



Instituto Universitário de Lisboa

Departamento de Ciências e Tecnologias da Informação

**Sistema de Gestão de Rega
Inteligente Suportado em IoT de
Baixo Custo**

Henrique Braga Ferreira

Dissertação submetida como requisito parcial para obtenção do grau
de mestre em

Engenharia de Telecomunicações e Informática

Orientador

Prof. Dr. Pedro Sebastião, Professor Auxiliar
ISCTE-IUL

Setembro 2018

"Work Hard, Play Harder"

Gretchen Wilson

Resumo

Na atualidade a Internet das Coisas faz parte do cotidiano de todos nós, havendo cada vez mais dispositivos conectados entre si. Face aos rápidos progressos nesta área, aquando da realização do trabalho de investigação e desenvolvimento apresentado nesta dissertação, constatou-se um enorme conjunto de trabalho associado a este tema, em particular relacionados com a aplicação de redes de sensores sem fios a diferentes projetos de IoT.

Porém, verificámos a ausência de sistema tão complexo como o desenvolvido, que consiste numa rede de sensores sem fios descentralizada, flexível, de forma modular e adaptável. Esta é capaz de ser implementada em diversos sistemas de IoT de médias e grandes dimensões. Isto tornou-se possível porque a comunicação entre os nós integrantes desta, é constituída por um módulo rádio que suporta a tecnologia LoRa. A rede desenvolvida tem como seu objetivo principal monitorizar elementos do ambiente onde está implementada, como a temperatura, humidade, luminosidade, entre outros, e atuar sobre o processo de controlo de modo a aumentar a eficiência do sistema em questão, conseguindo ainda aumentar a área de cobertura do mesmo.

Esta rede foi aplicada a um sistema de rega já existente, que foi substancialmente melhorado por via das alterações introduzidas. Alterações estas que permitem a rega individualizada, sem intervenção humana, das diferentes zonas do jardim de acordo com as suas características específicas, diminuindo assim os gastos desnecessários de água utilizada na rega e consequentemente a redução da pegada hídrica.

Palavras-chave: Internet das coisas, Rede de sensores sem fios, Arduino, ESP8266, Sistema de Rega, LoRa.

Abstract

Nowadays the Internet of Things is part of everyday life of all of us, with more and more devices connected to each other. In light of the progress made in this area, the research and development work presented in this dissertation revealed a huge set of work associated with this topic, in particular related to the application of wireless sensor networks to different IoT projects.

However, we note that there is no such complex system as the one developed, which is a decentralized, flexible and adaptable WSN. Which is capable of being implemented in several medium and large IoT systems. This has become possible because the communication between the nodes constituents of this, is carried out by a radio module that supports the LoRa technology. The main aim of the developed network is to monitor elements of the environment where it is implemented, such as temperature, humidity, luminosity, among others, and to act on the control processto increase the efficiency of the system in question, coverage.

This network was applied to an existing irrigation system, which has been substantially improved by the changes introduced. These changes allow individual watering without human intervention of the different areas of the garden according to their specific characteristics, thus reducing the unnecessary expenditure of water used in irrigation and consequently the reduction of the water footprint.

Keywords:Internet of Things, Wireless Sensor Network, Arduino, ESP8266, Irrigation System, LoRa;

Agradecimentos

O sucesso desta dissertação não seria possível sem o apoio e orientação de algumas pessoas e instituições, aos quais dirijo os meus sinceros agradecimentos. Em particular agradeço:

Ao Professor Pedro Sebastião, meu orientador, por todo o apoio prestado em termos de críticas, reflexões e ajudas concedidos ao longo de todo projeto desenvolvido, bem como pela sua paciência e disponibilidade constantes;

À Genviot, por me deixar realizar a dissertação no desenvolvimento de um produto inserido no âmbito da empresa e, em especial, ao seu cofundador, André Glória, por toda a sua ajuda, tanto na cedência antecipada de algum material bem como na partilha de conhecimentos. Este apoio serviu de alavanca à iniciação de todo o projeto;

Ao Instituto de Telecomunicação, pela ajuda financeira prestada na compra de material necessário a este projeto, sem a qual este nunca teria sido viável;

Aos meus amigos, tanto do ISCTE como fora do ISCTE, que de uma forma ou de outra me ajudaram, não só neste último ano, mas também durante todo o tempo do curso.

Dirijo um especial obrigado à minha namorada Sara, por toda a paciência que teve comigo mesmo quando as coisas não corriam como esperado e por ser uma fonte de força e esperança inesgotável.

Por último, quero agradecer aos meus pais, Gina e Mauro, pela educação, por sempre acreditarem em mim, nos bons e nos momentos menos bons, não só durante estes 5 anos, mas durante toda a minha vida.

A todos, muito obrigado!

Conteúdo

Resumo	v
Abstract	vii
Agradecimentos	ix
Lista de Figuras	xv
Abreviaturas	xvii
1 Introdução	1
1.1 Objetivos	1
1.2 Contribuições	3
1.3 Estrutura da Dissertação	3
2 Revisão de Literatura	5
2.1 Internet das Coisas	5
2.2 Protocolos de Comunicação	7
2.2.1 Internet	7
2.2.1.1 Message Queuing Telemetry Transport	7
2.2.2 Protocolos de Comunicação Sem Fios	9
2.2.2.1 Tecnologia Bluetooth	9
2.2.2.2 Tecnologia ZigBee	10
2.2.2.3 Tecnologia Wi-Fi	10
2.2.2.4 Tecnologia LoRa	10
2.2.3 Observações	11
2.3 Rede de Sensores sem Fios	12
2.4 Plataformas de Controlo	13
2.4.1 Plataforma Arduino	13
2.4.2 Módulo Wi-Fi ESP8266	14
2.4.3 Módulo Rádio RFM69HCW	15
2.4.4 Principais módulos usados a implementação	17
2.5 Sensores	17
2.5.1 Sensores de Humidade e Temperatura	18
2.5.1.1 Sensor DHT22	18
2.5.1.2 Sensor DS18B20	18

2.5.1.3	Sensor LM35	18
2.5.1.4	Sensor TMP36	20
2.5.2	Sensores de Humidade do Solo	21
2.5.2.1	Sensor de Humidade do Solo	21
2.5.2.2	Sensor de água de alta sensibilidade	21
2.5.3	Sensores de Luminosidade	21
2.5.3.1	Sensor de luz LDR	21
2.5.3.2	Foto Transístor ALS-PT19	22
2.5.4	Sensores a implementar nos nós da rede desenvolvida	22
2.6	Trabalhos Relacionados	23
2.6.1	Características inovadoras da rede a implementar	24
3	Arquitetura do Sistema	27
3.1	Arquitetura	27
3.2	Fluxogramas	30
3.2.1	Fluxograma relativo ao nó dos sensores	30
3.2.2	Fluxograma relativo ao nó do atuador	31
3.2.3	Fluxograma relativo ao <i>Gateway</i>	32
3.3	Diagramas Temporais de Mensagens	34
3.3.1	Diagrama temporal relativo ao Algoritmo de Troca de Mensagens	34
3.3.2	Diagrama temporal relativo à mensagem do tipo <i>Water</i>	37
3.3.3	Diagramas temporais relativos às mensagens do tipo <i>ConfirmToken</i> e <i>ConfirmBoard</i>	38
4	Implementação	41
4.1	Implementação do Sistema para Testes em Laboratório	42
4.2	Implementação do Sistema para Testes em Cenário Real	45
5	Resultados	51
5.1	Resultados da Implementação do Sistema para Testes em Cenário de Laboratório	51
5.2	Resultados da Implementação do Sistema para Testes em Cenário Real	53
5.3	Estudo da Potência de Receção do Sinal numa Ligação LoRa Ponto-a-Ponto	55
5.3.1	Arquitetura Utilizada para Medição da Potência de Receção	55
5.3.2	Resultados dos Teste da Potência de Receção do Sinal	56
5.3.3	Modelo de Propagação Empírico	59
6	Conclusões e Trabalho Futuro	61
6.1	Principais Conclusões	61
6.2	Trabalho Futuro	63
	Anexos	67

A Artigos Científicos Publicados	67
Bibliografia	77

Lista de Figuras

2.1	Elementos constituintes do IoT	6
2.2	Processo de Publish/Subscribe do MQTT	8
2.3	Topologia LoRaWAN	11
2.4	Nós constituintes de uma Rede de Sensores	12
2.5	Arduino Uno	14
2.6	Componentes ESP8266: a) chip ESP-12; b) Adafruit Huzzah ESP8266 breakout	15
2.7	Módulo rádio RFM69HCW	16
2.8	Temperatura em função da voltagem (Sensor LM35)	19
2.9	Temperatura em função da tensão (Sensor TMP36)	20
2.10	Resistência em função da luz incidente(ALS-PT19)	22
3.1	Arquitetura do Sistema proposto	28
3.2	Diagrama de Blocos Gateway	29
3.3	Diagrama de Blocos Atuador	29
3.4	Diagrama de Blocos Sensor	30
3.5	Fluxograma nó de Sensores	31
3.6	Fluxograma nó do Atuador	32
3.7	Fluxograma nó Gateway	33
3.8	Exemplo Algoritmo de Troca de Mensagens	36
3.9	Diagrama temporal relativo à mensagem do tipo <i>Water</i>	38
3.10	Diagramas temporais relativos às mensagens do tipo <i>ConfirmToken</i> e <i>ConfirmBoard</i>	39
4.1	Implementação da WSN em ambiente de Laboratório	42
4.2	Prototipo nó do Sensor	43
4.3	Prototipo nó do Atuador	44
4.4	Prototipo <i>Gateway</i>	44
4.5	Imagem do Jardim (Satélite)	46
4.6	Imagem do Jardim (Fotografia Real)	46
4.7	Prototipo do nó do sensor (Zona 1)	47
4.8	Prototipo do nó do sensor (Zona 2)	47
4.9	Prototipo do nó do atuador (Zona 3)	48
4.10	Prototipo do <i>Gateway (Indoor)</i>	49
5.1	Prototipo nó do Atuador (Simulação de Rega)	52

5.2	Diagrama de barras alusivo ao número de mensagens trocadas pelo <i>gateway</i> em ambiente de laboratório	53
5.3	Diagrama de barras alusivo ao número de mensagens trocadas pelo <i>gateway</i> em ambiente real	54
5.4	Cenários de teste 1, 2, 3 e 4 respectivamente	55
5.5	Arquitetura utilizada para medição da potência de recepção	56
5.6	Potência recebida em função da distância (cenário 1)	57
5.7	Potência recebida em função da distância (cenário 2)	58
5.8	Potência recebida em função da distância (cenário 3)	58
5.9	Potência recebida em função da distância (cenário 4)	59
5.10	Modelos de propagação empírico (<i>Indoor e Outdoor</i>)	60

Abreviaturas

ACK	A cknowledgement
API	A pplication P rogramming I nterface
BLE	B luetooth L ow E nergy
CSS	C hirp S pread S pectrum
FIFO	F irst I n F irst O ut
FSK	F requency - S hift K eying
FTDI	F uture T echnology D evelopments I nternational
HTTP	H ypertext T ransfer P rotocol
I/O	I nterface O utput
IDE	I ntegrated D evelopment E nvironment
IEEE	I nstitute of E lectrical and E lectronics E ngineers
IOT	I nternet of T hings
ISM	I ndustrial S cientific and M edical
LAN	L ocal A rea N etwork
LDR	L ight D ependent R esistor
LoRa	L ong R ange
LR-WPAN	L ow R ate- W ireless P ersonal A rea N etwork
M2M	M achine-to- M achine
MQTT	M essage Q ueuing T elemetry T ransport
QoS	Q uality of S ervice
SD	S ecure D igital
SoC	S ystem on C hip
SPI	S erial P eripheral I nterface
TCP	T ransmission C ontrol P rotocol
USB	U niversal S erial B us
Wi-Fi	W ireless F idelity
WLAN	W ireless L ocal A rea N etworks
WPAN	W ireless P ersonal A rea N etworks

WSN **Wireless Sensor Network**

Capítulo 1

Introdução

A genviot, empresa dedicada à criação de soluções tecnológicas para a sustentabilidade e poupança de recursos naturais e materiais, desenvolveu uma solução para sistemas de rega inteligentes, o genviotGarden, que quando aplicado a um sistema de rega comum é capaz de reduzir os consumos de água e de energia em cerca de 30%. Apesar do genviotGarden recolher online os dados meteorológicos (temperatura e humidade local) e receber informação de dois sensores ligados ao controlador, não atingiu um nível de precisão necessário à otimização da eficiência da rega.

A fim de atingir este desiderato propomo-nos desenvolver uma solução descentralizada de baixo custo e máxima eficiência, baseada em redes de sensores wireless, mediante a divisão do jardim em zonas de área reduzida, individualizando-as de acordo com as suas características específicas (vg. zonas de sombra, tipos de vegetação, declive, etc.), de modo a que a rega seja efetuada na exata medida da necessidade de irrigação da vegetação de cada uma das referidas zonas.

1.1 Objetivos

Para atingir o objetivo proposto efetuámos previamente um estudo comparativo dos tipos de sensores existentes no mercado, procedemos à integração dos mesmos no processo de rega, desenvolvendo uma rede de sensores sem fios (*Wireless Sensor Network*), utilizada para recolha de dados na área de atuação de cada sensor.

- **Estudo comparativo de vários tipos de sensores**

Para obter a máxima eficiência, o sistema necessita de recolher dados o mais precisos possível, tanto da temperatura e humidade ambiente, como da humidade do solo e luminosidade. Assim, é importante perceber quais são os sensores existentes no mercado, que podem ser incluídos no sistema, tendo sempre em mente o facto de se tratar de um sistema de baixo custo.

- **Integração dos dados no processo de rega**

O sistema na sua versão primitiva consegue automaticamente saber quais as necessidades específicas de rega para a grande área do jardim onde está a atuar, usando as fórmulas apresentadas de seguida, as quais permitem aferir-se, tendo em conta o tipo de expressores, área e condições meteorológicas, o tempo de rega do jardim.

A versão que apresentamos, na medida em que os dados recolhidos dizem respeito a mais áreas de dimensão reduzida e características específicas e individualizadas, permite ajustar o modelo de rega, corrigindo os dados recolhidos “online” com as informações recolhidas pelos sensores em cada zona, percebendo-se se os parâmetros recolhidos de temperatura e humidade ambiente e do solo, nas diversas áreas do jardim, estão dentro dos valores coletados anteriormente e caso isso não se verifique, ajustar a fórmula às necessidades de rega de cada zona em particular, transmitindo ao sistema o tempo exato de rega adequado às características de cada região.

Este processo permite criar um algoritmo capaz de analisar a mensagem do sistema relativa à rega, recolher os dados das zonas respetivas, analisar esses dados e perceber se existe ou não a necessidade de irrigação do solo, o que permite manter as plantas saudáveis, e ao mesmo tempo reduzir a quantidade de água gasta.

- **Desenvolvimento da rede de sensores sem fios**

Para se desenvolver a rede de sensores sem fios, em primeiro lugar, deve proceder-se ao estudo de diversos métodos de comunicação, e eleger aquele com mais potencial ao nível de alcance, número de nós, ritmo binário, hardware associado, preço e analisar o desempenho do mesmo no sistema a desenvolver. Segue-se o desenho e implementação de cada um dos nós constituintes da rede, ou seja, dos nós dos sensores, do nó do atuador e do *gateway*.

Por último, proceder-se-á à integração da rede desenvolvida com o sistema existente, genviotGarden, garantindo que todos os dados são recolhidos, sem perdas de pacotes.

1.2 Contribuições

Os resultados obtidos no desenvolvimento deste projeto resultam num artigo científico desenvolvido. Um deles foi admitido em uma conferencia internacional, em Londres, WorldS4 2018. A publicação deste artigo contribui para a sua disseminação na comunidade científica.

- H. Ferreira, A. Glória, and P. Sebastião, “LoRa peer-to-peer WSN architecture for large-scale IoT projects”, World Conference On Smart Trends in Systems, Security Sustainability 2018, WorldS4 2018.

Para além das contribuições científicas, esta dissertação também contribuiu para uma inovação tecnologia e transferência de tecnologia, na construção da rede de sensores sem fios bem como nos protótipos de cada nó constituintes da mesma.

A apresentação do primeiro artigo vai se realizar em Novembro de 2018 em Londres por um dos autores.

1.3 Estrutura da Dissertação

Esta dissertação é constituída por 6 capítulos. No primeiro capítulo é apresentada a importância e finalidade desta dissertação bem como a motivação que levaram à escolha deste tema. No segundo é feita uma abordagem teórica das tecnologias e material que podem ser utilizadas para a construção de uma rede deste tipo. No capítulo seguinte é apresentada a arquitetura do sistema, bem como a reação do mesmo às diferentes instruções do utilizador através de diagramas temporais de mensagens e fluxogramas. No capítulo quatro são amostrados dois tipos de implementação do sistema, tanto num ambiente controlado como num ambiente real. O capítulo cinco são os resultados e finalmente no último capítulo discutimos o resultado geral da dissertação e como este pode ser melhorado.

Capítulo 2

Revisão de Literatura

A Internet das Coisas ou em inglês “*Internet of Things*”, adiante designada por IoT , está cada vez mais presente no nosso quotidiano, constituindo a otimização da sua eficácia uma demanda constante, necessária à expansão da sua utilização. O desafio a que nos propomos é o de desenhar e implementar uma WSN de baixo custo capaz de se adaptar a projetos de IoT que procuram expansão. Este capítulo está dividido em 5 partes, onde analisamos as várias abordagens já antes estudadas para criar este tipo de rede. Na primeira parte, começamos por compreender como o IoT está organizado. Na parte seguinte focamo-nos no estudo de dois tipos de protocolos, sendo um deles destinado à conexão da rede à Internet e o outro à comunicação dos diversos nós constituintes da mesma. Nas terceira e quarta partes serão efetuadas comparações entre os diferentes micro- controladores e sensores existentes no mercado e elegíveis à utilização neste projeto. Finalmente, a última parte refere-se a alguns projetos que de alguma maneira podem estar relacionados com o projeto final.

2.1 Internet das Coisas

A Internet é um dos desenvolvimentos mais importantes da humanidade e a Internet das coisas (IoT) representará a próxima evolução da Internet [1] [2]. Com capacidade de reunir, analisar e distribuir os dados, o IoT consiste na conexão entre a Internet e uma variedade de dispositivos e sensores. O IoT, como mostrado

na Figura 1, pode ser dividido em seis elementos [3] que nos ajudam a entender o seu verdadeiro significado e funcionalidade, ou seja, identificação, detecção, comunicação, computação, serviços e semântica.



FIGURA 2.1: Elementos constituintes do IoT [3]

A identificação é necessária no IoT para nomear e combinar serviços de acordo com sua necessidade. Os objetos do IoT precisam ser endereçados, sendo por isso fundamental conhecer a diferença entre ID do objeto, como "Hum1" para um sensor de humidade particular, e o seu endereço na rede. A detecção é a capacidade de recolher dados do meio ambiente e enviá-los para serem posteriormente analisados por um servidor. Os dados podem ser a temperatura, a humidade, pressão ou qualquer outro parâmetro que possa ser medido.

A comunicação conecta objetos a dados comerciais e executa serviços específicos. No IoT são comumente usados protocolos de comunicação sem fio, apesar de em algumas circunstâncias se recorrer a protocolos com fio.

A computação representa o cérebro e a capacidade computacional do IoT, utilizando o processamento de hardware [por exemplo, microcontroladores, microprocessadores, sistema de chips (SoCs)] e softwares capazes de executar tarefas específicas.

Os serviços no IoT estão divididos em quatro categorias, a saber: serviços relacionados com a identificação, que lançam o terreno para outros serviços, pois todos os objetos do mundo real precisam e ser identificados no mundo virtual; serviços de agregação de informações, que reúnem e resumem a informação bruta que precisa de ser processada; serviços colaborativos, que utilizam os dados obtidos para tomar decisões; e serviços omnipresentes, que oferecem serviços cientes de colaboração a qualquer pessoa sob demanda, a qualquer hora e em qualquer lugar.

A semântica é a capacidade de recolher conhecimento de forma inteligente para fornecer os serviços necessários, incluindo a descoberta de recursos, a utilização dos mesmos, a modelagem de informações e o reconhecimento e análise de dados.

Existem ainda algumas barreiras [2] que podem impedir o desenvolvimento rápido do IoT, como por exemplo a necessidade do uso de práticas comuns, especialmente nas áreas de segurança e comunicação, ou consumo energético, pois os sensores são dispositivos que necessitam de energia para funcionarem, e por vezes esta é um recurso inexistente na área onde um dado sistema de IoT possa estar implementado, sendo necessário criar essa energia a partir do meio ambiente, utilizando, por exemplo, a luz solar ou vento. Mas dado o potencial da IoT, é apenas uma questão de tempo para que esses problemas sejam resolvidos.

2.2 Protocolos de Comunicação

2.2.1 Internet

Para que um projeto de IoT seja eficiente é fundamental um bom protocolo de comunicação capaz de conectar o sistema em questão à Internet [4], possibilitando a transferência de dados entre os sensores e os servidores, onde estes serão armazenados e analisados. Como a maior parte dos servidores se encontram na *cloud*, a opção de utilizar um protocolo sem fios (*wireless*) é fundamental, dado que a sua principal utilidade é a de transferência de dados entre sensores e servidores.

