artificial intelligence for energy conservation in buildings. *Advances in Building Energy Research*, 4.
- [Dragomir, 2014] Dragomir, E. G. (2014). Development of a multi-agent-based simulation system for air quality analysis. *Studies in Informatics and Control*, 23.

- [Durfee et al., 1991] Durfee, E., Lesser, V., and Corkill, D. (1991). Distributed Problem Solving. *The Encyclopedia of Artificial Intelligence, Second Edition*.
- [Espressif, 2019] Espressif (2019). Esp-idf programming guide. Available at <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html>. (Last accessed in 08/10/2022).
- [Fadda et al., 2019] Fadda, G., Fadda, M., Ghiani, E., and Pilloni, V. (2019). Communications and internet of things for microgrids, smart buildings, and homes. In *Distributed Energy Resources in Microgrids: Integration, Challenges and Optimization*, pages 243–273.
- [Farzaneh et al., 2021] Farzaneh, H., Malehmirchegini, L., Bejan, A., Afolabi, T., Mulumba, A., and Daka, P. P. (2021). Artificial intelligence evolution in smart buildings for energy efficiency.
- [Fearn, 2017] Fearn, N. (2017). Survey: Facilities managers look to iot for building performance boost. Disponível em <https://internetofbusiness.com/facilities-managers-iot-building-performance/>. (Last accessed in 27/04/2022).
- [Froufe et al., 2020] Froufe, M. M., Chinelli, C. K., Guedes, A. L. A., Haddad, A. N., Hammad, A. W., and Soares, C. A. P. (2020). Smart buildings: Systems and drivers. *Buildings*, 10(9).
- [Gomes et al., 2020] Gomes, L., Vale, Z. A., and Corchado, J. M. (2020). Multi-agent microgrid management system for single-board computers: A case study on peer-to-peer energy trading. *IEEE Access*, 8.
- [Horling and Lesser, 2004] Horling, B. and Lesser, V. (2004). A survey of multi-agent organizational paradigms. *Knowledge Engineering Review*, 19(4).
- [Hortelano et al., 2021] Hortelano, D., Olivares, T., and Ruiz, M. C. (2021). Providing interoperability in bluetooth mesh with an improved provisioning protocol. *Wireless Networks*, 27.
- [Huerta et al., 2020] Huerta, U., Baltazar, R., Pineda, A., Rocha, M., and Casillas, M. (2020). Data analysis of sensors in smart homes for applications healthcare in elderly people. volume 186.
- [i-SCOOP, 2017] i-SCOOP (2017). Smart Buildings and Facilities: Smart data, Insights and Integration. Disponível em <https://www.i-scoop.eu/internet-of-things-iot/facility-management-iot-smart-buildings/>.

(Last accessed in 27/04/2022).

- [Jennings, 1996] Jennings, N. (1996). Coordination techniques for distributed artificial intelligence.
- [Jia et al., 2019] Jia, M., Komeily, A., Wang, Y., and Srinivasan, R. S. (2019). Adopting Internet of Things for the development of smart buildings: A review of enabling technologies and applications. *Automation in Construction*, 101.
- [Kelly et al., 2013] Kelly, S. D. T., Suryadevara, N. K., and Mukhopadhyay, S. C. (2013). Towards the implementation of iot for environmental condition monitoring in homes. *IEEE Sensors Journal*, 13.
- [Kim and Ahn, 2013] Kim, B. Y. and Ahn, H. S. (2013). Coordination and control for building automation systems.
- [Lin et al., 2017] Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., and Zhao, W. (2017). A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet of Things Journal*, 4(5).
- [Lonzetta et al., 2018] Lonzetta, A. M., Cope, P., Campbell, J., Mohd, B. J., and Hayajneh, T. (2018). Security vulnerabilities in bluetooth technology as used in iot.
- [Ma et al., 2019] Ma, Z., Schultz, M. J., Christensen, K., Værbak, M., Demazeau, Y., and Jørgensen, B. N. (2019). The application of ontologies in multi-agent systems in the energy sector: A scoping review. *Energies*, 12.
- [Malone and Crowston, 2003] Malone, T. and Crowston, K. (2003). Toward an Interdisciplinary Theory of Coordination. *Massachusetts Institute of Technology (MIT), Sloan School of Management, Working papers*.
- [Mannino et al., 2021] Mannino, A., Dejacco, M. C., and Cecconi, F. R. (2021). Building information modelling and internet of things integration for facility management-literature review and future needs.
- [Mocanu et al., 2016] Mocanu, E., Nguyen, P. H., Gibescu, M., and Kling, W. L. (2016). Deep learning for estimating building energy consumption. *Sustainable Energy, Grids and Networks*, 6.
- [Mohanty et al., 2016] Mohanty, S. P., Choppali, U., and Kougiianos, E. (2016). Everything you wanted to know about smart cities: The Internet of things is the backbone. *IEEE Consumer Electronics Magazine*, 5(3).

