



## Sistema inteligente de comunicação longo alcance

**JORGE MIGUEL GOMES TAVARES**

novembro de 2018

# SISTEMA INTELIGENTE DE COMUNICAÇÃO LONGO ALCANCE

Jorge Miguel Gomes Tavares



Departamento de Engenharia Eletrotécnica  
Instituto Superior de Engenharia do Porto

2018



Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha da Unidade Curricular de Tese / Dissertação (TEDI), do 2º ano, do Mestrado em Engenharia Eletrotécnica e de Computadores

Candidato: Jorge Miguel Gomes Tavares, N.º 1110831, 1110831@isep.ipp.pt

Orientação científica: Lino Manuel Batista Figueiredo, lbf@isep.ipp.pt

Empresa: Smart-Helix

Supervisão: Bruno Canizes, João Soares



Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

13 de Novembro de 2018



## *Agradecimentos*

Aos meus pais agradeço toda o apoio incondicional nos piores e melhores momentos tendo sempre disponibilidade para me motivar e ajudar durante todo o trabalho. À minha irmã, agradeço a capacidade de me transmitir perseverança e sentido de responsabilidade durante cada etapa do trabalho.

Quero agradecer ao Eng.<sup>o</sup> João Soares e ao Eng.<sup>o</sup> Bruno Canizes pela orientação, ajuda técnica e conhecimentos partilhados bem como a resolução de problemas que foram surgindo ao longo do tempo. Também quero agradecer ao Eng.<sup>o</sup> Sérgio Neves pela oportunidade que me deu e pela disponibilidade que prestou na partilha do material.

Venho também agradecer ao Eng.<sup>o</sup> Lino Figueiredo pela sua orientação e colaboração na resolução de problemas, facultando conhecimentos técnicos que facilitaram a realização deste projeto.

Quero agradecer em especial à minha namorada Joana Soares, que sempre me apoiou nos momentos mais difíceis e festejou comigo nas minhas pequenas vitórias diárias. Este parágrafo é muito pequeno para descrever a sua importância na minha vida. Sinto que devo muito a ela por ter me aturado e contribuído de forma positiva com a sua presença e carinho.



## Resumo

A Internet das Coisas é uma revolução tecnológica que consiste na ideia de que todos os objetos do nosso quotidiano estão ligados à Internet. Isto acontece com a fusão do mundo real com o mundo virtual, permitindo interconetar diversos aparelhos e objetos diferentes, ajudando a facilitar e organizar a vida das pessoas. Este projeto consiste na implementação de uma tecnologia, que permita uma comunicação *mesh* a distâncias de pelo menos dois quilómetros.

Inicialmente foi elaborado um estudo profundo sobre o funcionamento desta comunicação e uma pesquisa sobre as alternativas para o conseguir. A comunicação funciona através de três estações, uma estação principal que recebe a informação de duas estações secundárias. Cada estação possui uma placa de sensores para recolher informações dos sensores e uma placa de comunicações para transmitir os dados. O processo de transmissão sem fios é assegurado por módulos rádio que enviam e recebem os dados.

A informação enviada para a estação principal corresponde a uma mensagem com os dados dos sensores. Depois esta mensagem é enviada para a Internet, sendo analisada para outros processos. As mensagens que circulam na rede podem ser mapeadas com rotas diferentes, utilizando nós intermédios para transportar a informação até ao destino. Depois de chegar à estação principal, o sistema escuta por novas mensagens e aguarda um tempo definido para depois adormecer.

Este sistema pretende complementar a recolha e tratamento de dados, possibilitando a comunicação de dispositivos em áreas amplas. A implementação do trabalho em ambiente industrial ou agrícola, irá favorecer o utilizador que não terá de se deslocar constantemente para a zona onde estará instalado. As potencialidades de comunicação poderão permitir o controlo de dispositivos através do acesso remoto por outras aplicações.

## Palavras-chave

Comunicação Longo Alcance, LoRa, LPWAN, Rede Mesh, Microcontrolador



## **Abstract**

The Internet of Things is a technological revolution that consists of the idea that all objects of our daily life are connected to the Internet. This happens by merging the real world with the virtual world, allowing to interconnect several different devices and objects. This project consists of the implementation of a technology that allows mesh communication at distances of at least two kilometers.

Initially it was elaborated an study about all the mechanisms of this communication and a research on alternatives to achieve this. Communication works through three stations, a main station that receives information from two secondary stations. Each station has a sensor board to collect sensor information and a communications board to transmit the data. The wireless transmission process is ensured by radio modules that send and receive data.

The information sent to the main station consists of a message with data from sensors. This message is then sent to the Internet and analyzed for other procedures. The messages that flow in the network can be mapped with different routes, using intermediate nodes to carry the information to the destination. After reaching the main station, the system listens for new messages and waits for a set time to fall asleep.

This system aims to complement the collection and processing of data, enabling the communication of devices in wide areas. The implementation of work in an industrial or agricultural environment will favor the user who won't have to constantly move to the area where it will be installed. Communication capabilities may also allow device control through remote access by other applications.

## **Keywords**

Long Range Communication, LoRa, LPWAN, Mesh Network, Microcontroller



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contextualização . . . . .	1
1.2	Objetivos . . . . .	2
1.3	Calendarização . . . . .	3
1.4	Organização . . . . .	4
<b>2</b>	<b>Conceitos Teóricos</b>	<b>5</b>
2.1	Princípios das comunicações rádio . . . . .	5
2.1.1	Espectro eletromagnético . . . . .	7
2.1.2	Fórmula de transmissão de Friis . . . . .	8
2.1.3	<i>Free Space Path Loss (FSPL)</i> . . . . .	9
2.1.4	<i>Link Budget</i> . . . . .	11
2.1.5	Zona de Fresnel . . . . .	12
2.2	Licenciamento e disponibilidade espectral . . . . .	13
2.2.1	Dispositivos de Pequena Potência (SRDs) . . . . .	14
2.2.2	Bandas ISM . . . . .	15
2.3	Antenas . . . . .	18
2.4	Topologia de rede . . . . .	20
2.5	Tecnologias LPWAN . . . . .	22
2.5.1	Contextualização . . . . .	22
2.5.2	Soluções comerciais . . . . .	24
2.6	Módulos de comunicação RF . . . . .	28

2.6.1	NRF24L01 . . . . .	28
2.6.2	RFM69HCW . . . . .	30
2.6.3	SX1278 . . . . .	33
2.6.4	Análise comparativa . . . . .	36
2.7	Ferramenta de previsão <i>Radio Mobile</i> . . . . .	37
<b>3</b>	<b>Arquitetura do sistema</b>	<b>43</b>
3.1	Requisitos do sistema . . . . .	43
3.2	Diagrama do sistema . . . . .	44
3.3	Hardware . . . . .	45
3.3.1	Módulo ESP8266 (NodeMCU) . . . . .	45
3.3.2	Módulo A6 GSM mini . . . . .	48
3.3.3	Sensores e conversor ADS1115 . . . . .	49
3.4	Software . . . . .	51
<b>4</b>	<b>Desenvolvimento</b>	<b>59</b>
4.1	Contextualização . . . . .	59
4.2	Hardware . . . . .	60
4.2.1	Placa de Comunicações . . . . .	65
4.2.2	Placa de Sensores . . . . .	66
4.3	Software . . . . .	67
4.3.1	Estação principal . . . . .	70
4.3.2	Estação secundária . . . . .	75
<b>5</b>	<b>Testes e resultados</b>	<b>81</b>
5.1	Cenários de Teste . . . . .	81
5.2	Ligação Ponto a Ponto . . . . .	82
5.2.1	Hardware . . . . .	82
5.2.2	Software . . . . .	84
5.2.3	Simulação e previsão . . . . .	93

5.2.4	Resultados . . . . .	98
5.2.5	Sumário . . . . .	102
5.3	Rede Mesh Simples . . . . .	105
5.3.1	Hardware . . . . .	105
5.3.2	Software . . . . .	106
5.3.3	Resultados . . . . .	112
5.4	Rede Mesh Integrada . . . . .	116
5.5	Análise do consumo de corrente . . . . .	121
5.6	Análise económica do sistema . . . . .	126
<b>6</b>	<b>Considerações finais</b>	<b>129</b>
6.1	Dificuldades . . . . .	129
6.2	Apreciação final . . . . .	130
6.3	Trabalho futuro . . . . .	131
	<b>Referências Documentais</b>	<b>138</b>
	<b>Anexo A</b>	<b>139</b>



# Lista de Figuras

Figura 1.1	Calendarização do projeto . . . . .	4
Figura 2.1	Representação de uma onda eletromagnética. [1] . . . . .	6
Figura 2.2	Modelo generalizado de um sistema transmissor-recetor. [2] . . . . .	7
Figura 2.3	Espectro eletromagnético [3] . . . . .	7
Figura 2.4	Classificação das ondas rádio. [4] . . . . .	8
Figura 2.5	Representação tridimensional da zona de Fresnel. [5] . . . . .	12
Figura 2.6	Representação mundial da alocação das principais frequências ISM. [4] . . . . .	15
Figura 2.7	Antena monopolo e padrão de radiação [6]. . . . .	18
Figura 2.8	Antena dipolo e seu padrão de radiação [6]. . . . .	19
Figura 2.9	a) Estrutura de rede árvore e b) em estrela. [7] . . . . .	21
Figura 2.10	a) Estruturas de rede mesh interligada totalmente e b) interligada parcialmente. [7] . . . . .	21
Figura 2.11	Comparação entre LPWAN e outras tecnologias sem fios. [8] . . . . .	23
Figura 2.12	Estrutura da rede LoRa/LoRaWAN. [9] . . . . .	25
Figura 2.13	Chaves de segurança LoRaWAN [10] . . . . .	26
Figura 2.14	Estrutura da rede Sigfox [8] . . . . .	27
Figura 2.15	Módulo NRF24L01 [11] . . . . .	29
Figura 2.16	Estrutura do pacote de dados NRF24L01 [12] . . . . .	29
Figura 2.17	Módulo RFM69HCW [13] . . . . .	31
Figura 2.18	Formato da mensagem RFM69 com comprimento fixo. [14] . . . . .	32
Figura 2.19	Módulo SX1278 [14] . . . . .	33

Figura 2.20	Taxas de codificação e as respectivas sobrecargas nos dados. [15]	34
Figura 2.21	Estrutura do pacote de dados LoRa [15].	35
Figura 2.22	Menu principal da ferramenta online [16].	38
Figura 2.23	Criação de uma simulação de ligação ponto a ponto [17].	39
Figura 2.24	Resultados de uma simulação - ligação ponto a ponto [17].	40
Figura 2.25	Representação gráfica de um mapa de cobertura [17].	41
Figura 3.1	Diagrama geral do sistema.	44
Figura 3.2	Diagrama de blocos do ESP8266 [18].	46
Figura 3.3	a) NodeMCU ESP-12E b) Wemos D1 mini [19].	48
Figura 3.4	Módulo A6 GSM mini [20]	48
Figura 3.5	a) DHT22 b) DS18B20 c) Cata-vento d) Anemómetro e) Pluviómetro f) HL-69.	49
Figura 3.6	Janela de visualização do menu Preferências.	52
Figura 3.7	Diagrama de herança das drivers	53
Figura 3.8	Diagrama de herança das managers	54
Figura 4.1	Circuito geral de uma estação.	60
Figura 4.2	Ligações entre placa de comunicações e placa de sensores.	61
Figura 4.3	Modos de arranque do NodeMCU.	62
Figura 4.4	Disposição dos leds de informação.	63
Figura 4.5	Informação sobre os estados da estação.	64
Figura 4.6	Circuito da placa de comunicações.	66
Figura 4.7	Circuito da placa de sensores [21].	67
Figura 4.8	Fluxograma referente ao cenário da rede <i>mesh</i> integrada.	70
Figura 5.1	Diagrama da ligação ponto a ponto.	83
Figura 5.2	Montagem experimental de uma estação para os testes.	83
Figura 5.3	Fluxograma referente à recolha de dados na ligação ponto a ponto.	85
Figura 5.4	Mapa das ligações ponto a ponto testadas em ambiente real.	93

Figura 5.5	Diagrama de uma ligação exterior e interior. . . . .	94
Figura 5.6	Zona de testes para uma potência de transmissão igual a 11 dBm. . . . .	97
Figura 5.7	Zona de testes visualizada em três dimensões. . . . .	97
Figura 5.8	Relação da média dos intervalos de: RSSI com a distância a) e SNR com a distância b). . . . .	103
Figura 5.9	Relação entre a taxa de entrega de pacotes (PDR) e a distância. . . . .	104
Figura 5.10	Visualização da encosta para a ligação ponto a ponto a 5.7 km. . . . .	104
Figura 5.11	Diagrama da rede mesh simples. . . . .	106
Figura 5.12	Fluxograma referente à rede <i>mesh</i> simples. . . . .	108
Figura 5.13	Representação física da rede simples. . . . .	113
Figura 5.14	Encaminhamento direto da mensagem e respetiva tabelas de rotas. . . . .	114
Figura 5.15	Encaminhamento alternativo da mensagem e respetiva tabelas de rotas. . . . .	115
Figura 5.16	a) Encaminhamento direto da mensagem b) Encaminhamento al- ternativo. . . . .	116
Figura 5.17	Representação física da rede mesh integrada. . . . .	117
Figura 5.18	Envio e receção da mensagem com sucesso a) sem sucesso b) Res- posta recebida c). . . . .	118
Figura 5.19	Visualização do <i>debugging</i> do processo da placa de sensores. . . . .	118
Figura 5.20	Visualização do <i>debugging</i> do processo (Estado A). . . . .	119
Figura 5.21	Visualização do <i>debugging</i> do processo (Estado C). . . . .	120
Figura 5.22	Esquema de ligações para a análise do consumo. . . . .	122
Figura 5.23	Consumo de corrente dos módulos rádio. . . . .	123
Figura 5.24	Consumo de corrente de uma estação. . . . .	124
Figura 6.1	Conversor ADS1115. . . . .	139
Figura 6.2	Formulário de criação de uma nova cobertura [17]. . . . .	140



# Lista de Tabelas

Tabela 2.1	Bandas de frequência para dispositivos não específicos de curto alcance na Europa [22][23] . . . . .	14
Tabela 2.2	Comparação entre as principais frequências ISM [24] . . . . .	17
Tabela 2.3	Tabela de comparação entre tecnologias LPWAN [25]. . . . .	28
Tabela 2.4	Tabela de comparação entre os módulos de comunicação [14]. . . . .	36
Tabela 5.1	Descrição das variáveis de espaço interior. . . . .	95
Tabela 5.2	Descrição das variáveis no espaço livre. . . . .	95
Tabela 5.3	Descrição dos pontos móveis e resultados da simulação. . . . .	96
Tabela 5.4	Resultados das ligações ponto a ponto para o RFM69HCW. . . . .	99
Tabela 5.5	Resultados das ligações ponto a ponto com SX1278 - Configuração Longo Alcance, Antena Monopolo. . . . .	100
Tabela 5.6	Resultados das ligações ponto a ponto com SX1278 (versão Lora Ra-02) - Configuração Longo Alcance, Antena <i>Rubberducky</i> . . . . .	101
Tabela 5.7	Resultados das ligações ponto a ponto com SX1278 (versão Lora Ra-02) - Configuração Médio Alcance, Antena <i>Rubberducky</i> . . . . .	102
Tabela 5.8	Componentes e custo da estação. . . . .	127
Tabela 5.9	Tabela comparativa de preços de uma estação com diferentes rádios.128	
Tabela 6.1	Tipos de aplicações para os dispositivos de pequena potência (SRD) [22] . . . . .	141



# Lista de Acrónimos

**AES** *Advanced Encryption Standard*

**CEPT** Conferência Europeia de Administração de Correios e Telecomunicações  
(*Conférence européenne des administrations des postes et des télécommunications*)

**CRC** *Cyclic Redundancy Check*

**ETSI** Instituto Europeu de Normas de Telecomunicações (*European Telecommunications Standards Institute*)

**EC** Comissão Europeia (*European Commission*)

**ISM** *Industrial-Scientific-Medical*

**ISEP** Instituto Superior de Engenharia do Porto

**ITU** União Internacional de Telecomunicações

**LPWAN** *Low Power Wide Area Networks*

**MEEC** Mestrado em Engenharia Eletrotécnica e de Computadores

**MIMO** Múltiplas Entradas e Múltiplas Saídas (*Multiple-Input and Multiple-Output*)

**RF** Radio Frequência

**RSSI** *Received Signal Strength Indicator*

**SPI** *Serial Peripheral Interface*

**SRD** *Short Range Devices*

**SNR** *Signal to Noise Ratio*

**TEDI** Tese/Dissertação

**UE** União Europeia

# Capítulo 1

## Introdução

Neste primeiro capítulo são apresentados os objetivos do projeto, bem como a sua contextualização, identificando o problema que se pretende resolver e os resultados que se pretendem obter. Dentro deste capítulo, encontra-se não só a calendarização do projeto, como também a organização da estrutura do relatório.

### 1.1 Contextualização

O projeto desenvolvido foi proposto pela empresa SmartHelix, sediada no Parque de Ciência e Tecnologia da Universidade do Porto (UPTEC) e enquadra-se no âmbito da unidade curricular Tese/Dissertação, do Mestrado em Engenharia Eletrotécnica e de Computadores – Área de Especialização de Automação e Sistemas, do Instituto Superior de Engenharia do Porto (ISEP).

A SmartHelix é uma empresa *startup* que centra a sua atividade na investigação e desenvolvimento de produtos inovadores na área da Energia e das Tecnologias de Informação. A sua missão consiste em incorporar os conhecimentos adquiridos na fase de pesquisa e de investigação, levando à criação e comercialização de soluções com valor de mercado. Conceitos, como a "Internet industrial" ou a "fábrica digital", são utilizados para descrever a SmartHelix, que pretende ser uma empresa digital de referência, trabalhando de forma mais próxima dos clientes, e fornecendo equipamentos e soluções

ajustadas à medida do cliente.

No futuro a empresa pretende apostar em áreas como a domótica, através da monitorização de consumo energético, controlo e atuação de equipamentos, ou até mesmo o desenvolvimento de aplicações para os transportes. No entanto, a agricultura um setor ainda pouco explorado pela Indústria 4.0 é o principal foco da empresa.

A digitalização dos processos agrícolas e análise em tempo real destes ecossistemas, implica que haja um conjunto de soluções inovadoras para esse efeito. O desenvolvimento de equipamentos como sistemas de *hidroponia/aquaponia*, e soluções para controlo e monitorização de uma produção agrícola doméstica e industrial, são projetos que estão a ser desenvolvidos e sobre os quais surgiu a proposta do estágio.

Nesse sentido, surgiu a oportunidade de resolver um dos problemas que afetam a manutenção das culturas agrícolas, que separadas por distâncias consideráveis, impossibilitam visitas regulares de técnicos especializados para monitorizar estas produções. Assim sendo, pretende-se implementar com tecnologia de comunicação sem fios, um sistema estruturado em rede *mesh* que responda em zonas de grande alcance (preferencialmente até dois quilómetros). O que se pretende obter é a comunicação entre equipamentos distanciados entre si, formando uma rede, que através do envio e receção de informação entre módulos, consiga manter a integridade dos dados e a sua segurança.

## 1.2 Objetivos

Este projeto foi desenvolvido tendo como objetivo principal, implementar uma tecnologia que permita uma comunicação *mesh* a distâncias de pelo menos dois quilómetros. A tecnologia deve utilizar microcontroladores que acoplados a diversos sensores, irão recolher a informação sobre o estado das plantações e enviar através da rede (módulos rádio, GSM ou equivalentes) dados relevantes para o utilizador. Um ou vários microcontroladores deverão fazer a ponte com a Internet, ou seja deverão fazer o envio dos dados monitorizados para a *cloud*.

No entanto, para garantir um desenvolvimento mais eficaz, procurou-se dividir o

projeto por diferentes etapas. Assim sendo para a realização do projeto foi necessário alcançar os seguintes objetivos parciais:

- Estudar os fundamentos da propagação de um sistema rádio;
- Analisar os microcontroladores com tecnologia WiFi para fazer a interface com a Internet;
- Entender o conceito de redes *mesh* e procurar soluções económicas para estabelecer uma rede de comunicação viável para longas distâncias;
- Estudo das vantagens e desvantagens dos diferentes tipos de tecnologias e protocolos existentes para as redes sem fios;
- Perceber o impacto da legislação europeia nas gamas de frequências e a importância do conceito de *Link Budget* nas comunicações;
- Implementar com diferentes módulos, um sistema emissor e um sistema recetor para avaliar a relação entre distância e qualidade de ligação;
- Desenvolver uma rede mesh com módulos para conseguir atingir as distâncias pré-definidas;
- Validação do sistema de comunicações através do controlo de pequenos equipamentos elétricos: o envio e a receção de uma trama com comandos externos pedidos por um utilizador.

### 1.3 Calendarização

De forma a cumprir prazos e minimizar pesquisa desnecessária durante a elaboração do protótipo final, foi construída uma tabela onde estão calendarizadas todas as tarefas realizadas, tais como pesquisa, elaboração de relatório e protótipo final, assim como prazos projetados para cada uma delas. Através deste diagrama da Figura 1.1 é possível ilustrar o avanço das várias fases do projeto, sendo que os intervalos de tempo representam

o início e fim de cada etapa. Apesar do planeamento e de algumas etapas terem sofrido alterações, todas foram cumpridas e concluídas.

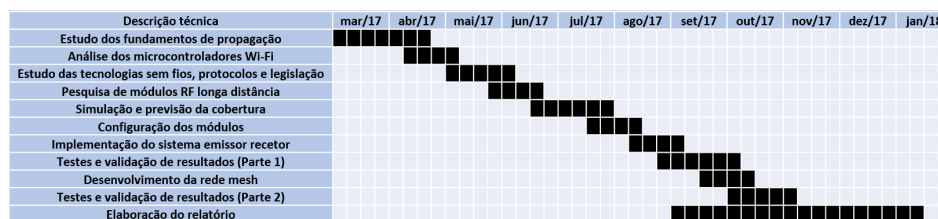


Figura 1.1: Calendarização do projeto

## 1.4 Organização

O relatório encontra-se organizado por 6 capítulos e estes separados por subcapítulos, permitindo assim estruturar a informação do projeto. No primeiro capítulo encontra-se a introdução que permite enquadrar o projeto e contextualizar a proposta que foi lançada, bem como apontar alguns objetivos para permitir um planeamento mais eficiente.

No segundo capítulo temos o estado da arte que consiste na pesquisa bibliográfica acerca do tema. Basicamente é explicado os conceitos relacionados com a propagação de ondas rádio, o que são as zonas de Fresnel, antenas e estudo das tecnologias LPWAN.

No terceiro capítulo está representado a arquitetura do sistema que espelha o estudo que foi efetuado nos componentes do projeto, ou seja, a análise prévia dos elementos principais e secundários do protótipo.

O desenvolvimento do protótipo está explicado no quarto capítulo. Aqui é analisado o sistema como um todo através de um diagrama geral, explicando os funcionamento do código com ajuda de fluxogramas.

No quinto capítulo são efetuados testes com cenários distintos para validar a solução final. Por último, no sexto encontram-se as conclusões, especificando as limitações existentes, uma apreciação final e melhorias num trabalho futuro.

## Capítulo 2

# Conceitos Teóricos

Neste capítulo será efetuado um estudo teórico sobre os conceitos que iremos abordar na componente prática do trabalho, servindo como base para o trabalho desenvolvido. Os subcapítulos estarão em conformidade com os objetivos do projeto, de forma a ser possível assimilar conceitos como a propagação de ondas eletromagnéticas, fundamentos sobre antenas e as componentes teóricas das redes sem fios.

Atualmente são utilizadas uma vasta gama de tecnologias de comunicação sem fios, sendo importantíssimo conhecer o mecanismo destas, os protocolos associados e a sua evolução com o tempo. Esta abordagem teórica é fundamental para perceber como determinados mecanismos e arquiteturas estão definidos, de forma a abordar as etapas seguintes com os conceitos bem assimilados.

### 2.1 Princípios das comunicações rádio

Nas telecomunicações, o termo rádio está definido como sendo "a transmissão e a recepção de sinais rádio frequência, através da propagação de ondas eletromagnéticas pelo espaço"[1]. A Figura 2.1 tenta ilustrar como estas ondas viajam pelo espaço, sendo os seus campos magnético e elétrico perpendiculares entre si, e perpendiculares à direção de propagação da onda [1],[26].

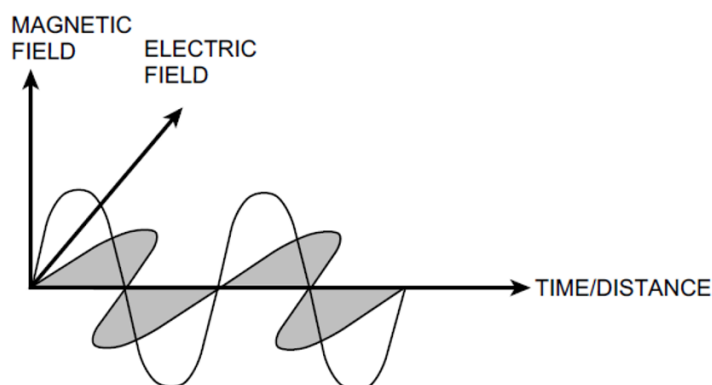


Figura 2.1: Representação de uma onda eletromagnética. [1]

A velocidade de propagação das ondas eletromagnéticas no vácuo é a mesma que a velocidade da luz, aproximadamente  $300 \times 10^6$  m/s, sendo o comprimento de onda  $\lambda$  dado pela seguinte equação:

$$\lambda = \frac{c}{f} \quad (2.1)$$

Onde,  $\lambda$  = comprimento de onda, em metros;  $c$  = velocidade da luz, em metros/segundo;  $f$  = frequência, em Hertz.

Estes sinais transportam informação codificada de forma analógica ou digital sob diversas formas como é o caso de imagens de televisão, voz, código ou operações comandadas à distância [1],[26]. Esta informação é codificada através de um modulador que varia a energia de transmissão à saída do transmissor, podendo este ser resultado de uma modulação em frequência ou amplitude. As ondas eletromagnéticas resultantes deste processo serão convertidas em energia à entrada do recetor e, posteriormente, reconvertidas em dados (voz, imagem, por exemplo) [2]. Na Figura 2.2 está representado um exemplo simples como estes sistemas funcionam. Com uma combinação adequada entre a antena, a linha de transmissão, o transmissor e o recetor é possível transmitir e receber dados continuamente.

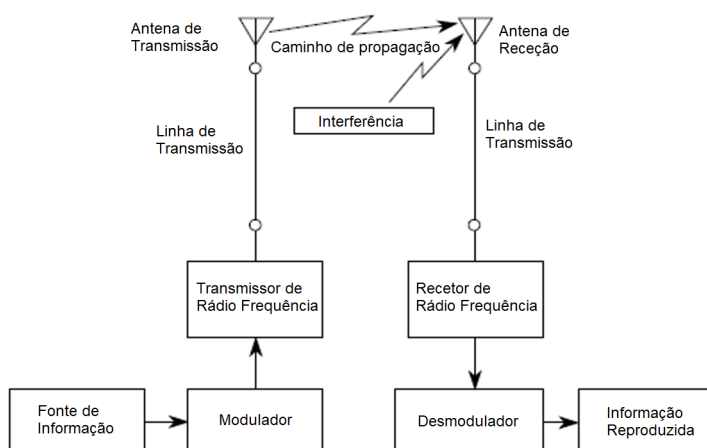


Figura 2.2: Modelo generalizado de um sistema transmissor-recetor. [2]

### 2.1.1 Espectro eletromagnético

As ondas eletromagnéticas estão agrupadas num espectro enorme. Por conseguinte, é conveniente ver onde é que cada uma das diferentes formas de radiações se encaixa dentro do espectro. É possível verificar que os sinais de rádio apresentam menor frequência, logo os seus comprimentos de onda são maiores em relação aos outros sinais. Acima do gama das ondas rádio, podem ser encontradas outras formas de radiação, como por exemplo a radiação infravermelha, visível e ultravioleta [27].