Tendo presente o baixo tamanho dos pacotes, os protocolos escolhidos deverão ter um bom desempenho, baixo consumo, baixa largura de banda e pouca latência.

Apesar de serem dois os protocolos desenhados para o IoT, nomeadamente o *Message Queuing Telemetry Transport* e o *Constrained Application Protocol*, considerando que o presente sistema consiste num aperfeiçoamento do sistema *geniotGarden*, a escolha recaiu sobre o primeiro dos protocolos mencionados utilizado com eficiência neste.

2.2.1.1 Message Queuing Telemetry Transport

Message Queuing Telemetry Transport (MQTT) é um protocolo de mensagens [3] [5] criado em 1999 e padronizado em 2013 no OASIS, que visa a conexão de dispositivos e redes integrados com aplicativos e *middleware*. Construído em cima do protocolo TCP/IP, ele usa um padrão de publicação/assinatura, com

um mecanismo de encaminhamento (um-para-um, um-para-muitos, muitos-para-muitos), para proporcionar flexibilidade e simplicidade de transição fazendo do MQTT um protocolo de conexão ideal para o IoT e *Machine-to-Machine* (M2M) . Este é adequado para dispositivos pequenos, baratos, de baixa potência e de baixa memória com largura de banda reduzida. Comparado ao HTTP , o MQTT foi projetado para ter um *overhead* reduzido em comparação com o anterior.

O MQTT é composto por três componentes: *Publisher*, *Broker* e *Subscriber*. O *Broker* permite a subscrição de um tópico específico de interesse, em vista à obtenção da informação publicada em referência ao mesmo. O *Publisher* atua como um gerador de dados para alguns tópicos, transmitindo essa informação ao *Subscriber* através do intermediário, o *Broker*.

A Figura 2.2 ilustra o processo usado pelo MQTT no método de inscrição / publicação.

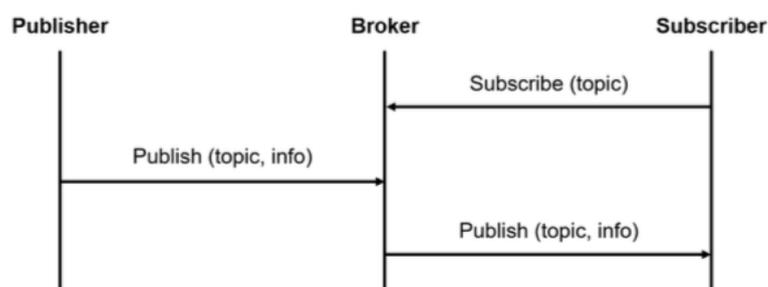


FIGURA 2.2: Processo de Publish/Subscribe do MQTT [6]

A fiabilidade no MQTT é garantida por três níveis de Qualidade de Serviço (QoS) [5]. No nível 0, a mensagem é entregue no máximo uma vez e nenhuma confirmação de recepção é necessária. No nível 1, a mensagem é entregue pelo menos uma vez e a confirmação da recepção da mensagem é necessária. No nível 2, um mecanismo de *handshake* de quatro vias é usado para a entrega de uma mensagem exatamente uma vez. Para fornecer a melhor comunicação entre clientes, o protocolo MQTT usa um conjunto de recursos [6], como sejam, o tempo *Keep Alive*, para evitar que conexões não utilizadas desperdicem dados; uma sessão duradoura, que armazena mensagens quando o *subscriber* se desconecta; e um alerta, enviado a todos os subscritores quando o *broker* sai.

2.2.2 Protocolos de Comunicação Sem Fios

Nesta secção é feita a análise teórica, com base em estudos já efetuados, sobre qual a tecnologia wireless (sem fios) que melhor se irá adaptar à proposta de dissertação. Será feita a comparação de tecnologias similares que são utilizadas por sensores e atuadores. Essa comparação irá recair maioritariamente sobre o ritmo binário de cada uma delas, frequência, alcance máximo que estas tecnologias atingem, quantos nós estas suportam, consumo energético que cada uma delas utiliza na troca de pacotes, vida útil da bateria e complexidade.

Os protocolos de comunicação sem fios oferecem tudo o que é necessário para a transferência de dados entre dispositivos sem a utilização de cabos. Os protocolos mais comuns sem fio são *Wireless-Fidelity (Wi-fi)*, *Bluetooth*, *ZigBee*, *Bluetooth Low Energy (BLE)* e *Long Range (LoRa)*.

Visto que o objetivo desta secção é descobrir o melhor protocolo de comunicação para projetos de IoT em larga escala, almeja-se encontrar um protocolo que seja de baixo custo, baixo consumo de energia e capaz de suportar comunicações entre vários nós. A tabela 2.1 contém algumas das principais características dos mesmos.

TABELA 2.1: Características dos principais protocolos de comunicação sem fio

Características	Wi-Fi	Bluetooth	ZigBee	LoRa
Ritmo Binário [Mbps]	11	1	0.25	0.11
Frequência [GHz]	2.4	2.4	2.4	0.868
Versão	802.11b	4.0	-	-
Alcance [m]	1-100	10-100	10-100	2000
Nós	32	7	65540	15000
Consumo [mA]	8.3-2.9	0.083-0.29	0.0083-0.83	0.0083-0.83
Complexidade	Alta	Média	Média	Baixa
Segurança	WPA/WPA2	128 bit	128 bit	128 bit

Na presente secção será ainda apresentada uma descrição detalhada sobre os protocolos mais relevantes à realização deste projeto, e por fim o protocolo sobre o qual recai a escolha, tendo em mente que o produto final será de baixo custo.

2.2.2.1 Tecnologia Bluetooth

Bluetooth é um protocolo de comunicação sem fios que utiliza o padrão IEEE 802.15.1, baseado num sistema de rádio que foi projetado para dispositivos de baixo custo e curto alcance, com o intuito de substituir os cabos, como por exemplo para impressoras, faxes, ratos para computador e teclados [7].

Este protocolo encontra-se na sua versão 4, BLE, que tem a vantagem de ter um reduzido consumo de energia e operar por longos períodos de tempo, o que o torna um ótimo candidato à escolha.

Outras duas vantagens prendem-se com o facto de um único coordenador ser capaz de controlar vários *slaves*, e de o protocolo ser capaz de garantir a auto-organização da rede aquando da desconexão de um dos nós [3].

2.2.2.2 Tecnologia ZigBee

ZigBee utiliza o padrão IEEE 802.15.4 e define especificações para WPAN de baixa taxa (LR-WPAN), que suporta dispositivos simples, que consomem pouca energia e normalmente operam a distâncias máximas de 10m. Este protocolo de comunicação a par do BLE permite redes auto-organizadas, com saltos múltiplos e fiáveis [8]. Apesar de os baixos consumos de energia do *Zigbee* tornarem este protocolo atrativo, o seu preço, superior aos outros protocolos referidos, tornam-no menos competitivo [4].

2.2.2.3 Tecnologia Wi-Fi

Wi-Fi utiliza o padrão IEEE 802.11 a/b/c para WLAN. Permite aos utilizadores acederem à Internet quando conectados a um ponto de acesso ou em modo *Ad-Hoc*. A arquitetura do IEEE 802.11 consiste na interação dos componentes de modo a fornecer LAN sem fio suportando a mobilidade da estação de forma fiável para as camadas superiores [8].

2.2.2.4 Tecnologia LoRa

LoRaWAN é um protocolo de comunicação bidirecional que usa a camada física LoRa para fornecer comunicações de longo alcance e de baixa potência. Para que isso seja possível, LoRa baseia-se na modulação CSS que possui as mesmas características de baixa potência da modulação FSK (presente em grande parte dos outros protocolos de comunicação sem fios), mas com um aumento significativo do alcance de comunicação. Com uma única estação base é possível cobrir até centenas de quilómetros quadrados [9].

O *LoRaWAN* usa uma topologia em estrela a viabilizar a sua comunicação de baixa potência, reduzindo a complexidade e aumentando a capacidade da rede, com o conseqüente aumento do tempo de vida. Isso é possível usando um *transceiver* (dispositivo que combina um transmissor e um recetor) e multi-modem no *gateway*, fornecendo assim a capacidade de receber mensagens em simultâneo [9].

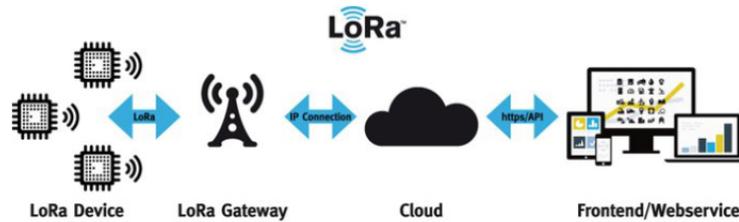


FIGURA 2.3: Topologia LoRaWAN [10]

Conforme apresentado na Figura 2.3, os nós não estão associados a um *gateway* específico. Como referido anteriormente, os dados são transmitidos para vários *gateways*, que, em seguida, encaminharão os pacotes recebidos do nó final para o servidor de rede, que irá então filtrar os pacotes redundantes, fazendo verificações de segurança e informando o *gateway* específico para enviar a confirmação [9].

Em resumo, este protocolo de comunicação tem mais vantagens do que desvantagens, sendo aquelas as relativas ao baixo consumo de energia, comunicações a longas distâncias e uma boa segurança com encriptação AES de 128-bits. A desvantagem prende-se apenas com o baixo tamanho de pacotes de apenas 55 bytes por mensagem.

Pelo que acima ficou exposto, este protocolo assume-se como um dos ideais para IoT [9].

2.2.3 Observações

Para desenhar e implementar a rede de sensores sem fios, dois tipos de comunicações serão utilizadas. A comunicação entre a rede e o servidor, e a comunicação entre os nós constituintes daquela.

Quanto à comunicação entre a rede e o servidor a escolha, como antes referido, recaiu sobre o MQTT, com bom desempenho, baixa largura de banda, consumo

energético e latência reduzidos, utilizado com eficiência no sistema genViotGarden, cuja performance nos propusemos melhorar.

Em referência à comunicação entre os nós da rede, e observando a tabela 2.1, que contém as principais características dos protocolos anteriormente mencionados na seção 2.2.2, nunca esquecendo que o objetivo é desenvolver um sistema de baixo custo, conclui-se que o protocolo ideal para a implementação da WSN proposta é o LoRa. Esta escolha surge reforçada nos estudos levados a cabo pelos autores em [4].

2.3 Rede de Sensores sem Fios

O IoT depende da capacidade de recolher, enviar e processar dados, e daí ser fundamental ao mesmo uma rede de sensores. Podem fazer parte destas redes qualquer dispositivo equipado com sensores, como *smartphones*, carros e até micro controladores, como os Arduinos.

As Redes de Sensores [11] são compostas por um número de nós sensores que se comunicam de forma multi-salto. Conforme apresentado na Figura 2.4, existem três tipos de nós constituintes da rede [12]:

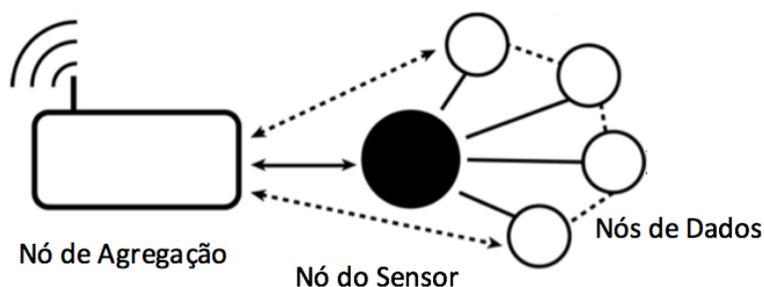


FIGURA 2.4: Nós constituintes de uma Rede de Sensores [11]

- **Nó de Agregação:** esses nós não leem os dados dos sensores, normalmente têm um dispositivo de comunicação e um *gateway* para o servidor. Estão preparados para receber dados de um ou mais nós de sensores e enviá-los aos servidores.

- **Nó do Sensores:** são sensores que armazenam dados e enviam-nos para outros nós. O armazenamento de dados pode ser usado para executar tarefas autónomas ou seguir informações quando as comunicações estão inativas. Estes nós exigem dispositivos adicionais com mais recursos, como o armazenamento, em que os cartões SD ou base de dados geralmente são usados.
- **Nó de Dados:** é o nível mais baixo de uma rede de sensores e é composto simplesmente por um sensor e um mecanismo de comunicação. Reúne as informações e envia-as para outro nó, seja o de agregação ou o de dados.

As redes de sensores são projetadas para economizar energia, porque esta é o recurso mais difícil de encontrar nos locais onde aquelas estão instaladas. Têm de ser escaláveis, dado o alto número de nós que uma rede pode ter; fiáveis, pois em algumas situações a rede pode enviar avisos caso algo esteja errado; e por fim robustas, já que os nós de sensores podem ser expostos a falhas ou riscos ambientais [11].

2.4 Plataformas de Controlo

IoT depende de um conjunto de dispositivos conectados numa rede controlada por uma aplicação usando um determinado protocolo de comunicação. Estes dispositivos têm de ser capazes de recolher e armazenar dados. Os mais comuns são micro controladores e dispositivos SoC , baseados num chip capaz de substituir um computador ou um circuito integrado, que contém um sistema de computação. Estes têm a vantagem de diminuir o custo de um sistema de computação, tornando-o mais barato para o consumidor final.

As plataformas de controlo mais utilizadas são placas baseadas em Arduino ou ESP8266 e os dispositivos SoC mais utilizados são o Raspberry Pi.

2.4.1 Plataforma Arduino

O Arduino foi criado em Itália em 2005 [13][14], sendo uma placa eletrónica e *open-source* bastante simples de implementar, tanto em termos de hardware como de software [15]. É uma placa simples de input/output [13], tendo a sua própria

IDE, sendo possível programa-la em várias linguagens de programação, sendo as mais comuns C e C++.

Os modelos Arduino mais comuns são o Uno e o Mega, representados na figura 2.5, que são baseadas num micro controlador ATmega. Estas são uma boa solução de baixo custo para projetos de eletrônica.



FIGURA 2.5: Arduino Uno [16]

Devido às suas características muito singulares o Arduino é único, pela sua adaptabilidade, correspondendo tanto ao seu Software *open-source* como ao seu Hardware. Estas placas são flexíveis e robustas, ideais para projetos relativamente pequenos, fáceis de programar devido à sua própria IDE e entrada USB . As únicas desvantagens das mesmas são a pequena memória interna, operarem numa única *Thread* e não serem capazes de correr o seu próprio sistema operativo [13][17].

2.4.2 Módulo Wi-Fi ESP8266

O chip ESP8266 é um módulo *WiFi* de baixo custo capaz de fornecer o protocolo TCP/IP completo embutido num microcontrolador, produzido pela Espressif Systems em 2014. Este módulo, quando conectado a outros dispositivos, como uma placa Arduino, permite a conexão à Internet e a troca simples de dados [14].

Existem vários lançamentos deste chip, com características diferentes, mas o mais comum é o ESP-12 [18], módulo WiFi baseado no processador core ESP8266 que permite a interface com sensores e atuadores através de pinos de I/O , oferecendo uma solução de rede WiFi completa e independente. É de baixo consumo de energia, baixo custo e tamanho reduzido.

Embora este chip possa funcionar por si só, os seus pinos não são compatíveis com a maioria dos formatos de placas, portanto, é comum ver o chip integrado numa placa de *breakout*, como mostra a Figura 2.6. O Adafruit Huzzah ESP8266 *breakout* foi projetado para permitir um fácil acesso ao chip ESP-12. Também contém um regulador de tensão e pinos FTDI para programar o chip, entre outras características.

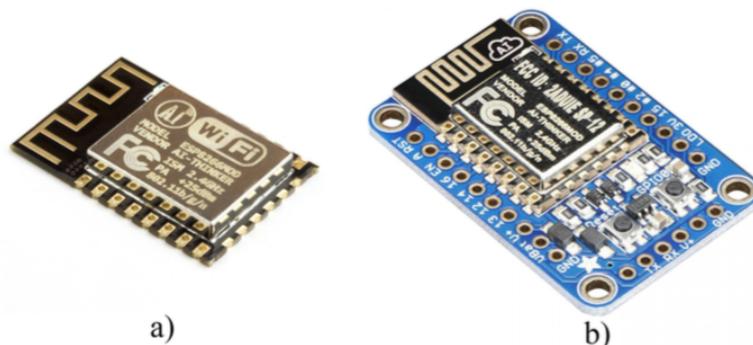


FIGURA 2.6: Componentes ESP8266: a) chip ESP-12; b) Adafruit Huzzah ESP8266 breakout [14]

Este chip é a solução ideal quando é necessária uma conexão à Internet, em projetos de IoT com as características supra referidas (baixo custo e tamanho reduzido). A única desvantagem é necessitar de um componente de hardware adicional para a sua utilização pelo consumidor final.

2.4.3 Módulo Rádio RFM69HCW

O RFM69HCW é a escolha ideal para o envio de dados em longas distâncias. A maioria dos chips de *WiFi*, *Bluetooth* e *Zigbee* utilizam a frequência de 2,4GHz, o que é ideal para transferências de dados rápidas, mas se o objetivo não é a velocidade de transferência de dados, mas sim a distância a que estes são trocados, é possível utilizar uma frequência livre de licenças como a 433 e 900 MHz.

Além disso, uma das vantagens desses rádios é que são mais simples que o *WiFi* ou BLE e não é necessário efetuar conexões complexas. Simplesmente, basta enviar dados sempre que for preciso, sendo que, qualquer outro módulo sintonizado na mesma frequência receberá os mesmos, podendo o recetor enviar uma resposta ao emissor. Estes módulos fazem a compactação de pacotes, correção de erros

e também são capazes de retransmitir automaticamente subtraindo esta tarefa ao emissor, evitando-se o desperdício de energia na manutenção de um link ou emparelhamento entre o transmissor e recetor. Se se pretender fazer um nó de sensor com o objetivo de transmitir dados numa cidade ou num campo, então esses módulos de rádio, associados a um Arduino ou outro micro controlador, são ideais.

Uma das poucas desvantagens do RFM69HCW é que o utilizador necessita de dois ou mais rádios com frequências correspondentes, para que seja possível a troca de pacotes, ao contrário do WiFi e Bluetooth, que na maioria dos casos são incluídos em computadores e telemóveis.

Na figura 2.7 está representado o módulo RFM69HCW na versão de 433 MHz. Estes são rádios que possuem muitas funcionalidades extras como criptografia e retransmissão automática, conseguindo alcançar distâncias de 500 m em linha reta sem obstáculos, utilizando apenas antenas omnidirecionais. A distancia de cobertura pode aumentar, até 5km, com a utilização de antenas diretivas.

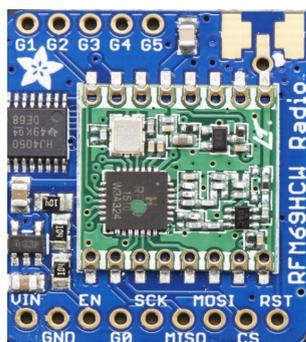


FIGURA 2.7: Módulo rádio RFM69HCW [19]

As principais características destes módulos são:

- Módulo baseado em SX1231 com interface SPI ;
- +13 a +20 dBm até 100 mW capacidade de saída de potência;
- Em corrente de 50mA (+13 dBm) a 150mA (+ 20dBm) para transmissões e 30mA durante escuta ativa do de rádio;
- Alcance de aproximadamente 500 metros, dependendo de obstruções, frequência, antena e saída de energia;

- Criação redes multi ponto com endereços individuais para cada nó;
- Mecanismo de encapsulamento de pacotes com AES-128;
- Bibliotecas de Arduino prontas a utilizar;
- Utiliza a licença amadora de largura de banda ISM ;
- Possibilidade de utilizar como antena um simples fio ou uma antena.

2.4.4 Principais módulos usados a implementação

Os componentes descritos anteriormente, constituem a implementação da rede de sensores sem fios que será o principal objeto desta dissertação de mestrado. Esta irá requerer um nó de agregação que será constituído por um ESP8266 e um RFM69HCW.

O ESP8266 será o responsável por exercer o protocolo MQTT, que irá fazer a comunicação entre o servidor e a rede.

Os outros nós da rede irão ter na sua constituição dois componentes principais um Arduino, um RFM69HCW e, ainda, alguns sensores que serão falados na secção 2.5. Toda a comunicação interna da rede, ou seja, entre os nós que a constituem, será levada a cabo pelo RFM69HCW. Todos os elementos da rede irão ter este componente integrado.

Estes componentes foram escolhidos sem esquecer que a rede a desenvolver será de baixo custo, sendo capazes de garantir a eficiência do sistema.

2.5 Sensores

Como foi referido acima, a deteção de dados é um dos elementos principais do IoT. Considerando que o objetivo desta dissertação é implementar uma WSN num sistema de rega já existente, que apenas recolhe os dados do ambiente online, esta secção incidirá na análise dos sensores de humidade e temperatura ambiente, humidade do solo e luminosidade existentes no mercado.