- [Morais et al., 2014] Morais, H., Sousa, T., Vale, Z., and Faria, P. (2014). Evaluation of the electric vehicle impact in the power demand curve in a smart grid environment. *Energy Conversion and Management*, 82.
- [Omoniwa et al., 2019] Omoniwa, B., Hussain, R., Javed, M. A., Bouk, S. H., and Malik, S. A. (2019). Fog/edge computing-based IoT (FECIoT): Architecture, applications, and research issues. *IEEE Internet of Things Journal*, 6(3).
- [Pakdeetrakulwong et al., 2016] Pakdeetrakulwong, U., Wongthongtham, P., Siri-charoen, W. V., and Khan, N. (2016). An ontology-based multi-agent system for active software engineering ontology. *Mobile Networks and Applications*, 21.
- [Pan et al., 2015] Pan, J., Jain, R., Paul, S., Vu, T., Saifullah, A., and Sha, M. (2015). An internet of things framework for smart energy in buildings: Designs, prototype, and experiments. *IEEE Internet of Things Journal*, 2.
- [Panchalingam and Chan, 2019] Panchalingam, R. and Chan, K. C. (2019). A state-of-the-art review on artificial intelligence for Smart Buildings.
- [Pi, 2022] Pi, R. (2022). Raspberry pi zero 2 w. Available at <https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/>. (Last accessed in 27/08/2022).
- [Pradeep et al., 2019] Pradeep, G., Chandra Shaker, P., and Prasad, S. V. (2019). Building management system. *International Journal of Innovative Technology and Exploring Engineering*, 8(11).
- [Qolomany et al., 2019] Qolomany, B., Al-Fuqaha, A., Gupta, A., Benhaddou, D., Alwajidi, S., Qadir, J., and Fong, A. C. (2019). Leveraging Machine Learning and Big Data for Smart Buildings: A Comprehensive Survey. *IEEE Access*, 7.
- [Reis and Oliveira, 2003] Reis, L. P. and Oliveira, E. C. (2003). Coordenação em sistemas multi-agente: Aplicações na gestão universitária e futebol robótico.
- [Rodrigues et al., 2018] Rodrigues, J. J., De Rezende Segundo, D. B., Junqueira, H. A., Sabino, M. H., Prince, R. M. I., Al-Muhtadi, J., and De Albuquerque, V. H. C. (2018). Enabling Technologies for the Internet of Health Things. *IEEE Access*, 6.
- [Rondon et al., 2020] Rondon, R., Mahmood, A., Grimaldi, S., and Gidlund, M. (2020). Understanding the performance of bluetooth mesh: Reliability, delay, and scalability analysis. *IEEE Internet of Things Journal*, 7.