Figura 2.3: Espectro eletromagnético [3]

A gama das ondas rádio vai desde a faixa LF (30-300 kHz) até à faixa EHF (30-300 GHz), sendo estas pouco energéticas no espectro eletromagnético. As mais utilizadas em rádio e televisão são as VHF (*Very High Frequency*) e as UHF (*Ultra High Frequency*). Na Figura 2.4 está representada esta classificação que separa as ondas rádio em gamas enunciados em diferentes diferentes [4].

<b>Name of Frequency Range and Abbreviation</b>	<b>Frequencies</b>
Very low frequency (VLF)	3–30 kHz
Low frequency (LF)	30–300 kHz
Medium frequency (MF)	300–3,000 kHz
High frequency (HF)	3–30 MHz
Very high frequency (VHF)	30–300 MHz
Ultrahigh frequency (UHF)	300–3,000 MHz
Superhigh frequency (SHF)	3–30 GHz
Extremely high frequency (EHF)	30–300 GHz

Figura 2.4: Classificação das ondas rádio. [4]

Nas bandas VHF e UHF, a propagação é dominada por alguns fenómenos como a reflexão nas superfícies planas, a propagação direta entre emissor e recetor, a dispersão por superfícies irregulares, a refração por causa das mudanças na atmosfera, a difração e a absorção. Para estas bandas, os efeitos da atmosfera como a precipitação (chuva, nevoeiro, neve e granizo) não afetam a transmissão a menos que estes efeitos se pronunciem muito no índice de refração. A partir da frequência de 6 GHz, estes efeitos são mais marcantes, tornando-se um fator limitante nas ligações de rádio [28].

### 2.1.2 Fórmula de transmissão de Friis

A fórmula de Friis é utilizada para encontrar a potência ideal recebida numa antena, a partir de informações básicas sobre a transmissão. A única dificuldade inerente da equação é o facto de ser calculada apenas para uma única frequência, apesar das transmissões normalmente serem compostas por muitas. A equação definida por H.T. Friis descreve o comportamento da onda em "campo aberto", sendo esta representada por [29]:

$$P_R = P_T \times G_T \times G_R \left(\frac{\lambda}{4\pi}\right)^2 \times \left(\frac{1}{d}\right)^n \quad (2.2)$$

Onde,

$P_R$  = potência recebida (Watts)

$P_T$  = potência transmitida (Watts)

$G_T$  = ganho da antena de transmissão (dBi)

$G_R$  = ganho da antena de recepção (dBi)

$\lambda$  = comprimento de onda (metro)

$d$  = distância que separa o transmissor do recetor

$n$  = expoente para as condições ambientais ( $n = 2$  define "campo aberto")

A potência do sinal recebido num determinado local varia em conformidade com quatro elementos principais: a potência do sinal original transmitido; o desempenho das antenas do transmissor e do recetor; o comprimento de onda relativo à frequência de operação; e a distância entre os pontos de comunicação. A equação pode ser expressa em diferentes formas, sendo utilizada consoante a necessidade dos cálculos, por exemplo para uma equação de *Path Loss* ou para avaliar a performance de um sistema de comunicação através do *Link Budget*. Alguns destes conceitos serão abordados nos subcapítulos seguintes [29].

Este modelo apenas considera as características físicas das ondas eletromagnéticas, sendo uma boa aproximação para as condições de campo aberto (*free space*). Contudo é recomendável alertar que as condições reais podem invalidar dados resultantes de uma abordagem completamente teórica [29].

### 2.1.3 *Free Space Path Loss (FSPL)*

A fórmula FSPL é utilizada para avaliar a atenuação do sinal de rádio no espaço livre, em função com a distância e a frequência de operação. Esta considera que não há obstáculos que possam fazer com que o sinal seja refletido, refratado ou que possa causar atenuação adicional. Normalmente, utiliza-se como uma primeira aproximação para cálculos de curto alcance, ou para uma área onde hajam poucas obstruções. Como tal, é uma

ferramenta valiosa para muitas pessoas que lidam com sistemas de radiocomunicações [30],[26],[29].

O sinal ao ser emitido por uma fonte isotrópica (por ex: transmissor) irá perder energia ao afastar-se da origem, isto ocorre quando uma onda eletromagnética viaja através de um campo de visão livre.

A equação 2.3 encontra-se representada com a disposição na forma logarítmica [30].

$$L = 32.45 + 20\log(f) + 20\log(d) \quad (2.3)$$

Onde,

L = atenuação do percurso (dB)

f = frequência (MHz)

d = distância (km)

Assim estes dados apenas olham a atenuação em si (contando como uma propagação direta) e não contêm nenhum fator relacionado à potência do transmissor, aos ganhos da antena ou aos níveis de sensibilidade do recetor. Esses fatores são parâmetros para o cálculo da conexão (*Link Budget*), que serão usados em modelos matemáticos e *softwares* para a previsão da cobertura rádio [30],[26].

O modelo de atenuação de interiores [31] estima o valor de perda de sinal dentro de uma sala, andar, ou qualquer outro espaço interior. A sua representação matemática encontra-se na equação 2.4, que difere um pouco da fórmula de *Free Space Path Loss*, com os valores do coeficiente e fator de atenuação tabelados para frequências de transmissão diferentes. No caso do coeficiente de atenuação por piso, este apresenta diferentes valores para andares diferentes. Por exemplo para um andar temos um valor de 4.5, 2 andares igual a 9.5 e 3 andares um valor de 12. Estes fatores não apresentam unidades porque são constantes empíricas, valores que são aproximações retiradas de estudos experimentais.

$$Lin = 20 \log_{10} f + N \log_{10} d + Pf(n) - 28 \quad (2.4)$$

Onde,

Lin = atenuação do percurso dentro de um edifício (dB)

f = frequência (MHz)

$d$  = distância (m)

$N$  = coeficiente de atenuação por distância

$Pf(n)$  = fator de atenuação por penetração por  $n$  pisos

Um radiador isotrópico é uma antena ideal que irradia potência com ganho unitário uniformemente em todas as direções, sendo normalmente utilizada com referência para o ganho de antenas em sistemas *wireless*. Assim é possível definir a EIRP (*Effective Isotropic Radiated Power*) como:

$$EIRP = Pt \times Gt \quad (2.5)$$

Que representa a máxima potência irradiada disponível de um transmissor, na direção de máximo ganho da antena, quando comparada com um radiador isotrópico. Outro conceito semelhante é o *Effective Radiated Power* (ERP), que compara a antena atual com uma antena de dipolo de meia onda [26]. Os ganhos das antenas são apresentadas em dBi (ganho em dB em relação a isotrópica) ou em dBd (ganho com respeito ao dipolo de meia onda).

#### 2.1.4 *Link Budget*

O termo *Link Budget* surgiu da necessidade de avaliar um sistema de comunicações, englobando as perdas na linha de transmissão, potência transmitida, sensibilidade e ganhos nas antenas. Estes cálculos são muito utilizados nos sistemas satélite e também em outras redes móveis para avaliar a cobertura do sinal em determinadas regiões. Este conceito foi abordado na subsecção 2.1.2 e tipicamente a equação é apresentada da seguinte forma:

$$P_R = P_T + G_T + G_R - (L_P + L_T + L_R + L_M) \quad (2.6)$$

Onde,

$P_R$  = potência recebida (dBm)

$P_T$  = potência transmitida (dBm)

$G_T$  = ganho da antena de transmissão (dBi)

$L_T$  = perdas de transmissão (dB)

$L_P$  = *Free Space Path Loss* (dB)

$L_M$  = perdas variadas (margem de desvanecimento, incompatibilidade na polarização,

entre outras) (dB)

$G_R$  = ganho da antena de receção (dBi)

$L_R$  = perdas na receção (dB)

A análise dos cálculos é essencial para estabelecer uma ligação RF robusta. Inicialmente colocam-se os elementos da equação que possuem valores fixos no sistema real, e substitui-se os valores restantes por estimativas. Estas alterações irão permitir identificar quais as restrições que existem consoante a variação de determinados elementos. Numa fase seguinte, esta equação pode ser usada para verificar o desempenho da ligação com dados reais medidos experimentalmente [29].

### 2.1.5 Zona de Fresnel

O método mais comum para conseguir uma simplificação computacional eficiente da ligação é usar o conceito de elipse de Fresnel e as atenuações da difração *knife-edge*.

Esta elipse representada na Figura 2.5 é o local de pontos onde, se houver uma superfície refletiva, o sinal refletido chegará ao ponto de receção com a mesma fase que o sinal original [32],[5].

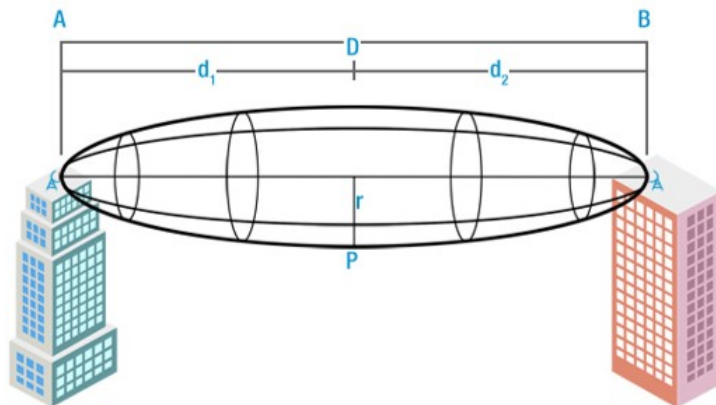


Figura 2.5: Representação tridimensional da zona de Fresnel. [5]

Em outros casos, o sinal refletivo chegará fora de fase com o original e assim provocará uma redução na força do sinal recebido. É recomendado que no planeamento deste tipo de ligações, cerca de 60 por cento deva estar livre de obstáculos. Nesta linha direta, para

qualquer ponto P, distanciado entre os extremos da ligação, o raio da elipse no plano vertical é expresso por [32],[5]:

$$Fn = \sqrt{\frac{n \times \lambda \times d1 \times d2}{d1 + d2}} \quad (2.7)$$

Onde,

$Fn$  = raio da 'n' zona de Fresnel (m)

$\lambda$  = comprimento de onda da frequência (m)

$d1$  = distância de A até P (m)

$d2$  = distância de B até P (m)

## 2.2 Licenciamento e disponibilidade espectral

Na Europa, a regulação da isenção de licença no espectro é gerida por três entidades, envolvendo a Comissão Europeia (EC - *European Commission*), o Comité das Comunicações Europeias (ECC - *European Communications Committee*) e o Instituto Europeu das Normas das Telecomunicações (ETSI - *European Telecommunications Standards Institute*). O ETSI teve origem na Conferência Europeia das Administrações de Correios e Telecomunicações (CEPT), tendo sido formalmente autonomizado em 1988 [33].

As autoridades nacionais de cada país (ANACOM em Portugal) estão responsáveis por adotar estas decisões e recomendações europeias. As normas para a utilização de bandas não licenciadas, estão agrupadas na Decisão (EU) 2017/1483 de 8 de Agosto de 2017 na Decisão de Alteração 2006/771/EC sobre a harmonização do espectro de rádio-frequência para o uso de dispositivos de curto alcance (*Short Range Devices - SRDs*) [34],[23].

A Recomendação ERC/REC 70-03, direcionada para o uso das SRDs, engloba as bandas que estão totalmente harmonizadas e outras. Como é uma recomendação, os países que a implementam não estão obrigados a compactuar com as alocações estabelecidas, só as decisões (ou diretivas) é que ficam sujeitas a harmonização. No subcapítulo 2.2.1 será abordado este conceito para os dispositivos em questão [23],[33].

### 2.2.1 Dispositivos de Pequena Potência (SRDs)

Os dispositivos de curto alcance, *Short Range Devices*, (SRDs) estão definidos num grupo de transmissores de rádio que permitem comunicação uni e bidirecional, que não causam interferência a outros equipamentos rádio. Estes dispositivos não são considerados um serviço de rádio, e utilizam antenas integradas, dedicadas ou externas, podendo as modulações estar sujeitas a padronização [23],[34].

Para além de não estarem sujeitos a licença, o CEPT recomenda que os limites impostos não sejam ultrapassados para existir pouca interferência entre aparelhos [22]. A Recomendação ERC/REC 70-03 descreve os requisitos para a gestão do espetro nas SRD's, abordando pontos como a alocação de frequências, o nível máximo de potência, o espaçamento entre canais (ou a largura de banda máxima), a modulação e ciclo de trabalho (*duty cycle*) [23],[22].

Tabela 2.1: Bandas de frequência para dispositivos não específicos de curto alcance na Europa [22][23]

Frequência (MHz)	ERP	<i>Duty Cycle</i>	Espaçamento entre canais
433.05 – 434.79 MHz	+10 dBm	<10%	ODTF
433.05 - 434.79 MHz	0 dBm	≤100%	ODTF
433.04 – 434.79 MHz	+10 dBm	≤100%	<25 kHz
868 – 868.6 MHz	+14 dBm	≤1%	ODTF para um ou mais canais
868.7 – 869.2 MHz	+14 dBm	≤0.1%	ODTF para um ou mais canais
869.4 – 869.65 MHz	+27 dBm	≤10%	ODTF para um ou mais canais
869.7 -870 MHz	+7 dBm; +14 dBm	≤100% para +7dBm e ≤1% para + 14dBm	ODTF para um ou mais canais
2400 – 2483.5 MHz	+10 dBm	≤100%	ODTF

\* ODTF = Ocupação de toda a faixa.

A Tabela 2.1 apresenta os valores estipulados para as frequências não licenciadas do uso geral, correspondendo ao Anexo 1 da Recomendação 70-03. Cada anexo corresponde

ao tipo de aplicação em que estas frequências são utilizadas, listadas na Tabela 6.1 do Anexo 6.3. De assinalar que na Tabela 2.1 estão selecionadas as bandas de frequência mais utilizadas no campo do IoT (*Internet of Things*) e na domótica [35].

### 2.2.2 Bandas ISM

As bandas ISM (*Industrial-Scientific-Medical*) foram criadas para processos de aproveitamento energético das ondas rádio, em equipamentos médicos, industriais ou científicos. No entanto, apesar do propósito inicial, a isenção de licença permitiu o surgimento de novas tecnologias de curto alcance e pequena potência (*Bluetooth*, *NFC*, *ZigBee*), que impulsionaram o uso destas bandas [24].

Neste campo existem outras bandas ISM, incluindo as frequências de 6.78 KHz, 13.56 KHz, 27.120 KHz, 40.62 MHz, 5.8 GHz, 24.125 GHz, 61.25 GHz, 122.5 GHz, e 245 GHz. A organização internacional para as telecomunicações ITU (*International Telecommunication Union*) divide o planeta em três regiões para permitir a gestão do espectro mundial. A Europa e a África fazem parte da Região 1, enquanto que a América faz parte da Região 2. A zona do Pacífico incluindo a Ásia e a Austrália fazem parte da Região 3 [24],[35]. Como é possível verificar na Figura 2.6, para esta distribuição geográfica existem alocações específicas para determinada região e outras alocadas mundialmente.

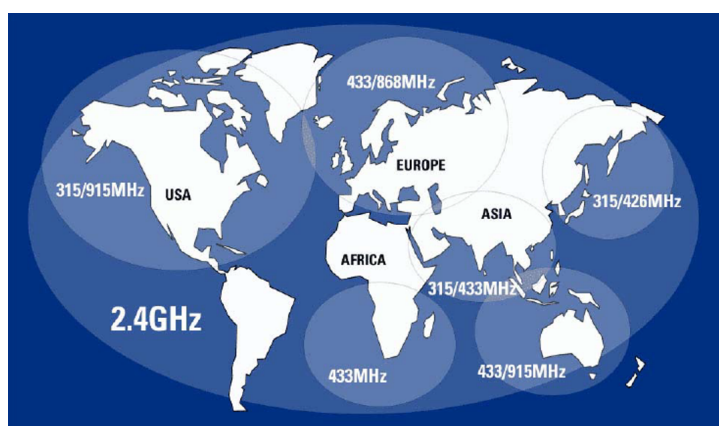


Figura 2.6: Representação mundial da alocação das principais frequências ISM. [4]

Após a identificação destas frequências principais, será abordado o impacto da utilização de cada uma, e em que aplicações ou equipamentos estão a ser implementados, de acordo com as regulamentações.

### **Frequências 433 MHz**

Esta banda é muito conhecida na Europa no âmbito do IoT por investigadores e entusiastas das telecomunicações, no desenvolvimento de projetos pequenos, como por exemplo estações meteorológicas remotas. A sua presença fica marcada em aplicações, onde aparelhos como interruptores, sensores de temperatura, sensores de contato de portas ou janelas, necessitam de leituras rápidas e transmissões curtas [24].

A banda de 433 MHz tem uma faixa de frequência de 433.050 MHz a 434.790 MHz, que inclui 69 canais com espaçamento de canal de 25 kHz. As restrições de potência e *duty cycle* levaram a que muitas destas aplicações migrassem para a banda 868 MHz, onde não existem tantas restrições técnicas [33].

### **Frequências 863-870 MHz**

Atualmente na Europa, esta banda foi alocada para as aplicações de faixa estreita com isenção de licença. Assim é segmentada em três faixas secundárias com a mais inferior a corresponder ao equipamento de áudio sem fios, a faixa intermédia para aplicações RFID e a banda mais superior a sistemas de alarme e a SRDs de uso geral [33]. Os dispositivos a operar nestas bandas estão limitados a 25 mW na Europa e um *duty cycle* até 1%. Nos Estados Unidos a faixa de frequências mais próxima destes valores é a banda 900 MHz. Para esta gama de frequências não existem restrições a nível da aplicação, nem no *duty cycle* [24].

### **Frequências 2.4 GHz**

Esta frequência está disponível mundialmente para os dispositivos *Wi-Fi*, Bluetooth e tecnologias ZigBee, implementando redes WLAN (*Wireless Local Area Networks*) ou outras redes móveis pessoais. Dependendo do tipo de sistema, a quantidade de canais

varia, podendo ser distribuídos um maior número de canais com menor espaçamento entre eles ou vice-versa. O potencial distância-penetração-custo aliado à isenção de licença deu um maior impulso para estas comunicações sem fios. Estes dispositivos estão limitados a 10 mW e sem limites para o *duty cycle* [24],[33].

Na Tabela 2.2 está apresentado uma pequena comparação entre estas bandas com a sua distribuição geográfica bem como outros detalhes técnicos (penetração, distância).

Tabela 2.2: Comparação entre as principais frequências ISM [24]

Banda	Prós	Contras	Região ITU
433 MHz	1) Maior alcance 2) Melhor penetração do sinal 3) Atenuação reduzida no percurso	1) Lotação em algumas frequências 2) Qualidade relativamente baixa e sistema de rádio caro	Reg.1(Europa)
868 MHz	1) Alcance razoável 2) Banda pouco congestionada	1) Planeamento espectral complexo	Reg.1(Europa).
915 MHz	1) Grande alcance até 1 km 2) Antenas pequenas	1) Aplicações limitadas 2) Necessita de melhor regulação para o uso ISM 3) Alto consumo de energia dos aparelhos	Reg.2(USA/CAN) Reg.3(AUS/NZ)
2.4 GHz	1) Alta taxa de transmissão 2) Sistemas rádio versáteis baratos	1) Alcance baixo pode ser aumentado com repetidores 2) Atenuação acentuada no percurso	Reg.1, Reg.2 and Reg.3

Com esta Tabela 2.2 é possível verificar que as regiões ITU têm diferentes bandas ISM para permitir o uso global e regional das frequências, procurando uma harmonização necessária. No entanto estabelecer uma regulação do espectro continua a ser um desafio constante para as necessidades globais.

O aumento do número de dispositivos a operar nestas bandas resultou num aumento da procura de espectro. Apesar do aumento da procura, as limitações na disponibilidade espectral tornam difícil conseguir solucionar estes problemas. No entanto a solução poderá passar por operar nestas bandas sem licença, de uma forma mais eficiente e económica.

A existência de dispositivos em ambientes com especificações técnicas idênticas, potencia a interferência entre aparelhos, pelo que o uso de técnicas de mitigação, pode facilitar a coexistência harmoniosa entre eles [24].

## 2.3 Antenas

As antenas são bastante importantes para qualquer sistema de comunicação rádio. É o componente que faz a ligação física entre a linha de transmissão, emitindo ondas eletromagnéticas para o espaço circundante. Os dispositivos pequenos costumam incorporar antenas mais pequenas, que têm de estar sintonizadas tanto no transmissor como no recetor para a frequência de operação.

A ideia é manter um tamanho ideal para que o desempenho não comprometa todo o sistema. Para isso existem dois tipos de antenas que os fabricantes usam como padrão; a antena monopolo de  $1/4$  de comprimento de onda e a antena de dipolo.

Normalmente a antena monopolo (ver Figura 2.7) é considerada o melhor exemplo prático de uma fonte isotrópica simples, considerada assim como referência para ganhos das outras antenas.

O desempenho da antena é medido pelo ganho "dB", e a antena isotrópica tem como ganho 0 dB. Os fabricantes avaliam as suas antenas em linha com as referências, no caso da antena isotrópica, o ganho fica com dBi e no caso do dipolo o ganho fica como dBd [29].

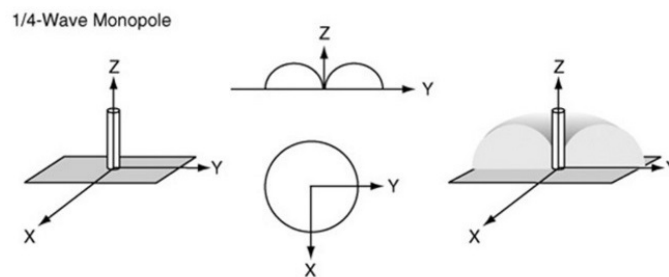


Figura 2.7: Antena monopolo e padrão de radiação [6].

Tendo em conta a equação 2.1, quanto maior a frequência, menor é o comprimento de onda, logo o tamanho da antena também será mais reduzido. Para uma frequência de 433.92 MHz, o comprimento de onda é igual a 0.69 m, e para 868 MHz, teremos um  $\lambda$  igual a 0.35 m. Felizmente, há maneiras de tornar a antena menor. Uma vez que a ressonância ocorrerá em frações de números inteiros ( $1/2$ ,  $1/3$ ,  $1/4$ , etc.) da frequência fundamental, antenas mais curtas podem ser usadas para enviar e recuperar o sinal.

Como acontece com tudo na engenharia, há um *trade-off*. Ao reduzir o tamanho da antena terá algum impacto na eficiência e impedância da antena, o que pode afetar o desempenho final do sistema [6],[29]. Na Figura 2.8 é possível visualizar a antena de dipolo com o seu padrão de radiação no espaço.

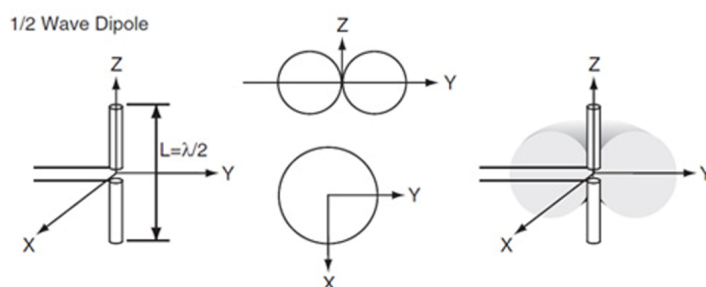


Figura 2.8: Antena dipolo e seu padrão de radiação [6].

O padrão de radiação varia consoante o tipo de antenas utilizadas, pelo que a diretividade da radiação, pode facilitar em aplicações onde seja necessário uma cobertura específica. Isto é, existem antenas com padrões verticais, horizontais ou omnidirecionais como é o caso da antena isotrópica. É fundamental que as antenas estejam polarizadas da mesma forma, ou seja, uma antena polarizada horizontalmente, não irá comunicar efetivamente com outra com polarização vertical.

Outro aspeto importante consiste na forma como a energia é transferida para a antena. Se a antena está desenhada para uma carga de 50 Ohms, então a sua impedância mínima deverá ser igual a 50 Ohms para melhores resultados. Num antena monopolo, a impedância varia se esta for curvada ou se existirem objetos por perto.

Estes valores de resistência e reactância são importantes, porque a transferência de

potência máxima ocorrerá quando as impedâncias de fonte e carga se combinarem. Se estes valores são diferentes, uma parte da energia enviada para a antena será refletida novamente na carga ou perdida como calor. Isso reduzirá a eficiência do sistema, reduzirá o alcance, aumentará a potência necessária para um determinado intervalo e reduzirá a vida da bateria [6].

## 2.4 Topologia de rede

Uma rede é constituída por nós que estando interligados entre si, podem apresentar diversas formas de se conectarem. A forma mais simples, com uma conexão direta, ponto a ponto, até a uma estrutura mais complexa como em árvore ou em formato de estrela. As topologias de rede são as diferentes formas que podemos ter na estrutura física, que faz a ponte de informação entre pontos distantes ou mais próximos consoante o formato pretendido.

### Rede Estrela

A topologia em estrela é uma estrutura simples que centraliza todas as operações num único nó. Tipicamente utilizadas em redes LAN, esta topologia permite que os nós secundários fiquem ligados ao nó principal, sem que haja grandes atrasos na comunicação.

Em caso de falha de conexão de um nó secundário, não há problemas nos restantes membros da rede, que continuam a sua operação com o nó principal. Pelo contrário, esta centralização permite que falhas ocorrentes no nó central, provocam sérios problemas à rede, causando instabilidade nas conexões secundárias [7],[36].

### Rede Árvore

A topologia em árvore pode ser construída a partir de uma rede estrela, ou de um conjunto delas. A sua configuração é muito parecida com a topologia de estrela, porque as ligações partem de um nó central, e ramificam-se entre os nós.

Isto é, porque existe um nó principal que estabelece pontes entre os nós secundários,

através de ramificações entre os níveis inferiores da hierarquia. Cada nó fica ligado ao nó seguinte, por ligações únicas ponto a ponto, que aumentam a quantidade de níveis e respetiva complexidade [7],[36].

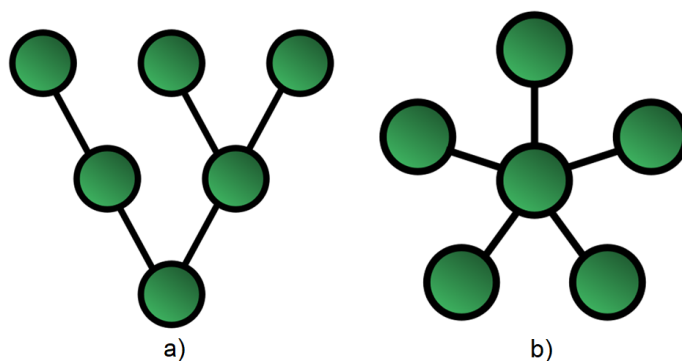


Figura 2.9: a) Estrutura de rede árvore e b) em estrela. [7]

## Rede Mesh

A topologia de rede *mesh* pode ser apresentada com duas configurações; uma em que a rede está completamente conetada, e outra em que está parcialmente interligada (ver Figura 2.10). Dependendo da configuração, as estruturas são constituídas por nós que mantêm ligações com todos os nós ou apenas com alguns deles [7],[36].