Para que o sistema desenvolvido seja uma mais-valia para o já existente, é necessário que os dados do ambiente sejam mais precisos, razão pela qual a recolha

de dados online será complementada através da implementação de um conjunto de sensores de humidade e temperatura do ar, humidade do solo e luminosidade, a escolher em função da melhor relação eficiência/preço. Considerando que o sistema genviotGarden visa ter um custo reduzido é imprescindível que os sensores tenham o menor custo possível e um bom desempenho, para uma boa eficiência do sistema.

Será feita a descrição detalhada dos principais sensores de baixo custo para cada uma das grandezas descritas acima. Os fatores a analisar em cada sensor são a eficiência, o preço e a simplicidade de ligação ao Arduino.

2.5.1 Sensores de Humidade e Temperatura

2.5.1.1 Sensor DHT22

DHT22 é um sensor de temperatura e humidade que permite fazer leituras de temperaturas entre -40.0°C a $+80.0^{\circ}\text{C}$ graus Celsius e humidade entre 0 a 100%, sendo muito fácil de usar com Arduino, Raspberry e outros micro controladores, em virtude de possuir apenas 1 pino com saída digital. Utiliza tecnologia exclusiva de recolha de sinal digital e tecnologia de deteção de humidade, assegurando a sua fiabilidade e estabilidade [20].

2.5.1.2 Sensor DS18B20

DS18B20 mede a temperatura de 9 bits a 12 bits e possui definições de alarme não voláteis definidas pelo utilizador. A sua utilização com placas Arduino e Raspberry também é muito simples, pois este apenas tem um pino com saída digital. O DS18B2 mede temperaturas entre -55.0°C a $+125.0^{\circ}\text{C}$ graus Celsius com precisão de 0.5°C na faixa de -10.0°C a $+85.0^{\circ}\text{C}$ [21].

2.5.1.3 Sensor LM35

O LM35 é um sensor de precisão, que apresenta uma saída de tensão linear relativa à temperatura em que ele se encontra no momento em que for alimentado por uma tensão de 4-20Vdc e GND, tendo à saída um sinal de 10mV para cada Grau Celsius de temperatura. Este sensor não necessita de qualquer calibração externa para

fornecer com exatidão valores de temperatura com variações $\frac{1}{4}^{\circ}\text{C}$ ou até mesmo $\frac{3}{4}^{\circ}\text{C}$ dentro da faixa de temperatura de -55.0°C a 150.0°C [22].

Utilizando as expressões 2.1 e 2.2 é possível medir o valor da temperatura ambiente com base do valor da tensão recolhido.

$$T_{A0} = V_o \left(\frac{5}{1023} \right) [V] \quad (2.1)$$

$$T_C = \left(\frac{T_{A0} \times \frac{5}{1023}}{10} \right) [^{\circ}\text{C}] \quad (2.2)$$

Onde:

- T_{A0} , representa a tensão em Volts
- V , representa a tensão de saída vinda do sensor em Volts
- T_C , representa a temperatura recolhida pelo sensor em graus Celsius.

Na figura 2.8 está representada a evolução da temperatura em função da voltagem.

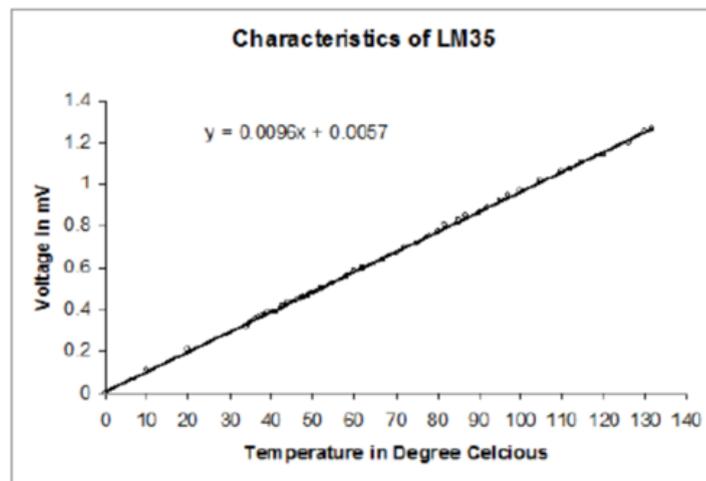


FIGURA 2.8: Temperatura em função da voltagem (Sensor LM35)[22]

2.5.1.4 Sensor TMP36

É um sensor preciso de temperatura de baixa tensão, que fornece uma saída de tensão linearmente proporcional à temperatura em graus Celcius. É capaz de identificar temperaturas de -40°C a $+125^{\circ}\text{C}$ sem necessidade de calibrações e com uma precisão de $\pm 1^{\circ}\text{C}$ a 25°C . O sensor destina-se a uma operação de alimentação única de 2.7V a 5.5V máxima com uma corrente atual abaixo de $50\mu\text{A}$, fornecendo uma Saída de 750mV a 25°C [23].

A expressão 2.3 ilustra o cálculo da temperatura a partir do valor da voltagem medido pelo sensor.

$$T_C = \frac{V - 500}{10} [^{\circ}\text{C}] \quad (2.3)$$

- T_C , representa a temperatura recolhida pelo sensor em graus Celsius.
- V , é o valor de tensão medido pelo sensor em mili Volts.

Na figura 2.9 está representada a evolução da temperatura em função da tensão.

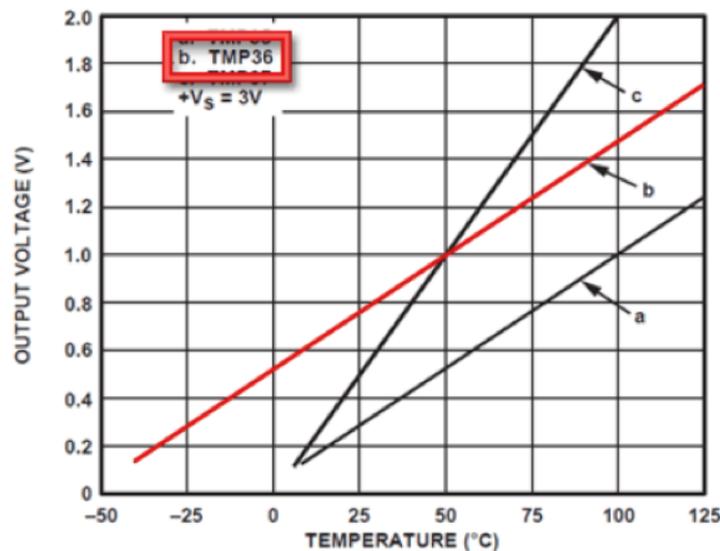


FIGURA 2.9: Temperatura em função da voltagem (Sensor TMP36) [23]

2.5.2 Sensores de Humidade do Solo

2.5.2.1 Sensor de Humidade do Solo

SparkFun Soil Moisture Sensor é um sensor simples para medir a humidade no solo e de materiais similares. O sensor de humidade do solo é bastante simples de usar. Os seus dois dentes expostos funcionam em conjunto como sondas para o sensor, atuando como uma resistência variável. Quanto mais água estiver no solo, melhor será a condutividade entre as resistências [24]. A integração deste sensor com placas como o Arduino é bastante simples.

2.5.2.2 Sensor de água de alta sensibilidade

É um sensor projetado para a deteção de água, que pode ser amplamente utilizado para detetar a precipitação, o nível da água em tanques e até mesmo fugas. Este pode avaliar o nível da água através de uma série de pontos de fios paralelos expostos para medir o tamanho de uma gota. O seu consumo de energia é reduzido e a sua sensibilidade é elevada. É compatível com o Arduino [25].

2.5.3 Sensores de Luminosidade

2.5.3.1 Sensor de luz LDR

Light Dependent Resistor (LDR) é usado para verificar a luminosidade e é composto por duas células foto correntes de sulfureto de cádmio, que podem imitar as respostas espectrais de um olho humano, em que a resistência cai com o aumento da intensidade da luz. O sensor pode ser usado com até 320V e tem uma corrente de alimentação máxima de 75mA fornecendo 400 Ω para 1000 LUX e 9k Ω para 10 LUX [26].

Na figura 2.10 está representado o gráfico que ilustra como a resistência do sensor evolui em função da luz incidente.

2.5.3.2 Foto Transistor ALS-PT19

É um sensor de luz ambiente de baixo custo, constituído por um fototransistor SMD em miniatura. Devido ao alto índice de rejeição da radiação infravermelha, a resposta espectral do sensor de luz ambiente é próxima da dos olhos humanos. Este é caracterizado por um fecho do espetro de acordo com o olho humano, sendo alimentado pela luz solar, tendo boa linearidade de saída em todo o alcance da iluminação, e baixa variação de sensibilidade, de acordo com várias fontes de luz. O seu bom desempenho garantido com temperatura entre -40°C e 85°C , alta faixa de tensão de alimentação, entre 2,5V e 5,5V [27].

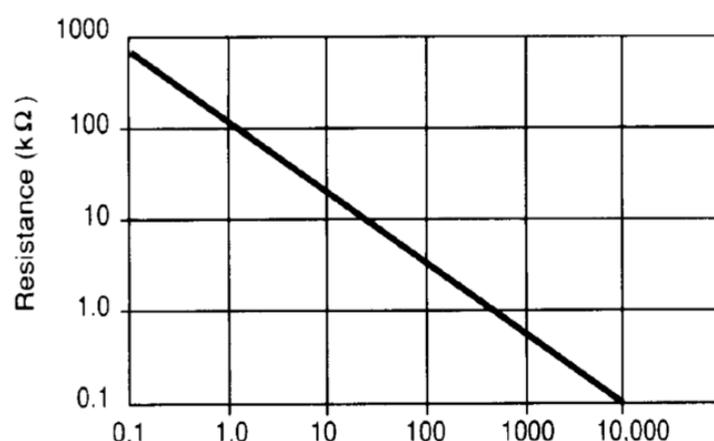


FIGURA 2.10: Resistência em função da luz incidente (ALS-PT19) [27]

2.5.4 Sensores a implementar nos nós da rede desenvolvida

O estudo efetuado dos diversos sensores de Temperatura e Humidade, Humidade do solo e Luminosidade podem ser resumidos nas tabelas 2.2 2.3 e 2.4.

TABELA 2.2: Comparação de Sensores de Humidade e Temperatura do ar

Humidade e Temperatura	Custo [€]	Precisão(a 25° C)	Gama de Valores [°C]	Simplicidade de Ligação
LM35	1.84	0.5	-55 / 150	Fácil
TMP36	1.84	0.5	-40 / 125	Fácil
DS18B20	7.87	0.5	-55 / 125	Média
DHT22	9.72	0.5	-40 / 80	Fácil
Si7021	10.95	0.4	-10 / 85	Média

Observando as tabelas e sem nunca esquecer que a rede que irá ser desenvolvida visa o baixo custo, o preço dos sensores constitui um dos pontos principais a ter em consideração.

TABELA 2.3: Comparação de Sensores de Humidade do solo

Humidade do Solo	Custo[€]	Precisão (25° C)	Gama de Valores[%]	Simplicidade de Ligação
Funduino Moisture/Drop	3.01	-	0 - 100	Média
SparkFun Soil Moisture	5.90	-	0 - 100	Fácil

TABELA 2.4: Comparação de Sensores de Luminosidade

Humidade do Solo	Custo [€]	Precisão (25° C)	Gama de Valores[LUX]	Simplicidade de Ligação
LDR	0.49	-	0 - 10000	Fácil
ALS-PT19	3.44	-	1 - 10000	Fácil

Por isso, em relação aos sensores de humidade e temperatura do ar os únicos sensores que possibilitam a construção de um sistema de baixo custo são o LM35 e o TMP36, por serem os mais baratos, como pode ser visto na tabela 2.2. Entre estes dois a escolha poderia recair em qualquer deles devido às suas características serem muito idênticas.

Em relação aos sensores para a humidade do solo, a escolha irá recair sobre o *SparkFun Soil Moisture*, por este ser um sensor exclusivo para coletar dados da humidade do solo, ao contrário do *Funduino Moisture/Drop*, mais adequado à medição da quantidade de água em reservatórios e verificação da pluviosidade. Apesar de ser o mais barato no mercado, a sua ligação ao Arduino é mais complexa e, por isso, menos adequada.

Por último, quando observamos a tabela relativa aos sensores de Luminosidade constatamos que o LRD é o sensor ideal, uma vez que o seu preço é menor do que o do ALS- PT19, apesar de ambos possuírem características semelhantes.

2.6 Trabalhos Relacionados

A área da rega não é nova para o mundo do IoT, pois já existem sistemas implementados que utilizam WSN para controlar a irrigação de jardins.

A maioria desses sistemas usa micro controladores, como Arduino e Raspberry Pi, combinados com um conjunto de sensores de humidade do solo, temperatura, humidade do ar e luminosidade para coletar dados do ambiente, para posterior análise. O principal objetivo destes sistemas é aumentar a eficiência do jardim ou campo agrícola, proporcionando maior conforto ao utilizador e gestão eficiente da água expressada.

Ruby Roselin em “*Smart Agro System Using Wireless Sensor Networks*” [28] propõe um sistema com dois propósitos: jardinagem e agricultura. O sistema utiliza um Arduino e sensores de temperatura e humidade do ar, humidade do solo e pluviosidade. O objetivo é irrigar o solo apenas quando a humidade deste for insuficiente, evitando regas desnecessárias.

O sistema apresentado em “*Smart Home Garden Irrigation System with Raspberry Pi*” [29] por Syaza Norfilsha utiliza um Raspberry Pi e uma aplicação *Android*. Esta possibilita ao utilizador controlar fluxo de água do sistema com base nos dados de sensores de humidade do solo e de chuva, sendo o utilizador alertado por email quando as condições ambientais estão críticas.

Dr M. Newlin Rajkumar desenvolveu um sistema de rega inteligente controlado por um Arduino, capaz de ativar as bombas de água automaticamente e aceder aos dados do campo agrícola em tempo real a partir do seu telemóvel [30].

Anindita Mondal desenhou e desenvolveu uma WSN de baixo custo para ser aplicada à agricultura. Tendo em vista uma maior redução dos custos desenhou novos nós de sensores, em vez de os comprar no mercado [31].

Outra WSN construída para a agricultura foi desenvolvida por Trifun Savić, que a aplicou a um projeto de rega já existente. Esta rede baseia-se num Arduino e sensores que monitorizam a qualidade ambiental. Os nós desta WSN comunicam através do ZigBee e enviam os dados para o servidor para serem analisados mais tarde [32].

Sowmya desenvolveu uma WSN flexível e robusta para controlo remoto da qualidade da água. Esta consiste num coordenador e vários nós de sensores equipados com micro controladores, módulos de Internet e sensores que coletam os parâmetros da qualidade da água, como por exemplo o pH. Este dados são enviados para um servidor onde são analisados [33].

2.6.1 Características inovadoras da rede a implementar

A WSN que será desenvolvida nesta dissertação destaca-se das referidas anteriormente pela utilização de um protocolo de comunicação sem fios que se encarregará da comunicação entre os nós da rede. Isto é uma grande vantagem, pois não há limitação de comprimento dos fios. Esta WSN será dinâmica, ou seja, será sempre

possível adicionar novos nós de sensores diferentes dos existentes sem que isso afete o seu desempenho.

Na construção e implementação da WSN não será utilizado *ZigBee* pois este implica custos desnecessários. Em vez disso será utilizado outro componente, mais barato e com maior alcance, o RFM69HCW, que utiliza o protocolo *LoRa*.

Capítulo 3

Arquitetura do Sistema

Neste capítulo será apresentada a arquitetura do sistema em três secções, procedendo-se na primeira à definição de todos os componentes constituintes da rede de sensores sem fios desenvolvida, bem como à correlação existente entre cada um dos nós constituintes; na segunda será feita a esquematização dos processos e algoritmos através de fluxogramas, que irão ilustrar de forma simples a transição das informações entre todos os elementos da rede; e na última será feita a apresentação e explicação dos diagramas temporais relativos às trocas de mensagens das fases cruciais da rede, com especial enfoque num algoritmo de troca de mensagens desenvolvido para evitar que alguma delas seja perdida e que com ela se percam informações necessárias ao bom funcionamento do sistema.

3.1 Arquitetura

Uma visão de alto nível da arquitetura do sistema proposto é representada na Figura 3.1.

Como pode ser visto, esta é constituída por 3 nós principais: um *Gateway*, nós de sensores e atuadores.

Esta rede é flexível e adaptável, conforme descrito anteriormente, na medida em que é sempre possível adicionar vários nós de sensores e atuadores, em função do tamanho do jardim e, conseqüentemente, do número de zonas que ele possui, ao contrário do *Gateway* que é único em cada rede.

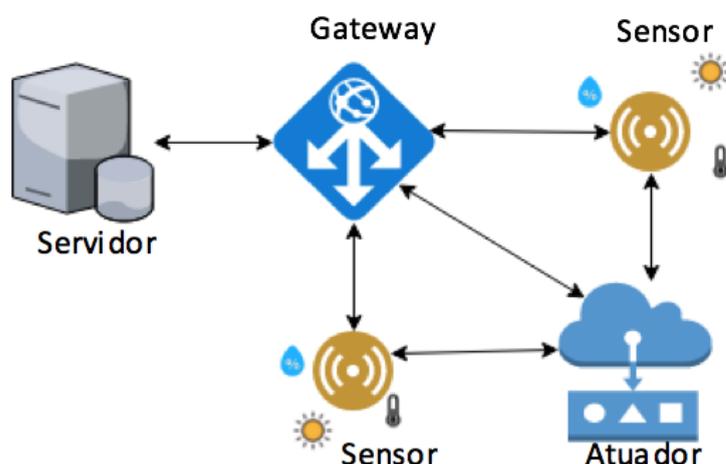


FIGURA 3.1: Arquitetura do Sistema proposto

Toda a rede é controlada com um único *Gateway* que funciona como nó de agregação e que está em constante comunicação com um servidor online.

Esta comunicação entre o *Gateway* e o servidor é feita através do MQTT, por via de uma conexão *WiFi*, e é complementado com um conjunto de nós de sensores e atuadores que comunicam com o *Gateway* e entre si utilizando uma ligação ponto-a-ponto *LoRa* através do RFM69HCW.

Como dito anteriormente, a arquitetura proposta é composta por três tipos de nós. Esses nós são descritos detalhadamente nos pontos que se seguem.

- **Gateway:** é composto apenas por 2 elementos, um ESP8266 e um RFM69HCW, e tem como principal recurso a capacidade de comunicar com o servidor, pois é o único elemento capaz de se conectar à Internet. Isto faz com que este nó seja responsável por reencaminhar as mensagens vindas do servidor com destino aos outros nós da rede e vice-versa, mensagens estas que podem conter informações relativas às configurações do sistema, tempos de irrigação e dados dos sensores em tempo real para que estes sejam armazenados no servidor para serem analisados. Este nó também é responsável pela gestão do algoritmo de troca de mensagens que irá ser falado mais à frente.

O *Gateway* deve estar posicionado numa posição estratégica, de modo a estar dentro do alcance tanto do ponto de acesso à Internet, como dos outros nós da rede.

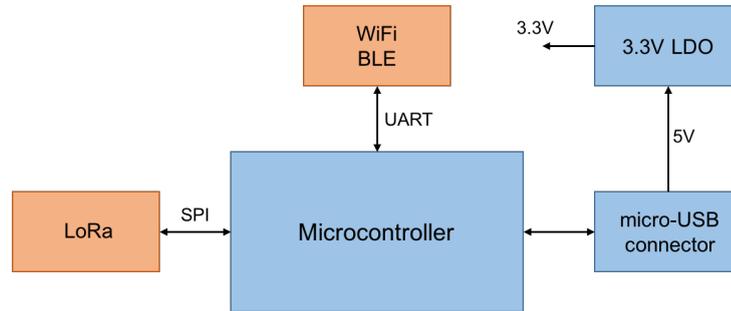


FIGURA 3.2: Diagrama de Blocos Gateway

- Nó do Atuador:** é o nó mais complexo da rede, capaz de controlar bombas de água, motores ou luzes. Os principais componentes deste nó são um Arduino, um RFM69HCW, um shift-register - capaz de aumentar as portas digitais do Arduino entre 4 e 32 e um Darlington Transistor Array, responsável por fornecer a energia necessária ao atuador, já que o Arduino pode fornecer apenas 5V @ 40mA e a maioria dos atuadores requer pelo menos 12 / 24V @ 100mA. Como ele precisa de uma fonte de voltagem de saída para garantir a funcionamento correto dos seus atuadores deve ser colocado próximo a uma tomada elétrica ou conectado a um painel solar ou sistema de bateria.

O Atuador é capaz de comunicar com os outros nós da rede de maneira bidirecional, forma de comunicação muito importante pois só assim é possível ligar ou desligar as bombas de água com base nos dados dos sensores ou instruções do utilizador.