- [Russel and Norvig, 2012] Russel, S. and Norvig, P. (2012). *Artificial intelligence—a modern approach 3rd Edition*.
- [Santur et al., 2017] Santur, Y., Karaköse, E., Karaköse, M., and Akın, E. (2017). Deep learning based artificial manager for smart city.
- [Sethi and Sarangi, 2017] Sethi, P. and Sarangi, S. R. (2017). Internet of Things: Architectures, Protocols, and Applications.
- [Shafique et al., 2020] Shafique, K., Khawaja, B. A., Sabir, F., Qazi, S., and Mustaqim, M. (2020). Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT Scenarios.
- [Shaikh et al., 2016] Shaikh, P. H., Nor, N. B. M., Nallagownden, P., Elamvazuthi, I., and Ibrahim, T. (2016). Intelligent multi-objective control and management for smart energy efficient buildings. *International Journal of Electrical Power and Energy Systems*, 74.
- [Silva et al., 2018a] Silva, B. N., Khan, M., and Han, K. (2018a). Internet of Things: A Comprehensive Review of Enabling Technologies, Architecture, and Challenges.
- [Silva et al., 2018b] Silva, B. N., Khan, M., and Han, K. (2018b). Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities.
- [Sousa et al., 2012] Sousa, T., Morais, H., Vale, Z., Faria, P., and Soares, J. (2012). Intelligent energy resource management considering vehicle-to-grid: A simulated annealing approach. *IEEE Transactions on Smart Grid*, 3.
- [Sovacool and Furszyfer Del Rio, 2020] Sovacool, B. K. and Furszyfer Del Rio, D. D. (2020). Smart Home Technologies in Europe: A Critical Review of Concepts, Benefits, Risks and Policies.
- [Srinivasan et al., 2017] Srinivasan, R., Islam, M. T., Islam, B., Wang, Z., Sookoor, T., Gnawali, O., and Nirjon, S. (2017). Preventive maintenance of centralized hvac systems: Use of acoustic sensors, feature extraction, and unsupervised learning. volume 3.
- [Systems, 2021] Systems, E. (2021). Esp32 series datasheet. (Last accessed in 26/08/2022).

- [Terán et al., 2014] Terán, J., Aguilar, J. L., and Cerrada, M. (2014). Collective Learning in Multi-Agent Systems Based on Cultural Algorithms. *CLEI Electronic Journal*, 17(2).
- [TransformaInsights, 2020] TransformaInsights (2020). Global IoT market will grow to 24.1 billion devices in 2030, generating \$1.5 trillion annual revenue - Transforma Insights. Disponível em <https://transformainsights.com/news/iot-market-24-billion-usd15-trillion-revenue-2030>. (Last accessed in 01/05/2022).
- [Viswanath et al., 2016] Viswanath, S. K., Yuen, C., Tushar, W., Li, W. T., Wen, C. K., Hu, K., Chen, C., and Liu, X. (2016). System design of the internet of things for residential smart grid. *IEEE Wireless Communications*, 23.
- [Wang et al., 2010] Wang, Z., Yang, R., and Wang, L. (2010). Multi-agent control system with intelligent optimization for smart and energy-efficient buildings.
- [Wei and Li, 2011] Wei, C. and Li, Y. (2011). Design of energy consumption monitoring and energy-saving management system of intelligent building based on the internet of things.
- [Wei et al., 2020] Wei, K., Zhang, L., Guo, Y., and Jiang, X. (2020). Health Monitoring Based on Internet of Medical Things: Architecture, Enabling Technologies, and Applications. *IEEE Access*, 8.
- [Whitmore et al., 2015] Whitmore, A., Agarwal, A., and Da Xu, L. (2015). The Internet of Things—A survey of topics and trends. *Information Systems Frontiers*, 17(2).
- [Wooldridge and Jennings, 1995] Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2).
- [Yang and Wang, 2012] Yang, R. and Wang, L. (2012). Multi-agent based energy and comfort management in a building environment considering behaviors of occupants.
- [Zanella et al., 2014] Zanella, A., Bui, N., Castellani, A., Vangelista, L., and Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet of Things Journal*, 1(1).
- [Zeadally et al., 2019] Zeadally, S., Siddiqui, F., and Baig, Z. (2019). 25 years of bluetooth technology. *Future Internet*, 11.