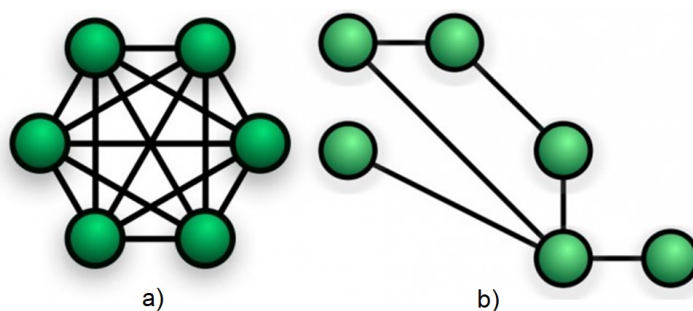


Figura 2.10: a) Estruturas de rede mesh interligada totalmente e b) interligada parcialmente. [7]

A interligação total entre os nós da rede, aliado ao mapeamento de rotas para as mensagens, permite que não haja grandes falhas de comunicação. Assim é possível

que seja calculado uma rota através da conectividade, velocidade e tarefas pendentes. A necessidade de algoritmos de configuração de rotas, em todos os membros pode ser uma desvantagem, pois obriga a uma maior complexidade do sistema. Uma alternativa será utilizar a rede parcialmente conectada, para facilitar este processo [7],[36].

## 2.5 Tecnologias LPWAN

Neste subcapítulo é elaborado uma abordagem sobre as tecnologias LPWAN *Low Power Wide Area Networks* que estão a surgir no campo do IoT. Este último termo descreve que todos os objetos podem estar ligados à Internet através de uma ligação rádio. A ligação pode abranger objetos, que comunicam com pequenas taxas de transmissão, a distâncias de metros a quilómetros. Assim procura-se contextualizar este tipo de tecnologias e as suas aplicações, que pretendem revolucionar o mundo.

### 2.5.1 Contextualização

O termo LPWAN corresponde às redes de comunicação sem fios que utilizam equipamentos de pequena potência, mas com grande área de cobertura. Estas tecnologias permitem a monitorização de sistemas localizados em regiões de difícil acesso, onde a manutenção dos aparelhos fica a cabo de mão de obra humana. Não só é arriscado, como tem custos que poderiam ser reduzidos [37],[38].

Portanto existem algumas características fundamentais que devemos seguir para implementar uma rede LPWAN [38]:

- O aparelho deve funcionar em modo de baixa potência para evitar substituições dispendiosas de baterias.
- As implementações devem ser baratas e de ampla aceitação; ou seja, sem o uso de cartões SIM, e a instalação fácil com manutenção mínima.
- O nível de atividade varia de aplicação para aplicação, mas para limitar o consumo, deve-se acordar o aparelho, só quando é necessário.

- A infraestrutura deve permitir a instalação a um nível nacional, com boa cobertura e possibilidade de mover para outros países.
- A transferência de informação entre o objeto e o utilizador, deve estar completamente segura, tornando-se robusta a bloqueios externos.
- Os dispositivos estarão numa posição estacionária, fixados a determinado ponto. No entanto o ambiente onde são inseridos podem haver muito movimento, logo a modulação precisa de ser robusta.

Na Figura 2.11 é possível observar uma comparação entre as tecnologias LPWAN e algumas das outras tecnologias sem fios. Neste paradigma, estas redes conseguem fazer compromissos que as tecnologias tradicionais não conseguem.

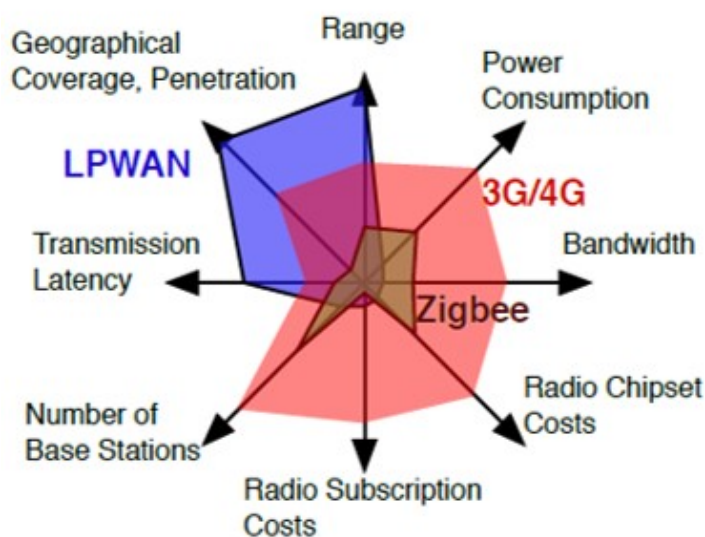


Figura 2.11: Comparação entre LPWAN e outras tecnologias sem fios. [8]

As redes sem fios como é o caso o ZigBee, Bluetooth, RFID, entre outras, simplesmente não abrangem a área que se pretende, porque estão tipicamente alocadas a aplicações com distâncias reduzidas. Com o WiFi, a complexidade para dar uma cobertura abrangente, implicaria uma alocação de recursos extraordinária, com um elevado número de *gateways* e dispositivos. Os custos associados à subscrição e implementação de um

sistema com rede móvel, não possibilitam a criação de um sistema com muitos objetos. Para além disso, neste tipo de redes o consumo energético é alto, para aplicações de áudio e voz [39],[8]. Assim para este tipo de redes, serão apresentadas algumas das tecnologias existentes neste ramo.

### 2.5.2 Soluções comerciais

Nesta subsecção serão identificados algumas das tecnologias emergentes, e as suas particularidades técnicas mais relevantes. Assim procura-se descrever os principais responsáveis no desenvolvimento das redes LPWAN, tentando fazer uma pequena comparação entre eles.

#### LoRa/LoRaWAN

A LoRa Alliance é uma associação sem fins lucrativos que pretende definir um padrão global aberto para as comunicações IoT. A LoRaWAN está disponível em muitos países em todo o mundo, e define-se como um protocolo aberto MAC (*Medium Access Control*), baseado na modulação LoRa [39],[8].

Este último consiste na camada física da sua arquitetura protocolar, que é implementada nos seus produtos pela empresa Semtech. A LoRa define diferentes fatores de propagação para fazer o compromisso entre alcance e transferência de dados. A taxa de transferência varia de 3 bps até 37.5 kbps, dependendo do fator de propagação e da largura do canal [39],[37].

É possível receber dados simultaneamente, com diferentes fatores de propagação, numa estação única, permitindo assim uma topologia de rede em estrela. Com esta aliança, os seus parceiros foram para uma abordagem mais aberta, abrindo a possibilidade da criação de redes LPWA públicas e privadas, com uma estrutura idêntica à Figura 2.12 [39],[38].

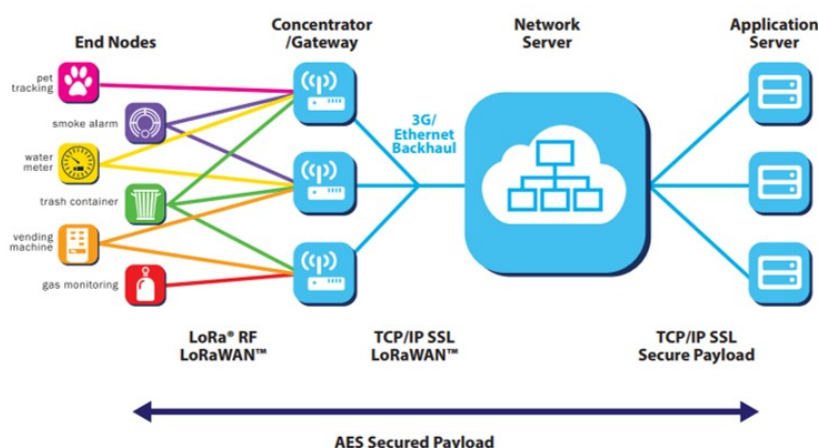


Figura 2.12: Estrutura da rede LoRa/LoRaWAN. [9]

A estrutura é constituída por vários elementos: os nós finais são os elementos da rede onde a deteção ou controlo é realizado. A ponte de ligação *gateway* recebe as comunicações dos nós finais e depois transfere-os para o sistema *backhaul*. Os *gateways* estão conectados ao servidor de rede usando conexões de IP padrão. Desta forma, os dados usam um protocolo padrão, mas podem ser conectados a qualquer rede de telecomunicações, seja pública ou privada.

O *Network Server* gere a rede, atuando para eliminar pacotes duplicados e adaptar taxas de dados. Depois um computador remoto pode controlar as ações dos pontos finais ou apresentar os seus dados através de aplicações. Para atender as mais diversas necessidades no protocolo LoRaWAN foram definidas três classes de dispositivos (*end-points*), responsáveis pela interação direta com o meio [9],[39],[38],[10].

- **Classe A - Dispositivos bidirecionais:** Esta classe permite a comunicação bidirecional onde cada transmissão do nó para a *gateway* (upload), é seguida por duas janelas de receção de dados intercaladas (download). Isto é, os módulos só podem receber dados da *gateway* em janelas de tempo pré determinadas e imediatamente após realizarem uma transmissão. Se a *gateway* não conseguir enviar dados dentro destes intervalos, este tem que esperar até à próxima transmissão.
- **Classe B - Dispositivos bidirecionais com janelas de receção agendadas:**

Para além de possuir as funcionalidades da Classe A, estes módulos podem abrir outras janelas de receção com horários programados. O módulo e o *gateway* trocam informações (beacon way) de forma a que o *gateway* saiba quando o módulo esta pronto para receber dados.

- **Classe C - Dispositivos bidirecionais com janelas de receção sempre abertas:** Os módulos estão continuamente a escutar a *gateway*, fechando as janelas apenas na transmissão. Nesta classe o módulo está sempre apto a receber dados, pelo que não é recomendado o uso de baterias devido ao consumo de energia.

Todos os módulos devem obrigatoriamente funcionar em classe A. Para garantir a privacidade dos dados existem três chaves de segurança. Na Figura 2.13 encontra-se representado como estas chaves são utilizadas na estrutura da rede.

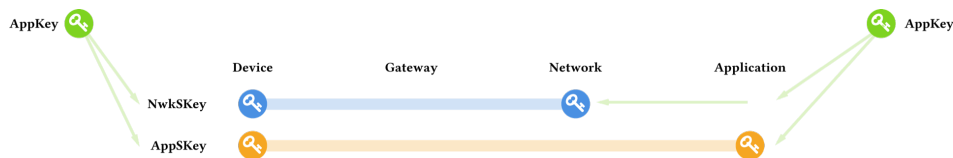


Figura 2.13: Chaves de segurança LoRaWAN [10]

A *Application Key* é apenas conhecida pelo dispositivo e pela aplicação. Quando um dispositivo conecta-se à rede, são criadas duas chaves de encriptação, uma que é partilhada pela rede (*Network Session Key*), e a outra que é privada (*Application Session Key*).

Os dados transmitidos (*payload*) são encriptados usando o algoritmo AES de 128 bits, com uma chave conhecida por *AppSKKey*. A chave *NwkSKKey* é conhecida pela rede e pelo dispositivo, sendo usada para validar a integridade da mensagem e garantir que o pacote recebido não foi alterado.

De assinalar que o servidor de rede pode apenas verificar a integridade do pacote e nunca revelar o seu conteúdo, pois não tem a chave *AppSKKey*. A duração de cada chave varia consoante a sessão da ligação [9],[10].

## Sigfox

A SigFox é uma variação de um sistema móvel, que permite que os dispositivos conectem a um ponto de acesso, com uma faixa extremamente pequena. Uma tecnologia proprietária desenvolvida por uma empresa francesa, que apresenta a tecnologia a operar na banda de frequência 868 MHz, com um espectro dividido em 400 canais de 100 Hz [39],[8].

Esta tecnologia usa um método de transmissão de rádio chamado codificação de mudança de fase binária (BPSK). O recetor apenas escuta uma pequena fatia de espectro, permitindo assim atenuar o efeito do ruído. Esta tecnologia necessita de um rádio de baixo custo e uma estação mais sofisticada para gerir a rede [39],[38].

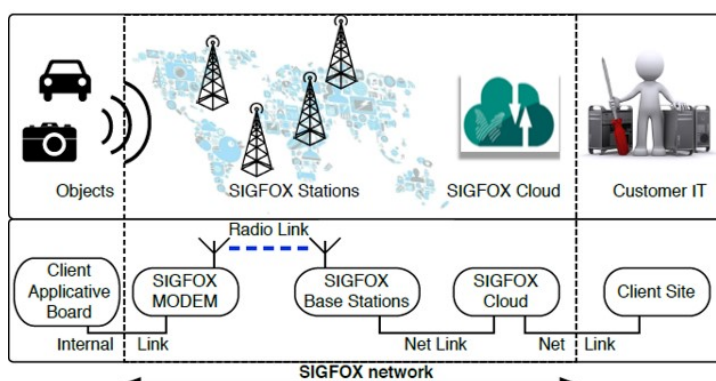


Figura 2.14: Estrutura da rede Sigfox [8]

Cada dispositivo final pode enviar até 140 mensagens por dia, com um tamanho de *payload* de 12 bytes e uma taxa de transferência até 100 bps. A Sigfox afirma que cada ponto de acesso pode lidar com até um milhão de dispositivos finais, com uma área de cobertura de 30-50 km em áreas rurais e 3-10 km em áreas urbanas [39],[37],[8].

## *Weightless*

O *Weightless*-SIG é um grupo de interesse especial que apresenta a tecnologia *Weightless* em três padrões específicos, oferecendo conectividade sem fio para redes LPWAN. O *Weightless*-N (unidireccional, custo mais baixo), *Weightless*-P (comunicação bi-direccional para uma distância um pouco menor e baixo custo) e *Weightless*-W (também bidireccional,

para alcances maiores que 5 quilômetros; mais caro, mas ainda mais barato do que o móvel). O *Weightless-P* também suporta o espectro licenciado [40],[39].

Na Figura 2.3 apresenta-se uma tabela de comparação entre as principais tecnologias LPWAN. Como é possível verificar existem diversas tecnologias, com os seus padrões bem definidos e abordagens de mercado distintas. O sucesso destas tecnologias irá depender da capacidade de oferecer um conectividade de baixa potência para um número elevado de dispositivos. Tendo em conta que a sua distribuição estará dispersa por áreas muito amplas, é possível conseguir um baixo custo global na implementação [39].

Tabela 2.3: Tabela de comparação entre tecnologias LPWAN [25].

	UNB	UNB	NB	LoRa	NB-Fi
<b>Implementation</b>	SIGFOX	Weightless-N	Weightless-P	LoRa	WAVIoT
<b>Frequency Range</b>	Sub-GHz	Sub-GHz	Sub-GHz	Sub-GHz	Sub-GHz
<b>Modulation</b>	BPSK	DBPSK	FDMA/TDMA	CSS	DBPSK
<b>Channel Width</b>	100 Hz	200 Hz	12.5 kHz	125 kHz	100 Hz
<b>Typical Range</b>	10-50 km	5 km	2 km	22 km	17 km
<b>Typical Data Rate</b>	100 bps	30-100 kbps	0.2-100 kbps	0.3-50 kbps	10-100 bps

## 2.6 Módulos de comunicação RF

Neste subcapítulo foi elaborado uma abordagem sobre os módulos de comunicação rádio-frequência. A gama de frequências assiste equipamentos com frequências inferiores a 3 GHz. Os fabricantes originais da tecnologia detêm os direitos sobre o produto, criando o *software* e *hardware* necessário. Apesar de existirem tecnologias proprietárias, existem empresas dedicadas à replicação dos equipamentos.

### 2.6.1 NRF24L01

O dispositivo NRF24L01 é um módulo transmissor/recetor que opera na faixa de frequências ISM, 2.4 - 2.4835 GHz, e é adequado para aplicações de baixa potência. O sistema precisa apenas de um microcontrolador e de alguns componentes eletrónicos. É possível operar e configurar o NRF24L01, através da interface SPI (*Serial Peripheral Interface*)

que facilita o acesso ao mapa de registos (responsável pelos registos de configuração e acessibilidade aos modos de operação do aparelho). Na Figura 2.15, está representado o módulo com uma antena integrada PCB, e os seus pinos constituintes [12],[11].



Figura 2.15: Módulo NRF24L01 [11]

O NRF24L01 é constituído por um mecanismo embutido, baseado na comunicação por pacotes e suportando vários modos de operação, desde o modo manual até a uma operação mais autónoma. A modulação GFSK está integrada no sistema rádio, e existe a possibilidade de configurar alguns parâmetros, como a potência de transmissão, a taxa de transferência de dados, entre outros [12],[11].

O pacote de dados está representado na Figura 2.16, sendo constituído por um *preamble*, um endereço, um campo de controlo de dados, o conteúdo (*payload*) e o CRC.



Figura 2.16: Estrutura do pacote de dados NRF24L01 [12]

O *preamble* serve para preparar o desmodulador do recetor para a entrada de informação. O endereço assegura que o pacote é detetado e recebido pelo recetor, evitando comunicações acidentais com outros módulos. No campo de controlo é definido o tamanho do conteúdo e a identificação do pacote, que permite verificar se o pacote recebido é novo, ou retransmitido [12]. Algumas características principais estão listadas a seguir:

- Potência de transmissão máxima ( $P_{saída}$ ), igual a 4 [4 a -18 dBm];
- Sensibilidade elevada com -98 dBm, para 250 kbps;

- Distância de comunicação até 100 metros;
- Modulação GFSK;
- Modo de repouso absoluto até 900 nA, Modo de espera (*Standby*) até 26 uA;
- Taxa de transferência de dados 250 kbps, 1 Mbps, 2 Mbps;
- Pacotes de dados até 32 bytes;
- Temperatura de funcionamento: -40°C a 85°C;

O processo comunicativo é simples, pois começa com o transmissor a enviar informação para o recetor, esperando este por uma mensagem de confirmação (ACK). No lado do recetor, o pacote recebido envia a confirmação, e entra rapidamente em modo de receção. Na situação de insucesso, ou seja, não haver uma confirmação para o transmissor, inicia o processo de retransmissão, aguardando o retorno de uma mensagem ACK validada [12].

### 2.6.2 RFM69HCW

O módulo RFM69HCW consegue transmitir dados nas frequências da faixa ISM, através de canais de frequência programáveis. Como existe a possibilidade de escolher entre uma faixa mais estreita para comunicar, este aparelho consegue limitar o seu consumo, oferecendo poupança de energia. Dependendo do modo de operação, a configuração oferece alto desempenho quando está a transmitir e poupança nas horas sem trabalho. Aliado a isto, a potência de transmissão pode ser regulada, atendendo às necessidades da aplicação [14].

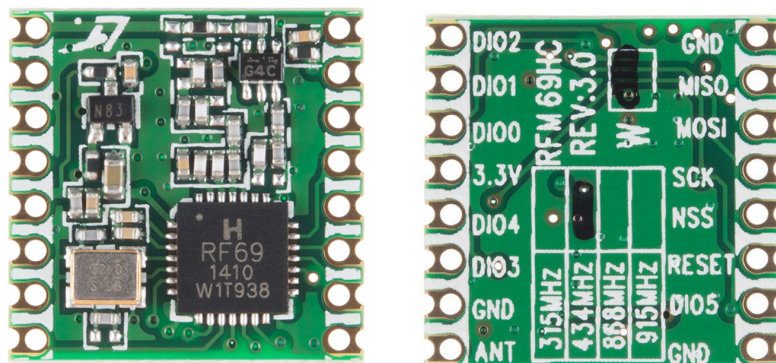


Figura 2.17: Módulo RFM69HCW [13]

O RFM69HCW é capaz de transmitir até 100 mW e até 300 kbps, permitindo uma implementação diferente por aplicação. Por exemplo, potenciar o alcance transmitido, através do aumento da potência de transmissão e redução da taxa de dados. Pelo contrário, nas redes de curto alcance, os dados podem viajar mais depressa, consumindo mais energia da bateria.

Uma fonte de alimentação de 1.8 a 3.6 V é fundamental para o funcionamento do dispositivo, que pode consumir até 130 mA quando está a transmitir. Quando está no modo de receção, usa 16 mA e tem várias opções de descanso que usam menos corrente do que isso. Algumas das características principais encontram-se listadas a seguir [14],[13]:

- Potência de transmissão máxima (Psáida), igual a 100 mW, ou 20 dBm (programável de -18 a 20 dBm em passos de 1 dB);
- Sensibilidade elevada com -120 dBm, para 1.2 kbps;
- Distância de comunicação até os 2000 metros;
- Modulações FSK, GFSK, MSK, GMSK e OOK;
- Modo de repouso absoluto até 0.1 uA, Modo de espera (*Standby*) até 1.5 mA;
- Taxa de transferência de dados FSK até 300 kbps;

- Faixa dinâmica de RSSI (115 dB);
- Pacotes de dados com CRC-16, AES-128 (*Advanced Encryption Standard*), FIFO de 66 bytes;
- Temperatura de funcionamento: -40°C a 85°C;
- Certificações FCC, ETSI.

Para economizar energia e permitir que múltiplos módulos partilhem a mesma frequência, o RFM69HCW envia pequenos pacotes de dados, de forma rápida e repetida. Esta ilusão de uma conexão contínua, é possível porque o transmissor está ligado somente pela fração de segundo que leva para transmitir cada pacote.

Cada *payload* pode conter até 255 bytes de dados, se não for ativado a encriptação AES. Caso contrário a mensagem está limitada a 64 bytes, ou seja (65 bytes de *payload*). Estes dados podem ser caracteres de texto ou números binários. No que diz respeito ao RFM69, é apenas um bloco de dados. Se for preciso enviar mais de 64 bytes, basta enviar mais pacotes.

Como é possível verificar na Figura 2.18, o formato da mensagem corresponde ao formato de comprimento fixo e este depende inteiramente da configuração. O módulo apresenta três tipos de pacotes de dados para uma aplicação mais específica, existindo o modo de comprimento fixo, variável e ilimitado [14],[13].

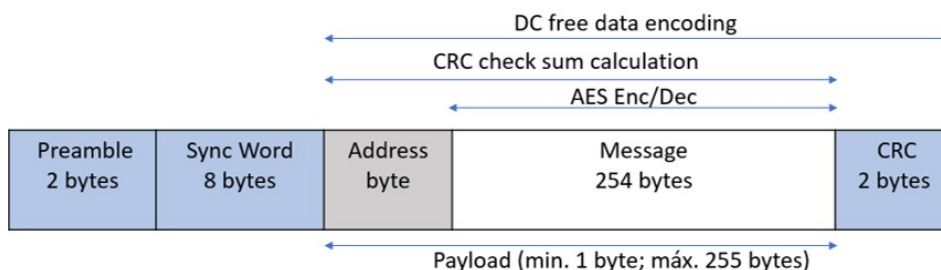


Figura 2.18: Formato da mensagem RFM69 com comprimento fixo. [14]

Em aplicações onde o comprimento do pacote é fixado antecipadamente, este modo de operação pode ser de interesse para minimizar a quantidade de dados. Todos os nós,



pequenas alterações na gama de fatores de propagação e taxa de transferência de dados. Conforme indicado na subsecção 2.5.2, este dispositivo utiliza um protocolo de modulação proprietário, baseado na dispersão de cada bit, em mais elementos de modulação e num maior espectro [15],[41],[42].

Na prática, cada byte (ou símbolo de informação) é representado por mais elementos de modulação (chip), ou seja o chamado fator de propagação (ou *spreading factor*), consiste no número de chips usados para cada símbolo. Quanto maior o fator de propagação, melhor a receção, isto é, a relação sinal / ruído (SNR) fica melhor. O *spreading factor* pode variar de 64 para 4.096. O valor padrão é 128 e pode ser modificado no registo atribuído [15],[41].

Outro benefício é o fato de cada fator de propagação, ser ortogonal, ou seja, vários sinais transmitidos podem ocupar o mesmo canal sem haver interferência.

Para aumentar a robustez das interferências, é implementado um método de deteção e correção, utilizando um conjunto de taxas de codificação. Conforme a Figura 2.20, esta taxa pode ser alterada em resposta às condições do canal - a taxa de codificação pode opcionalmente, ser incluída no cabeçalho do pacote para uso pelo recetor [15],[41].

Se a taxa de código é  $k / n$ , para cada  $k$  bits de informações úteis, o codificador gera um total de  $n$  bits de dados, dos quais  $n-k$  são redundantes.

<b>CodingRate (RegTxCfg1)</b>	<b>Cyclic Coding Rate</b>	<b>Overhead Ratio</b>
1	4/5	1.25
2	4/6	1.5
3	4/7	1.75
4	4/8	2

Figura 2.20: Taxas de codificação e as respetivas sobrecargas nos dados. [15]

Tendo isto em conta, na construção dos pacotes podem existir dois tipos de formatos de pacotes, os explícitos e implícitos. O pacote explícito inclui um cabeçalho curto que contém informações sobre o número de bytes, taxa de codificação e um CRC usado no pacote. A estrutura do formato é mostrada na Figura 2.18 [15],[41].

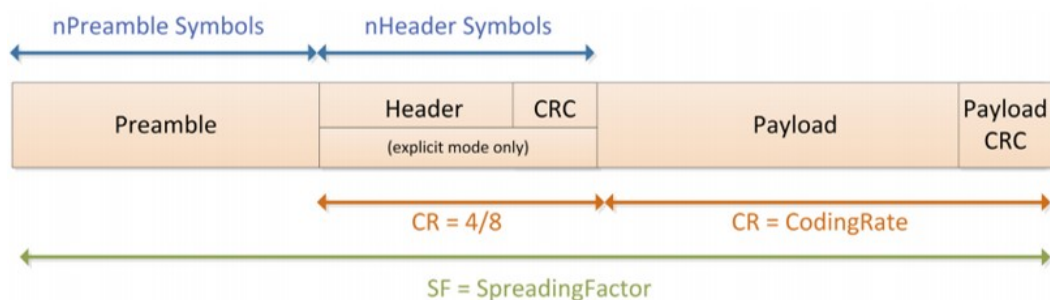


Figura 2.21: Estrutura do pacote de dados LoRa [15].

Quando há necessidade de reduzir o tempo de transmissão de um pacote, e já se sabe em avanço os parâmetros iniciais, aplica-se o modo de cabeçalho implícito. Basicamente é apenas removido o campo *header* com os restantes campos (tamanho de *payload*, presença de CRC e taxa de codificação) a serem configurados manualmente em ambos os pontos de comunicação. O *preamble* é utilizado para sincronizar o recetor com o fluxo de dados na entrada, podendo apresentar até 65535 símbolos diferentes. O campo *payload* tem um tamanho máximo de 255 bytes por pacote (mais um para o CRC)[15],[41]. Algumas das características principais encontram-se listadas a seguir:

- Potência de transmissão máxima (Psaída), igual a 100 mW, ou 20 dBm;
- Sensibilidade elevada com -148 dBm;
- Alcance testado entre 3 a 5 quilómetros;
- Modulações FSK, GFSK, MSK, GMSK, OOK, LoRa;
- Modo de repouso absoluto até 0.2 uA, Modo de espera (*Standby*) até 1.8 mA;
- Taxa de transferência de dados FSK até 300 kbps;
- Faixa dinâmica de RSSI (127 dB);
- Valor máximo de *Link Budget* de 168 dB;
- Pacote de dados até 256 bytes com CRC;

- Sensor de temperatura incorporado e indicador de bateria fraca;
- Temperatura de funcionamento:  $-40^{\circ}\text{C}$  a  $85^{\circ}\text{C}$ ;

A maior parte dos parâmetros são totalmente configuráveis via interface SPI, através do acesso aos registos de configuração. O utilizador pode decidir sobre a largura de banda, fator de propagação e taxa de correção de erros, aumentando assim a sua flexibilidade [15],[41].