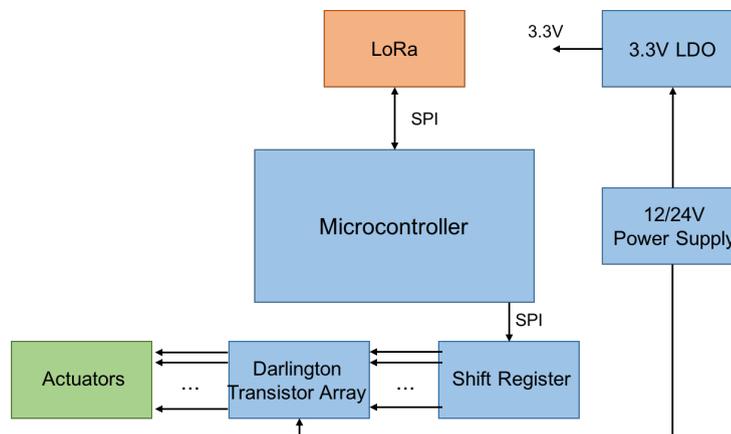


FIGURA 3.3: Diagrama de Blocos Atuador

- **Nó do Sensor:** é composto por um micro controlador Arduino ou modelo alternativo equivalente, um RFM69HCW e uma matriz de sensores. O seu principal objetivo é coletar os dados sobre as condições ambientais e enviá-las para o servidor para processamento dos mesmos. Estes dados são enviados de forma automática ou quando requeridos pelo atuador. Este nó, já que apenas é acordado por curtos períodos de tempo, é alimentado por baterias.

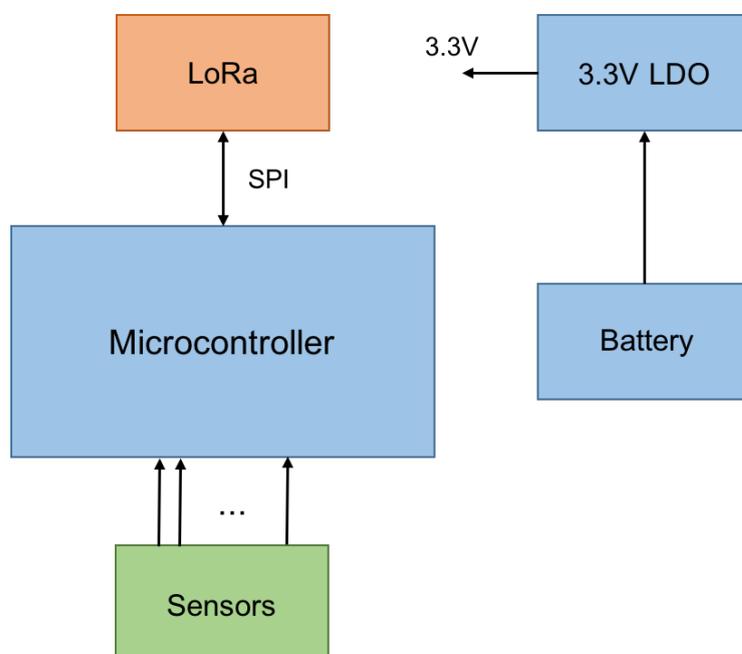


FIGURA 3.4: Diagrama de Blocos Sensor

3.2 Fluxogramas

3.2.1 Fluxograma relativo ao nó dos sensores

Conforme o fluxograma da Figura 3.5 é possível ver que o nó que contém os sensores é o mais simples da rede desenvolvida.

Este nó reage a dois tipos de mensagens, uma com o cabeçalho de *ConfirmBoard*, proveniente do *Gateway*, e outra com o cabeçalho de *GiveData* por parte do Atuador.

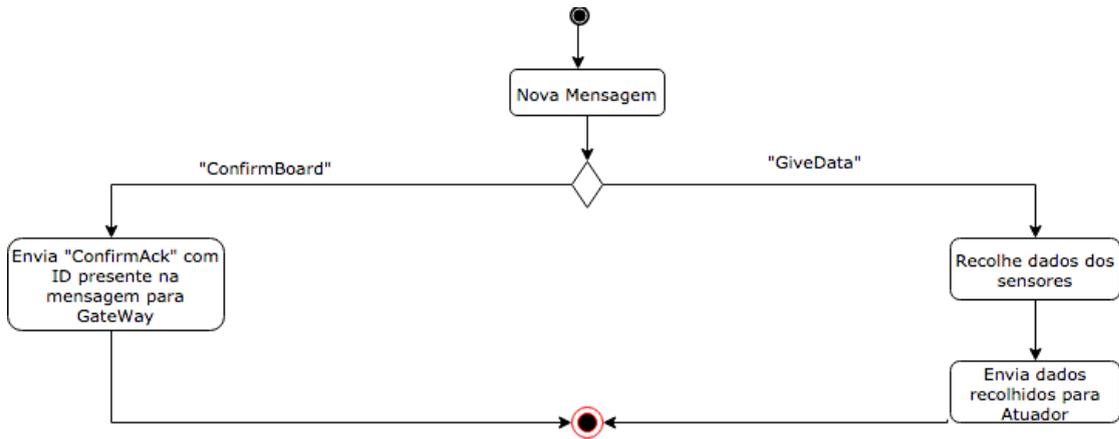


FIGURA 3.5: Fluxograma nó de Sensores

Quando o nó do sensor é ligado à corrente remete uma mensagem de *ConfirmBoard* junto do seu ID para o *Gateway*. Este quando recebe essa mensagem verifica na sua base de dados se o ID do nó que se ligou se encontra presente na mesma e em caso positivo é enviado *ConfirmBoard* de volta. O sensor quando recebe esta mensagem apenas necessita enviar uma mensagem de *ConfirmAck* para o remetente, contendo o ID presente na mensagem que recebeu.

Após a receção da mensagem de *GiveData* enviada pelo Atuador o nó dos sensores recolhe os dados relativos à temperatura e humidade do ar, humidade do solo e luminosidade, reenviando-os ao atuador.

3.2.2 Fluxograma relativo ao nó do atuador

Como já foi referido anteriormente o Atuador é o nó responsável pela ligação das bombas de água, ou *led's*. Como pode ser visto na figura 3.6 este reage a 3 tipos de mensagens provenientes tanto do *Gateway* como dos nós que contêm os sensores.

O atuador recebe mensagens do *Gateway* dos tipos *Water* e *ConfirmBoard*.

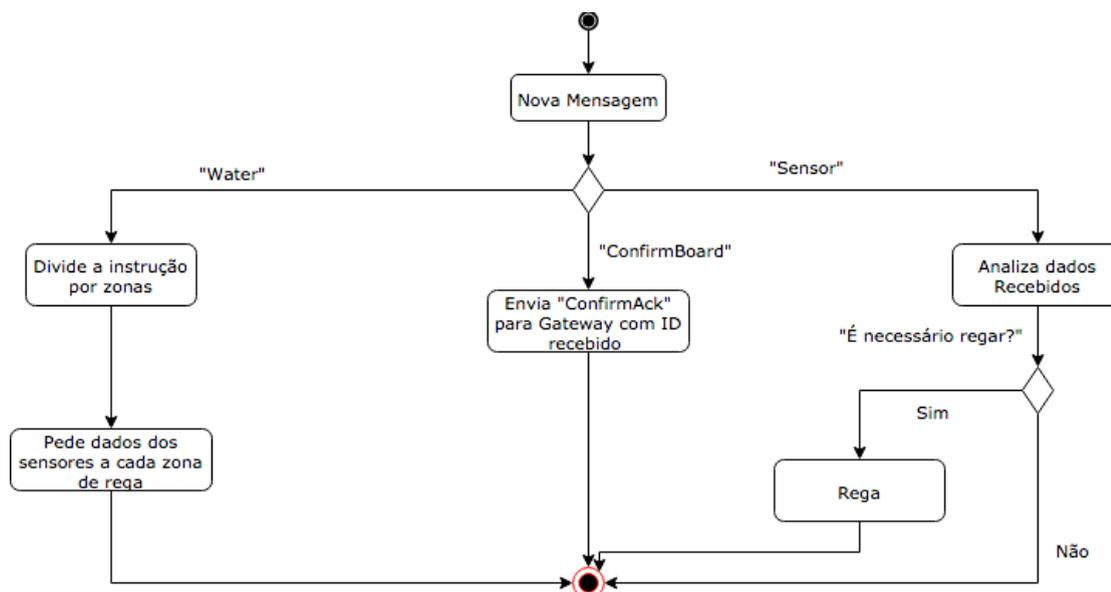


FIGURA 3.6: Fluxograma nó do Atuador

No momento em que o Atuador é ligado à corrente remete uma mensagem de *ConfirmBoard* junto do seu ID para o *Gateway*. Por sua vez, o *Gateway* quando recebe essa mensagem verifica na sua base de dados se o ID do nó que se ligou se encontra presente na mesma e, na afirmativa, reenvia um *ConfirmBoard*. O sensor quando recebe esta mensagem apenas necessita de enviar uma mensagem de *ConfirmAck* para o remetente, contendo o ID presente na mensagem que recebeu.

Sempre que o Atuador recebe uma mensagem com o tópico *Water*, seguida de uma instrução relativa ao tempo de rega de cada zona, divide-a solicitando a cada zona, separadamente, os dados dos sensores referentes à mesma.

Quando o Atuador recebe os dados anteriormente solicitados a cada zona com o tópico “*Sensor*”, analisa-os e verifica se a irrigação é necessária na zona em referência e, na afirmativa, em que medida. Decorrido o tempo de rega fixado, ou em caso de desnecessidade de irrigação, o sistema passa para a próxima zona repetindo o processo anteriormente descrito na zona seguinte.

3.2.3 Fluxograma relativo ao *Gateway*

O *Gateway* é o único nó do nosso sistema capaz de comunicar com o servidor por ser o único componente apto a suportar uma ligação *WiFi*. Para além disso, é o único nó que comunica com todos os outros nós da rede.

Como pode ser visto na figura 3.7, o *Gateway* reage a 5 tipos de mensagens, tanto por parte do servidor, como por parte dos outros nós da rede.

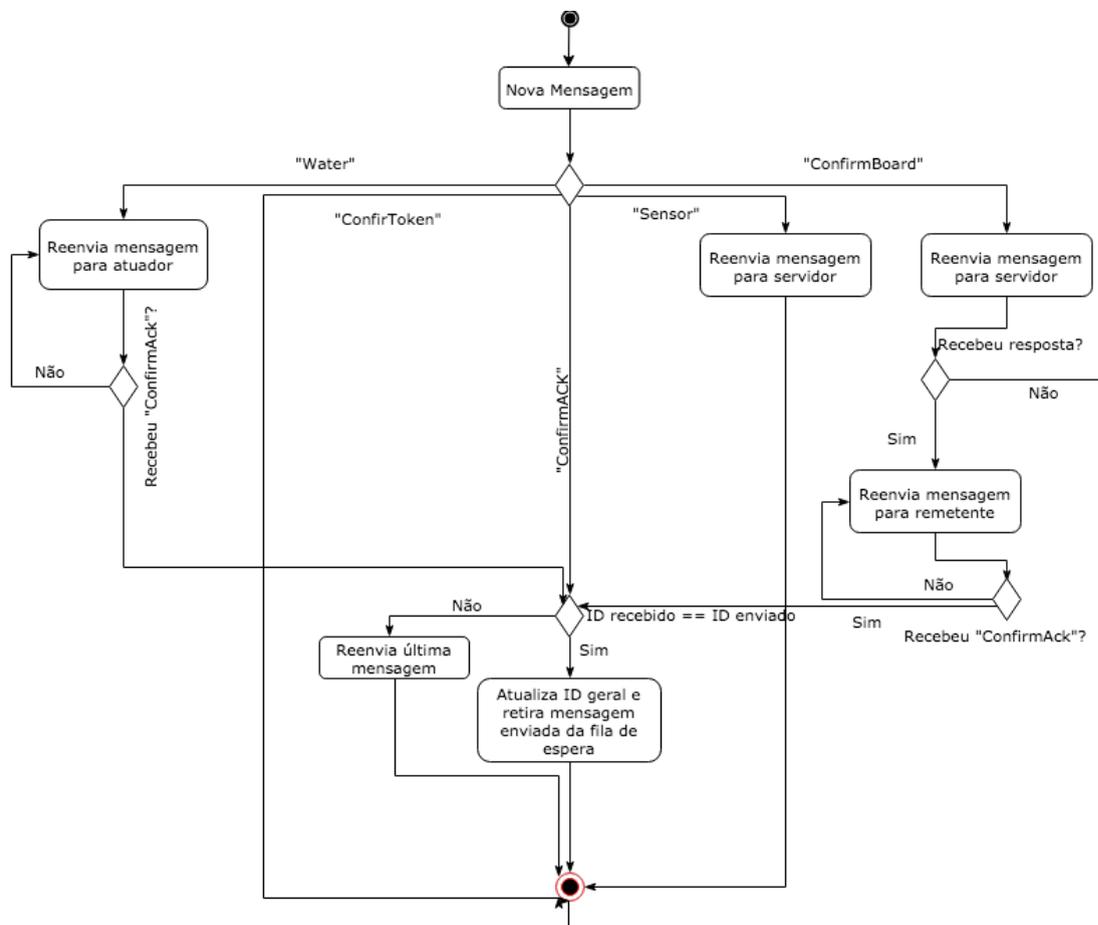


FIGURA 3.7: Fluxograma Gateway

O *Gateway* quando é acionado envia para o servidor uma mensagem com o tópic *ConfirmToken*, seguido do identificador geral do jardim. Por sua vez, o servidor irá perscrutar na base de dados pelo referido identificador e, detetando-o, reenvia para o *Gateway* uma mensagem de *ConfirmToken*, habilitando este nó a intermediar a ligação dos restantes nós.

Quando ocorre a ligação dos nós em referência, cada um deles envia para o *Gateway* uma mensagem de *ConfirmBoard*, a qual é imediatamente reenviada para o servidor que irá analisar se o nó que se ligou está presente na base de dados e, na afirmativa, reenvia uma confirmação para o mesmo.

Quando o utilizador faz o pedido através da plataforma para ser efetuada a rega do jardim, o *Gateway* recebe uma mensagem com o cabeçalho *Water*. Esta

é enviada pelo servidor sendo prontamente reencaminhada pelo *Gateway* para o Atuador. Como este tipo de mensagens é enviado por *Broadcast* este nó tem de garantir que a mesma chega corretamente ao destino, em razão do que fica a aguardar a receção de uma mensagem de *ConfirmAck*, com o mesmo ID da mensagem que foi anteriormente enviada, procedendo ao reenvio da mesma quando a receção não chega a ocorrer. Todo este processo relativo ao algoritmo de trocas de mensagens será explicado em detalhe mais à frente.

Quando o *Gateway* recebe uma mensagem de *ConfirmAck* é imperativo que faça a verificação do ID, que está contido na mesma, e veja se corresponde ao ID da última mensagem enviada. Em caso positivo, o ID geral é atualizado e a mensagem é removida da fila de espera de mensagens, onde estão todas as mensagens que foram recebidas para posteriormente serem enviadas. Em caso negativo, é reenviada a mensagem que se encontra na primeira posição da fila de espera, ou seja, a última mensagem enviada.

3.3 Diagramas Temporais de Mensagens

Nesta secção serão ilustradas e explicadas as principais trocas de mensagens entre os nós da rede. Nesta secção também será explicado o algoritmo de troca de mensagens quando essas são enviadas por *Broadcast*.

Mensagens enviadas por *Broadcast* são mensagens que são enviadas para todos os nós da rede. A biblioteca do RFM69HCW para o Arduino permite fazer o envio deste tipo de mensagens, mas não garante que estas sejam recebidas. Para isso foi necessário desenvolver um algoritmo que garantisse essa fiabilidade no envio deste tipo de mensagens.

Alguns campos das mensagens apresentadas nos diagramas seguintes foram ocultados devido aos direitos intelectuais da Genviot.

3.3.1 Diagrama temporal relativo ao Algoritmo de Troca de Mensagens

Como antes referimos, os nós do sistema proposto comunicam entre si através de uma ligação LoRa ponto-a-ponto, usando os módulos RFM69HCW. Estes módulos

combinados com uma biblioteca pronta a utilizar para Arduino, *RadioHead* [34], permitem atribuir a cada nó da rede um endereço de identificação único. Esta API permite que os nós comuniquem entre si de duas maneiras:

- Envio de uma mensagem diretamente para um nó específico, utilizando o endereço de identificação único;
- Envio de uma mensagem por *Broadcast* para todos os nós que se encontrem dentro do alcance de comunicação.

No primeiro caso, a API garante fiabilidade na troca de mensagens, enviando uma mensagem ACK automática, enquanto que no segundo caso não é possível garantir que a mensagem seja entregue ao nó pretendido. Além disso, a API fornece transações encriptadas e uma função *sleep*, para reduzir o consumo de energia dos nós.

A rede desenvolvida faz uso de ambas as maneiras de comunicação. Quando uma mensagem é enviada a partir do *Gateway* para todos os nós o envio é realizado por *Broadcast*, mas no sentido contrário é utilizado o endereço de identificação único do *Gateway*. Assim, a API só garante a fiabilidade de metade do sistema desenvolvido.

Como a fiabilidade e eficiência do sistema de IoT são fundamentais ao sucesso deste, é necessário garantir que todas as mensagens sejam efetivamente entregues. Para este efeito foi desenvolvido um algoritmo de confirmação de troca de mensagens por *Broadcast*, desenvolvido em cima da API já disponível. Desta forma, é possível assegurar que os nós pretendidos recebam a mensagem que é enviada para todos os nós, que constituem a rede. Para a implementação deste algoritmo foi aplicado o método *Firsts in, Firsts out* (FIFO), conforme ilustrado na Figura 3.8.

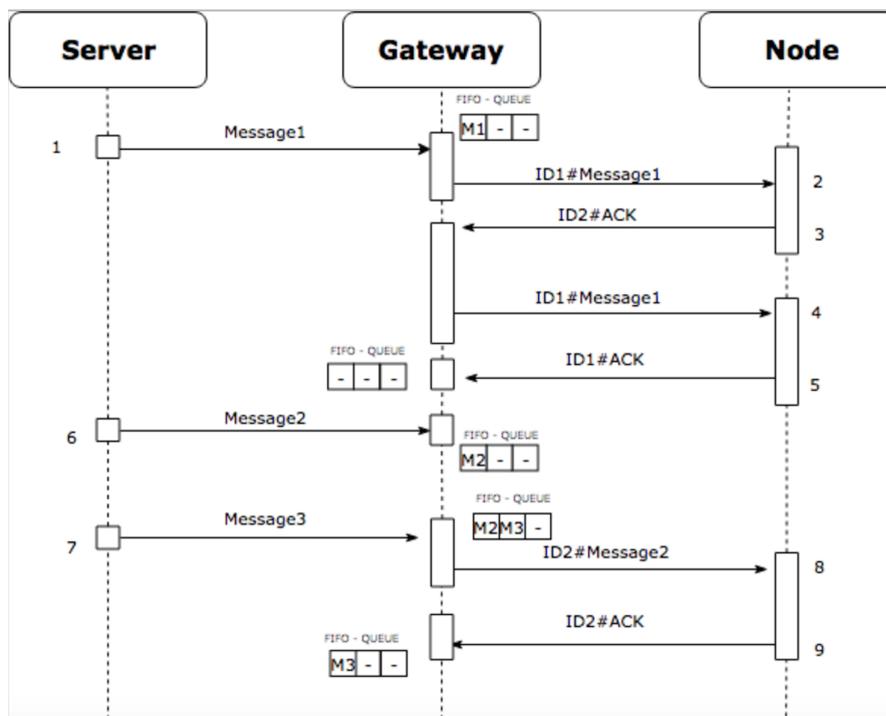


FIGURA 3.8: Exemplo Algoritmo de Troca de Mensagens

Na figura 3.8 encontra-se um exemplo de como esse algoritmo funciona. Para melhor percepção do mesmo, a seguir enuncia-se a sua forma de processamento:

1. O servidor envia uma mensagem para o *Gateway*. Como é possível ver essa mensagem encontra-se na primeira posição da fila;
2. O *Gateway* seleciona a mensagem que está em primeiro lugar da fila e acrescenta-lhe o ID geral, e posteriormente envia-a por *Broadcast*;
3. O nó recebe a mensagem e envia de volta uma mensagem de ACK com o ID presente na mensagem anterior. Quando o *Gateway* recebe uma mensagem de ACK compara o ID presente nesta com o último ID enviado;
4. Como o ID recebido não é igual ao último enviado, o *Gateway* volta a reenviar a última mensagem;
5. O nó recebe a mensagem e envia de volta uma mensagem de ACK com o ID. Desta vez o ID contido na mensagem de ACK é igual ao último enviado, o que determina a remoção da mensagem da fila de espera e a incrementação do ID;

6. O servidor envia uma mensagem para o *Gateway*. Como a fila está vazia a mensagem é colocada na primeira posição.
7. O servidor envia outra mensagem para o *Gateway*, como a fila não está vazia esta mensagem é colocada atrás da outra.
8. *Gateway* seleciona outra vez a mensagem que está em primeiro lugar da fila de espera e adiciona-lhe o ID. Depois envia-a por *Broadcast*;
9. O nó envia uma mensagem de ACK com o ID recebido para o *Gateway*. Como o ID recebido nessa mensagem é igual ao último ID enviado a mensagem é removida da fila de espera. Da observação da figura em análise resulta que apenas a mensagem 3 se encontra na fila de espera.