#### 2.6.4 Análise comparativa

Como foi analisado, existem diversos módulos RF para este tipo de aplicações. Dentro das possibilidades foram analisados alguns deles, para teste e validação de resultados. Assim dentro destas escolhas, foram analisados os módulos NRF24L01, os RFM69HCW e os SX1278. Na Tabela 2.4 encontra-se uma pequena tabela com esta análise prévia, que resume as características principais dos aparelhos.

Tabela 2.4: Tabela de comparação entre os módulos de comunicação [14].

	<b>NRF24L01</b>	<b>RFM69HCW</b>	<b>SX1278</b>
<b>Frequência</b>	2.4 GHz	433 MHz	433 MHz
<b>Alcance *testados</b>	100 metros	até 2 km	3 a 5 km
<b>T_trans</b>	1 Mbps e 2 Mbps	1.2 - 300 kbps	0.018 - 300 kbps
<b>I_min/I_máx</b>	900 nA / 320 uA	0.1 uA / 1.5 mA	0.2 uA / 1.8 mA
<b>RX máx/TX máx</b>	12.3 mA / 11.3mA	16 mA / 130 mA	12 mA / 120 mA
<b>P_máx_trans</b>	4 dBm	20 dBm	20 dBm
<b>Sensibi_recetor</b>	-85 dBm	-120 dBm	-127 dBm
<b>Custo (euros)</b>	0.82 - 1.81 euros	2.11 - 2.65 euros	3.04 - 7.56 euros

Como é possível verificar a média da potência máxima de transmissão ( $P_{\text{máx\_trans}}$ ), permite atingir até os 20 dBm com uma sensibilidade muito acentuada para a receção. Isto é, a sensibilidade corresponde ao nível de intensidade que o recetor pode receber em determinado ponto. O valor negativo indica que o módulo pode receber sinais com pouca intensidade.

As modulações correspondem a tipos de modulações digitais, com a variação de amplitude (ASK ou OOK) ou frequência (FSK e GFSK). O campo  $I_{\text{mín}}/I_{\text{máx}}$ , diz respeito

à corrente mínima e máxima possível nos modos de operação de descanso. Tipicamente existem dois modos, o modo de repouso absoluto e o modo de espera. Também podem ser implementados outro tipo de modos intermédios para permitir um desempenho razoável aliado a poupança energética.

No campo Rx máx/Tx máx, diz respeito ao consumo máximo de corrente, quando este está a receber dados ou simplesmente a transmitir. A média de preços para um módulo mantêm-se abaixo dos 5 euros, pelo que implementar uma rede com inúmeros nós, torna-se economicamente viável para o custo final do projeto.

O módulo NRF24101 apresenta algumas vantagens, como o baixo custo, taxas de transmissão elevadas e consumos reduzidos no trabalho e no descanso. No entanto comparado com os restantes módulos, não possibilita comunicações acima dos 100 metros, devido à antena PCB e a limitação da potência de transmissão ( $4 \text{ dBm} < 20 \text{ dBm}$ ).

O módulo RFM69HCW tem características muito idênticas ao módulo SX1278, pois ambos utilizam a mesma frequência de operação e têm consumos de corrente parecidos. A vantagem do módulo RFM69HCW é ser mais barato que o SX1278 e apresentar correntes de descanso ligeiramente inferiores as do seu semelhante. Como ponto negativo temos o alcance que foi testado experimentalmente até aos 2 km.

Este valor é inferior ao testado pelo SX1278 que seguramente é um modelo interessante para futuros testes dentro de um alcance até 5 km. O módulo SX1278 tem o maior nível de sensibilidade e consegue transmitir e receber dados com menos corrente que a do RFM69HCW.

## 2.7 Ferramenta de previsão *Radio Mobile*

A ferramenta *Radio Mobile* é um programa gratuito e poderoso para traçar padrões de rádio frequência e facilitar na previsão do desempenho de sistemas de rádio. É possível produzir mapas virtuais a cores, na escala de cinza e raios-x através de dados de elevação de terreno disponíveis livremente [17][16].

O autor Roger Coudé com o VE2DBE, produziu esta ferramenta, que é baseada no

modelo de propagação ITM (Irregular Terrain Model) (também conhecido por modelo Longley-Rice) e permite simular uma operação na gama entre os 20 MHz e os 20 GHz. A versão online está disponível na Internet e permite o acesso a qualquer utilizador, existindo também uma versão para Windows, Linux e Mac [17],[16].

O desempenho individual da unidade é especificado como um Sistema Operacional de Rádio, que calcula através de parâmetros iniciais a cobertura na receção do sinal. Todos as ligações de rádio entre unidades podem ser examinadas, para avaliar tanto a ligação como ajustar as configurações iniciais (procurando obter a melhor simulação possível). Após o registo do utilizador nesta ferramenta online, é possível entrar num menu intuitivo (ver Figura 2.22) que permite aceder a várias opções [17],[16].

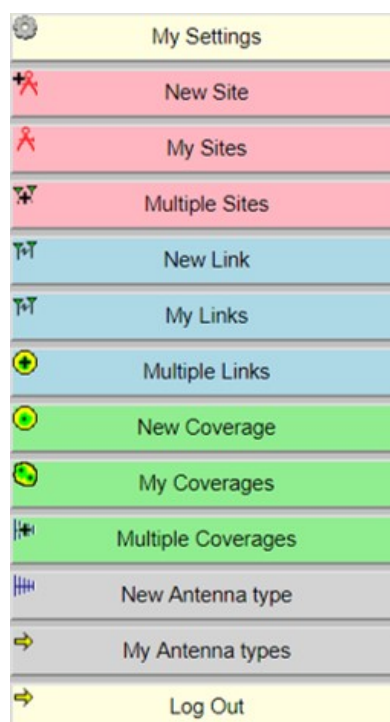


Figura 2.22: Menu principal da ferramenta online [16].

A aplicação usa informação digital do terreno com um modelo matemático para simular transmissões entre dois pontos fixos (*Radio Link* - Campos a azul) ou entre um ponto fixo e um móvel (*Radio Coverage* - Campos a verde). Esta informação está agrupada em

três bases de dados: densidade populacional, elevação e cobertura do terreno. O modelo matemático utiliza o modelo de propagação Longley-Rice, o método de dois raios e a estimativa *Path Loss*. Estas previsões vão ao encontro do planeamento de ligações, referido na subsecção 2.1.4, que permitem avaliar teoricamente o seu comportamento [17],[16].

A simulação de uma ligação entre dois pontos geográficos necessita da introdução de parâmetros iniciais como é apresentado na Figura 2.23.

New Link		
From: Trans_MeioEMerlães*		
From	Transmissor	
Antenna height (m above ground)	6	19.69 ft
To: MeioEstrMerlães		
To	MeioEstrMerlães	
Antenna height (m above ground)	1.5	4.92 ft
Description	Trans_MeioEMerlães**	
Frequency (MHz)	433	
Tx power (Watts)	0.013	11.14 dBm
Tx line loss (dB)	3	
Tx antenna gain (dBi)	2	
Rx antenna gain (dBi)	2	
Rx line loss (dB)	0.5	
Rx threshold (µV)	0.5	-113.02 dBm
Required reliability (%)	70	
Use land cover	<input checked="" type="checkbox"/>	
Use two rays	<input checked="" type="checkbox"/>	
<input type="button" value="Define as default values"/> <input type="button" value="Restore original values"/>		
<input type="button" value="Submit"/>		

Figura 2.23: Criação de uma simulação de ligação ponto a ponto [17].

A ligação Trans\_MeioEMerlães tem dois pontos: um que está 6 metros acima do solo que servirá como ponto fixo (topo do telhado de uma casa), enquanto que o outro consiste no ponto móvel a 1.5 m do solo (correspondendo à altura média de um adulto).

Os ganhos e perdas do recetor e transmissor também são inseridos, bem como a sensibilidade do recetor designada por *threshold*. Antes de definir estas ligações, o utilizador necessita de estabelecer os pontos geográficos no mapa, atribuindo um nome ou uma de-

signação genérica. Os resultados da simulação demonstram o perfil geográfico da ligação com as zonas de Fresnel e uma previsão das zonas obstruídas como árvores ou edifícios.

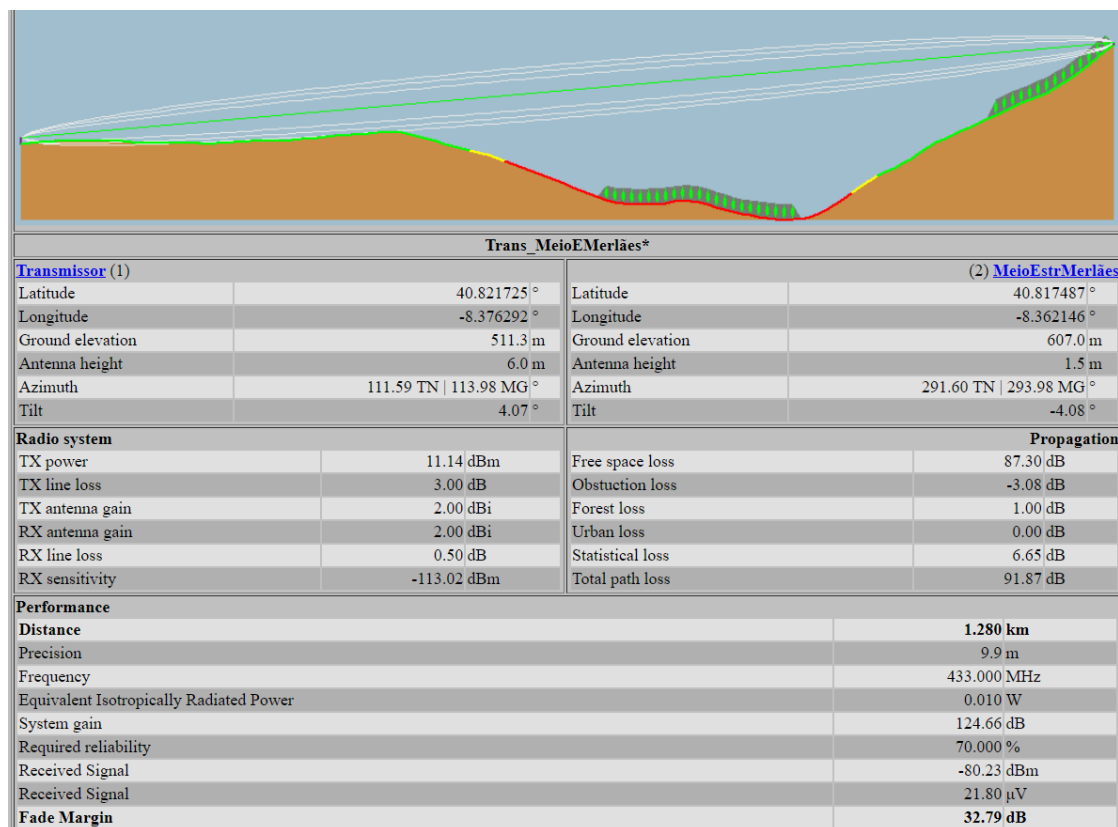


Figura 2.24: Resultados de uma simulação - ligação ponto a ponto [17].

Neste caso para uma distância de 1.3 km e um grau de confiança de 70 %, o nível de intensidade esperada é de -80,23 dBm.

A intensidade do sinal recebido varia consoante estes obstáculos, que bloqueiam as comunicações e impedem que o sinal chegue ao outro ponto de comunicação. Como indicado na Figura 2.24 as zonas sem árvores apresentam-se com uma linha verde e as zonas cobertas por árvores estão a vermelho. O pequeno traço amarelo representa a transição entre zonas livres e obstruídas. Esta linha tricolor reflete o funcionamento da ligação nos pontos intermédios do percurso, ou seja, zonas cobertas por floresta situadas em encostas sem linha de visão (assinaladas a vermelho) não são pontos seguros para

comunicar com um módulo rádio.

O programa produz também uma imagem colorida da cobertura de uma unidade rádio ou um conjunto de estações que mostram os níveis esperados de sinal na recepção. Os níveis são exibidos usando uma das seguintes unidades especificadas pelo utilizador: S-units, dBm, uV, uV/m.

Se for necessário executar uma previsão da cobertura numa área específica, o autor retorna para o menu apresentado na Figura 2.22 e seleciona a opção *New Coverage*. Depois surge um formulário (apresentado na Figura 6.2 do Anexo 6.3) para introdução de parâmetros iniciais com configurações para a criação da imagem da cobertura (ver Figura 2.25).

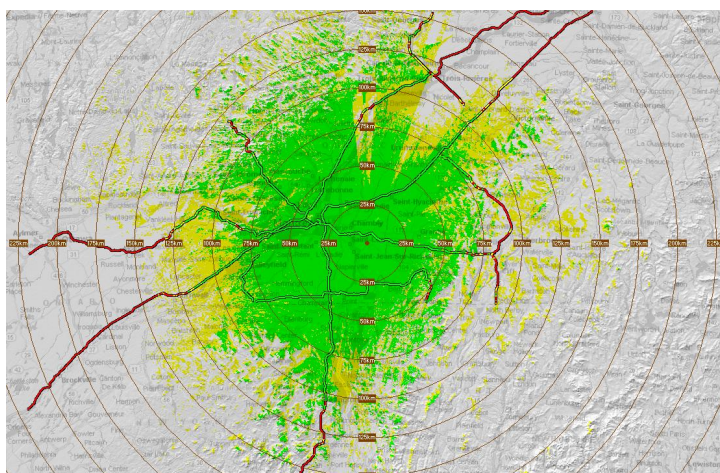


Figura 2.25: Representação gráfica de um mapa de cobertura [17].

Como se trata de simulação, os resultados retirados são apenas aproximações, pelo que o utilizador deve acautelar-se com as discrepâncias existentes em transmissões reais.



## Capítulo 3

# Arquitetura do sistema

Neste capítulo iremos abordar a arquitetura do sistema, isto é, analisar os requisitos do sistema e esboçar um diagrama geral que mostra de uma maneira simples os componentes e módulos a incorporar no protótipo. Para além disso, o estudo irá debruçar-se sobre as ferramentas de hardware e/ou software, procurando explicar a estrutura geral com as alternativas escolhidas para a implementação.

### 3.1 Requisitos do sistema

O principal objetivo do sistema consiste em desenvolver um protótipo com uma comunicação de longo alcance (distâncias até 2 km), para uma solução agrícola ou industrial, ou seja, uma aplicação que possa ser implementada numa área ampla. A ideia consiste em enviar dados dos sensores de várias estações para uma interface web.

Como a localização das estações poderá situar-se em zonas remotas, um dos requisitos do sistema consiste em utilizar a comunicação rádio para interagir com dispositivos, através da troca de informação ou dados de controlo. A tecnologia deverá funcionar em rede com vários dispositivos interligados entre si numa estrutura de rede *mesh* ou estrela.

Este sistema será constituído por três estações com funções idênticas entre si. Assim a implementação da rede permitirá estabelecer ligações entre dispositivos diferentes. O fluxo de mensagens irá assim variar consoante o destinatário pretendido. Cada estação

tem de ser independente, para transmitir as informações dos sensores, mesmo que não haja Internet para uma ligação Web. Assim, esta tem que ser dotada de funcionalidades de rádio ou outro tipo de aparelho transmissor. Desta forma a rede de comunicações funcionará com uma estação central ligada à Internet, e as restantes (estações secundárias) sem esse acesso. Os materiais necessários englobam o hardware, onde se agrupam os módulos rádio, microcontroladores, eventuais atuadores e sensores; e o software disponível para configurar as ligações.

O sistema final será testado para verificar o comportamento energético, analisando o consumo de corrente dos módulos rádio e da estação como um todo. Esta implementação será analisada para comparar em termos de custos, fazendo uma abordagem económica do sistema. Neste sentido perceber de que forma o sistema final pode competir com outras soluções existentes no mercado.

## 3.2 Diagrama do sistema

De acordo com a análise de requisitos descrita na secção 3.1 foi elaborado o diagrama da Figura 3.1. Neste gráfico está representada a estrutura do sistema que será utilizada para verificar o funcionamento das ligações.

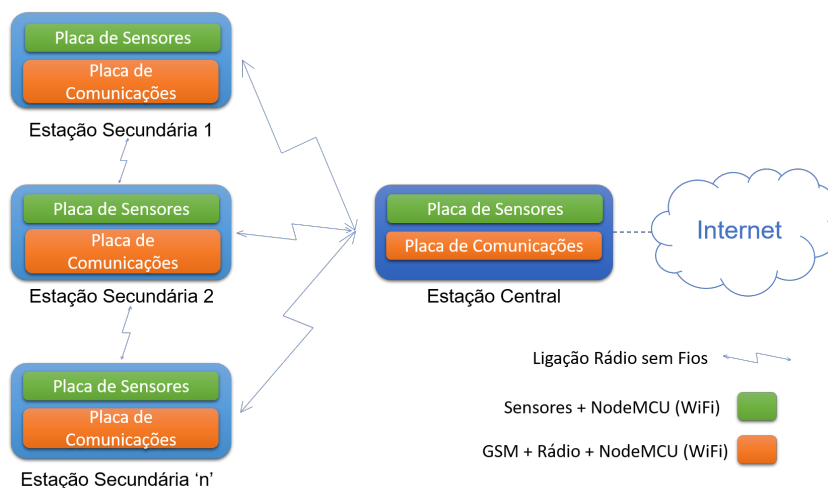


Figura 3.1: Diagrama geral do sistema.

Na Figura 3.1 temos representados quatro módulos do sistema com a possibilidade para aumentar a rede para 'n' estações secundárias. Os módulos secundários têm a mesma função, portanto todos eles comportam-se como repetidores desta rede. Uma estação secundária corresponde a um nó, que poderá transportar a informação por intermediários, até ao destinatário (estação central). Se a estação 1 estiver fora do alcance da estação central, existe a alternativa de mapear uma rota pela estação 2, até à estação central. Assim é possível, enviar e receber dados simultaneamente com a atribuição de rotas atualizadas para os nós.

Cada estação é constituída por duas placas: uma com um conjunto de sensores e outra com as comunicações. A placa de sensores recolhe informação através dos sensores de temperatura, humidade e vento, enquanto que a placa de comunicações lida com os dispositivos rádio. As operações são controladas por um NodeMCU que possui uma placa WiFi integrada. Após a recolha, é criada uma mensagem única que envia a informação dos sensores para a Internet (*cloud*). Se tal não for possível, é enviada essa informação com um módulo rádio, que a transmite para um destinatário com acesso à web (estação central). A incorporação do módulo GSM é considerado um requisito para o sistema, que funcionará como último recurso para salvaguardar as estações na transmissão de dados.

## 3.3 Hardware

Neste subcapítulo iremos analisar alguns dos componentes integrantes do protótipo para permitir uma abordagem genérica e facilitar a compreensão dos materiais utilizados. Esta análise mais simples consiste em focar nos aparelhos mais importantes do sistema, deixando a parte mais técnica e específica para os próximos capítulos.

### 3.3.1 Módulo ESP8266 (NodeMCU)

O ESP8266EX é um *chip* produzido por um fabricante chinês, a Espressif, e consiste numa solução Wi-Fi altamente integrada com um design compacto e um desempenho satisfatório. Este aparelho permite ser aplicado a outros micro-controladores como um

adaptador de Wi-Fi, através de interfaces SPI/I2C/UART. As suas capacidades de rede permitem que este possa ser executado como um aplicação autónoma ou como um escravo para uma unidade MCU (*MicroController Unit*) *host*. A sua *cache* integrada otimiza o desempenho e a memória do sistema [18].

Além das funcionalidades Wi-Fi, o ESP8266EX também integra uma versão aprimorada do processador, Tensilica's L106 Diamond, de 32 bits e SRAM integrado. Também pode ser aplicado com sensores externos e outros dispositivos através dos 17 GPIOs (dos quais apenas 11 GPIOs estão disponíveis para uso), resultando num custo reduzido na implementação [18]. O seu interior integra também interruptores de antena, conversor RF, amplificador de potência, amplificador de baixo ruído, filtros e uma unidade para a gestão da energia (ver Figura 3.2).

Este circuito integrado foi sendo implementado numa vasta gama de modelos, que apresentam diferenças, no que toca à quantidade de pinos disponíveis e o tamanho do módulo. Até agora os modelos existentes, estão numerados e podem ir do ESP-01, até ao ESP-12 [19] [43].

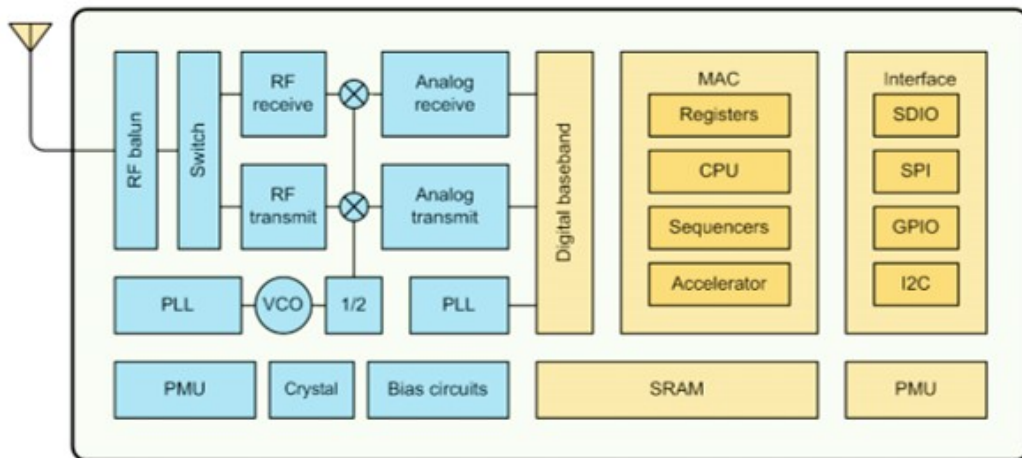


Figura 3.2: Diagrama de blocos do ESP8266 [18].

Algumas das suas características principais estão apresentadas na seguinte lista [18]:

- Micro-controlador de 32 bits, com uma velocidade de *clock* de 80 MHz, e ADC de

10 bits;

- Normas 802.11 b/g/n com a pilha protocolar TCP/IP;
- Wi-Fi 2.4 GHz, suporta WPA/WPA2;
- Suporta modos de operação STA/AP/STA+AP (STA - Station; AP - Access Point);
- Suporta função para ligação inteligente para os aparelhos Android e iOS;
- Suporta SDIO 2.0, SPI, UART, I2C, I2S, IR, PWM, GPIO's;
- Modo de repouso absoluto <10uA, Corrente de fuga no modo de espera < 5uA;
- Tempo de acordar e transmitir pacotes < 2ms;
- Consumo de energia elétrica em modo de espera < 1.0mW;
- Temperatura de funcionamento: -40°C a 125°C;
- Certificações FCC, CE, TELEC, WiFi Alliance e SRRC.

Dependendo do modelo, existem algumas designações que especificam o tipo de aparelho que estamos a falar. No entanto os aparelhos não variam muito, podendo ter ligação USB ou não, para a programação dos mesmos. As variações mais recentes deste aparelho, utilizam o módulo ESP-12E, uma versão melhorada do último da gama que é incorporado nas placas de desenvolvimento NodeMCU ou Wemos da AI-Thinker (Figura 3.3) ou da Adafruit para versões mais oficiais [19] [43].

O fabricante varia, mas tipicamente estas placas correspondem a *firmwares* baseados em eLua para o ESP8266. As versões mais recentes, como o ESP-14, ainda não foram implementadas em muitas placas de desenvolvimento [19] [43].

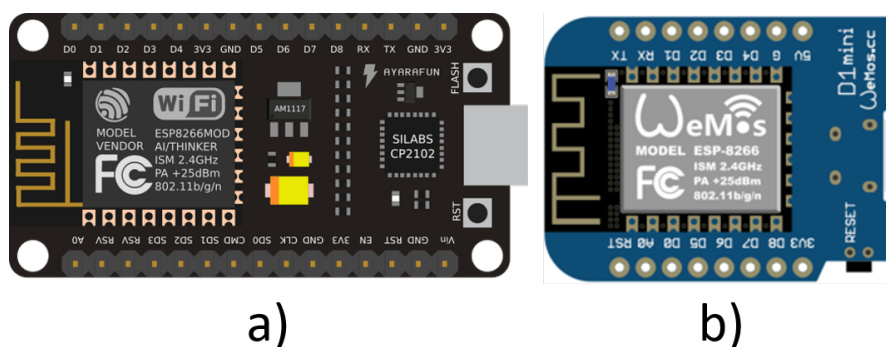


Figura 3.3: a) NodeMCU ESP-12E b) Wemos D1 mini [19].

### 3.3.2 Módulo A6 GSM mini

O módulo A6 GSM mini é uma placa de desenvolvimento que se baseia no dispositivo GPRS/GSM A6 da Ai-Thinker. O aparelho está disponível para GPRS e transmissão remota de dados através de mensagem SMS. Deste modo consiste numa versão mais compacta e de baixo consumo energético. No modo de suspensão, o consumo de corrente é menor que 3 mA, sendo alimentado por uma fonte externa de 5 V [20][44].

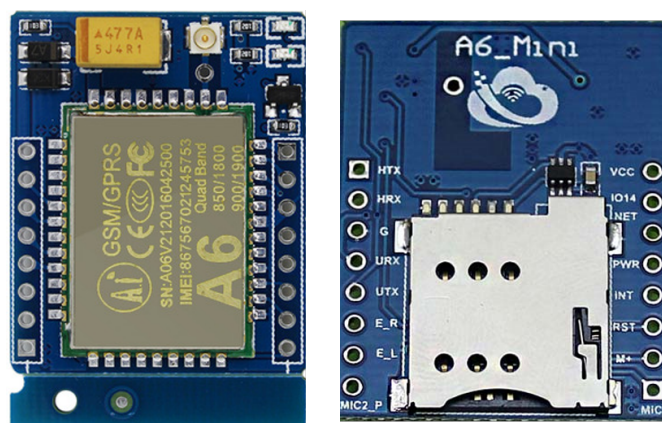


Figura 3.4: Módulo A6 GSM mini [20]

Algumas das características do aparelho encontram-se listadas a seguir [20] [44]:

- Temperatura operacional de -30 a 80 °C;
- Interface de comunicação UART;
- Suporta as quatro bandas GSM/GPRS, incluindo 850, 900, 1800, 1900 MHz;
- Suporta chamadas de voz, SMS, tráfico de dados GPRS (Download de 85,6 Kbps e Upload de 42,8 Kbps);
- Suporta os comandos padrão GSM 07.07, 07.05;
- Suporta áudio digital e analógico para codificação de voz HR, FR, EFR, AMR;
- Possui certificação ROHS, FCC, CE e CTA.

Este aparelho possui umas dimensões de 23 mm de comprimento e 17 mm de largura, possuindo uma sensibilidade de -108 dB. Os pinos disponíveis permitem comunicar com o microcontrolador através das portas Rx/Tx, suportando um conjunto de comandos AT da Ai-Thinker [20] [44].

### 3.3.3 Sensores e conversor ADS1115

Os sensores são introduzidos para levantar os dados relativos às condições atmosféricas e do solo (ver Figura 3.5). Este grupo é constituído pelo sensor DHT22, um sensor de temperatura e humidade atmosférica; o sensor DS18B20, um termómetro digital que vem encapsulado para recolher a temperatura de líquidos; um pluviómetro; um anemómetro; um cata-vento; e um sensor da humidade do solo.

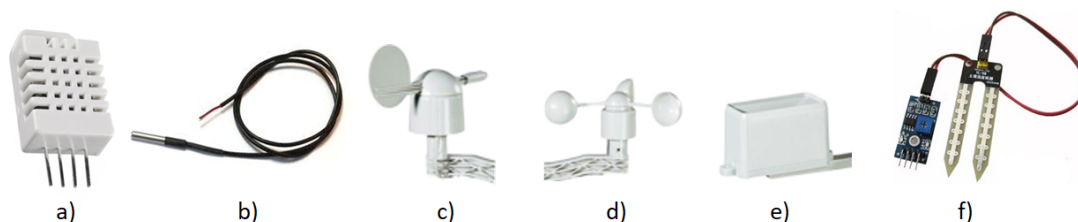


Figura 3.5: a) DHT22 b) DS18B20 c) Cata-vento d) Anemómetro e) Pluviómetro f) HL-69.

Basicamente o sistema incorpora uma estação meteorológica, com acessibilidades à rede WiFi, para análise e tratamento de dados. A placa de sensores foi construída pelo Miguel Mira [21], que desenvolveu esta parte do projeto, criando o sistema bem como o *software* para a recolha de dados.

O sensor DHT22 permite medir a humidade relativa do ar entre 0 a 100 %, e também permite a recolha da temperatura atmosférica entre -40 a 80°C.

O sensor DS18B20 consegue analisar a temperatura do solo através do seu encapsulamento fechado e à prova de água, recolhendo temperaturas compreendidas entre -55 a +125°C. Este aparelho tem a particularidade de utilizar apenas um cabo para enviar os dados (protocolo *OneWire*), permitindo assim reduzir na quantidade de pinos da comunicação.

Tanto o anemómetro como o pluviómetro são dispositivos que utilizam interrupções externas sempre que há uma variação do estado lógico do pino. A função da interrupção externa permite converter o número de impulsos em dados reais (velocidade em km/h ou precipitação em mm/h).

A direção do vento é recolhida pelo cata-vento, um sensor analógico que gera uma tensão entre 0 a 3V, consoante a posição cardinal do sensor. Deste modo, o autor utiliza um conversor externo ADS1115, para a conversão do valor analógico para digital, que utiliza a comunicação I2C para a ligação ao microcontrolador.

Também é incluído um sensor de humidade do solo HL-69, um sensor analógico que retorna um valor de tensão em função da humidade dos seus conetores. Como no cata-vento, esta leitura de tensão é inserida no conversor ADS1115, retornando um valor entre 0 a 32767.

O conversor ADS1115 potencia as capacidades do microcontrolador, pois permite adicionar mais 4 entradas analógicas ao microcontrolador que apenas possui uma. A comunicação I2C permite acoplar mais do que um conversor, através da partilha das portas SDA/SCL (até 4 sensores em conjunto).

Na Figura 6.1 do Anexo 6.3 é possível verificar as quatro ligações com o respetivo endereçamento das linhas de comunicação. Para além da resolução de 16 bits, o incremento

máximo de conversores resulta em 16 entradas analógicas.

## 3.4 Software

A programação das placas de desenvolvimento baseia-se no ambiente integrado de desenvolvimento (IDE) do Arduino. Esta ferramenta consiste num editor de linguagem C, com capacidade para compilar e programar de uma forma intuitiva as placas Atmega ou equivalentes. O ESP8266 produzido pela EspressIf possuiu um ambiente de desenvolvimento pré-definido, que foi mais tarde integrado no Arduino IDE para permitir um acesso mais amplo a toda a comunidade.

O desenvolvimento de protótipos potenciou o uso deste ambiente gráfico, pelo facto deste não apresentar custos para os seus utilizadores. A comunidade aberta disponibiliza também bibliotecas destinadas a diferentes tipos de projetos, portanto a implementação torna-se mais rica e interessante.

Assim sendo, para programar o ESP8266 (ou seja o NodeMCU), é necessário descarregar uma versão oficial do programa (versão mais recente 1.8.1) e instalar no sistema operativo. Depois da instalação, o *software* precisa de uma extensão adicional que pode ser descarregada da página GitHub ou então inserir o endereço URL nas configurações. Para tal, após abrir o programa, é necessário:

- Selecionar no menu Ficheiro > Preferências > Configurações;
- Escrever o URL `http://arduino.esp8266.com/stable/package_esp8266com_index.json` no campo URL Adicionais do Gestor de Placa;
- Ir a Ferramentas > Placa > Gestor de Placas e procurar por 'esp8266'. Selecionar a versão mais recente e instalar.
- (\*Adicionar bibliotecas) Ir a Rascunho > Incluir Biblioteca > Gerir Bibliotecas ou Adicionar Biblioteca ZIP.

O pacote de instalação ESP8266 engloba as definições para programar os módulos ESP e suas placas de desenvolvimento (incluindo o NodeMCU, o WeMos D1, Adafruit

HUZZAH, entre outras). Dependendo da placa, pode ser necessário instalar as *drivers* dos conetores USB-Serial correspondentes (CH340 para o NodeMCU e CH2102 para Adafruit HUZZAH). Outras placas com um chip FTDI não necessitam das *drivers*, pelo que funcionam logo que haja conexão.

Na Figura 3.6 está representado a janela de visualização do menu Configurações, onde será colocado o URL no campo indicado.

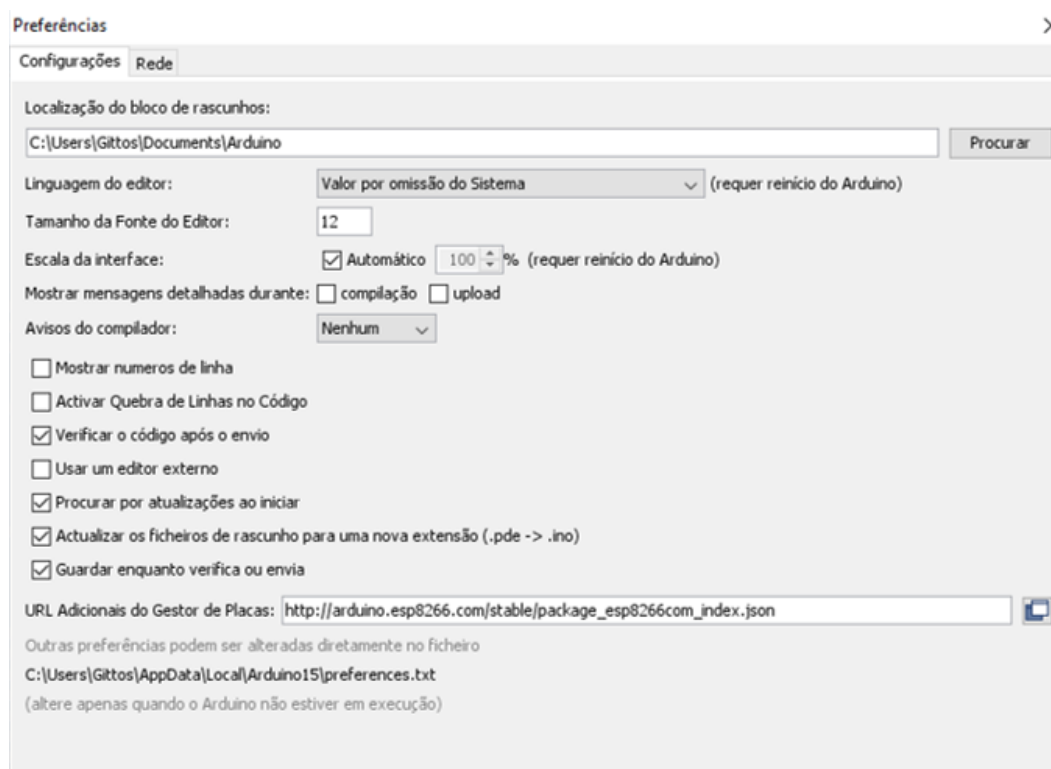


Figura 3.6: Janela de visualização do menu Preferências.

A última etapa significa a instalação de bibliotecas que permitem aceder a conteúdo externo de outros projetos. Nesta etapa foi adicionado um pacote de bibliotecas designado por RadioHead para o envio e receção de mensagens em sistemas rádio. Esta ferramenta consiste em dois tipos de classes: os *drivers* e os gestores(*managers*).

Os *drivers* permitem o acesso de baixo nível aos diferentes pacotes de rádio e outros tipos de transporte de mensagens (consistem nos módulos físicos que podem ser utiliza-

dos), enquanto que os *managers* correspondem à camada mais alto nível da comunicação. As aplicações são geridas pelos *managers* que utilizam os *drivers* para aceder aos módulos rádio na troca de mensagens. Num programa estas duas classes têm que ser inicializadas, no entanto a *driver* pode funcionar sozinha com uma comunicação insegura.

A classe `RHGenericDriver` define as funções que devem ser fornecidas por qualquer driver. Os diferentes tipos de driver implementarão todas as funções abstratas e, poderão substituir outras funções nesta subclasse, ou talvez adicionar novas funções exigidas por determinado driver.

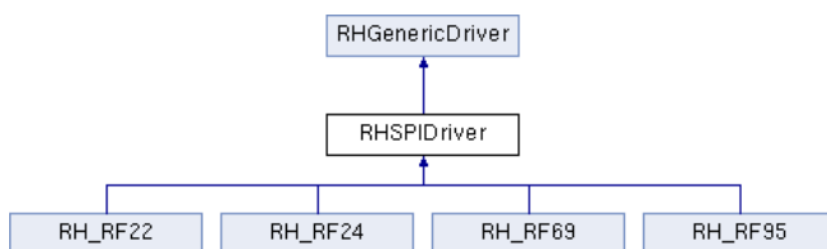


Figura 3.7: Diagrama de herança das drivers

Dentro das *drivers* disponíveis temos: `RH_RF22`, para a família RF22, `RH_RF69` para os módulos RFM69, `RH_NRF24` para os módulos Nordic nRF24101 e suas variantes, bem como o *driver* `RH_RF95` para a família RFM95/96/97/98.

No campo dos *managers* existem quatro versões:

- no **RHDatagram**, as mensagens não são fiáveis, mas apresentam um tamanho variável, com mecanismos para retransmissão.
- no **RHReliableDatagram**, as mensagens endereçadas, podem ser retransmitidas e confirmadas com ACKs.
- no **RHRouter**, as mensagens "saltam" do nó origem para o nó destino, com encaminhamento manual.
- no **RHMesh**, o método de salto é utilizado com a descoberta automática de rotas para os nós.

Todos os *drivers* têm a mesma API idêntica, ou seja, o utilizador pode usar qualquer *driver* com qualquer um dos *managers* descritos anteriormente. A API é composta por uma série de funções acessíveis somente por programação, e que permitem utilizar características do software menos evidentes ao utilizador tradicional.

A Figura 3.8 representa a estrutura da arquitetura, mostrando a herança dentro da classe base RHDatagram. Este método de organização permite reaproveitar código com o uso de determinadas funções em processos mais específicos. Quando uma classe herda da outra procura guardar métodos utilizados na classe-pai, e especializar-se com métodos (operações) diferentes.

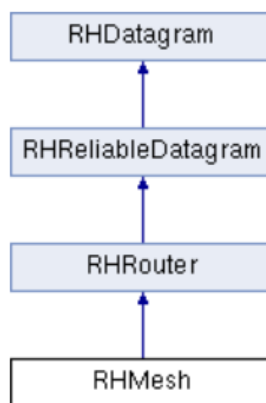


Figura 3.8: Diagrama de herança das managers

A superclasse RHDatagram tem a classe RHReliableDatagram como classe filho e a classe RHRouter tem como filho a classe RHMesh. A biblioteca RHMesh utiliza diferentes formatos de mensagens, tais como:

- **MeshApplicationMessage** (tipo de mensagem `RH_MESH_MESSAGE_TYPE_APPLICATION`). Define uma mensagem da camada de aplicação criada para o construtor RHMesh.
- **MeshRouteDiscoveryMessage** (tipos de mensagens `RH_MESH_MESSAGE_TYPE_ROUTE_DISCOVERY_REQUEST` e `RH_MESH_MESSAGE_TYPE_ROUTE_DISCOVERY_RESPONSE`). Transporta mensagens de descoberta de

rota (*broadcast*) e respostas (*unicast*).

- **MeshRouteFailureMessage** (tipo de mensagem `RH_MESH_MESSAGE_TYPE_ROUTE_FAILURE`). Notifica as mensagens de erro causadas por falhas nas rotas.

A herança permite usar classes já definidas para derivar novas classes. Uma nova classe herda propriedades (dados e métodos) da classe base. O membro da classe derivada pode usar os membros (funções ou variáveis) públicos (*public*) e protegidos (*protected*) da sua classe base (como se fossem declarados na própria classe). Deste modo as funções de membro público existentes na classe `RHMesh` são:

- **RHMesh**(`RHGenericDriver& driver, uint8_t thisAddress = 0`); Construtor.  
 /driver - A interface rádio para transportar as mensagens.  
 /thisAddress - O endereço para atribuir um endereço para este nó. Predefinido com valor 0.
- `uint8_t sendtoWait(uint8_t* buf, uint8_t len, uint8_t dest, uint8_t flags = 0)`;  
 Envia uma mensagem para o nó destino. Inicializa o cabeçalho da mensagem e chama a função `route()` que procura na tabela de encaminhamento o próximo nó "salto" (i.e. nó que serve de intermediário à mensagem) para entregar. Se nenhuma rota é conhecida, inicia a descoberta de rotas e espera por uma resposta. Se existir rota, envia a mensagem para esse nó e espera por um ACK (acknowledgement) do próximo salto.  
 /buf - O conteúdo da mensagem  
 /len - O seu tamanho em bytes  
 /dest - O endereço do nó destino. Se o endereço for `RH_BROADCAST_ADDRESS` (255), a mensagem será transmitida para todos os nós mais próximos.  
 /flags - Marcadores opcionais para utilização na camada de aplicação ou nas subclasses. O recetor pode recuperar estes marcadores com `recvFromAck()`.  
 /return O código do resultado:

- RH\_ROUTER\_ERROR\_NONE - Mensagem foi encaminhada e entregue ao próximo nó "salto";
  - RH\_ROUTER\_ERROR\_NO\_ROUTE - Não existe uma rota para dest na tabela de encaminhamento local;
  - RH\_ROUTER\_ERROR\_UNABLE\_TO\_DELIVER - Não é possível entregar para o próximo salto (geralmente porque não reconhece devido a estar fora do alcance).
- bool **recvfromAck**(uint8\_t\* buf, uint8\_t\* len, uint8\_t\* source = NULL, uint8\_t\* dest = NULL, uint8\_t\* id = NULL, uint8\_t\* flags = NULL);

Inicia o recetor para depois processar e encaminhar mensagens endereçadas a outros nós, entregando-as para esses endereços. Se houver uma mensagem válida disponível para esse nó (ou RH\_BROADCAST\_ADDRESS), envia uma confirmação para o último endereço do salto (bloqueando até que isso seja concluído) e copia o conteúdo da mensagem (payload) para buf (sucesso = retorno positivo). Esta é a função preferida para obter mensagens endereçadas a este nó. Se a mensagem não for transmitida, confirme ao remetente antes de retornar.

/buf - Localização da cópia da mensagem recebida;

/len - Espaço disponível em buf. Define o número de bytes copiados.

/source - Se não for NULL, a variável é definida para o endereço de origem (SOURCE).

/dest - Se não for NULL, a variável é definida para o endereço de destino (DEST).

/id - Se não for NULL, a variável é definida como um identificador (ID).

/flags - Se não for NULL, a variável é definida para os marcadores (SOURCE).

/return - O código do resultado: *true* se uma mensagem válida foi recebida para esse nó e copiada para buf.

- bool **recvfromAckTimeout**(uint8\_t\* buf, uint8\_t\* len, uint16\_t timeout, uint8\_t\* source = NULL, uint8\_t\* dest = NULL, uint8\_t\* id = NULL, uint8\_t\* flags = NULL);

Esta função bloqueia até que uma mensagem válida da camada de aplicação esteja disponível para esse nó ou o tempo limite esgote. É muito semelhante à `recvfromAck()`, com os parâmetros entre ambas as funções são idênticos. O único parâmetro diferente é o campo *timeout*.

*/timeout* - Tempo máximo de espera em milisegundos.

As funções de membro protegido do `RHMesh` consistem em:

- virtual void **peekAtMessage**(RoutedMessage\* message, uint8\_t messageLen);

Função interna que inspeciona mensagens a serem recebidas e ajusta (se necessário) a tabela de rotas. É chamada por `recvfromAck()` após receber a mensagem de `RHReliableDatagram`.

*/message* - Apontador para a mensagem `RHRouter` que será recebida;

*/messageLen* - Tamanho da mensagem em bytes.

- virtual uint8\_t **route**(RoutedMessage\* message, uint8\_t messageLen);

Isto é virtual, o que permite a utilização da função `route()` por outras subclasses. Esta função é chamada por `sendtoWait` depois do cabeçalho da mensagem ser preenchido.

*/message* - Apontador para a mensagem a ser enviada;

*/messageLen* - Tamanho da mensagem em bytes.

- virtual bool **doArp**(uint8\_t address);

Tenta criar uma rota para um endereço específico. Esta função procura um endereço durante 4 segundos e bloqueia quando há sucesso. É virtual logo as subclasses podem sobrepor-se à implementação na superclasse.

*/address* - O endereço físico para encontrar rota

*/retorno* - *true* se o endereço possui uma nova rota, adiciona-a na tabela de encaminhamento.

- virtual bool **isPhysicalAddress**(uint8\_t\* address, uint8\_t addresslen);

Testa se o endereço fornecido de tamanho `addresslen`, é idêntico ao endereço físico do nó. É chamado por `recvfromAck()` para testar se uma mensagem do tipo descoberta de rota é para este nó (`RH_MESH_MESSAGE_TYPE_ROUTE_DISCOVERY_REQUEST`).

`/address` - Endereço do endereço físico que está a ser testado;

`/addresslen` - Tamanho do endereço em bytes.

`/retorno` - *true* se o endereço físico do nó é idêntico ao endereço teste.

## Capítulo 4

# Desenvolvimento

Ao longo deste capítulo será demonstrado a montagem do módulo assim como os fluxogramas que serviram de base para a construção da rede. Num primeiro ponto será abordado a montagem geral do sistema e posteriormente uma abordagem específica aos componentes das estações. Dentro de cada subcapítulo serão explicados todos os aspectos técnicos utilizados bem como os procedimentos e testes realizados no processo.

### 4.1 Contextualização

No capítulo 3 foi definido um diagrama geral (Figura 3.1) para entender a estrutura do sistema e a maneira como cada estação encontra-se interligada. De acordo com o diagrama, no sistema existem três estações com a capacidade de enviar e receber informações entre si, através dos módulos rádio.

Esta interdependência beneficia os nós da rede, que em caso de desconexão podem suportar-se dos restantes elementos, para voltarem a comunicar. O encaminhamento das mensagens será gerido pelas bibliotecas com funções específicas para o efeito. As mensagens de estações secundárias seguem diretamente para o destinatário, dentro da sua cobertura, e fora dela através de nós intermediários. Assim, as mensagens circulam consecutivamente para um destinatário (estação central) que as possa remeter para a *cloud*.

## 4.2 Hardware

Tendo em conta o diagrama geral, a estação central e as estações secundárias são constituídas pelos mesmos componentes e dispositivos, logo as ligações serão idênticas. A diferença entre estação central e secundária deve-se ao tipo de funções que executam e não ao material que as constitui.

A estrutura aponta para o uso de dois NodeMCU, porque a necessidade de aumentar o número de sensores, impede a coexistência de um único aparelho. Deste modo, para o conjunto de sensores requeridos, é inevitável a utilização de outro NodeMCU em conjunto. Assim, é possível separar os dois circuitos, com uma unidade a trabalhar para os sensores, e outra a tratar da informação para o módulo rádio.

Na Figura 4.1 encontra-se apresentada a estrutura de uma estação, separada por quatro ligações entre ambas as unidades. Para visualizar estas ligações com mais detalhe, é apresentada na Figura 4.2, uma visão mais aproximada desta zona. A ligação número 1 estabelece a ligação entre o RESET e o GPIO16.

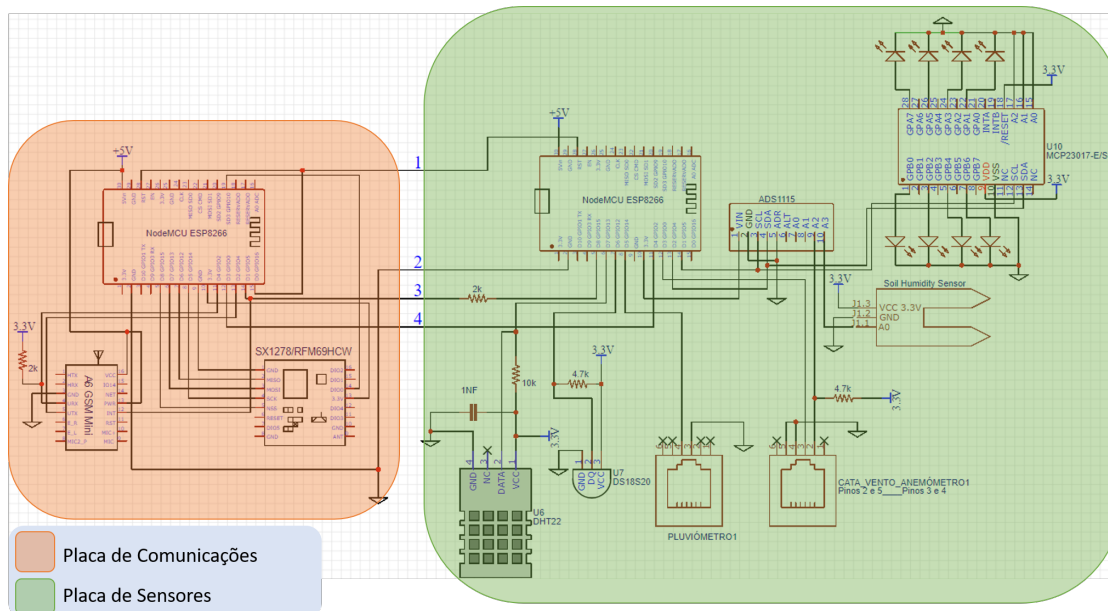


Figura 4.1: Circuito geral de uma estação.

Esta conexão permite despertar o NodeMCU das comunicações, que através da mesma ligação acorda também o NodeMCU dos sensores. Durante o funcionamento, o pino RST está num nível alto (HIGH), no entanto quando recebe um sinal baixo, o NodeMCU é reiniciado. Sempre que o tempo de sono (*sleep timer*) terminar, o GPIO16 envia um sinal de nível baixo (LOW) para reiniciar o aparelho. A segunda ligação corresponde à ligação terra que é comum às duas placas.

A terceira ligação corresponde a uma interrupção externa criada para adormecer o NodeMCU da placa de sensores durante um período específico. Quem comanda o período de sono é o NodeMCU da placa de comunicações, porque o NodeMCU da placa de sensores continua a funcionar, necessitando de um sinal externo para adormecer.

A quarta ligação permite estabelecer uma comunicação bidirecional entre as unidades que estarão perto uma da outra. A comunicação será responsável pela troca de mensagens entre a placa de sensores e a placa de comunicações. Mais concretamente o envio da mensagem da placa de sensores para a placa de comunicações e o envio dos estados da estação, no sentido inverso. Este processo de comunicação está explicado na secção 4.3.2 nas Listagens 4.6 e 4.7.

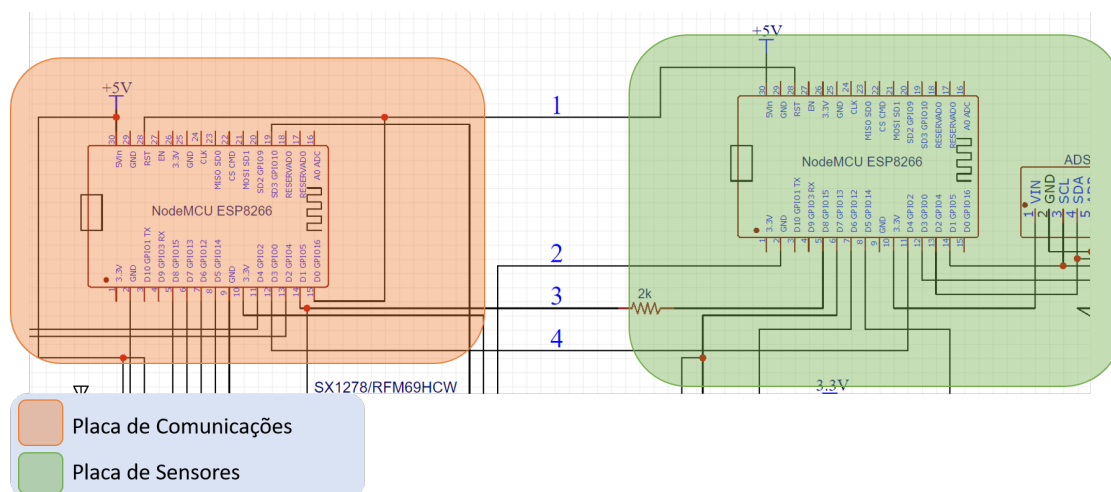


Figura 4.2: Ligações entre placa de comunicações e placa de sensores.

Tendo em conta a quantidade de sensores e dispositivos emparelhados, é necessário

uma fonte externa DC de 5 V e um regulador de tensão com a capacidade para dividir a tensão em 3.3 V. Os pinos GPIO (*General Purpose Input/Output*) permitem uma interface entre os microprocessadores e os periféricos externos, podendo ser definidos como entrada ou saída de sinais digitais. Estes pinos operam em 3.3 V e não possuem tolerância a 5 V, possuindo *pull-up* interno.

Os GPIO6-11 correspondem a pinos que têm funções específicas em processos internos do MCU. Os GPIO1 e 3 são utilizados para a realização do *debugging* através da comunicação UART (Rx0/Tx0), e os GPIOs 4 e 5 podem ser usados como pinos de entrada ou saída após configuração prévia.

Cada GPIO tem um limite máximo de corrente igual a 12 mA, pelo que se for necessário uma corrente maior, é recomendável utilizar um transistor como *driver* de corrente. O mapa do NodeMCU apresenta alguns pinos com limitações, como é o caso do GPIO16, que permite acordar o módulo, os pinos GPIO1/GPIO3 utilizados na comunicação UART e a combinação GPIO0/2/15.