Concluindo, é possível assegurar a fiabilidade do sistema e garantir o correto funcionamento da WSN desenvolvida, certificando-se que nenhuma mensagem é perdida num ambiente controlado e que num ambiente real a taxa de perda de mensagens neste algoritmo é cerca de 1%.

3.3.2 Diagrama temporal relativo à mensagem do tipo *Water*

Este tipo de mensagem é solicitado por parte do servidor e é relativa à rega dos jardins.

Chega ao *Gateway* uma mensagem com o cabeçalho *water* seguido de uma instrução relativa ao tempo de rega de cada zona, com base nos valores recolhidos online. Esta mensagem é reencaminha do atuador, que divide a instrução e envia uma mensagem de *GiveData* com destino aos sensores, para que estes lhe enviem os seus dados, a fim de serem comparados com os recolhidos online.

Depois de todo o processo que este tipo de mensagem desenrola, o objetivo principal é o ajustamento dos tempos de rega em cada zona, de acordo com os dados recolhidos pelos sensores.

Este processo está ilustrado na figura 3.9 e desenrola-se da seguinte forma:

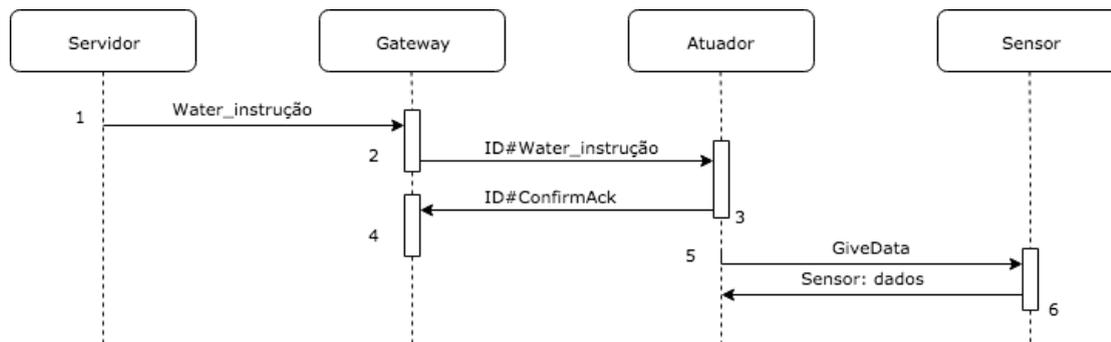


FIGURA 3.9: Diagrama temporal relativo à mensagem do tipo *Water*

1. O servidor envia para o *Gateway* uma mensagem com o cabeçalho *water* e uma instrução relativa aos tempos de rega de cada zona. Esta mensagem é adicionada à fila de espera de mensagens;
2. Quando a mensagem recebida ficar em primeiro lugar da fila, é-lhe acrescentado um ID que é geral, pois esta é enviada por *Broadcast* para todos os atuadores existentes;
3. Esta é recebida pelo Atuador que cria uma mensagem de *ConfirmAck* com o ID que recebeu, e envia-a de volta para o *Gateway*;
4. A mensagem de *ConfirmAck* é recebida e o algoritmo de troca de mensagens irá verificar se esta está correta.
5. O atuador após receber uma mensagens de *Water* envia para todos os nós de sensores da sua zona uma mensagem de *GiveData*, para os dados de cada zona específica sejam analisados;
6. O nó do sensor quando recebe uma mensagem de *GiveData* prontamente recolhe os dados dos seus sensores e envia-os com o cabeçalho "Sensor" para que estes sejam analisados.

3.3.3 Diagramas temporais relativos às mensagens do tipo *ConfirmToken* e *ConfirmBoard*

Este tipo de mensagens são desencadeados sempre que um dos nós da rede se liga. O *ConfirmToken* é referente à ligação do *Gateway*, e o *ConfirmBoard* refere-se à

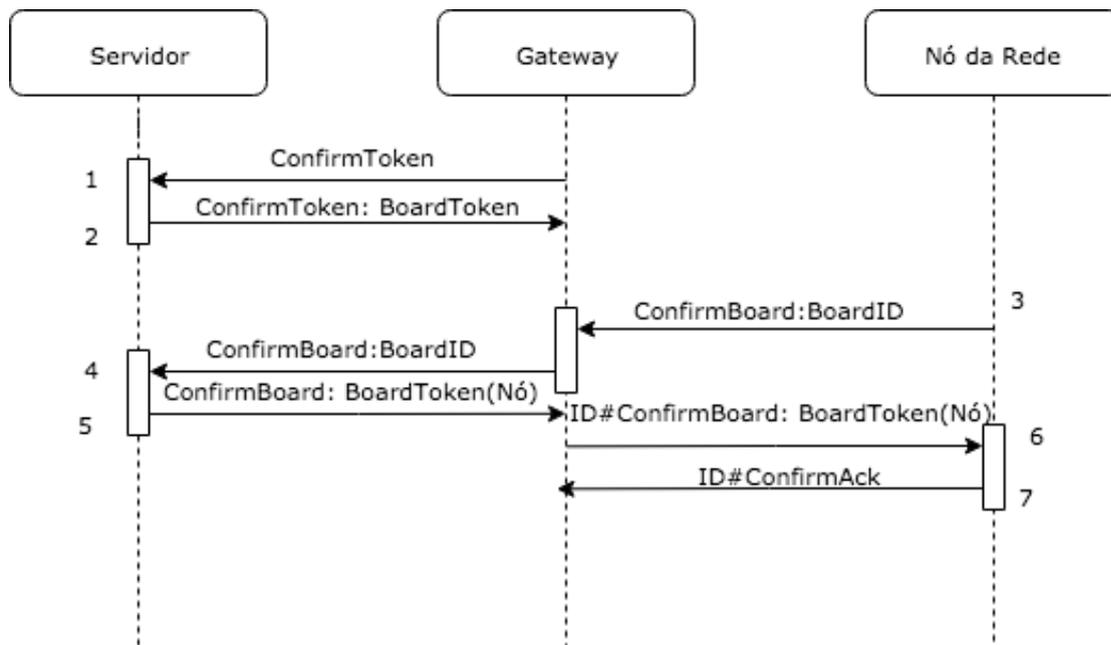


FIGURA 3.10: Diagramas temporais relativos às mensagens do tipo *ConfirmToken* e *ConfirmBoard*

ligação dos restantes nós da rede. Os nós têm de estar previamente registados da base de dados do sistema.

Este processo está ilustrado na figura 3.10 e desenrola-se da seguinte forma:

1. Quando o *Gateway* se liga, envia para o servidor o identificador geral de toda a rede, neste caso o *Token*, numa mensagem com o cabeçalho *ConfirmToken*;
2. Quando o servidor recebe a mensagem de *ConfirmToken*, verifica na sua base de dados se esse *Token* existe, em caso afirmativo reenvia para o *Gateway* uma mensagem de *ConfirmToken* com o identificador único do *Gateway*, neste caso o seu *BoardToken*;
3. Quando um nó da rede se liga envia para o *Gateway* uma mensagem com o cabeçalho *ConfirmBoard* seguida do seu identificador único;
4. Quando a recepção deste tipo de mensagem o *Gateway* apenas serve de intermediário, e reenvia-a para o servidor;
5. O servidor novamente verifica de este identificador se encontra na sua base de dados e em caso afirmativo envia uma mensagem com o cabeçalho de *ConfirmBoard*, para o *Gateway*, contendo o identificador do nó que se está a ligar;

6. Novamente o *Gateway* apenas serve de intermediário entre o Servidor e o nó da rede enviando a mensagem por *Broadcast*;
7. Quando o nó da rede recebe uma mensagem com o cabeçalho *ConfirmBoard* constrói uma mensagem de *ConfirmAck* com o ID contido nessa mensagem recebida;
8. A mensagem é enviada diretamente para o *Gateway* para avisar a recepção da mensagem anterior.

Capítulo 4

Implementação

A rede desenvolvida foi projetada para se conseguir adaptar a projetos de IoT em busca de expansão, nomeadamente para as atividades agrícolas, de jardinagem, complexos de piscinas ou casas, sem afetação do seu desempenho e eficiência, com capacidade de monitorização de áreas maiores, mantendo o baixo custo, o que o torna mais acessível ao grande mercado e, conseqüentemente, mais rentável.

Esta dissertação tem como objetivo geral aplicar a WSN desenvolvida ao principal produto da empresa Genviot, o GenviotGarden, permitindo a sua aplicação a jardins de pequenas e grandes dimensões, sem limitações que advenham do tamanho dos fios.

Neste capítulo são ilustrados dois tipos de implementações a que a rede foi sujeita, numa primeira fase, em cenário de laboratório e, numa segunda, em cenário real, bem como o processo de validação da mesma em cada um dos cenários. A primeira implementação foi feita num ambiente perfeito de laboratório, sem que fatores externos interferissem na comunicação entre os nós da rede. Após verificação das validações feitas em ambiente de laboratório, foi executada num ambiente real a segunda implementação, adaptação ao GenviotGarden.

As validações em qualquer um dos ambientes obedecerem aos procedimentos seguintes:

1. Envio de mensagens padrão de modo a verificar se os nós estavam todos a comunicar sem problemas;

2. Posterior envio pelo servidor de um conjunto de mensagens em vista a confirmar-se se a rede tinha o comportamento esperado na execução de cada uma das instruções dadas.

4.1 Implementação do Sistema para Testes em Laboratório

Depois de escolhidos os materiais e de esquematizada toda a mecânica da rede, esta foi testada num ambiente controlado de laboratório para ver como se comportava sem ter fatores externos que pudessem diminuir a eficiência da mesma, como por exemplo, vegetação a fazer de obstáculos, ou elevações do terreno.

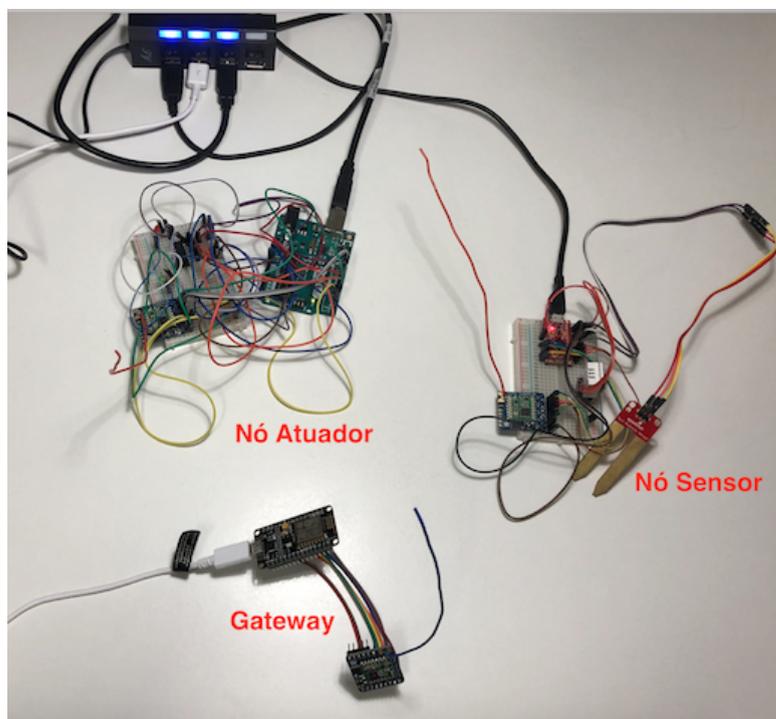


FIGURA 4.1: Implementação da WSN em ambiente de Laboratório

Como é possível ver na figura 4.1 os nós da rede foram colocados perto uns dos outros, para que fosse garantido que nada interferia na comunicação feita pelos rádios e ao mesmo tempo que todos os nós estivessem dentro do alcance uns dos outros, garantindo assim a inexistência de fatores externos a afetar a comunicação dos nós da rede.

Nas figuras 4.2, 4.3 e 4.4 é possível ver em pormenor a constituição dos elementos da rede.

- **Nó do Sensor:** como é possível ver na figura 4.2, o nó do sensor é constituído por um micro controlador alternativo e equivalente ao Arduino de tamanho reduzido, o módulo do rádio RFM69HCW e dois tipos de sensores, um de Humidade e Temperatura ambiente (o DHT22) e um sensor para medir a humidade do solo (o *Soil Moisture Sensor da SparkFun*).

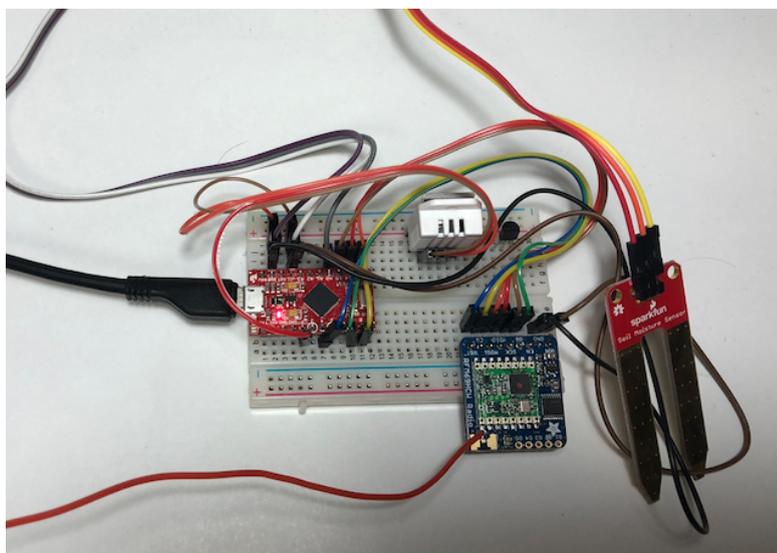


FIGURA 4.2: Prototipo nó do Sensor

- **Nó do Atuador:** este nó está representado na figura 4.3, sendo possível constatar, pelas ligações que tem, que é o nó mais complexo da rede, quando comparado aos outros. Dos componentes que o constituem, facilmente se vê o Arduino, o módulo rádio RFM69HCW, o *Shift Register*, que permite controlar mais portas lógicas do que as já existentes no Arduino, e dois *led's*, um vermelho e um verde, que neste caso representam as bombas de água. Quando um destes acende simula a abertura das mesmas

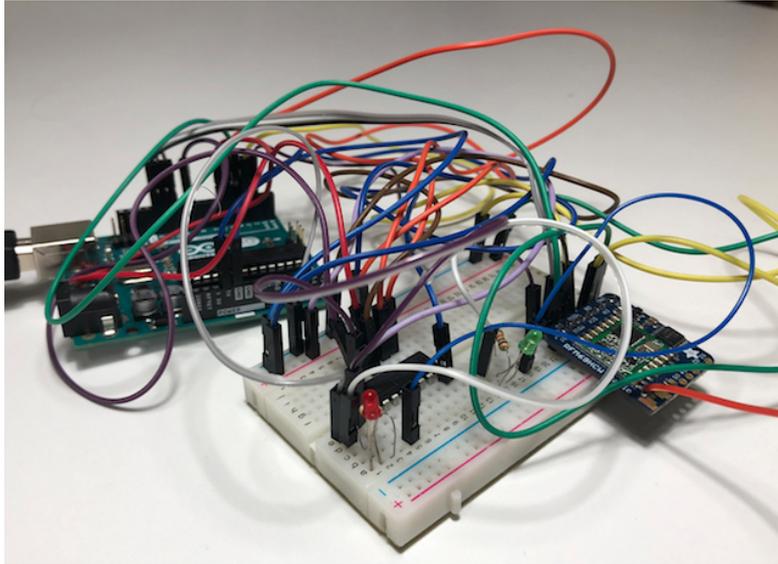


FIGURA 4.3: Prototipo nó do Atuador

- **Gateway:** em termos de componentes facilmente se vê que este nó é o mais simples, na medida em que é constituído apenas por dois elementos: o módulo rádio RFM69HCW e a placa de acesso à Internet ESP8266, que podem ser vistos na Figura 4.4.

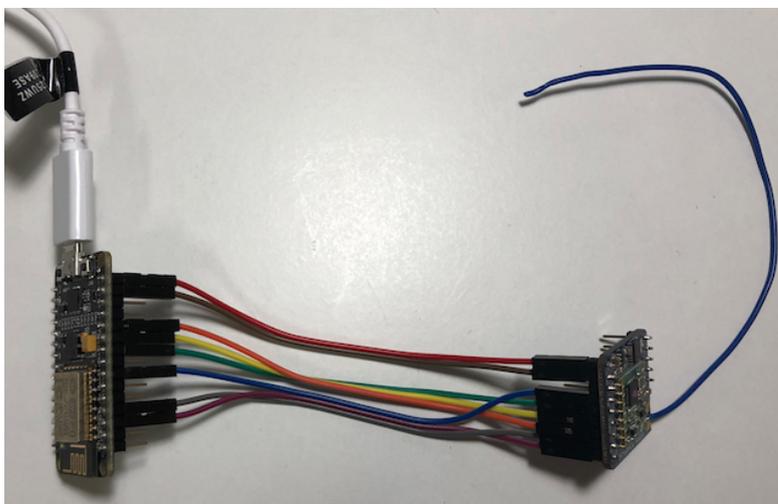


FIGURA 4.4: Prototipo *Gateway*

4.2 Implementação do Sistema para Testes em Cenário Real

Após a rede ter sido implementada e testada num ambiente de laboratório, foi verificado que a WSN desenvolvida proporcionava fiabilidade na troca de mensagens. Então, foi escolhido um cenário real para que fosse implementada e verificada a manutenção da fiabilidade na troca de mensagens.

Como um dos principais objetivos do IoT é a sustentabilidade e gestão dos recursos naturais, foi escolhido um sistema de irrigação de jardins GenviotGarden, sobre o qual a WSN foi desenvolvida, foi implementada e testada. O objetivo desta implementação foi controlar a quantidade de água utilizada na rega do jardim, com base nos dados de sensores em tempo real. Com estes é calculado, no atuador, o tempo de rega que cada zona do jardim necessita de ser regada, de modo a minimizar a quantidade de água utilizada na irrigação, sem afetar o seu desempenho.

Procedeu-se à divisão do jardim escolhido em zonas, cada uma delas com um conjunto de características que as diferenciasses, como por exemplo, o tipo de plantas, zonas de sombra ou elevações no solo, fazendo com que cada zona tenha características de temperatura, humidade do ar e humidade do solo diferentes e em consequência tempos de irrigação diferentes.

Para alcançar o objetivo proposto de um sistema de irrigação perfeito, o jardim escolhido foi dividido em 3 zonas, dando origem uma rede de 3 nós mais um Gateway.

As Figura 4.5 e 4.6 ilustram o jardim bem como a sua divisão por zonas.

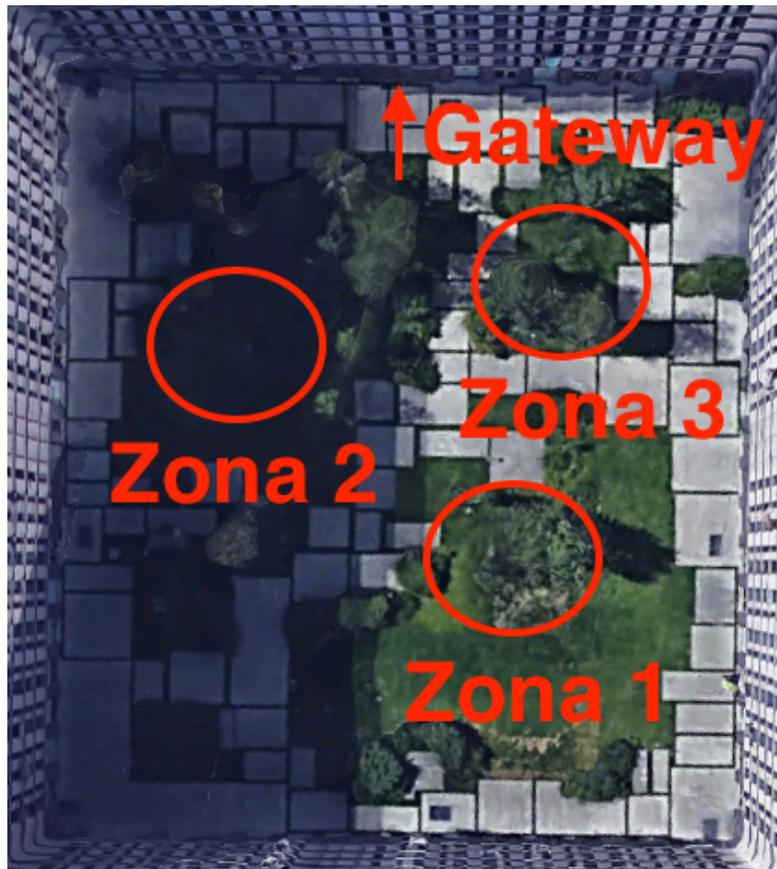


FIGURA 4.5: Imagem do Jardim (Satélite)



FIGURA 4.6: Imagem do Jardim (Fotografia Real)

A distância entre os nós da rede a implementar necessitava de ser relativamente grande de modo a que as condições ambientais variassem, a justificar a sua divisão em zonas, para se atestar a eficiência máxima da rega em casa zona.

Na zona 1 e 2 foram colocados nós com sensores, equipados com sensores de temperatura e humidade do ar e do solo. Como foi mostrado na secção 4.2, os sensores escolhidos para constituírem os nós dos sensores foram o DHT22 e o *Soil Moisture Sensor* da SparkFun. As Figuras 4.7 e 4.8 mostram os nós dos sensores implementados no jardim.

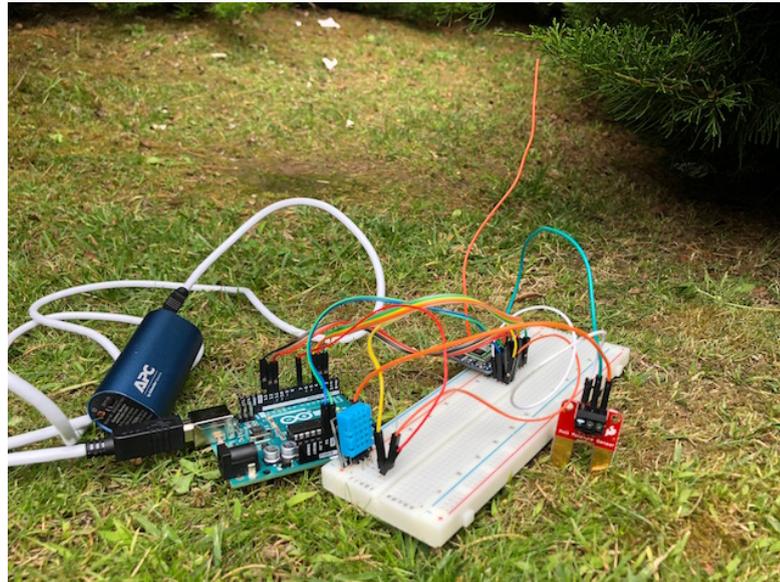


FIGURA 4.7: Prototipo do nó do sensor (Zona 1)

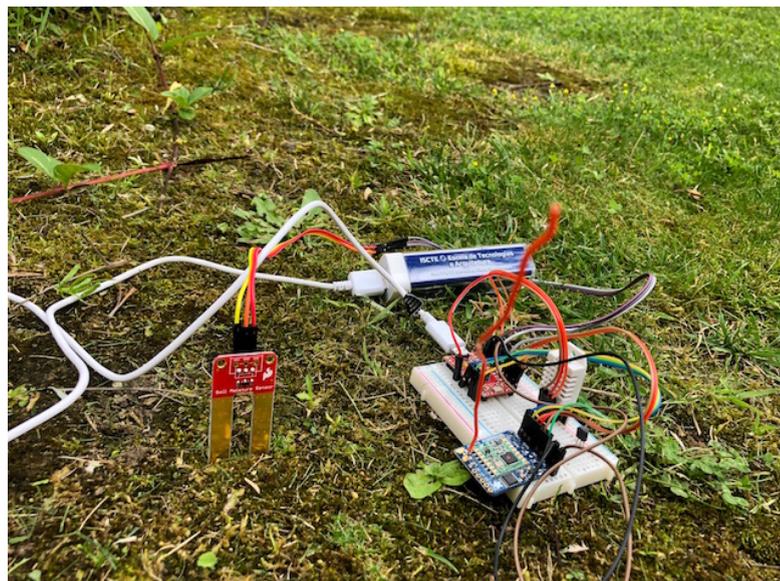


FIGURA 4.8: Prototipo do nó do sensor (Zona 2)

O nó que contém o atuador foi colocado na zona 3 do jardim, como é possível ver na Figura 4.9. A escolha deste local foi estratégica pois este necessitava de estar próximo das válvulas destinadas à irrigação, para controlo das mesmas.

Foi concebido um controlo de válvulas individuais para cada uma das zonas do jardim e este sistema de válvulas foi conectado ao atuador, para que este tivesse o poder de controlar a irrigação de cada uma das zonas, individualmente.

O atuador requisita os dados ambientais dos sensores e analisa-os. Após a análise dos mesmos, os tempos de irrigação de cada zona são ajustados de acordo com a necessidade de cada uma. Se o terreno se encontrar suficientemente húmido devido, por exemplo, a períodos de precipitação próximos do momento da recolha dos dados, não havendo por isso necessidade de irrigação, o tempo desta pode ser ajustado para 0 segundos.



FIGURA 4.9: Prototipo do nó do atuador (Zona 3)

Finalmente o *Gateway* foi colocado dentro do edifício perto de um ponto de acesso à Internet, como é possível ver na Figura 4.10. A sua colocação era fulcral para o bom funcionamento do sistema, pois este precisava de ter ligação à Internet e ao mesmo tempo ser capaz de comunicar com todos os nós da rede. Na Figura 4.10 é possível ver o *Gateway* dentro de um escritório com visão para o jardim.



FIGURA 4.10: Prototipo do *Gateway (Indoor)*

Capítulo 5

Resultados

Depois de implementada a WSN, tanto num ambiente controlado como num ambiente real, foi obtido um conjunto de resultados, cuja análise trataremos no presente capítulo, que dividimos em 3 partes, reportando-se a primeira aos resultados da implementação da WSN em laboratório, e a segunda à implementação do sistema num ambiente real.

Reservamos a terceira parte à exposição dos resultados dos testes efetuados à Potência de Recepção do sinal, aquando de uma ligação LoRa ponto-a-ponto entre dois RFM69HCW.

5.1 Resultados da Implementação do Sistema para Testes em Cenário de Laboratório

Depois de montados e colocados os elementos nos respetivos sítios, foi necessário averiguar se estes conseguiam comunicar uns com os outros. Para o efeito, foram enviadas mensagens padrão do servidor para os nós constituintes da rede e vice-versa. Depois de todas as mensagens terem sido corretamente trocadas a WSN estava pronta a ser testada.

Então, pelo servidor foram enviadas mensagens ao Gateway a simular instruções de irrigação de uma determinada zona, correspondente ao nó do sensor que havia sido aí implementado, que, por sua vez, a reencaminhou para o Atuador, que dividiu a mensagem pelo número de zonas existentes (no caso só havia uma

zona), o qual, finalmente, solicitou dados aos sensores da zona, constatando-se que nesta última etapa o Atuador recalculou com os dados recebidos do sensor o tempo de rega, ajustando o que constava na mensagem inicial de irrigação, na medida em que no termo deste processo a luz que simula a válvula da zona de irrigação manteve-se acesa durante o tempo que a válvula devia estar aberta a regar.

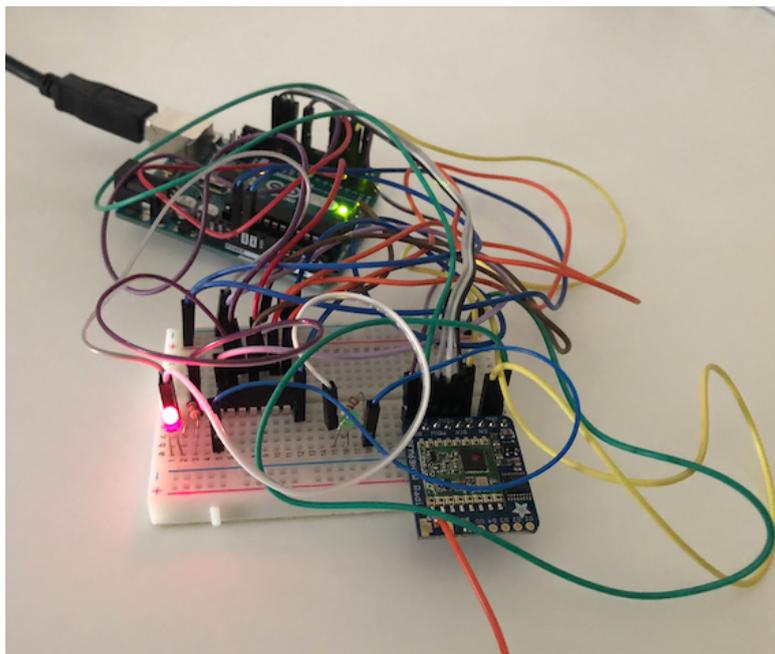


FIGURA 5.1: Prototipo nó do Atuador (Simulação de Rega)

Por outro lado, para verificar se o algoritmo de troca de mensagens tinha sido corretamente implementado, foram enviadas 15 mensagens de *Broadcast* do *Gateway* para os nós da rede, verificando-se que por a rede estar implementada num ambiente perfeito, sem obstáculos a obstruir os módulos rádio, todas as mensagens enviadas via *Broadcast* tiveram resposta, ou seja, foram recebidas 15 mensagens de *ConfirmACK*.

Para além disso, foram enviadas pelo nó do sensor para o *Gateway*, 89 mensagens, durante o tempo em que a rede esteve montada, constatando-se que todas elas foram corretamente reencaminhadas para o servidor. Nos ensaios efetuados no laboratório verificou-se 100% na taxa de sucesso das mensagens trocadas.

A Figura 5.2 mostra o número de mensagens trocadas em função do tipo de cada uma.

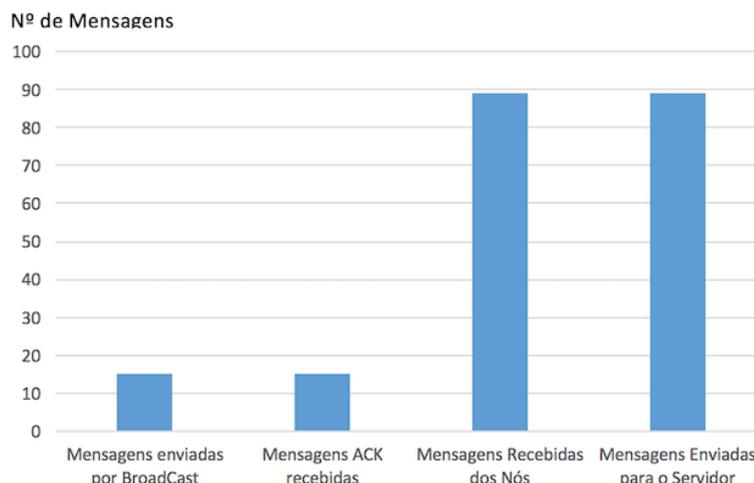


FIGURA 5.2: Diagrama de barras alusivo ao número de mensagens trocadas pelo *gateway* em ambiente de laboratório

Analisando o gráfico ilustrado acima, facilmente se percebe que em ambiente de laboratório a rede desenvolvida mostrou muita fiabilidade na troca de mensagens.

5.2 Resultados da Implementação do Sistema para Testes em Cenário Real

Depois de implementada com sucesso em laboratório, a rede desenvolvida foi sujeita a ligeiras adaptações, como seja a relativa à troca dos *led's* por bombas de água, para ser possível a sua implementação no GenviotGarden e sua posterior validação.

Com este fim, após divisão do jardim escolhido por zonas e implementação de um nó em cada uma delas, procedeu-se à verificação da comunicação entre os nós, para o que foram enviadas mensagens padrão do servidor com destino a todos os nós constituintes da rede e vice-versa. Apenas quando se confirmou que todas as mensagens padrão foram trocadas com sucesso é que se procedeu ao envio de mensagens relativas à rega do jardim.

Com todo o sistema implementado, foi possível ver que o envio das mensagens de irrigação foi feito com sucesso por parte do servidor para o *Gateway*, e deste para o atuador, como foi explicado anteriormente, na medida em que o atuador procedeu à redistribuição das instruções de rega pelas duas zonas, e que em consequência as

válvulas de cada uma das zonas mantiveram-se abertas durante o tempo descrito na mensagem.

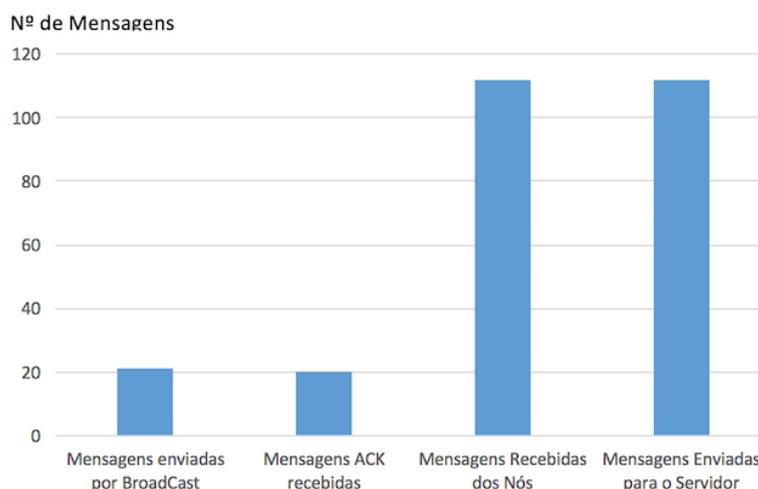


FIGURA 5.3: Diagrama de barras alusivo ao número de mensagens trocadas pelo *gateway* em ambiente real

Como pode ser visto na Figura 5.3, foram enviadas pelo *Gateway* via *Broadcast* 21 mensagens, mas apenas foram recebidas 20 Mensagens de *ConfirmACK*. Isto deveu-se ao facto de ter ocorrido um erro durante a transmissão da mensagem número 11. Como o *Gateway* não recebeu a mensagem de *ConfirmACK* correspondente ao ID 11 procedeu, posteriormente, quando detetou que o ID presente na mensagem de confirmação não correspondia ao ID da última mensagem enviada, ao envio de nova mensagem.

Além disso, é possível verificar que todas as mensagens enviadas pelos nós para o *Gateway* foram reencaminhadas com sucesso para o servidor.

Apesar de ter havido perda de uma mensagem, com estes números verificou-se que a WSN desenvolvida, a par da implementação em laboratório, foi implementada com sucesso num sistema real já existente e garantiu a fiabilidade na troca de mensagens, na medida em que o algoritmo desenvolvido assegurou que nenhuma informação fosse perdida.

5.3 Estudo da Potência de Recepção do Sinal numa Ligação LoRa Ponto-a-Ponto

Para avaliar a potência de recepção do sinal quando um pacote é trocado entre dois RFM69HCW, um transmissor e um recetor durante uma ligação LoRa ponto-a-ponto, bem como a distância máxima possível de uma ligação deste tipo, foram executados testes para determinação da distância máxima a que se poderia manter os rádios, e efetuadas medições da potência da recepção do sinal em cada interação de distância.

Este estudo foi feito em 4 cenários diferentes, entre eles:

1. Num corredor, onde a transmissão é feita em linha reta e sem qualquer tipo de obstáculo;
2. No mesmo corredor, mas desta vez com uma porta fechada entre o transmissor e o recetor;
3. Num jardim numa área sem obstáculos;
4. No mesmo jardim, mas desta vez numa área com bastantes obstáculos, mais propriamente árvores.



FIGURA 5.4: Cenários de teste 1, 2, 3 e 4 respetivamente

5.3.1 Arquitetura Utilizada para Medição da Potência de Recepção

A arquitetura que foi pensada e implementada para a execução destes testes foi bastante simples e pode ser facilmente aplicada a qualquer projeto de IoT em grande escala.

Esta está representada na Figura 5.5 e é constituída por um transmissor e um recetor, ambos contendo um RFM69HCW conectado a um Arduino.



FIGURA 5.5: Arquitetura utilizada para medição da potência de recepção

Para auxílio nos testes foi utilizada uma biblioteca para o Arduino, chamada *RadioHead*. Com esta biblioteca é possível usar uma API com algumas funções bastante simples. Na mesma existem 255 nós em 256 redes possíveis e o tamanho máximo da mensagem é 61 bytes. Finalmente, esta biblioteca garante a fiabilidade na troca de mensagens com *ACKs* automáticos com a função *sendWith-Retry()* [34].

Para chegar a um valor de potência de recepção mais fidedigno foram registadas 3 amostras da mesma para cada distância, ou seja, foram enviadas 3 mensagens intervaladas de 2 segundos entre os dois rádios e registada a potência com que o sinal chegava ao recetor.

5.3.2 Resultados dos Teste da Potência de Recepção do Sinal

- **Cenário 1**

Observando o gráfico da Figura 5.6, é possível verificar que à medida que a distância entre o transmissor e o recetor aumenta o valor da potência da recepção do sinal diminui. Isto acontece porque a potência do sinal decresce à medida que a distância aumenta, decréscimo progressivo que se revela mais drástico nos primeiros 10m.

Os testes foram efetuados até à distância máxima de 100m, por ser este o comprimento do corredor.

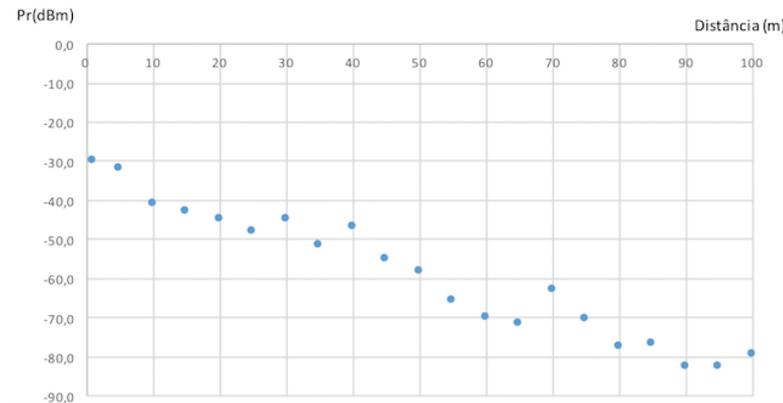


FIGURA 5.6: Potência recebida em função da distância (cenário 1)

- **Cenário 2**

Indo de encontro com o cenário anterior, verifica-se no gráfico da Figura 5.7 que com o aumento da distância entre transmissor e o recetor o valor da potência de recessão do sinal diminui.

Como neste caso há um obstáculo entre o transmissor e o recetor, era de esperar que o valor da potência de receção do sinal fosse menor para a mesma distância, em comparação com o cenário anterior. Esta diferença é mais acentuada para distâncias maiores, como por exemplo, no cenário 1 o valor da potência de receção do sinal para a distância de 70m é de cerca de -61dBm ao contrário do cenário 2 que é de cerca de -69dBm.

Apesar de haver um obstáculo a tapar a linha de vista entre o transmissor e o recetor, verifica-se que num ambiente *indoor* a comunicação é sempre assegurada.

Como estas amostras foram recolhidas no mesmo corredor do cenário anterior a limitação do tamanho do corredor manteve-se.

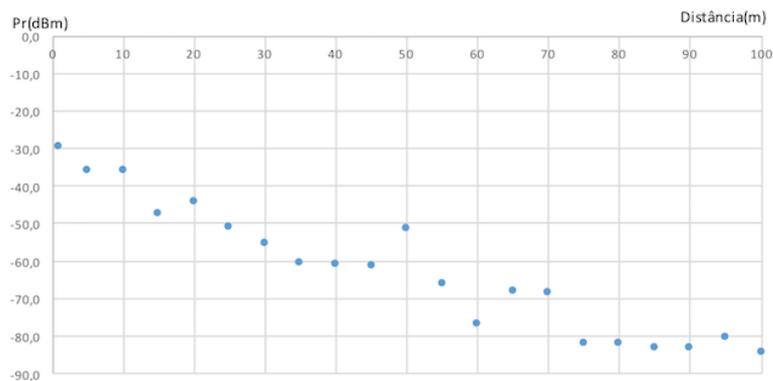


FIGURA 5.7: Potência recebida em função da distância (cenário 2)

• **Cenário 3**

Como pode ser observado na Figura 5.8, à semelhança dos cenários anteriores, o valor da potência de recepção diminuiu quando a distância entre o transmissor e o recetor aumentou.

Verifica-se uma diferença muito significativa entre a distância de 1m para 5m. Há uma queda brusca de aproximadamente -37dBm para -63dBm respetivamente. Isto não se verifica nos outros dois cenários anteriores devido à dispersão do sinal ser muito maior na rua do que dentro de um corredor onde as paredes serviam de refletores.

Neste cenário devido à presença de obstáculos, mais precisamente árvores, a comunicação foi perdida aos 80m.