Quando o NodeMCU arranca, os valores dos pinos GPIO0/2/15, são examinados. A combinação do estado destes três pinos condiciona a forma como o dispositivo é programado. De acordo com a Figura 4.3, o arranque pode ser realizado através da UART ou do SPI, pelo que é importante manter estes pinos livres de acesso ou mantendo-os no nível lógico apropriado.

GPIO15	GPIO0	GPIO2	Mode
0V	0V	3.3V	Uart Bootloader
0V	3.3V	3.3V	Boot sketch (SPI flash)
3.3V	x	x	SDIO mode (not used for Arduino)

Figura 4.3: Modos de arranque do NodeMCU.

Relativamente ao GPIO15, este pino funciona como SS (*Slave Select*) no SPI alternativo, que permite ativar um dispositivo sempre que o seu estado esteja a 0.

A comunicação Serial é utilizada para comunicações entre NodeMCU e outros dispositivos (Arduino, Bluetooth). Esta comunicação com o computador é possível através

do conversor serial USB presente no hardware deste aparelhos. A biblioteca padrão do Arduino IDE possui uma classe (`Serial`) que contém funcionalidades para ler dados através da função `read` ou escrever utilizando a função `print`, `println` e `write`. Sempre que é efetuado um pedido, o monitor Serial consiste numa interface gráfica que facilita a comunicação entre o NodeMCU e um computador.

O utilizador necessita da comunicação Serial da interface Arduino para retirar informação sobre o estado de variáveis, processos ou funções. Desta forma o utilizador necessita sempre de um computador para observar essas etapas, conseguindo perceber qual o erro existente na construção do programa. Para visualizar as etapas dos processos sem aceder ao computador, foi criada uma interface de monitorização (ver Figura 4.4) com um expansor de pinos e um conjunto de leds.

Na placa de sensores temos um dispositivo que permite a expansão do número de pinos definidos pelo NodeMCU. Mais conhecido como MCP23017, este aparelho consiste em dois conjuntos de portas de entrada e saída de 16 bits (`PORTA` e `PORTB`). As duas sequências somam 16 entradas/saídas que podem ser utilizadas através da comunicação I2C. Até 8 dispositivos em paralelo podem ser acoplados aos pinos `SDA/SCL` inseridos para maximizar os pinos disponíveis (máx: 128 portas).

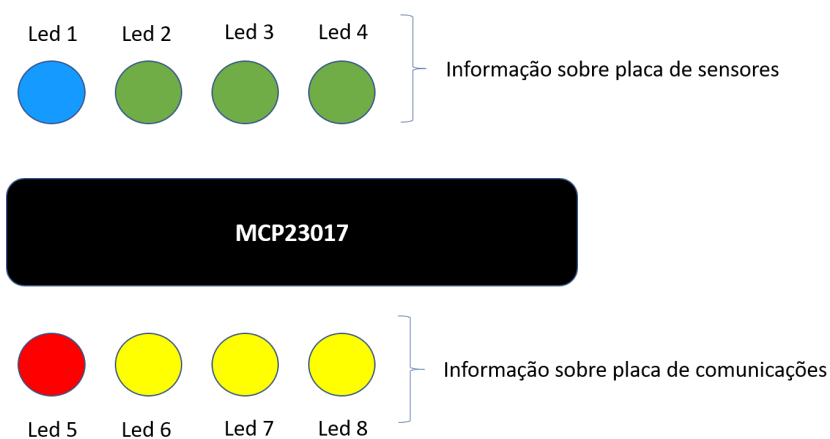


Figura 4.4: Disposição dos leds de informação.

Esta disposição permite que o utilizador consiga verificar o funcionamento de uma

estação sem necessitar de aceder a este monitor Serial. Esta configuração tenta facilitar a visualização, com uma barra de leds para as informações da placa de sensores e outra para a placa de comunicações. Assim para perceber este processo é apresentado na Figura 4.5 as informações relativas a uma e a outra placa.

Estado	Significado	Led 1	Led 2	Led 3	Led 4
0	Ligar à rede WiFi	... 1/0/1 ...	0	0	0
1	Sincronizar relógio com a hora global	1	1	0	0
2	Leitura de dados pelos sensores	1	1	1	0
3	Envio do url para a placa de comunicações	1	1	1	1
4	10 segundos finais antes de adormecer	... 1/0/1 ...	1	1	1
Estado	Significado	Led 5	Led 6	Led 7	Led 8
A	Enviou e recebeu dados	1	1	1	0
B	Enviou mas não recebeu nada	1	1	0	0
C	Não conseguiu enviar nem receber nada	1	0	0	0
D	Mensagem recebida	1	0	0	1
E	Receber mensagens	1	0	0	0
F	10 segundos finais antes de adormecer	... 1/0/1 ...	0	0	0

Figura 4.5: Informação sobre os estados da estação.

Cada estado significa uma sequência diferente, começando com os processos dos estados numéricos (relativo à placa de sensores) e depois com os processos dos estados alfabéticos (relativos a placa de comunicações). O programa da estação começa na placa de sensores, porque existe a criação de uma string com as recolhas de sensores. Depois existe a transmissão de dados a longa distância para a estação principal, com a placa de comunicações.

O estado de um led é caracterizado por estar ligado (ON = 1) ou desligado (OFF = 0). Se existir alternância entre ligado e desligado, o estado muda entre 1 ou 0 como apresentado nos estados 0, 4 e F.

### 4.2.1 Placa de Comunicações

A placa de comunicações permite estabelecer uma ponte entre duas estações. Um dispositivo rádio (RFM69HCW ou SX1278) estará ligado ao ESP8266, através da placa de desenvolvimento NodeMCU ESP12E. A conexão física entre o NodeMCU e o rádio (RFM69HCW ou SX1278) só é possível através da utilização de uma interface SPI.

O hardware NodeMCU fornece dois barramentos SPI, que mapeiam para os pinos geralmente identificados como SPI e HSPI. Como é utilizado um tipo de placa de desenvolvimento que utiliza memória flash, a SPI é tipicamente usada para comunicação com o chip flash.

A alternativa passa por utilizar uma segunda interface SPI existente no NodeMCU, também designada de HSPI (Hardware SPI). De acordo com a folha de fabricante, os pinos HSPI são usados para estas ligações e a alimentação fornecida pela placa, é redirecionada para o módulo rádio. A ligação restringe também o uso do pino DIO0, que servirá como pino de interrupção no código. Na Figura 4.6 está representado este circuito, em que o módulo RFM69HCW ou o SX1278, são módulos idênticos pelo que as conexões físicas serão semelhantes para ambos os casos.

A ligação entre o pino D0 (GPIO16) e o RESET existe para despertar o módulo durante um tempo definido, acordando após esse período de descanso. Quando o microcontrolador acorda, faz automaticamente um RESET e recomeça o ciclo programado para o aparelho.

No módulo A6 mini são utilizados os pinos Rx/Tx para comunicar, sendo invertidos na conexão com o MCU. Isto é, o pino Rx do módulo GSM, liga ao pino Tx do MCU e o pino Tx do módulo ao pino Rx do MCU. Quando este módulo é ligado, o pino PWR precisa de uma tensão superior a 2 V durante cerca de dois segundos. O pino INT funciona para suspender o módulo e encontra-se ligado ao pino 0. Se o nível lógico da porta INT for alto, o aparelho fica ligado, se for baixo entra em modo de suspensão. No seu modo de suspensão, a comunicação UART não pode ser utilizada.



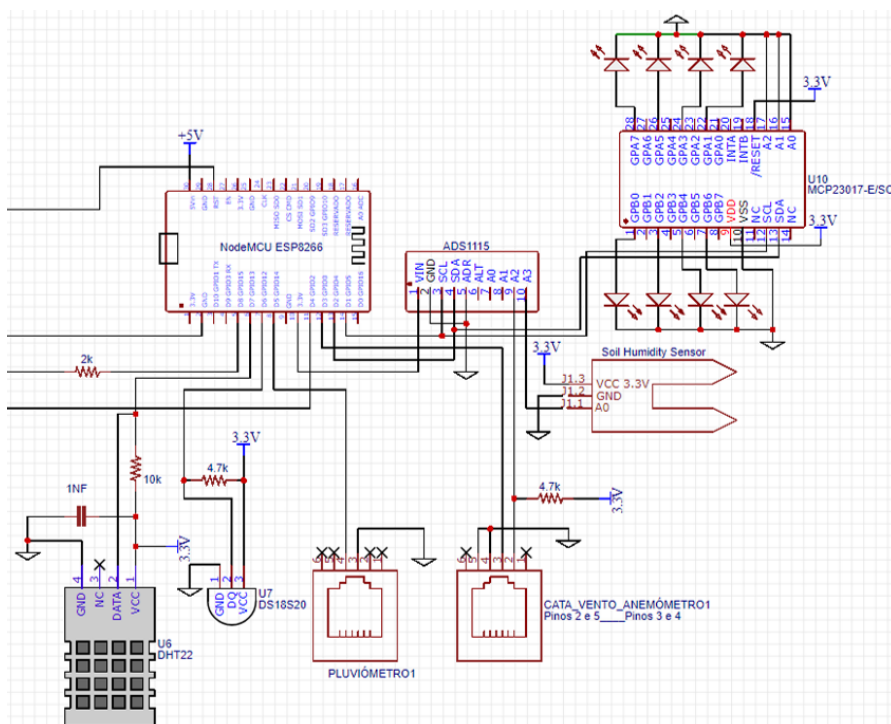


Figura 4.7: Circuito da placa de sensores [21].

De acordo com a estrutura, o cata-vento e o anemómetro partilham o mesmo conector RJ-11, enquanto que o pluviómetro utiliza apenas um para enviar os dados. A tensão de alimentação de 3.3 V depende da inserção de um regulador de tensão externo, ou simplesmente do uso da alimentação regulada pelo microcontrolador. Os detalhes mais específicos sobre a aplicação dos sensores e respetivas ligações estão descritas no documento do autor [21].

### 4.3 Software

Na rede *mesh* integrada, o sistema utiliza duas placas eletrónicas por estação. Os processos que ocorrem dentro da placa de sensores consistem nas leituras analógicas/digitais dos aparelhos e a construção de uma hiperligação com os parâmetros recolhidos. Inicialmente tenta-se estabelecer uma conexão WiFi, e caso esta seja bem sucedida, a placa de sensores faz um pedido HTTP com essa hiperligação. Se o resultado do pedido for posi-

tivo (código 200 - OK), o microcontrolador não necessita de enviar mais, adormecendo durante 'n' segundos. O modo de sono profundo foi introduzido para existir poupança energética.

Tanto a estação central como as secundárias possuem uma placa de comunicações com este software desenvolvido. A partir do fluxograma da Figura 4.8, são descritos de uma forma sequencial as etapas do funcionamento do programa.

1. Inicializa a *driver* e o *manager* do rádio com as configurações iniciais (verificação de ligações, definição da frequência de operação, potência de transmissão, configuração do modo de operação) bem como a configuração de variáveis globais;
2. Verificação da conexão WiFi após um conjunto limitado de tentativas;
3. Se o dispositivo estiver ligado ao WiFi, inicia o processo de escuta de mensagens externas através do módulo rádio;
  - Se existir receção de mensagens, procede-se ao encaminhamento das mensagens para a *cloud*;
  - Se as mensagens forem enviadas com sucesso, adormece durante um tempo limite, caso contrário mostra uma mensagem de erro e esgotado o tempo de escuta, este adormece;
4. Se o dispositivo não conseguir conectar-se ao WiFi, inicia o processo de receção e envio da mensagem interna (hiperligação dos sensores) através do módulo rádio;
  - A receção da mensagem interna envolve uma ligação UART, pelo que é iniciada uma interface SoftwareSerial para enviar os dados, utilizando um cabo;
  - Depois de receber a mensagem, existe o envio da informação por rádio até à estação central;
  - Quando a mensagem é enviada com sucesso, aguarda-se por uma confirmação ACK do recetor, apresentado o conteúdo da informação no monitor;

- Se esta rota estiver definida, existe a representação atualizada da tabela de rotas;
- Caso apareça um erro resultante da falha de envio ou a ausência de resposta, visualiza-se uma mensagem de erro e volta à escuta da mensagem interna.
- Se o tempo limite de escuta esgotar, o aparelho volta a adormecer durante um tempo pré-definido;
- Uma resposta positiva revela que a mensagem foi confirmada e de seguida, envia-se uma mensagem SMS para o número do utilizador;
- O conteúdo da mensagem SMS consiste na hiperligação enviada pelo módulo rádio;
- Depois o aparelho espera pela receção de mensagens externas, entrando em modo de escuta;
- Se existir uma mensagem externa que necessite de ser transmitida para a estação central, há o reencaminhamento para a rede;
- Este processo implica a renovação do mapa de rotas que apresentará rotas diferentes para mensagens que precisam de chegar ao destino (estação central);
- Repetir este ciclo até acabar o tempo limite, entrando assim em modo de sono profundo;
- Sempre que o aparelho acorde, existe um impulso RESET automático ao sistema, levando assim ao recomeço do ciclo;

O software desenvolvido foi instalado na placa de comunicações da estação central e das estações secundárias. A diferença está no conjunto de processos que serão executados pelas estações. Como a estação central necessita de uma conexão constante ao WiFi, os processos de trabalho são diferentes e seguem o padrão estabelecido pelo primeiro, segundo e terceiro pontos da secção 4.3 (lista anterior).

Relativamente à estação secundária, a ligação WiFi não é necessária porque este tipo de estações foram concebidas para transmitir dados com ajuda do módulo rádio. Para isso, o NodeMCU percorre o primeiro, segundo e quarto pontos indicados na secção 4.3.

O fluxograma representado na Figura 4.8 ilustra o processo da rede mesh integrada.

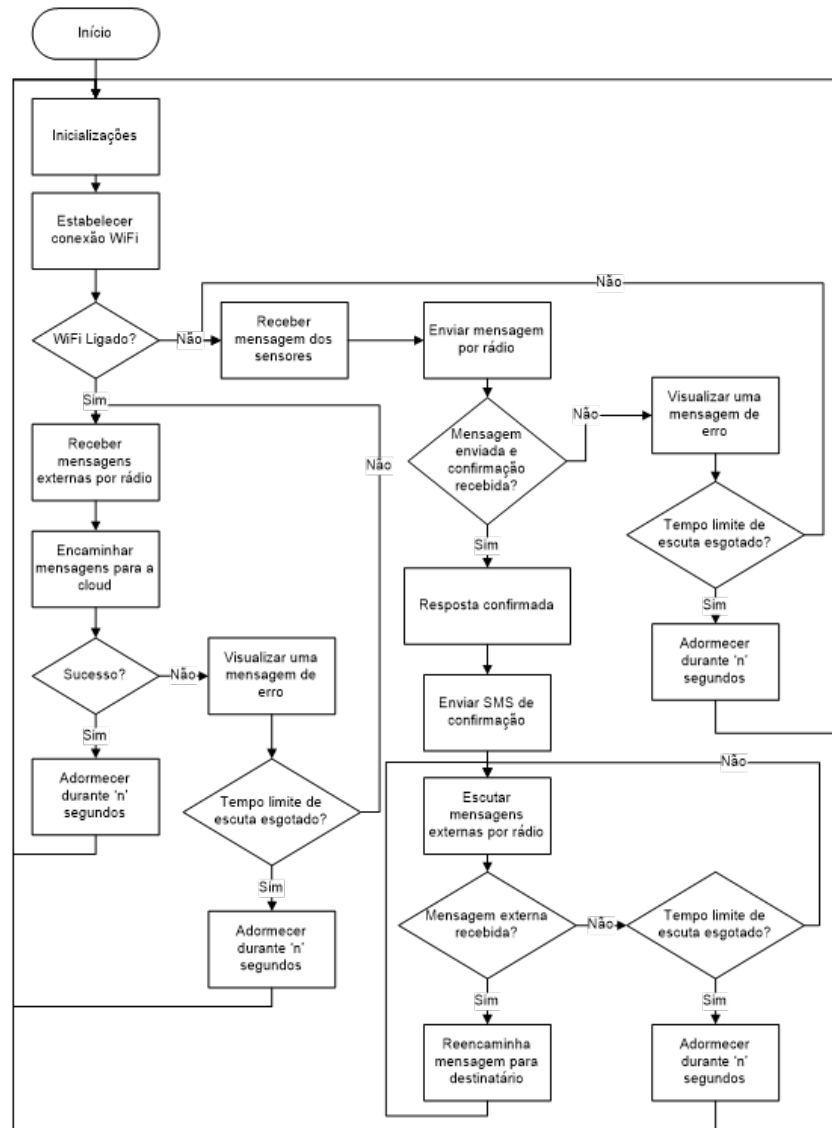


Figura 4.8: Fluxograma referente ao cenário da rede *mesh* integrada.

### 4.3.1 Estação principal

Tendo em conta o fluxograma anterior, as inicializações correspondem à declaração de variáveis e definição de funções que antecedem a realização de processos dentro da função `loop()`. A função `setup()` incorpora algumas dessas funções preparatórias, sendo uma

delas a preparação da *driver* do dispositivo e o *manager* que gere que tipo de bibliotecas podem ser utilizadas pela driver. Na Listagem 4.1 está representado como é realizada essa inicialização, com a utilização do pino 15 correspondendo ao *Slave Select* ou SS e a interrupção DIO0 para o pino 4 do microcontrolador.

Listagem 4.1: Inicializa a *driver* e *manager*.

```
1 (...)  
2 RH_RF95 driver(15,4); //(SS/DIO0)  
3  
4 RMesh manager(driver, MESH1_ADDRESS);  
5 (...)
```

Na Listagem 4.2 está representado na linha 2, a verificação da ligação WiFi, com a utilização de uma variável de estado WL\_CONNECTED (sucesso na conexão). Este estado permite validar a ligação à rede WiFi através do comando WiFi.begin() com as credenciais indicadas no campo *ssid* e *password* (linha 5).

Listagem 4.2: Estabelecer uma conexão WiFi

```
1 (...)  
2 if (WiFi.status() != WL_CONNECTED)  
3     {  
4         WiFi.mode(WIFI_STA);  
5         WiFi.begin(ssid,password);  
6         Serial.println("");  
7         retries = 0;  
8         while ((WiFi.status() != WL_CONNECTED) && (retries < 20))  
9             {  
10                retries++;  
11                delay(500);  
12                Serial.println(".");  
13            } (...)  
14     }
```

O Wifi está ligado na estação principal, logo existe a procura de mensagens externas das estações secundárias. O papel da estação principal é recolher essas mensagens que são propagadas via rádio e posteriormente encaminha-las para a Internet. Na linha 3, a função *recvfromAck* valida se recebeu uma mensagem válida para este módulo (nó da rede). Se uma mensagem for recebida, copia a mensagem para *buf* com um tamanho indicado por *\*len*. O remetente possui um endereço numerado por um valor atribuído no campo *\*from*.

Listagem 4.3: Recolher mensagens externas por rádio

```

1 (...
2  memset(buf, 0, len);
3  if(manager.recvfromAck(buf, &len, &from)) //Rececao de mensagem
4  {
5      Serial.print("got message from : 0x");
6      Serial.print(from, HEX);
7      Serial.print(": ");
8      Serial.println((char*)buf);
9      Serial.println("****RoutingTable****");
10     manager.printRoutingTable();
11     (...) //Processos para sincronizacao da hora global
12     recebe_reply = 1;
13     url_comunica = 'D'; //Estado "Recebeu Mensagem"
14     time_t timenow = now(); //Armazenar a hora atual
15     itoa((int)timenow,timestring,10);
16     strcat((char*)back,timestring);
17     Serial.println((char*)back);
18     //Enviar uma resposta de ACK para o remetente
19     if(manager.sendtoWait(back, sizeof(back), from) != RH_ROUTER_ERROR_NONE)
20     Serial.println("sendtoWait failed");
21     memset(back, 0, sizeof(back));
22     strcpy((char*)back,"Ack_from_node1_time:");
23 } (...

```

Na linha 19, a função *sendtoWait* envia uma mensagem de resposta (*Acknowledge - ACK*) para o nó origem (*from = remetente*), com o conteúdo inserido no campo *back*. Neste ponto existe uma verificação do envio da mensagem e no caso de não existir erro na rota (*RH\_ROUTER\_ERROR\_NONE*), o remetente recebe essa resposta. Deve certificar-se que esta função é chamada com frequência suficiente para não perder nenhuma mensagem. Na linha 10, a função *printRoutingTable* imprime a tabela de rotas atualizada, sempre que chega uma nova mensagem.

Na rede circulam mensagens constituídas pela informação dos sensores das estações secundárias. Uma mensagem é caracterizada por uma hiperligação HTTP, com a informação encapsulada numa *string*.

Na Listagem 4.4 existe uma parte inicial para assegurar que existe conexão ao servidor da empresa SmartHelix. Depois no campo *SHCustom:sh0d9ff02* corresponde a um identificador da placa, ou seja, uma referência gerada aleatoriamente para etiquetar o aparelho (*tag*).

Listagem 4.4: Hiperligação com os dados dos sensores.

```
1 http://smarthelix.cloudapp.net/prototype-web/api/push/SHCustom:sh0d9ff02?  
2 ts=19700101000332000&data=iter,6;t22,18.00;h22,62.50;wind_s,0.00;wind_d,1.99;  
3 rain_1h,0.00;rain_24h,0.00;hum,100.00!
```

No ponto *ts* (início da linha 2) está representado o *timestamp* que corresponde a um conjunto de caracteres com o formato da data e hora para sincronização de relógios. No campo *data=iter,6* indica o número de iterações realizadas no programa, que correspondem a seis ciclos de programa.

Os valores dos sensores caracterizam-se pela sequência seguinte, com a temperatura (*t22*) e a humidade do ar (*h22*); a velocidade (*wind\_s*) e a direção do vento *wind\_d*; a pluviosidade num dia e numa hora (*rain\_1h*, *rain\_24h*); e a humidade do solo (*hum*). Nem todos os sensores foram utilizados na criação desta hiperligação, por isso é que existem variáveis sem valores.

Na Listagem 4.5 é feito o encaminhamento das mensagens para a *cloud*, através do envio das hiperligações com pedidos HTTP. Inicialmente é declarado um objeto da classe *HTTPClient*, que é designado por *http* (linha 2). Esta classe fornece os métodos para criar e enviar o pedido HTTP.

Na linha 4, é chamado o método *begin* no objeto *http*, passando o URL (no campo *\*buf*) que se pretende conectar. A etapa seguinte passa pelo envio do pedido, chamando o método *GET* (linha 6). Esse método retorna o estado da operação, que é importante no tratamento de erros. Se o valor for maior que 0, então é um código HTTP padrão. Se o valor for menor que 0, então é um erro do cliente relacionado com a conexão.

Listagem 4.5: Encaminhar mensagens para a cloud.

```

1 (...
2   HTTPClient http;
3   Serial.print("[HTTP] begin ");
4   http.begin((char*)buf);
5   Serial.print("[HTTP] GET...");
6   int httpCode = http.GET();
7   if(httpCode > 0) {
8       Serial.printf("[HTTP] GET...code: %d \n", httpCode);
9       if(httpCode == HTTP_CODE_OK) {
10          String payload = http.getString();
11          Serial.println(payload);
12      }
13  } else {
14      Serial.printf("[HTTP] GET... failed, error: %s\n", ...
15                    http.errorToString(httpCode).c_str());
16  }
17  if(httpCode == 200){
18      cont=cont+1;
19      Serial.print("Success send: "); Serial.println(cont);
20  }
21  http.end();

```

Na linha 7, se o código for maior que 0, é imprimido o conteúdo da resposta chamando o método *getString* no objeto HTTP. Se o código for igual a 200, o pedido foi bem sucedido, caso contrário retorna o valor de erro através do método *errorToString*. O método *end* é muito importante para fechar a conexão TCP e libertar os recursos.

### 4.3.2 Estação secundária

Na placa de sensores existem vários sensores para a recolha dos dados, que inserem a informação numa mensagem representada na Listagem 4.4. Depois essa mensagem é enviada por comunicação serial (Listagem 4.6) através da pino 2, que servirá como pino TX da ligação. Na linha 1 esse pino é definido com o objeto *serialsender* da classe *SoftwareSerial* (o pino 30 é descartável).

A biblioteca *SoftwareSerial* serve para aproveitar pinos digitais para efetuar comunicações serial. Depois de configurar os pinos, na linha 2 a comunicação é iniciada através do objeto *begin* que especifica qual a velocidade de transmissão pretendida. Na linha 3 existe o envio da mensagem através da função *write* que remete a mensagem para a placa de comunicações.

Listagem 4.6: Enviar mensagem dos sensores

```
1  SoftwareSerial serial_sender(30,2);
2  serial_sender.begin(115200);
3  serial_sender.write(URL);
4  Serial.println(URL);
5  Serial.println("[Dados dos sensores enviados!]);
```

Dentro da placa de comunicações da estação secundária não existe acesso para encaminhar a informação para a Internet, portanto o fluxograma segue um percurso diferente da estação principal. Inicialmente faz os processos de inicializações do microcontrolador e do rádio, tentando estabelecer uma conexão por WiFi. Após esgotar as tentativas para ligar à rede WiFi, este procura receber a hiperligação criada para os sensores (ver Listagem 4.7).

Listagem 4.7: Receber mensagem dos sensores

```
1  SoftwareSerial serialreceiver(0,30);
2  serialreceiver.begin(115200);
3  while(received<0){
4  delay(1);
5  if(serialreceiver.available()){
6      Serial.print("[Data: ");
7      serialreceiver.readBytesUntil('!',url_sensor,200);
8      received = 1;
9  }
```

Para isso utiliza a comunicação *serial* para receber os dados da placa de sensores na placa de comunicações. Esta última atua como ponto RX da comunicação, utilizando apenas um pino (pino 0) para o efeito.

Assim na linha 1 é definido com o objeto *serialreceiver* da classe `SoftwareSerial` (o pino 30 é descartável). Na linha 5 existe uma verificação da disponibilidade da porta série, para que sempre que existir uma nova mensagem, é possível recolhe-la para uma variável. Se existir uma mensagem para leitura na porta série, guarda-a num buffer designado por *url\_sensor*.

Na linha 7, a função *readBytesUntil* recebe a mensagem até terminar o número de caracteres lidos ou até detetar um carácter específico (!). Dentro deste ciclo existe uma *flag* (*received*) para que quando existir uma mensagem lida, o programa deixa de escutar as novas mensagens. Posteriormente esta mensagem é enviada pelo módulo rádio que a remete para o destinatário (estação principal).

Na Listagem 4.8 existe o envio da mensagem por rádio através da função *sendtoWait*, que encaminha a mensagem (campo *\*data*) para o destinatário (endereçado por *adres*). Se não existir erros, envia a mensagem e espera pela resposta, do destinatário inicial. Na linha 7, a função *recvfromAckTimeout* aguarda um tempo limite até receber uma mensagem de ACK válida. Se a mensagem de ACK for recebida, é apresentada a tabela de rotas atualizada e o valor de RSSI (a intensidade do sinal recebido).

Listagem 4.8: Enviar mensagem por rádio

```

1     Serial.print("\n[Send message to mesh ");
2     Serial.println(adres);
3     if(manager.sendtoWait((uint8_t*)data, sizeof(data), adres) == ...
        RH_ROUTER_ERROR_NONE)
4     { //Now wait for a reply from the ultimate router
5         Serial.println("[Sending SUCESS...]");
6         memset(buf, 0, len);
7         if(manager.recvfromAckTimeout(buf, &len, 3000, &from))
8         {
9             Serial.print("[got reply from : 0x");
10            Serial.print(from, HEX);
11            Serial.print(": ");
12            Serial.println((char*)buf);
13            Serial.print("RSSI: ");
14            Serial.println(driver.lastRssi(), DEC);
15            Serial.println("****RoutingTable****");
16            manager.printRoutingTable();
17            Serial.println("[Enviar SMS.....]");
18            SendMessage();
19            url_comunica = 'A';
20        }(...)