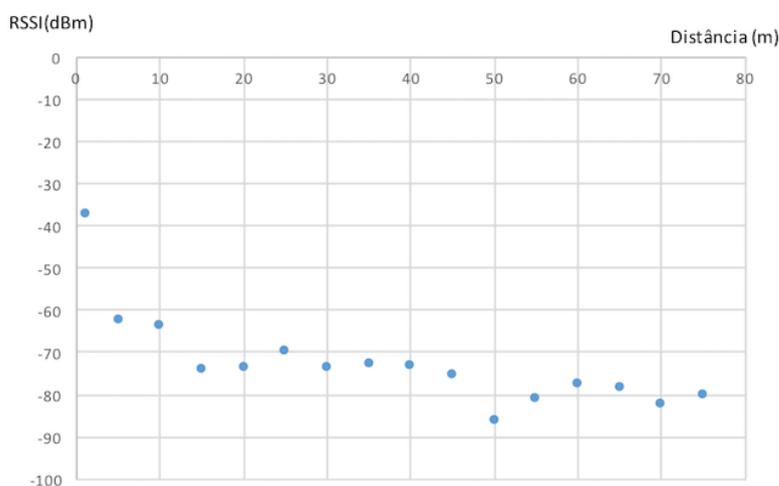


FIGURA 5.8: Potência recebida em função da distância (cenário 3)

• **Cenário 4**

O maior objetivo deste cenário era testar a distância máxima possível de comunicação entre dois RFM69HCW numa ligação LoRa ponto-a-ponto, já que no primeiro cenário havia a limitação do tamanho do corredor.

No gráfico da Figura 5.9 é possível ver que os maiores decréscimos da potência de receção de sinal são entre os primeiros 15m. Esta decresce de -30dBm para -76dBm. A partir daí o decréscimo apesar de progressivo mantém-se regular em função da maior distância.

Indo contra as especificações vistas anteriormente do RFM69HCW, é possível ver que a distância máxima possível de comunicação entre o transmissor e o recetor foi de apenas 190m, ao contrário dos 500m especificados pelo fabricante, do que se conclui que o módulo rádio, em referência, fica muito aquém do esperado. Valores mais altos poderiam ser obtidos se fosse implementada nos módulos rádio, tanto no transmissor como no recetor, uma antena diretiva.

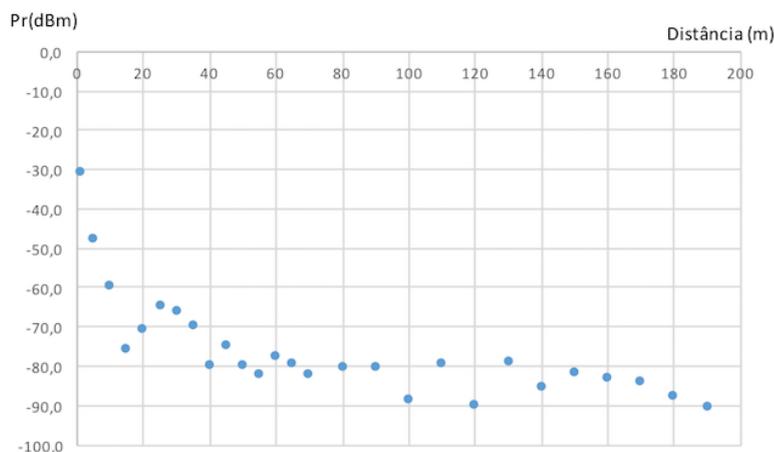


FIGURA 5.9: Potência recebida em função da distância (cenário 4)

5.3.3 Modelo de Propagação Empírico

Baseado nos resultados anteriormente obtidos, é possível formular um conjunto de equações representadas em (1), fazendo uma regressão linear representada na Figura 5.10, que poderá ser utilizada em futuras aplicações para verificar se uma WSN que utilize esse tipo de módulos rádio consegue operar em condições similares.

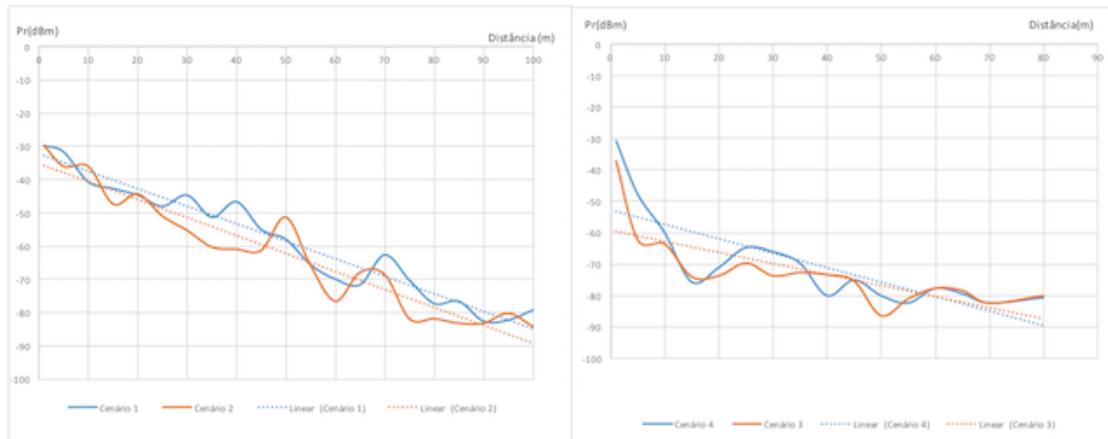


FIGURA 5.10: Modelos de propagação empírico (*Indoor e Outdoor*)

$$P_R = \alpha \times d + \beta \quad (5.1)$$

TABELA 5.1: Parâmetros do modelo de propagação para os 4 cenários apresentados

	α	β
<i>Indoor</i> sem obstáculos	-0,539	-35,271
<i>Indoor</i> com obstáculos	-0,527	-31,271
<i>Outdoor</i> sem obstáculos	-0,350	-63,217
<i>Outdoor</i> com obstáculos	-0,161	-63,217

Para cada cenário foram apresentadas 4 equações que derivam da fórmula apresentada em (5.1), em que o valor de cada variável está representado na tabela 5.1.

Capítulo 6

Conclusões e Trabalho Futuro

6.1 Principais Conclusões

A presente dissertação versa sobre o desenvolvimento e implementação de uma rede de sensores sem fios de baixo custo, capaz de ser aplicada a qualquer projeto de IoT, que procure a expansão para grande escala. Pretendia-se uma rede economicamente acessível, eficiente, versátil e fiável, pois estas características são fundamentais para qualquer projeto de IoT.

A fiabilidade do sistema na troca de mensagens foi assegurada através de um algoritmo, que controla o envio e receção daquelas, e procede ao reenvio das mesmas quando não confirma a receção correta da última mensagem enviada, impedindo a perda de mensagens entre os nós da rede e, conseqüentemente, a diminuição do desempenho do sistema, ou falha do mesmo. A WSN foi desenvolvida, implementada e testada em dois ambientes, sendo um deles um cenário perfeito (ambiente de laboratório) e o outro um cenário real (sujeito a distorções). Quando esta foi implementada e testada num ambiente de laboratório, como já se esperava, a comunicação ocorreu sem problemas, na medida em que não existiam fatores externos que pudessem influenciar o alcance dos nós da rede. Como foi visto na Secção 5.1 do capítulo 5, a fiabilidade da troca de mensagens foi 100% eficaz, pois todo o envio de mensagens com destino ao Servidor foi concretizado com sucesso. Verificou-se também que o algoritmo de troca de mensagens via Broadcast foi muito eficiente, pois todas as mensagens enviadas receberam a sua confirmação por ACK.

A WSN desenvolvida foi também implementada num cenário real, onde foi aplicada a um sistema de irrigação de jardins já existente, o Genviotgarden, que se pretendeu aperfeiçoar de modo a possibilitar uma irrigação diferenciada de zonas do jardim, em razão das suas características específicas, atinentes ao tipo de vegetação (relva, árvores, arbustos, flores), declive e elevação do terreno, luminosidade (zonas soalheiras e zonas de sombra), fazendo corresponder a irrigação efetivamente realizada às necessidades de rega concretas de cada uma das zonas, de modo a minimizar a quantidade de água utilizada.

Os objetivos traçados foram alcançados, pois a rega individualizada de cada zona foi feita durante o tempo presente na mensagem Water, enviada pelo servidor para o Gateway, sendo que neste cenário o algoritmo desenvolvido, a par da implementação em laboratório, mostrou-se fiável, na medida em que nenhuma mensagem enviada por Broadcast foi perdida.

Conclui-se, do exposto, ser a rede desenvolvida uma solução flexível e eficaz, capaz de se adaptar a projetos de IoT em grande escala e de melhorar a eficiência do sistema, de forma simples e com baixo custo. Além disso, o algoritmo de troca de mensagens apresentado para a comunicação LoRa ponto-a-ponto provou ser bastante fiável, por evitar a perda de mensagens entre os nós constituintes da rede, não prejudicando o seu desempenho.

Em relação ao estudo da potência de receção numa ligação LoRa ponto-a-ponto em função da distância entre o transmissor e o recetor, como esperado, foi possível verificar, como se viu nos quatro cenários de teste apresentados, que o valor da potência tem tendência a diminuir quando a distância entre eles aumenta.

Observou-se também que tanto nos cenários *indoor* como *outdoor* o valor da potência é sempre menor para a mesma distância quando existem obstáculos, em comparação com o cenário sem obstáculos. Isto deve-se ao facto dos obstáculos atenuarem o sinal.

Por fim, constatou-se que à mesma distância o valor de potência de receção é maior num cenário *indoor* do que no cenário *outdoor*, o que acontece porque naquele o teste é feito dentro de um edifício, num corredor, contexto em que a dispersão do sinal é quase nula, em virtude do mesmo ser refletido nas paredes e desta forma chegar facilmente ao recetor. Quando estes testes são feitos na rua (*outdoor*) muita potência do sinal é perdida na atmosfera, razão, pela qual, o valor da potência de receção do sinal é menor.

Depois dos testes efetuados, foi possível obter o modelo de propagação empírico, que define a qualidade do sinal em diferentes ambientes. Este pode ser utilizado em futuros projetos, com condições ambientais semelhantes, para verificar a qualidade do sinal.

6.2 Trabalho Futuro

Pretende-se, no futuro, implementar a rede desenvolvida num jardim ainda maior, com pelo menos 20 nós, e verificar se a fiabilidade na troca de mensagens se mantém e também validar modelos de propagação que tenham árvores, como obstáculos, e verificar a variabilidade que estas impõem, em função da época do ano.

Também seria interessante aplicar este sistema a ambientes novos, como sejam as habitações, para verificação dos consumos energéticos ou fugas de gás e, a grandes complexos de piscinas, na monitorização da qualidade da água.

Anexos

Anexo A

Artigos Científicos Publicados

LoRa peer-to-peer WSN architecture for large-scale IoT projects

1st Henrique Ferreira

Instituto de Telecomunicações)

ISCTE-Instituto Universitario de Lisboa

Lisboa, Portugal

hbfaa@iscte-iul.pt

2nd Andre Gloria

Instituto de Telecomunicações

ISCTE-Instituto Universitario de Lisboa

Lisboa, Portugal

afxga@iscte.pt

3rd Pedro Sebastião

Instituto de Telecomunicações

ISCTE-Instituto Universitario de Lisboa

Lisboa, Portugal

pedro.sebastiao@iscte-iul.pt

Abstract—This article presents a proposal for a Wireless Sensor Network (WSN) that can be applied to large scale Internet of Things projects, with multiple sensor and actuator nodes, where efficiency and reliability are crucial. The basis of this network is composed of a LoRa point-to-point connection between nodes assembled with low-cost components, such as Arduino compatible microcontrollers and sensors. The network architecture, message exchange algorithm and hardware are discussed in this paper, as well as the results of real case scenario implementation of the proposed system, in a garden irrigation system.

Index Terms—Internet of Things, Wireless Sensor Network, Arduino, ESP8266, Irrigation System, LoRa

I. INTRODUCTION

Nowadays, the need to connect everything to the Internet is growing, not only to send information to servers for processing and storage, but also to provide total control of physical devices on the Internet. It is estimated that by 2020, more than 200 billion smart devices will be connected to the Internet, providing 45% of all traffic just for Machine-to-Machine (M2M) communications [1] [2].

Designed to play a fundamental role in the daily lives of human beings, IoT (Internet of Things) is increasingly being integrated into every service or product, as it increases our quality of life [3]. Everywhere from Smart Homes, where users can control their thermostats or lights with a Smartphone, to mail shipping, where real time sensors can tell the condition of the package, IoT is improving the efficiency and overall satisfaction of process that until a few years back were done manually or not even considered [2] [4].

IoT, as shown in Figure 1, can be divided into six elements [5] that help us understand its true meaning and functionality. These are, identification, detection, communication, computing, services and semantics. Combining these elements offers IoT the capability of gathering, analyzing and distributing data, with a sensor network being the main feature of any IoT project.

With the growth of IoT, there is a need to expand the projects scale, being necessary to implement a reliable WSN (Wireless Sensor Network). Implementing a WSN in an existing garden irrigation system is capable of improving the its efficiency by reducing up to 30% in the amount of water used. In order to achieve this goal, the authors intend to apply the proposed WSN to an irrigation system by dividing the



Fig. 1. IoT Elements [5]

garden in small area, individualizing them according to their specific characteristics (eg. shade areas, types of vegetation, slope ...), so that water is carried out to the exact amount for the irrigation of the vegetation of each of these zones. To obtain this level of savings the system must work with a high level of precision and reliability and one of its key points is the communication, that, as can be seen in Figure 1, is one of the main elements of IoT, so a communication protocol capable of supporting the message exchange among the elements of the network is necessary.

Therefore, the authors will start this paper with a small comparison between the main wireless communication protocols that can be used in the construction of this type of networks with special attention in LoRa. Then it will be detailed a proposal for the design/development of a low cost wireless sensor network, that can be applied to any IoT project, with the main goal of increasing the system efficiency through a more robust wireless message exchange and overall software and hardware revision. Since having a trustworthy network is very important in a IoT system, because the lost of a single message can imply the failure of the process, also a proposal of an message exchange algorithm will be presented, alongside with the overall description of the WSN, including an explanation of the system architecture and its most important components. Finally, the results of the implementation in a real environment will be presented.

II. RELATED WORK

The literature shows a variety of projects [6]–[10] implementing IoT systems using WSN for irrigation control. Most of these systems use microcontrollers, such as Arduinos or Raspberry Pi, combined with a set of environmental sensors, which data is used to increase the garden or agricultural fields efficiency, including user comfort and water savings.

Most of them were implemented using single nodes with multiple sensors attached, using large amounts of wire to spread them across the fields. Also, some were found that use full WSN, such as Maksudjon Usmonov, who developed a low-cost WSN based on LoRa [11], capable of controlling an existing drip irrigation system with the aim of improving it in order to automate it by totally or partially eliminating human interaction, or Rajinder Kumar Math and Nagaraj Dharwadkar how developed a low-cost WSN using Arduino boards and ZigBee [12] in order to increase the efficiency and accuracy of an agricultural field based on real time data from air humidity and soil moisture sensors which are then sent to a server and can be monitored on the owner’s computer.

Our proposal for a WSN stands out from the others by the use of a wireless communication protocol for communication between the nodes of the network and also the ability to perform in any conditions or specifications without the need of hardware or software changes. This is a great advantage as there is no area limitation due to wires or the need to on-site maintenance, since everything is configured online. Also the implementation of our network will not be using ZigBee, as found in almost every project previously stated, because it entails unnecessary costs, using instead a new protocol with better specifications. Finally the WSN presented in this paper is dynamic, it is always possible to add new sensor nodes with different sensors from the existing ones without this affecting its performance.

III. COMMUNICATIONS PROTOCOLS

A good communication protocol is fundamental to the efficiency of an IoT project [13]. Although the word Internet is contained in IoT, the use of an Internet connection is not always needed. With the amplified use of Sensor Networks and applications in the most varied environments, the need of different protocols, both wired and wireless, is growing. Wireless protocols provide everything that is necessary for communication between devices and are increasingly used to create Sensor Networks. The most used protocols are Wi-Fi, ZigBee, Bluetooth and more recently technologies like LoRa have been progressively used.

TABLE I
MAJOR COMMUNICATION WIRELESS PROTOCOLS CHARACTERISTICS

Feature	Wi-Fi	Bluetooth	ZigBee	LoRa
Based Data Rate [Mbps]	11	1	0.25	0.11
Frequency [GHz]	2.4	2.4	2.4	0.868
Version	802.11b	4.0	-	-
Range [m]	1-100	10-100	10-100	2000
Nodes/Master	32	7	65540	15000
Power Consumption [mA]	100-350	1-35	1-10	1-10
Complexity	High	Medium	Medium	Low
Security	WPA/WPA2	128 bit	128 bit	128 bit

Observing Table I and paying attention to the previously mentioned characteristics, as well as the objective of the project, it is possible to conclude that the ideal protocol for the construction of a network of low-cost sensors is LoRa. In addition, this choice is also supported by the authors’ study in [13].

LoRa is a long range low power wireless technology platform that uses unlicensed radio spectrum in the industrial, scientific, and medical radio band [14]. LoRa aims to eliminate repeaters, reduce device cost, increase battery lifetime on devices, improve network capacity, and support a large number of devices. It is a physical layer used for long-range communication. For this to be possible, LoRa is based on CSS (Chirp Spread Spectrum) modulation that has the same low power characteristics as FSK modulation (present in most other wireless protocols), but with a significant increase in communication reach. With a single base station, it is possible to cover up to hundreds of square kilometers [15]. This communication protocol has more advantages than disadvantages [16]. It has the advantages of low power consumption, long communication range and highly secure, with 128-bit AES encryption. This protocol has just one disadvantage, low packet size, only 55 bytes per message (11 kbps) [16].

IV. WIRELESS SENSOR NETWORK

IoT depends on the ability to collect, send and process data, so the need for sensor networks is critical. These type of networks, as shown in Figure 2, are composed by a number of nodes that communicate in a multi-hop way, being these nodes any device equipped with sensors, such as smart-phones or cars or even amateur micro controllers, as Arduino or Raspberry Pi. These must be scalable, reliable and robust. Scalable due the high number of nodes that this network can have, reliable because in some cases these networks can send warnings in case something is wrong and robust because sensor nodes can be exposed to failures or bad environmental conditions [3]. As shown in Figure 2, there are three types of sensor nodes [17]:

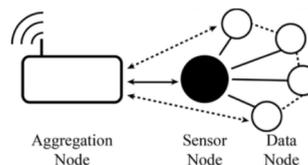


Fig. 2. Sensor Network Nodes [17]

- **Sensor Node** is the lowest level by a sensor and a communication system. They get the information and send it to another node.
- **Data Node** this node store information and send it to another node. Data storage can be used to perform autonomous tasks or follow information when communications are inactive. These nodes require additional devices with more features, such as storage, where SD (Secure Digital) or database cards are usually used.
- **Aggregation Node** dont read the sensor data, they normally have a communication device and a Gateway to the server. They are used to collect data from one or more sensor nodes and send it to the servers.

V. SYSTEM ARCHITECTURE

An high level view of the proposed system architecture is represented in Figure 3. As can be seen, it is constituted by a gateway, actuators, and sensor nodes. This system is flexible and adaptable, as described in the following sections, it is possible to add several sensor nodes and actuators depending on the size of the garden and consequently the number of zones it has.

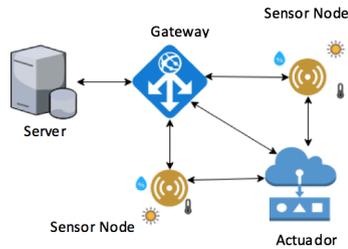


Fig. 3. System Architecture

The network is controlled by a single gateway, acting as an aggregation node, that is in constant communication with an online server, using Message Queuing Telemetry Transport (MQTT), a messaging protocol suitable for small, lowpower, low-power, low-bandwidth devices [5], [18], via an WiFi connection, and is complemented with a set of sensor and actuator nodes, that communicate with the gateway and between them using a LoRa peer-to-peer connection.

As said before, the proposed architecture is composed by three types of nodes. These nodes are described, in detail, in the following subsections.

A. Gateway

Composed by only two elements, as shown in Figure 5 and 13, an ESP8266, a low-cost microcontroller capable of providing WiFi and a TCP/IP stack [19], and a RFM69HCW, a simple radio module capable of transmitting data between a multi-point network, with individual node addresses and encryption, using the LoRa protocol to achieve a range of approx 500 meters [20]. This node has, as a key feature, the ability to communicate with the server, as it is the only element with an Internet connection. This feature make it responsible for exchange the messages between the server and the multiple nodes in the network, from system configurations or irrigation times sent from the server to real-time sensor data to store in the server, being this node the main responsible for the broadcast messaging algorithm. The gateway must be positioned in a strategic position where it can be in both range of a good quality WiFi Access Point and all the other nodes in the network.

B. Actuator Node

Being responsible for controlling actuators, such as water pumps, motors or lights, makes this the most complex node

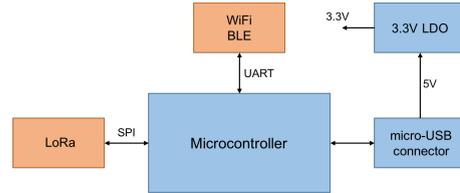


Fig. 4. Gateway node block diagram

in the network. Figure 8 and 12 shows the main components, an Arduino microcontroller, a RFM69HCW, a shift-register, capable of extending the microcontroller digital ports between 4 and 32, and a Darlington Transistor Array, responsible for providing the necessary energy to the actuator, since the Arduino can only provide 5V@40mA and most actuators require at least 12/24V@100mA. As it needs an output voltage supply, to guarantee the correct functionality of its actuators, this node is intended to be placed near electric socket or attached with a solar panel and battery system. Able to communicate with the other nodes in a bidirectional way, it is capable of turn on/off actuators based on sensor data or from user input.