```

Estes valores são apresentados através da função *print* que permite a visualização dos dados no monitor Serial, sempre que são chamadas as variáveis. Na linha 6, a função *memset* serve para limpar o *buffer* (buf) através do preenchimento da memória com zeros. Na linha 18 da Listagem 4.8 existe uma função para enviar um SMS de confirmação para o utilizador.

A função *SendMessage* está representada na Listagem 4.9 com a utilização de comandos AT. Inicialmente existe a definição dos pinos que estabelecem a comunicação UART entre o módulo GSM e o NodeMCU. Depois é enviado um comando para definir o modo de texto (AT+CMGF=1) e outro para introduzir o número (AT+CMGS="+35191\*\*\*\*\*"), bem como o conteúdo da mensagem (linha 8). Os delays existentes entre o envio de cada

comando AT servem para dar tempo ao módulo GSM para preparar as configurações e depois enviar a mensagem.

Listagem 4.9: Enviar mensagens SMS de confirmação

```
1  mySerial.println("AT");
2  delay(1000);
3  Serial.println("[Sending SMS]");
4  mySerial.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode
5  delay(1000);
6  mySerial.println("AT+CMGS=\"+351913748599\"\\r"); // Replace with number
7  delay(1000);
8  String mensagem = "Estacao Nr2 Dados";
9  mySerial.println(mensagem); // The SMS text you want to send
10 delay(100);
11 mySerial.println((char)26); // ASCII code of CTRL+Z
12 delay(1000);
```

O tempo ativo definido é igual a 70 segundos e corresponde ao tempo de ciclo do NodeMCU. Se este tempo for superior, a função *deepSleep* permite que o módulo adormeça durante um tempo definido. A sincronização do tempo de relógio é definida com *setTime* nas inicializações.

Na Listagem 4.10 é armazenado o tempo para uma variável *timenow*, para ser inserida na memória RTC. Este processo de escrita é efetuado através da passagem da estrutura *rtctime* para a função *system\_rtc\_mem\_write* (linha 7). Esta memória permite guardar dados, mesmo em períodos longos de descanso.

O tamanho da informação é assegurado no terceiro parâmetro e para guardar os dados no início do segmento *User\_Data*, o endereço será igual a 64 (256/4). Como estão a ser usados dois NodeMCUs numa estação, é importante adormecer com sincronização. Para isso foi gerado uma interrupção na placa de sensores que é ativada, sempre que o nível lógico do pino 5 fica baixo (LOW). Para além de ativar uma interrupção, o estado do pino 5 coloca o módulo GSM em modo de descanso. Na linha 8, a função *sleep* permite

adormecer o módulo rádio SX1278.

Listagem 4.10: Adormecer o módulo

```
1  if(millis() > 70000)
2      {
3          Serial.print(millis()); Serial.println("ms");
4          time_t timenow = now();
5          rtctime.timestamp=(int)timenow;
6          rtctime.flag++;
7          system_rtc_mem_write(64, &rtctime, 4);
8          driver.sleep();
9          digitalWrite(5,LOW);
10         Serial.print("DORMIR AGORA!!"); //DORMIR 1 MINUTO!!!!
11         ESP.deepSleep(60e6); // 60e6 is 60 seconds
12     }
```



# Capítulo 5

## Testes e resultados

De forma a validar a implementação e resultados obtidos no capítulo anterior, foram analisados alguns casos de estudo que comprovam a eficácia do trabalho realizado. Este capítulo engloba testes de implementação do sistema emissor-recetor e respetivas análises, como a rede *mesh*, ligação ponto a ponto e análise do consumo.

### 5.1 Cenários de Teste

Antes de partir para os resultados, foram efetuados alguns testes dentro de cenários específicos. Para isso o sistema final teve que ser alterado para condicionar as variáveis que o autor pretendia analisar. Com isto procedeu-se à alteração do sistema com a implementação de *hardware* e *software* diferente para realizar processos de *debugging* e testes de desempenho. Assim foram executados os seguintes cenários:

- Cenário 1 - Ligação Ponto a Ponto;
- Cenário 2 - Rede Mesh Simples;
- Cenário 3 - Rede Mesh Integrada;
- Cenário 4 - Análise do consumo de corrente;
- Cenário 5 - Análise económica do sistema;

Os cenários 1 e 2 foram criados para realizar testes de desempenho aos módulos rádio. O primeiro cenário é caracterizado por testes de ligação simples, onde existe apenas dois intervenientes, enquanto que no segundo existem três intervenientes.

O cenário 2 existe para testar o encaminhamento das mensagens com três módulos que estão interligados numa rede simples. Esse nome deve-se ao fato de não serem usados todos os componentes de *hardware* e *software* utilizados na rede integrada.

Nos dois primeiros cenários existiram alterações nas montagens dos componentes bem como no código introduzido para realizar esses testes.

O cenário 3 diz respeito ao sistema final que foi explicado no capítulo 4, pelo que nesta secção serão apresentados apenas os resultados. O cenário 4 e 5 representam respetivamente os testes aos consumos de corrente e uma análise económica ao sistema final. No cenário 4 é introduzido um sensor de corrente que permite realizar medições desse consumo ao longo do tempo. No cenário 5 existe uma análise sobre o impacto dos custos do sistema final, em comparação com outras tecnologias de comunicação.

## 5.2 Ligação Ponto a Ponto

O primeiro cenário consiste no caso mais simples que servirá para avaliar o desempenho das ligações. Este estudo irá incidir nos dois módulos rádio que estarão situados em posições distintas. Uma estação é controlada através de uma aplicação web, que permitirá ao utilizador acionar o módulo e visualizar os dados no telemóvel.

### 5.2.1 Hardware

Na Figura 5.1 está representado o diagrama para a ligação ponto a ponto. O caminho livre significa que existe campo de visão entre a estação 1 e a 2. Neste cenário existe a integração de um *smartphone* com acesso à Internet (dados móveis). A estação 1 será móvel, enquanto que a estação 2 estará numa posição estacionária para receber os dados enviados.

As componentes do sistema consistem num NodeMCU e num módulo rádio (SX1278

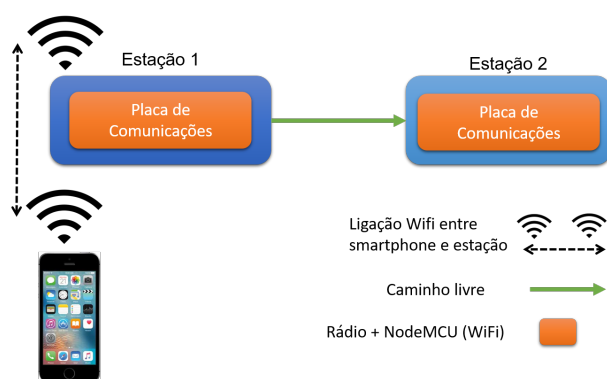


Figura 5.1: Diagrama da ligação ponto a ponto.

ou RFM69HCW). Os testes serão isolados, realizando-se separadamente a análise ao desempenho do módulo RFM69HCW e depois ao SX1278. Os dois modelos são iguais na ligação dos pinos, estabelecendo uma ligação SPI entre o módulo rádio e o NodeMCU. Na Figura 5.2 está representado esta montagem experimental de uma estação que servirá de base para o cenário 1 e para o cenário 2.

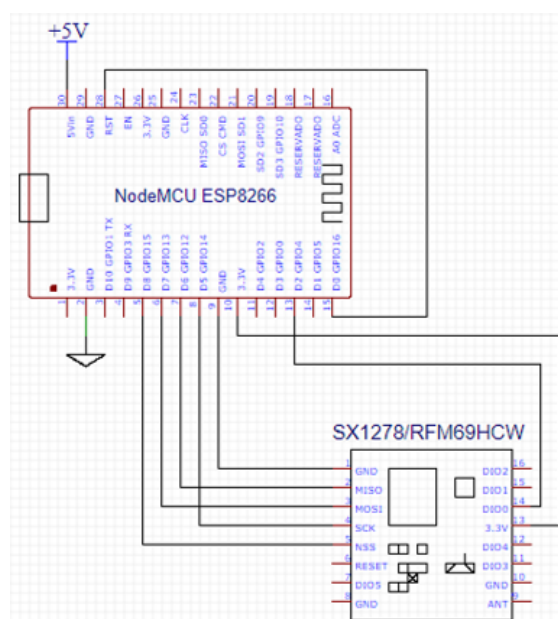


Figura 5.2: Montagem experimental de uma estação para os testes.

### 5.2.2 Software

Na ligação ponto a ponto existem dois dispositivos separados por uma distância variável. Um dispositivo móvel que monitoriza a informação e outro dispositivo fixo onde os dados não são observados em tempo real. Este cenário será utilizado apenas para testes dos módulos.

Para que fosse possível observar esta informação e registar estes dados em tempo real, procedeu-se a elaboração de um fluxograma para estruturar o processo de implementação. O programa encontra-se instalado na placa de comunicações e executa as seguintes funções:

- Inicializa a *driver* do módulo rádio e respetivas configurações iniciais (verificação de ligações, definição da frequência de operação, potência de transmissão, configuração do modo de operação);
- Configura o módulo NodeMCU como *Access Point* e define as credenciais de acesso;
- Criar um servidor HTTP e atualiza a página;
- Verifica qual a opção pretendida para a aplicação, encaminhando a ligação para ativar o rádio através de uma página *web*;
- Se selecionada a opção Enviar, envia uma mensagem para o recetor e espera por uma confirmação da sua receção dentro de um tempo limite;
- Se selecionada a opção Apagar, apaga as informações recolhidas dos últimos pacotes;
- Se selecionada a opção Gravar, mostra uma tabela com os dados recolhidos dos testes, ou seja o estado de cada mensagem, percentagem de ruído e indicador da potência recebida;
- Repetir novamente a execução do programa, atualizando a página;

O fluxograma representado na Figura 5.3 descreve o processo de recolha de dados da ligação ponto a ponto. A aplicação foi criada para o utilizador interagir com o módulo rádio sem a instalação de outros equipamentos (monitores ou *displays*). O método implica a utilização de um telemóvel com acesso à Internet e a aproximação deste à placa, onde o nó está inserido.

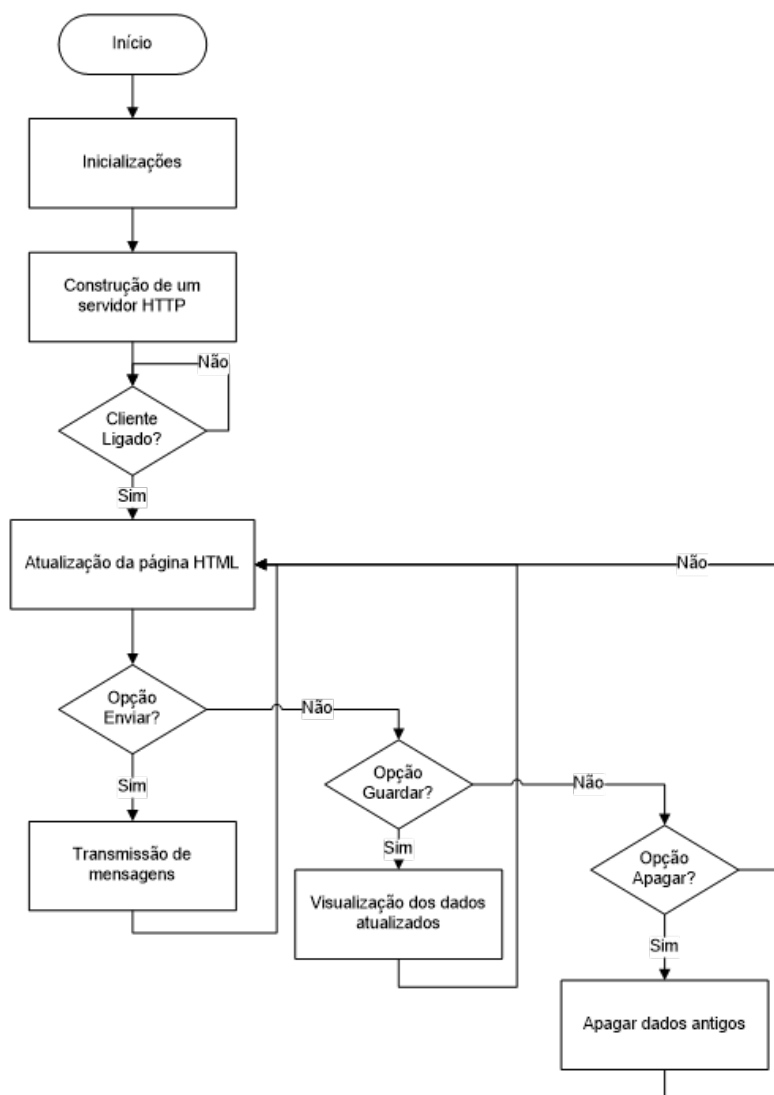


Figura 5.3: Fluxograma referente à recolha de dados na ligação ponto a ponto.

De acordo com o fluxograma, existem dois processos que correspondem às inicia-

lizações e construção de um servidor HTTP, respetivamente. No processo inicial são introduzidas bibliotecas com funções para configurar um servidor e lidar com os pedidos HTTP. Para permitir uma inicialização direta dos objetos, o compilador necessita de uma função para chamar quando a variável é criada, isto é, um construtor (ver Listagem 5.1). Como argumento para o construtor da classe `ESP8266WebServer`, é passado o valor da porta onde o servidor estará à escuta (porta padrão = 80). No construtor da classe `RH_RF95`, os argumentos são os valores dos pinos do NodeMCU. O primeiro é o pino de seleção do escravo (*Slave Select* - SS) e o segundo corresponde ao pino da interrupção que estará conectado à linha de interrupção DIO0.

Listagem 5.1: Construtores das classes `ESP8266WebServer` e `RH_RF95`

```
1  ESP8266WebServer server(80);  
2  
3  RH_RF95 rf95 (15,4);
```

Como é importante configurar o servidor HTTP antes de executá-lo, a maior parte da codificação é realizada na função de configuração (`setup()`). Esta configuração está representada na Listagem 5.2. Na linha 2 do código configura-se a taxa de transferência (bits por segundo) para transmissão serial.

A interface do ponto de acesso softAP é configurada por três parâmetros na função `softAPConfig`. Estes valores são do tipo `IPAddress` e definidos da seguinte forma, respetivamente: `*local_ip` - endereço IP do ponto de acesso virtual; `*gateway` - endereço IP do gateway; `*sub-rede` - máscara de sub-rede.

Depois na linha 8 configura-se um ponto de acesso (softAP) para estabelecer uma rede WiFi com as respetivas credenciais (ssid e password).

Na linha 13 introduz-se o método `on` ao objeto declarado, para especificar o código a executar sempre que for pedido. O método consiste em definir uma função de manipulação (no segundo parâmetro), juntamente com o URL que acionará a execução dessa função. Isso é exemplificado entre a linha 13 até à linha 17.

Listagem 5.2: Função setup e função loop

```
1 void setup() {
2   Serial.begin(115200);
3   Serial.println("Configuring access point...");
4
5   WiFi.mode(WIFI_AP_STA);
6   WiFi.softAPConfig(apIP, apIP, IPAddress(255, 255, 255, 0));
7
8   WiFi.softAP(ssid, password);
9   IPAddress myIP = WiFi.softAPIP();
10  Serial.print("AP IP address: ");
11  Serial.println(myIP);
12
13  server.on ( "/", handleRoot );
14  server.on ( "/envia=1", handleRoot);
15  server.on ( "/apaga=1", handleRoot);
16  server.on ( "/grava=1", handleRoot);
17  server.on ( "/inline", []() {
18    server.send ( 200, "text/plain", "this works as well" );
19  } );
20  server.onNotFound ( handleNotFound );
21  server.begin();
22  Serial.println("HTTP server started");
23
24  while (!Serial) ;
25  if (!rf95.init())
26    Serial.println("init failed");
27 }
28 void loop() {
29   server.handleClient();
30 }
```

Na linha 14, a função *handleRoot* é ativada sempre que a string `"/envia=1"` é adicionada ao URL do servidor. Quando não há ligação Wi-Fi, é executada a função *handleNotFound*, dentro do método *onNotFound* listado na linha 20.

Para definir uma resposta ao pedido, é utilizado na linha 18, o método *send* do objeto *server*. Neste caso é enviado o código 200, que corresponde à resposta “OK” com o tipo de conteúdo como "text / plain". O conteúdo corresponde a uma simples mensagem que corresponde ao terceiro parâmetro do método ("this works as well"). Depois de terminar as pré-configurações, o servidor é criado no método *begin* (linha 21). Na linha 25, o módulo rádio arranca se o retorno da inicialização der positivo, caso contrário termina com uma mensagem de erro. Finalmente, para atender à entrada de pedidos HTTP, é preciso chamar o método *handleClient* do objeto *server*, na função principal (linha 29).

Na Listagem 5.3 está representada a transmissão de uma mensagem rádio, quando existe a confirmação selecionada na página HTML. Nesta página existem três hiperligações definidas nas linhas 2, 3 e 4. Para efetuar a criação de uma hiperligação é necessário utilizar o atributo *href*, que especifica o URL da página para onde é encaminhado. A função *strcpy* copia essa string para uma variável do mesmo tipo, para posteriormente ser introduzida na página.

Listagem 5.3: Envio da mensagem por transmissão rádio - Parte 1

```
1  (...)
2  strcpy(ledText1, "<a href='/?envia=1\''>Enviar</a>");
3  strcpy(ledText2, "<a href='/?apaga=1\''>Apagar</a>");
4  strcpy(ledText3, "<a href='/?grava=1\''>Gravar</a>");
5  if(server.arg("envia").toInt()==1)
6  {
7      Serial.println("Sending to rf95_server");
8      uint8_t data[] = "Hello World!";
9      rf95.send(data, sizeof(data));
10     rf95.waitPacketSent();
11     uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
12     uint8_t len = sizeof(buf);
13     (**)
```

Se for selecionada a opção Enviar (linha 5), existe uma verificação do sucesso dessa

operação. Para isso é utilizado o método *arg*, que recebe como argumento de entrada uma string contendo o nome do parâmetro em consulta. Se não for encontrado, a função retornará uma string vazia. Se encontrado, a função retornará o valor do parâmetro de consulta. A função *toInt* transforma essa string resultante num valor inteiro, que condicionará o ciclo da linha 5. Na linha 9, o utilizador transmite uma mensagem (*data*) com o método *send* do objeto *rf95*, aguardando que o pacote seja enviado pelo método *waitPacketSent*.

Na Listagem 5.4 existe a receção do ACK, que resulta do envio inicial da mensagem para o destinatário. A parte 2 corresponde à continuação do código da Listagem 5.3.

Na linha 2 (*waitAvailableTimeout*) entra-se em modo de receção e bloqueia até que uma mensagem recebida esteja disponível ou dentro do tempo limite (3 segundos). Na linha 4, se a mensagem de resposta (ACK) chegar, o método *recv* recebe esse conteúdo passando-o para a variável *buf*. O intensidade do sinal recebido (RSSI) é recolhida pelo método *lastRssi*, que retorna o último valor de RSSI, bem como o nível de ruído (SNR) com o método *lastSNR* (linhas 12 e 13).

Listagem 5.4: Envio da mensagem por transmissão rádio - Parte 2

```

1      (**)
2      if (rf95.waitAvailableTimeout(3000))
3      { txGood++;
4        if (rf95.recv(buf, &len))
5        { Serial.print("RSSI: ");
6          Serial.println(rf95.lastRssi(), DEC);
7          rxGood++;
8          Serial.print("p:"); Serial.println(p);
9          if(p < qtd)
10         { r++;
11           vetor_rssi[0]=r; vetor_snr[0]=r;
12           vetor_rssi[p] = rf95.lastRssi();
13           vetor_snr[p] = rf95.lastSNR();
14         } p++; (...)} (...)}

```

Na Listagem 5.5 existe a construção da página HTML com a utilização da função *sprintf*. Esta função compõe uma string com o mesmo texto que seria apresentado se o conteúdo fosse usado em *printf*. O conteúdo apresentado no terceiro parâmetro é armazenado na string *html* com capacidade até 2000 caracteres.

Listagem 5.5: Construção da página HTML

```

1  sprintf ( html, 2000,
2  "<html>\ <head>\
3      <meta http-equiv='refresh' content='1'/'>\
4      <title>WifiAP</title>\<style>\
5      body { background-color: #e5e5e5; font-size: 40px;}\
6      h1 { Color: #7b68ee; }\ p,a { font-size: 40px } \
7      table { font-size: 40px;text-align:center; }\</style>\ </head>\ <body>\
8      <h1>Radio Wifi Acess Point</h1>\
9      <p>Uptime: %02d:%02d:%02d</p>\
10     <p>Rssi: %d dBm</p>\
11     <p>SNR: %d dB</p>\
12     <p>TX Good: %d RX Good: %d </p>\
13     <p>PDR: %d % </p>\
14     <p>%s %s %s</p>\
15     <p><table> %s </table></p>\
16     <p><a href=\"javascript:window.location.reload();\">!Refresh!</a> </p>\
17     </body>\
18 </html>",
19     hr, min % 60, sec % 60, rf95.lastRssi(), rf95.lastSNR(), ...
20     txGood,rxGood, pdr, ledText1,ledText2,ledText3, table10);
server.send ( 200, "text/html", html ); (...)
```

Na linha 19, as variáveis *ledText1*, 2 e 3 correspondem às hiperligações; e *txGood* e *rxGood* ao número de pacotes transmitidos e recebidos com sucesso, respetivamente. A variável *pdr* corresponde à percentagem da entrega de mensagens e a *table10* corresponde à tabela criada para a recolha de dados dos testes.

Quando é selecionada a opção Gravar, é apresentada uma tabela com os dados dos

testes recolhidos para análise. Na Listagem 5.6 é possível observar a construção da tabela de dados. A string representada na linha 5 corresponde ao código HTML que será incorporado no código fonte da página (ver Listagem 5.5).

Listagem 5.6: Construção da tabela de recolha de dados - Parte 1

```

1  (...)
2  if(server.arg("grava").toInt()==1)
3  {
4      Serial.print("n:"); Serial.print(n);
5      String table = "<tr><th>PDR</th><th>minRSSI</th>
6      <th>maxRSSI</th><th>minSNR</th><th>maxSNR</th></tr>";
7      if (n < 5)
8      {
9          for(int k = 0; k < n; k++)
10         {
11             table += "<tr><td>";
12             table += leitura[k].pdrval;
13             table += "</td><td>";
14             table += leitura[k].minrssival;
15             table += "</td><td>";
16             table += leitura[k].maxrssival;
17             table += "</td><td>";
18             table += leitura[k].minsnrval;
19             table += "</td><td>";
20             table += leitura[k].maxsnrval;
21             table += "</td></tr>";
22             strncpy(table10, table.c_str(), sizeof(table10));
23             table10[sizeof(table10) - 1] = 0;
24         } } (**)

```

Os testes consistem em ciclos de 11 mensagens transmitidas. Estas mensagens apresentam valores de RSSI e SNR consoante a ligação estabelecida. Num ciclo é analisado o número de mensagens confirmadas pelo destinatário, e recolhido o valor mínimo e má-

ximo de RSSI e SNR. Assim é possível obter um intervalo de valores da intensidade do sinal e o nível de ruído existente para cada ligação. Desta forma um ciclo irá corresponder a uma linha da tabela de dados. Não há limite para o número de ciclos que se podem realizar, no entanto existe a necessidade de impedir que a página fique demasiado grande. Isto é, consiga imprimir caracteres até à sua capacidade máxima.

Na linha 7 da Listagem 5.6 se o número de linhas for inferior a 5, existe a introdução de novas linhas na tabela, caso contrário as linhas de teste antigas são removidas e incluídas novas. Na Listagem 5.7 verifica-se este processo com a introdução de novas linhas, sempre que o índice da tabela ( $n$ ) for superior a quatro (linha 1). A utilização de ciclos *for* permite que a tabela seja construída linha a linha, com uma linha a representar um ciclo de teste. As colunas correspondem aos valores do teste: média de PDR (`pdrval`), valor mínimo de RSSI (`minrssival`), valor máximo de RSSI (`maxrssival`), valor mínimo de SNR (`minsnrval`), valor máximo de SNR (`maxsnrval`).

Listagem 5.7: Construção da tabela de recolha de dados - Parte 2

```
1  (**) else{
2      for(int k = (n-4); k < n; k++)
3      {
4          table += "<tr><td>";
5          table += leitura[k].pdrval;
6          table += "</td><td>";
7          table += leitura[k].minrssival;
8          table += "</td><td>";
9          table += leitura[k].maxrssival;
10         table += "</td><td>";
11         table += leitura[k].minsnrval;
12         table += "</td><td>";
13         table += leitura[k].maxsnrval;
14         table += "</td></tr>";
15         strncpy(table10, table.c_str(), sizeof(table10));
16         table10[sizeof(table10) - 1] = 0; } } (...)
```

### 5.2.3 Simulação e previsão

Neste caso prático foi utilizado um conjunto de pontos para avaliar o desempenho da ligação com um nó fixo (designado de Recetor) e um nó móvel. Cada ponto é atribuído com uma pequena designação para facilitar a leitura pelo utilizador. Como é possível visualizar na Figura 5.4, a representação destes pontos e respetivas ligações, marcadas num mapa que cobre a região de testes. A zona geográfica pertence à cidade de Vale de Cambra, uma localidade constituída por uma área extensa de baixa altitude com montanhas e colinas cercadas à sua volta.

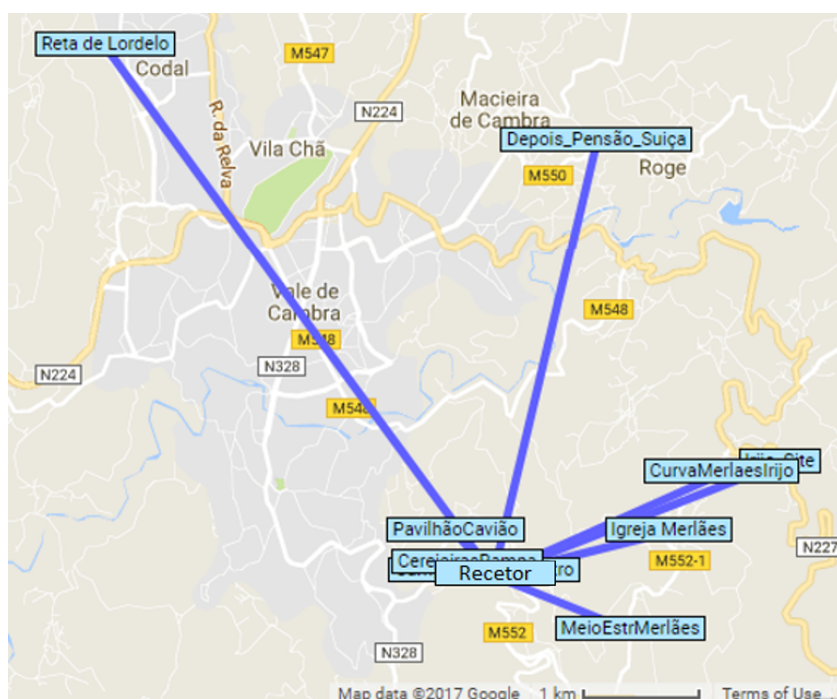


Figura 5.4: Mapa das ligações ponto a ponto testadas em ambiente real.