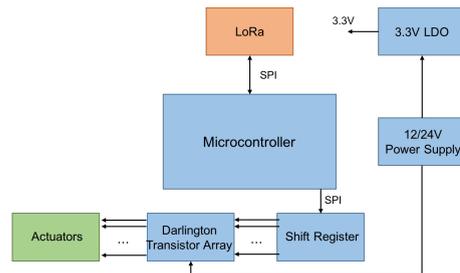


Fig. 5. Actuator node block diagram

C. Sensors Node

Composed, as presented in Figure 7 and 11, by an Arduino microcontroller, an RFM69HCW and an array of sensors, its main objective is to harvest information about the environment and send them to the server for processing or when required by the actuator node. This node, since it is only awake for short periods of time, is powered by batteries.

VI. MESSAGE EXCHANGE ALGORITHM

As previously stated, the nodes of our proposed system communicate with each other through a peer-to-peer LoRa connection, using the RFM69HCW modules. Combined with a ready-to-use Arduino library, RadioHead [21], it's possible to assign each node with a unique address for identification inside the network. This API allows the nodes to communicate in two different ways:

- Send a message directly to a specific node, using the destination unique address;

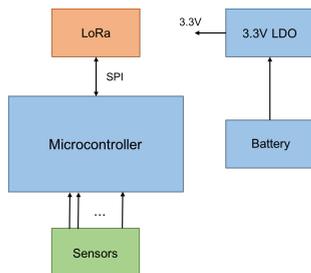


Fig. 6. Sensor node block diagram

- Send a broadcast message to every node in range of communication.

In the first scenario, the API ensures some fidelity in the exchange of messages, with an automatic ACK response, whereas the second one does not guarantee that the message are delivered to the intended node. Also, the API provides encrypted transactions and a sleep function, to reduce the nodes power consumption.

Since our network makes use of both communication methods, from gateway to other nodes it uses broadcast messages and vice-versa uses the gateway unique address, only half of our system is reliable based on the raw API. As stated before, in a IoT system, reliability and efficiency are key features, so in order to guarantee that all message are delivered an acknowledgment message exchange algorithm was implemented on top of the broadcast API mechanism. This way is possible to ensure that the intended nodes received the broadcast message. To implement this algorithm, a Firsts In, Firsts Out (FIFO) method was applied, as shown in Figure 8.

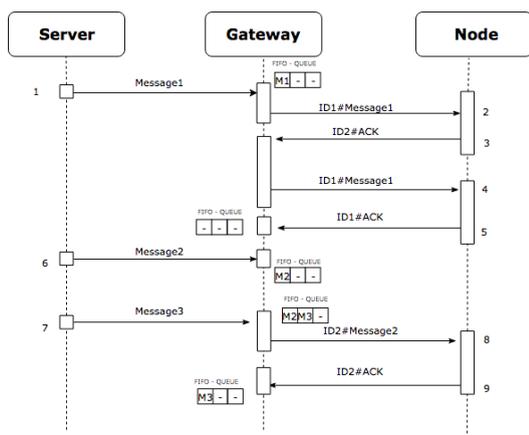


Fig. 7. Algorithm Diagram

FIFO, is an algorithm of organizing and manipulating a data buffer, in which the oldest input is processed first [22]. In a

FIFO queue, the elements are queued and removed in order of arrival. The fundamental idea of the queue is that we can only insert a new element at the end of the queue and we can only remove the element from the beginning [22].

Represented in Figure 8 are the message exchange algorithm between the server, gateway and a generic node. Also the queue organization is represented. The algorithm is performed as follows:

- 1) A message comes from the Server and it's received in the gateway. As it's possible to see this message is the first in the queue;
- 2) The gateway selects the message that is at the head of the queue and adds an id that is general. Then send it by broadcast;
- 3) The node receives a message and send back an ACK message with the ID. When the gateway receives an ACK message compares the ID that comes in this message with the last ID sent.
- 4) As the ID received isn't equals to the last ID sent the gateway sends the message again;
- 5) The node receives a message and send back an ACK message with the ID. This time as the ID that came in the ACK message is the same as the last ID sent, the message is removed from the queue and the ID is incremented;
- 6) The Server send a message to the gateway. As the queue was empty the new message is placed on the head of it;
- 7) The Server send another message to the gateway and this message goes behind the other;
- 8) Again the gateway selects the message that is at the head of the queue and adds the general id. Then send it by broadcast;
- 9) The nodes send an ACK message with the ID received and send it back. As the ID received by the gateway is equal to the last one sent the message is removed from the queue, now message 3 is at the head of the queue.

In this way, it is possible to obtain the necessary reliability to guarantee the correct functioning of the proposed system in terms of the exchange of messages within the network, ensuring that there are no lost messages.

VII. IMPLEMENTATION

Finished the design of our proposed adaptive WSN, capable of controlling large IoT projects with high performance and efficiency, it was necessary to test our design in a real environment. After the implementation and some laboratory tests, that ensure the reliability of our system, a real case scenario was chosen to conduct a study. Being the team main investigation areas IoT for sustainability and natural resources management, a garden irrigation system was the ideal project to implement the proposed system. The goal of this implementation was to control the amount of water used, based on real time sensor data, that helps calculate the amount of water an irrigation zone really needs to stay healthy. Since the garden has 3 irrigation zones, each with a different set of plants and environmental changes among them (shade zones or

ground elevations), to achieve a perfect irrigation system each zone must have different irrigation times based on real time conditions. So, to achieve the proposed goal, a network of 3 nodes, plus a gateway, were included in the garden. Figures 9 and 10 shows the chosen garden and the different zones where nodes were inserted.



Fig. 8. Satellite Image of the Garden



Fig. 9. Garden

In zones 1 and 2 sensors nodes attached with air temperature and humidity and soil moisture sensors were placed. DHT22, a low cost temperature and humidity sensor [23] that allow temperature readings between -40 to +80 degrees Celsius and humidity between 0 to 100%, being very easy to use with Arduino, Raspberry and other micro controllers because it has only 1 pin with digital output. The soil moisture sensor, a simple breakout for measuring the moisture in soil and similar materials. This one has two pads, that work like probes. The conductivity of these pads will be greater as more water is in the soil which will lead to a higher value of SIG output [24]. The distance between nodes is relatively big so that environmental conditions in these zones can make a difference in the overall efficiency of the irrigation system.

Zone 3 was the chosen place for the actuator node, since it was near the irrigation system valve control. Connecting it to the actuator node, granted the controlled of individual valves

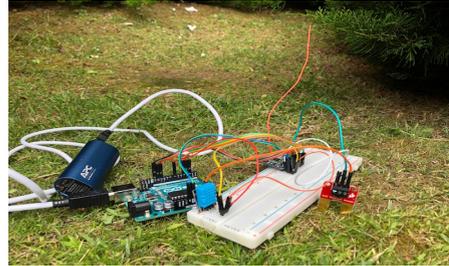


Fig. 10. Sensor 1

for each irrigation zones where a sensor node was placed. The actuator collects the environmental data from the sensors and then checks them. After analyzing the data, the irrigation times of each zone are adjusted according to the need of each one, in the last case the irrigation time can be adjusted to 0s, not having to irrigate it.

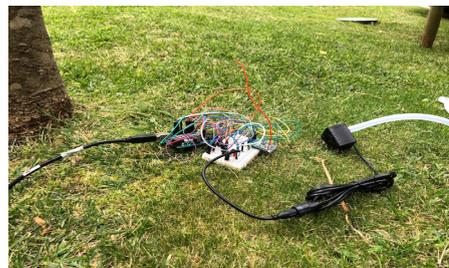


Fig. 11. Actuator Node at irrigation zone

Finally, the gateway was placed inside the surrounding building, near an WiFi Access Point, therefor access to the Internet is always guaranteed, and in range with all the network nodes.

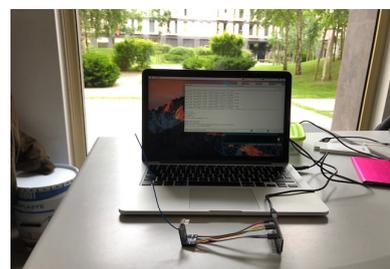


Fig. 12. Gateway indoor

VIII. RESULTS AND DISCUSSION

After the implementation of the complete network and the required adjustments to adapt to the irrigation project, the

system was tested. Firstly the objective was to conclude if all the nodes can communicate with each other, so a set of standard message was sent from the server to the individual nodes and vice versa. Then, the full system adapted to the irrigation scenario was tested.

With the whole system implemented, it was possible to see that the sending of irrigation messages was done successfully by the gateway. Subsequently this message was sent back to the actuator that turned on the valves during the time described in the message.

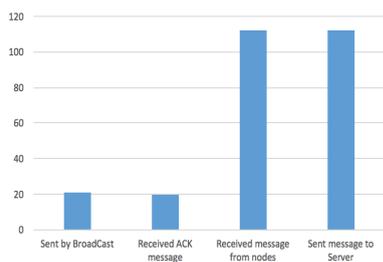


Fig. 13. Message sent and received by gateway

As it can be seen in Figure 14, were sent by broadcast 21 message and received 20 ACK by the gateway, this is due to the fact that an error occurred during the sending of message number 11. As the gateway did not receive the ACK corresponding to ID 11 it had to send the message again, the difference between the broadcast messages and the received ACK is due to this. Also, it's possible to verify that all messages sent by the nodes to the gateway have been successfully forwarded to the server.

IX. CONCLUSIONS AND FUTURE WORK

In this article, a design for a new adaptive WSN for application within large-scale IoT projects, with a focus on exchanging messages through a reliable LoRa peer-to-peer communication algorithm, was presented and the development and implementation of the proposed system was tested in a real case scenario. After the application of the proposed architecture in a real irrigation system, it was possible to conclude that the system can achieve the proposed goal, being a reliable solution for controlling large-scale IoT projects, improving the efficiency of the environment where it was placed in a simple and low-cost way. Also the proposed Message Exchange Algorithm for LoRa peer-to-peer communication was proved to be reliable, with no lost messages between all nodes. In the future, the authors would like to continue testing and tuning this system within other environments, such as household consumptions or swimming pool controls. The implementation in a bigger garden its also been planned, with at least 20 nodes.

ACKNOWLEDGMENT

The authors would like to acknowledge ISCTE-Instituto Universitario de Lisboa and Instituto de Telecomunicações for providing the components and conditions to support the work developed and presented in this paper.

REFERENCES

- [1] J. Gantz and D. Reinsel, "THE DIGITAL UNIVERSE IN 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East," *Idc*, vol. 2007, no. December 2012, pp. 1–16, 2012.
- [2] D. Lee, D. Lough, S. Midkiff, N. Davis, and P. Benchoff, "The next generation of the internet: aspects of the ipv6," *IEEE Network*, vol. 12, no. April, pp. 28–33, 1998.
- [3] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [4] D. Evans, "The Internet of Things - How the Next Evolution of the Internet is Changing Everything," *CISCO white paper*, no. April, pp. 1–11, 2011.
- [5] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [6] R. R. A. "Smart Agro System Using Wireless Sensor Networks," pp. 400–403, 2017.
- [7] M. N. Rajkumar, S. Abinaya, and V. V. Kumar, "Intelligent irrigation system - An IOT based approach," *IEEE International Conference on Innovations in Green Energy and Healthcare Technologies - 2017, IGEHT 2017*, pp. 1–5, 2017.
- [8] A. Mondal, I. S. Misra, and S. Bose, "Building a low cost solution using wireless sensor network for agriculture application," *2017 International Conference on Innovations in Electronics, Signal Processing and Communication (IESC)*, pp. 61–65, 2017.
- [9] T. Savi, "WSN Architecture for Smart Irrigation System," pp. 1–4, 2018.
- [10] V. N. R. Vjiet, "Implementation of Wireless Sensor Network for Real Time Overhead Tank Water Quality Monitoring," *2017 IEEE 7th International Advance Computing Conference Implementation*, pp. 546–551, 2017.
- [11] M. Usmonov and F. Gregoretti, "Design and Implementation of a LoRa Based Wireless Control for Drip Irrigation Systems," 2017.
- [12] R. K. Math and N. V. Dharwadkar, "A Wireless Sensor Network Based Low Cost and Energy Efficient Frame Work for Precision Agriculture," *International Conference on Nascent Technologies in the Engineering Field (ICNTE-2017)*, 2017.
- [13] A. Gloria, F. Cercas, and N. Souto, "Comparison of communication protocols for low cost Internet of Things devices," *South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference, SEEDA-CECNSM 2017*, 2017.
- [14] A. J. Wixted, P. Kinnaird, H. Larjani, A. Tait, A. Ahmadiania, and N. Strachan, "Evaluation of LoRa and LoRaWAN for wireless sensor networks," *Proceedings of IEEE Sensors*, vol. 0, pp. 5–7, 2017.
- [15] J.-s. Lee, Y.-w. Su, and C.-c. Shen, "A Comparative Study of Wireless Protocols :," *IECON Proceedings (Industrial Electronics Conference)*, pp. 46–51, 2007.
- [16] T. M. Workgroup, "What is it?," no. November, 2015.
- [17] C. Bell, *Beginning Sensor Networks with Arduino and Raspberry Pi*. Apress, 1st ed., 2013.
- [18] D. Thangavel, X. Ma, A. Valera, H. X. Tan, and C. K. Y. Tan, "Performance evaluation of MQTT and CoAP via a common middleware," *IEEE ISSNIP 2014 - 2014 IEEE 9th International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Conference Proceedings*, no. April, pp. 21–24, 2014.
- [19] Adafruit, "Adafruit HUZZAH ESP8266 breakout." [Online] Available: <https://learn.adafruit.com/adafruit-huzzah-esp8266-breakout/overview>, (visited 15/06/2018).
- [20] Adafruit, "Adafruit RFM69HCW Transceiver Radio Breakout - 433 MHz - RadioFruit." [Online] Available: <https://www.adafruit.com/product/3071>, (visited 10/06/2018).
- [21] LowPowerLab, "RFM69 Library." [Online] Available: <https://github.com/LowPowerLab/RFM69>, (visited 10/06/2018).
- [22] Robert L. Curse, *Data structures and program design*. second ed., 1984.
- [23] L. Aosong Electronics Co, "Dht22 (Am2302)," vol. 22, pp. 1–10, 2015.

[24] SparkFun, "SparkFun Soil Moisture Sensor." [Online] Available: <https://www.sparkfun.com/products/13322>, (visited 15/06/2018).

Bibliografia

- [1] D. Evans, “The Internet of Things - How the Next Evolution of the Internet is Changing Everything,” *CISCO white paper*, no. April, pp. 1–11, 2011. [Online]. Available: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:The+Internet+of+Things+-+How+the+Next+Evolution+of+the+Internet+is+Changing+Everything#0>
- [2] D. Lee, D. Lough, S. Midkiff, N. Davis, and P. Benchoff, “The next generation of the internet: aspects of the ipv6,” *IEEE Network*, vol. 12, no. April, pp. 28–33, 1998.
- [3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [4] A. Gloria, F. Cercas, and N. Souto, “Comparison of communication protocols for low cost Internet of Things devices,” *South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference, SEEDA-CECNSM 2017*, 2017.
- [5] D. Thangavel, X. Ma, A. Valera, H. X. Tan, and C. K. Y. Tan, “Performance evaluation of MQTT and CoAP via a common middleware,” *IEEE ISSNIP 2014 - 2014 IEEE 9th International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Conference Proceedings*, no. April, pp. 21–24, 2014.
- [6] N. De Caro, W. Colitti, K. Steenhaut, G. Mangino, and G. Reali, “Comparison of two lightweight protocols for smartphone-based sensing,” *IEEE SCVT*

- 2013 - *Proceedings of 20th IEEE Symposium on Communications and Vehicular Technology in the BeNeLux*, pp. 0–5, 2013.
- [7] E. Ferro and F. Potortì, “Bluetooth and Wi-Fi wireless protocols: A survey and a comparison,” *IEEE Wireless Communications*, vol. 12, no. 1, pp. 12–26, 2005.
- [8] J.-s. Lee, Y.-w. Su, and C.-c. Shen, “A Comparative Study of Wireless Protocols :,” *IECON Proceedings (Industrial Electronics Conference)*, pp. 46–51, 2007.
- [9] T. M. Workgroup, “What is it?” no. November, 2015.
- [10] A. F. X. d. Glória, “The use of Sensor Networks to create smart environments,” 2017. [Online]. Available: <https://repositorio.iscte-iul.pt/handle/10071/14628>
- [11] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [12] Charles Bell, *Beginning Sensor Networks with Arduino and Raspberry Pi*.
- [13] M. Banzi, *Getting Started with Arduino*, 2nd ed., O’Reilly, Ed. Make:Books, 2011.
- [14] E. Community Wiki, “Getting Started with ESP8266.”
- [15] A. Grygoruk and J. Legierski, “IoT gateway – implementation proposal based on Arduino board,” vol. 8, pp. 1011–1014, 2016. [Online]. Available: <https://fedcsis.org/proceedings/2016/drp/283.html>
- [16] Amazon, “Arduino Uno R3 Microcontroller.” [Online]. Available: [https://www.amazon.com/Arduino-Uno-R3-Microcontroller-A000066/dp/B008GRTSV6\(visited27/09/2018\)](https://www.amazon.com/Arduino-Uno-R3-Microcontroller-A000066/dp/B008GRTSV6(visited27/09/2018))
- [17] A. Nayyar and V. Puri, “A review of Arduino board’s, Lilypad’s & Arduino shields,” *2016 International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 1485–1492, 2016.
- [18] AI Thinker, “ESP-12E WiFi Module.” [Online]. Available: [https://www.kloppenborg.net/images/blog/esp8266/esp8266-esp12e-specs.pdf\(visited27/09/2018\)](https://www.kloppenborg.net/images/blog/esp8266/esp8266-esp12e-specs.pdf(visited27/09/2018))

- [19] Adafruit, “Adafruit RFM69HCW Transceiver Radio Breakout - 433 MHz - RadioFruit.” [Online]. Available: <https://www.adafruit.com/product/3071>(visited27/09/2018)
- [20] Aosong, “Temperature and Humidity Module,” *Temp, Humidity and Dew point measurement expert*, vol. 1, pp. 1–11, 2015.
- [21] Maxim Integrated, “Datasheet DS18B20,” *Maxim Integrated*, vol. 92, p. 20, 2015. [Online]. Available: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf><https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf><https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- [22] L. Self-heating and L. I. Output, “LM35 Precision Centigrade Temperature Sensors,” no. November, pp. 1–13, 2013. [Online]. Available: <http://www.ti.com/lit/ds/symlink/lm35.pdf>(visited27/09/2018)
- [23] F. B. Diagram, P. I. N. Configurations, and G. Description, “Low Voltage Temperature Sensors,” 2005.
- [24] SparkFun, “SparkFun Soil Moisture Sensor.” [Online]. Available: <https://www.sparkfun.com/products/13322>(visited27/09/2018)
- [25] P. Rog, Z. Surwhfwv, W. K. H. Qlfnho, K. D. V. Vhyhudo, D. Ryhu, P. Frqyhqwlrqdo, D. Q. G. Fkhdshu, V. Sodwqlqjv, V. Dv, and F. Srlqvw, “High Sensitivity Moisture Sensor.”
- [26] E. Efficiency, E. Circuit, L. Dependent, and R. Circuits, “Light dependent resistor (ldr),” *Energy*, pp. 1–3, 2010.
- [27] Everlight, “Ambient Light Sensor Surface - ALS-PT19,” 2013. [Online]. Available: <https://www.adafruit.com/products/2748>(visited27/09/2018)
- [28] R. R. A, “Smart Agro System Using Wireless Sensor Networks,” pp. 400–403, 2017.
- [29] Syaza Norfilsha Binti Ishak, “Smart Home Garden Irrigation System With Raspberry Pi,” *Ieee*, vol. 16, no. June, p. 24, 2008.
- [30] M. N. Rajkumar, S. Abinaya, and V. V. Kumar, “Intelligent irrigation system - An IOT based approach,” *IEEE International Conference on Innovations*

- in Green Energy and Healthcare Technologies - 2017, IGEHT 2017*, pp. 1–5, 2017.
- [31] A. Mondal, I. S. Misra, and S. Bose, “Building a low cost solution using wireless sensor network for agriculture application,” *2017 International Conference on Innovations in Electronics, Signal Processing and Communication (IESC)*, pp. 61–65, 2017. [Online]. Available: [http://ieeexplore.ieee.org/document/8071865/\(visited27/09/2018\)](http://ieeexplore.ieee.org/document/8071865/(visited27/09/2018))
- [32] T. Savi, “WSN Architecture for Smart Irrigation System,” pp. 1–4, 2018.
- [33] V. N. R. Vjiet, “Implementation of Wireless Sensor Network for Real Time Overhead Tank Water Quality Monitoring,” *2017 IEEE 7th International Advance Computing Conference Implementation*, pp. 546–551, 2017.
- [34] L. Felix Rusu, “RFM69 Library.”