A zona representada corresponde a uma área de cerca de 28 quilómetros quadrados, estando o ponto fixo (recetor) a 511.3 metros acima do nível do mar. A área de testes encontra-se inserida geograficamente num vale, pelo que os nós móveis são pontos das encostas, em linha de visão com o nó fixo. Este mapa de ligações corresponde à bateria de testes que foi realizada para ambos os módulos (SX1278 e RFM69HCW). A imple-

mentação e configuração foi estabelecida no capítulo 4, portanto para ajustar valores, serão avaliadas as ligações no campo da simulação e depois retirar resultados a partir dos testes experimentais.

Para esta simulação existem poucas perdas na linha e ganhos não muito significativos nos pontos de ligação. A posição do recetor (RX) está a 7.5 metros acima do nível do solo e o transmissor a 1.5 metros, comportando-se como ponto móvel. Na Figura 5.5 encontra-se um diagrama com a estrutura para uma ligação exterior e uma ligação interior. Para as ligações interiores foram considerados os pisos de uma habitação, enquanto que as exteriores correspondem às ligações definidas no mapa da Figura 5.4.

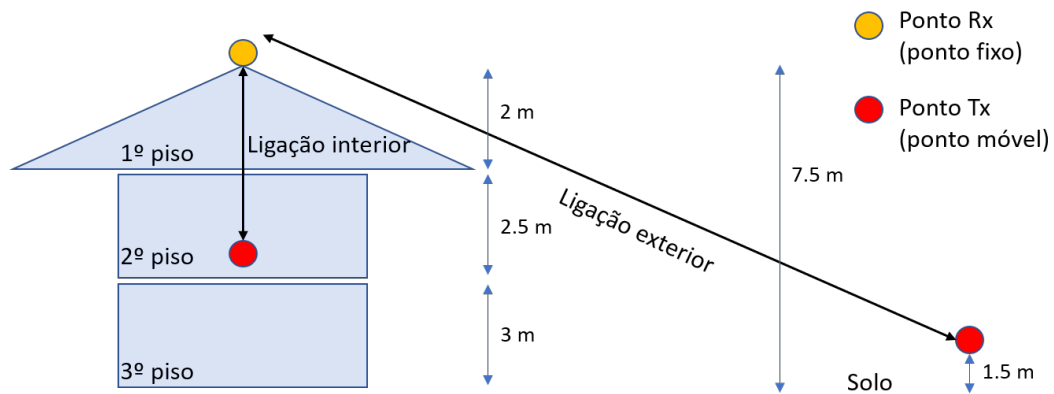


Figura 5.5: Diagrama de uma ligação exterior e interior.

A sensibilidade do módulo rádio corresponde ao limite máximo de RSSI que este ponto poderia receber. O cálculo do RSSI passa pelos seguintes passos: Aplicar a fórmula da atenuação no espaço livre para as ligações exteriores (ver equação 2.3) ou a fórmula de atenuação para os espaços interiores (ver equação 2.4); Depois utilizar a equação de *Link Budget* (ver equação 2.6) com os parâmetros definidos para a simulação. O valor da potência recebida corresponde ao valor de RSSI, pois é o indicador da potência recebida.

Portanto para um andar com distância de 2 metros entre os pontos de comunicação, considerando as variáveis da Tabela 5.1 da equação 2.4, a atenuação interna é 39.16 dB (Lin). Depois ao considerar as variáveis da Tabela 5.1 e utilizando a equação 2.6, a potência recebida será de -31.82 dBm.

Tabela 5.1: Descrição das variáveis de espaço interior.

Var_ equação 2.4		Var_ equação 2.6	
<b>f</b>	433 MHz	<b>P_T</b>	11.14 dBm
<b>N</b>	33	<b>G_T</b>	2 dBi
<b>d</b>	2 m	<b>G_R</b>	2 dBi
<b>Pf(n)</b>	4.5	<b>Lin</b>	39.16 dB
		<b>L_T</b>	3 dB
		<b>L_R</b>	0.5 dB
		<b>L_M</b>	4.3 dB

Para uma ligação exterior, com 240 metros e 5.61 dB de outras perdas, a única diferença é a utilização da fórmula da atenuação de espaço livre (ver equação 2.3): (Nesta equação, a distância está em quilómetros e não em metros!).

Considerando as variáveis da Tabela 5.2 e recorrendo à equação 2.3, a atenuação é 72.78 dB. Depois considerando as variáveis da Tabela 5.2, utilizando a equação 2.6, a potência recebida será de -66.75 dBm.

Tabela 5.2: Descrição das variáveis no espaço livre.

Var_ equação 2.3		Var_ equação 2.6	
<b>f</b>	433 MHz	<b>P_T</b>	11.14 dBm
<b>d</b>	0.24 km	<b>G_T</b>	2 dBi
		<b>G_R</b>	2 dBi
		<b>Lin</b>	72.78 dB
		<b>L_T</b>	3 dB
		<b>L_R</b>	0.5 dB
		<b>L_M</b>	5.61 dB

Na Tabela 5.3 encontram-se os resultados da simulação para os pontos móveis. O valor simulado de RSSI é uma indicação da força do sinal que se espera obter naquele ponto específico.

Os resultados apresentados levam em conta a atenuação causada por obstruções (edifícios, florestas, etc) e outras perdas derivadas por um modelo estatístico. O somatório destas perdas influenciará a força do sinal na receção, levando a uma diminuição da margem de desfasamento.

Este valor indica a quantidade que se pode reduzir do sinal recebido, sem interferir

Tabela 5.3: Descrição dos pontos móveis e resultados da simulação.

	Descrição	Distância (metros)	RSSI Simu (dBm)	Fade Margin (dB)
<b>A</b>	Ponto Inicial	1	-19,87	93,15
<b>B</b>	Campo Fundo	77	-55,05	57,97
<b>C</b>	Cerejeiras Rampa	240	-66,75	46,27
<b>D</b>	Pavilhão Cavião	498	-79,4	33,62
<b>E</b>	Meio Estrada Merlães	1280	-80,23	32,79
<b>F</b>	Igreja Merlães	1567	-83,56	29,46
<b>G</b>	Curva Merlães Irijó	2166	-105,12	7,9
<b>H</b>	Irijo Site	2664	-89,71	23,31
<b>I</b>	Depois Pensão Suíça	3841	-92,09	20,93
<b>J</b>	Reta Lordelo	5657	-96,49	16,53
	<b>Interiores</b>	(*distância vertical)		
<b>A1</b>	Piso Um (2 m)	2	-31,82	81,20
<b>A2</b>	Piso Dois (2.5 m)	4.5	-48,45	64,57
<b>A3</b>	Piso Três (3 m)	7.5	-58,27	54,75

com a comunicação. Se esta tolerância for muito reduzida, o sinal pode perder-se facilmente se existir uma atenuação muito forte pelo caminho. Como é possível verificar, o ponto G apresenta uma margem reduzida em comparação com o ponto inicial (A), estando teoricamente mais suscetível a perdas de sinal do que o inicial.

Os pontos interiores correspondem aos pisos da habitação, que apresentam uma altura distinta entre si, e comportam-se como meios de obstrução para avaliação da penetração do sinal.

Como se pode observar pela Figura 5.6, a zona de testes abrange o vale inteiro, com o verde a corresponder a regiões com sinal forte (a partir de 10 microVolt), e regiões a amarelo com o sinal mais fraco (de 1 microVolt até 9.99 microVolt). As regiões sem nenhuma marcação são zonas onde o sinal fica muito fraco, logo marca-se como regiões sem sinal. Tipicamente zonas em linha com o campo de visão, correspondem a regiões a verde, e regiões com algumas obstruções a amarelo.

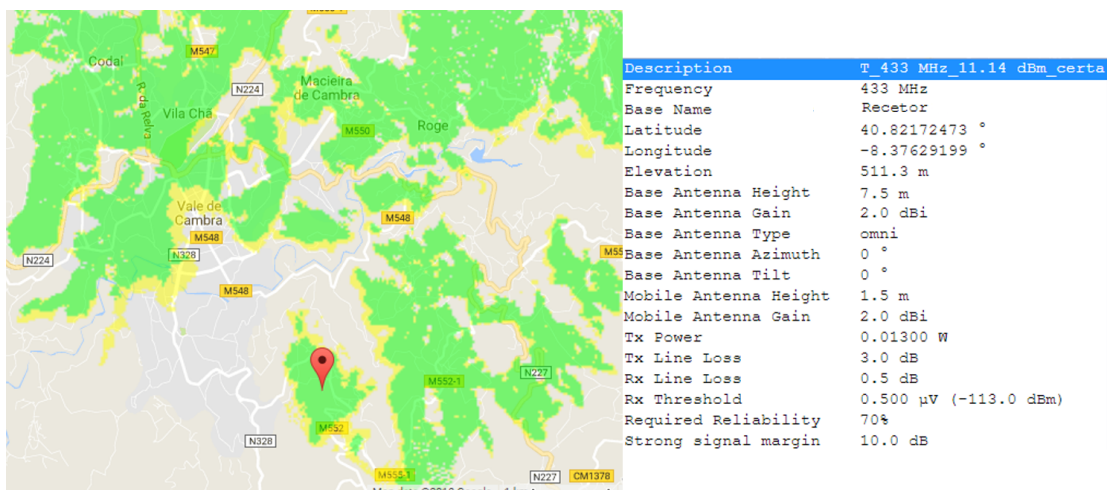


Figura 5.6: Zona de testes para uma potência de transmissão igual a 11 dBm.

Esta imagem vista de cima do mapa, pode ser aplicada num mapa do Google Earth, através de um ficheiro kmz. O formato do ficheiro é retirado do software *Radio Mobile Online* no menu *My Coverages*. Este ficheiro permite visualizar dados geográficos através de elementos como polígonos, sobreposições do solo ou marcadores. Deste modo, é possível verificar que na Figura 5.7, as encostas do vale são zonas válidas, porque estamos em linha de visão direta com o ponto recetor.

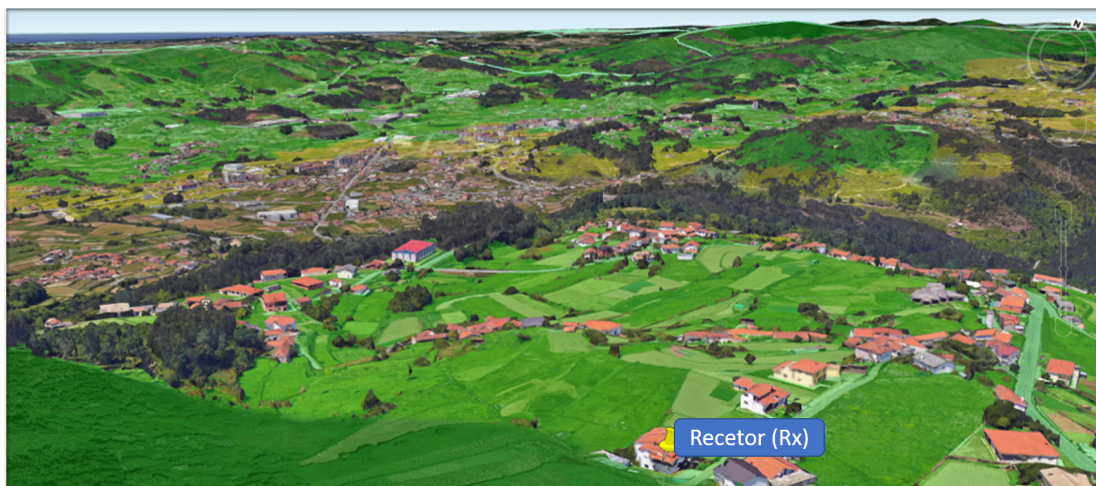


Figura 5.7: Zona de testes visualizada em três dimensões.

### 5.2.4 Resultados

Tendo isto em conta, foram realizados quatro tipos de testes para as ligações estabelecidas, variando o tipo de configuração e/ou módulo por teste. Em cada ligação, o utilizador procedia ao envio de mensagens e receção das confirmações.

Para validar os dados, foram realizadas cerca de 50 transmissões por ligação. A ideia consistiu em separar essas transmissões em 4 ciclos separados, com 11 pacotes por ciclo, ou seja um total de 44 pacotes enviados por ponto.

A estrutura dos testes permitia facilitar o trabalho ao utilizador, pois permitia realizar pequenos ajustes no terreno, bem como conseguir medições mais rápidas e eficientes. As medições totais resultaram numa média dos valores mínimos e máximos de RSSI, e da média em percentagem da taxa de entrega de pacotes (PDR).

As características físicas do terreno foram os fatores mais limitantes para o sucesso dos testes, pois as obstruções impedem que o sinal consiga ser detetado.

Na Tabela 5.4 encontram-se os resultados das ligações ponto a ponto para o módulo RFM69HCW. Nos testes foram utilizadas duas antenas monopolo de fio com um comprimento igual a 17.3 cm (uma por aparelho). Os parâmetros de configuração dos pacotes utilizam a seguinte matriz: Modulação GFSK, *Whitening*, Taxa de Transmissão = 250 kbps, Desvio de Frequência = 250 kHz.

De acordo com os resultados, todos os pontos assinalados tiveram uma taxa de entrega perto dos 100 %, pelo que apenas os pontos G e H tiveram uma taxa de entrega bastante reduzida. Estes pontos estão limitados pelas obstruções das árvores, que enfraquecem o sinal e obrigam à perda de muitos pacotes. A atenuação dos pontos G e H inserida por estes obstáculos revela que o sinal mantém-se tanto ou mais fraco, em comparação com pontos mais distantes e sem obstáculos. Assim, tendo em conta a simulação, estes resultados mantêm-se dentro dos intervalos de RSSI fixados pela análise teórica.

Como os resultados do RFM69HCW estavam dentro do esperado, partiu-se para o segundo teste, que consistiu na utilização do módulo SX1278. Ambos os pontos possuem antenas de monopolo de 17 cm e possuem uma configuração de longo alcance que segue a seguinte matriz: Largura de Faixa (*Bandwidth*) = 125 kHz, *Coding Rate* = 4/8, *Spreading*

Tabela 5.4: Resultados das ligações ponto a ponto para o RFM69HCW.

Ponto	D (m)	minRSSI (dBm)	maxRSSI (dBm)	PDR (%)
<b>A</b>	1	-28	-16	100
<b>B</b>	77	-55	-54	100
<b>C</b>	240	-67	-64	100
<b>D</b>	498	-79	-77	100
<b>E</b>	1280	-76	-75	100
<b>F</b>	1570	-84	-81	100
<b>G</b>	2170	-97	-96	4.5
<b>H</b>	2670	-91	-91	13.5
<b>I</b>	3840	-90	-84	100
<b>J</b>	5660	-90	-84	100
Andar	D (m)	minRSSI (dBm)	maxRSSI (dBm)	PDR (%)
<b>A1</b>	2	-41	-39	100
<b>A2</b>	4.5	-50	-46	100
<b>A3</b>	7.5	-60	-49	100

$Factor = 4096$ , Potência de Transmissão = 11 dBm. As mensagens foram analisadas para os diferentes pontos e os seus resultados apresentados na Tabela 5.5.

Para este módulo, os resultados apontam para muitos pacotes perdidos, com zonas a não conseguirem transmitir dados. Em comparação com o teste anterior, nota-se uma atenuação acentuada nos pontos acima dos 1500 metros. Isto deveu-se à fraca qualidade da antena e respetiva soldadura, pois o autor inseriu um pingo de solda demasiado grande junto ao pino da antena. A soldadura da antena com a placa tem que ser bem executada, para não deixar pingos de solda em excesso. Qualquer metal ou substância condutora que esteja em contato com a zona da antena, perto dos seus terminais, comporta-se como um fio condutor e transmite ondas rádio. Este fenómeno provoca uma redução no desempenho das comunicações que podem falhar com estas alterações, bem como o manuseamento indevido das antenas.

No ponto D, o utilizador não detetou nenhum sinal, por causa da falta de robustez da antena e da fraca sensibilidade do recetor. A ligação no ponto E possibilitou a transmissão de algumas mensagens, porque a linha de visão é limpa sem obstruções, que permitiu uma

Tabela 5.5: Resultados das ligações ponto a ponto com SX1278 - Configuração Longo Alcance, Antena Monopolo.

Ponto	D (m)	minRSSI (dBm)	maxRSSI (dBm)	minSNR (dB)	maxSNR (dB)	PDR (%)
<b>A</b>	1	-67	-63	8	12	100
<b>B</b>	77	-88	-84	6	11	100
<b>C</b>	240	-104	-99	6	11	100
<b>D</b>	498	*	*	*	*	0
<b>E</b>	1280	-113	-109	4	8	58.5
<b>F</b>	1570	*	*	*	*	0
<b>G</b>	2170	*	*	*	*	0
<b>H</b>	2670	*	*	*	*	0
<b>I</b>	3840	*	*	*	*	0
<b>J</b>	5660	*	*	*	*	0
Andar	D (m)	minRSSI (dBm)	maxRSSI (dBm)	minSNR (dB)	maxSNR (dB)	PDR (%)
<b>A1</b>	2	-81	-75	8	11	100
<b>A2</b>	4.5	-104	-92	8	11	100
<b>A3</b>	7.5	-105	-92	8	11	100

taxa de entrega a rondar os 60 %. Com o aumento da distância, menor é a relação SNR, que indica que o sinal está mais sujeito ao ruído de fundo. Relativamente aos pisos da habitação, os dados revelam que a relação SNR não sofre muitas alterações com intervalos idênticos para mínimos e máximos. Os intervalos de RSSI variam em conformidade com a distância e o nível de penetração, obrigando a uma perda de intensidade para pisos mais afastados ou distâncias maiores.

Na Tabela 5.6 encontram-se os resultados do terceiro teste efetuado para o módulo SX1278. A configuração dos pacotes mantêm-se idêntica ao segundo teste, pelo que a única diferença está na inserção da antena *Rubberducky*.

Esta antena comporta-se como um monopolo, porque o fio condutor passa a estar enrolado como uma mola em formato helicoidal. O formato permite proteger esta hélice numa cápsula de plástico ou borracha. Ambas as antenas têm um ganho de 2 dBi com uma potência de transmissão igual a 11 dBm.

Tendo em conta as mensagens transmitidas, é possível verificar que existem perdas bastante acentuadas nas ligações. A força do sinal vai perdendo intensidade com o au-

Tabela 5.6: Resultados das ligações ponto a ponto com SX1278 (versão Lora Ra-02) - Configuração Longo Alcance, Antena *Rubberducky*.

Ponto	D (m)	minRSSI (dBm)	maxRSSI (dBm)	minSNR (dB)	maxSNR (dB)	PDR (%)
<b>A</b>	1	-68	-65	7	11	100
<b>B</b>	77	-102	-95	7	11	100
<b>C</b>	240	-136	-116	-8	2	100
<b>D</b>	498	-134	-118	-9	2	100
<b>E</b>	1280	-132	-120	-7	2	100
<b>F</b>	1570	-139	-131	-13	-6	90
<b>G</b>	2170	-149	-143	-22	-10	47
<b>H</b>	2670	-146	-144	-20	-18	29
<b>I</b>	3840	-148	-137	-21	-11	100
<b>J</b>	5660	-138	-129	-12	-4	100
Andar	D (m)	minRSSI (dBm)	maxRSSI (dBm)	minSNR (dB)	maxSNR (dB)	PDR (%)
<b>A1</b>	2	-96	-81	8	12	100
<b>A2</b>	4.5	-102	-97	7	11	100
<b>A3</b>	7.5	-103	-95	6	12	100

mento da distância. Como os parâmetros de configuração não estão alterados a existência de intervalos reduzidos de RSSI, deve-se à existência de perdas na linha de transmissão, junto aos conectores das antenas. A interferência do ruído está bem presente, porque os níveis da relação SNR apresentam valores negativos em ambos os limites, chegando a um mínimo de -22 dB.

Para os pontos G e H a entrega ficou abaixo dos 50 %, com mínimos de RSSI a chegar a um extremo de -149 dBm, indicando assim um valor muito baixo para a intensidade do sinal. Apesar da atenuação forte e a interferência evidenciada pelos resultados, a configuração apresentada permitiu atingir um alcance elevado, com a confirmação de todos os pacotes transmitidos para uma distância a rondar os 6 quilômetros.

No último teste foi utilizado o mesmo aparelho (SX1278) mas com uma configuração de médio alcance: Largura de Faixa (*Bandwidth*) = 125 kHz, *Coding Rate* = 4/5, *Spreading Factor* = 128, Potência de Transmissão = 11 dBm. O autor conseguiu atingir os seguintes resultados apresentados na Tabela 5.7.

Os resultados experimentais revelam que a partir dos 1.3 kms não existem entregas

Tabela 5.7: Resultados das ligações ponto a ponto com SX1278 (versão Lora Ra-02) - Configuração Médio Alcance, Antena *Rubberducky*.

Ponto	D (m)	minRSSI (dBm)	maxRSSI (dBm)	minSNR (dB)	maxSNR (dB)	PDR (%)
A	1	-67	-61	9	10	100
B	77	-113	-108	6	9	100
C	240	-120	-115	1	5	100
D	498	-130	-119	-4	-2	100
E	1280	-132	-129	-6	-3	90
F	1570	*	*	*	*	0
G	2170	*	*	*	*	0
H	2670	*	*	*	*	0
I	3840	*	*	*	*	0
J	5660	*	*	*	*	0
Andar	D (m)	minRSSI (dBm)	maxRSSI (dBm)	minSNR (dB)	maxSNR (dB)	PDR (%)
A1	2	-95	-88	9	10	100
A2	4.5	-96	-94	7	10	100
A3	7.5	-107	-95	8	10	100

bem sucedidas, porque todos os pacotes transmitidos não receberam confirmação da sua chegada. Até aos primeiros mil metros, os pacotes recebidos, apontaram para uma deteção de ruído nos pontos D e E. As medições efetuadas dentro da habitação, mostram que os pacotes foram entregues com sucesso, não havendo muita interferência à custa do ruído.

### 5.2.5 Sumário

De acordo com os resultados obtidos é possível fazer uma análise aos testes realizados. Os testes foram ordenados por ordem alfabética, considerando o primeiro teste como teste A, ou seja, resultado\_A, e por aí adiante. A distância entre os pontos permitiu verificar que com configurações diferentes se obtêm resultados diferentes.

Na Figura 5.8 estão representados os resultados dos quatro testes realizados para as variáveis RSSI e SNR. Tendo em conta a relação RSSI/distância, o primeiro teste (teste A) apresenta uma curva com valores menos negativos que a curva apresentada pelo terceiro teste (teste C).

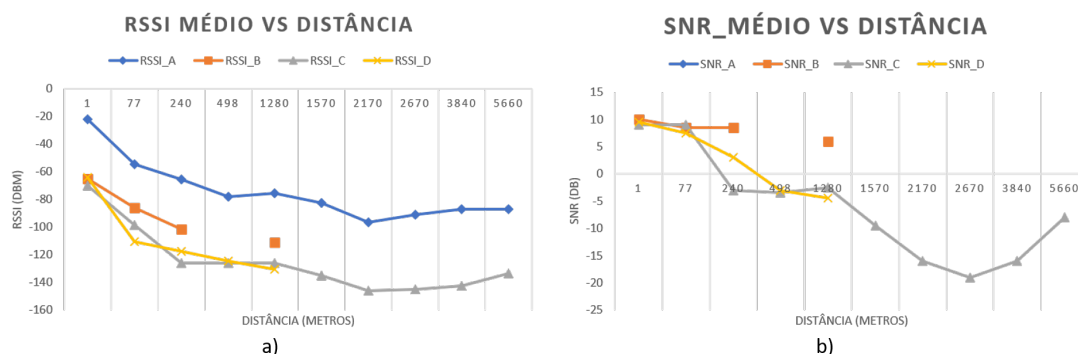


Figura 5.8: Relação da média dos intervalos de: RSSI com a distância a) e SNR com a distância b).

Ainda no gráfico 5.8 a), a curva RSSI\_C revela que o SX1278 apresenta uma comportamento muito similar ao RFM69HCW (RSSI\_A). As curvas RSSI\_B e RSSI\_D representam as falhas existentes nas comunicações até aos 1300 metros. A situação mais peculiar apresentada foi na curva RSSI\_B com o ponto a 498 metros, que foi incapaz de comunicar. Para as curvas laranja (teste B), o mau manuseamento das antenas e a falta de robustez dos módulos estão na origem deste problema, pelo que procurou-se evitar isso com os restantes testes.

No gráfico 5.8 b), o RFM69HCW não foi possível obter dados sobre o SNR, pois este não possui métodos para a recolha dessa variável, impedindo a sua visualização experimental.

A alteração do módulo SX1278 e a sua versão que incorpora uma antena encapsulada, aliada à configuração de longo alcance permitiu atingir resultados positivos. Na curva SNR\_C, as obstruções do ponto a 2670 metros interferiram com o sinal, levando a uma presença de ruído muito notória. Depois o teste D consistiu na validação do módulo SX1278 (utilizado no teste C) com uma configuração de médio alcance. Este teste revelou que não se consegue comunicar a distâncias superiores a 1300 metros, revelando que não existem melhorias no nível de ruído nem no nível de intensidade.

A percentagem de pacotes entregues está representada na Figura 5.9, com quatro curvas características. As curvas PDR\_A e PDR\_C indicam que os respetivos testes foram um sucesso, porque as taxas de entrega mantiveram-se perto dos 100 %, levando

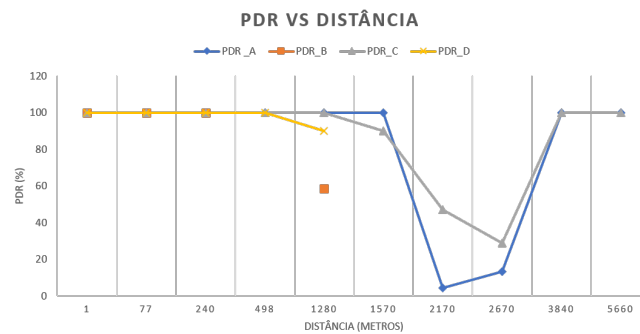


Figura 5.9: Relação entre a taxa de entrega de pacotes (PDR) e a distância.

a que grande parte das mensagens fossem entregues. Os pontos situados entre os 2000 e os 3000 metros foram os mais fustigados, com taxas de entrega abaixo dos 50 %. Os restantes testes foram bloqueados por atenuações existentes junto aos conetores das antenas, bem como outras perdas por obstáculos ou interferências por ruído.

O teste C revelou que o SX1278 (versão LoRa Ra-02) é capaz de comunicar a níveis de intensidade muito inferiores, e com ruído muito acentuado, apresentando um nível de entrega de pacotes semelhante ao do RFM69HCW. Com o SX1278 (versão LoRa Ra-02) é possível analisar três variáveis com uma antena mais resistente a fatores externos. Para além disso, o RFM69HCW não possui uma capacidade para alterar a configuração consoante as necessidades do sistema.



Figura 5.10: Visualização da encosta para a ligação ponto a ponto a 5.7 km.

Contudo se as ligações não apresentarem quaisquer obstáculos dentro do seu campo de visão, é possível comunicar com os módulos rádio a 5.7 km. (ver Figura 5.10).

### 5.3 Rede Mesh Simples

O sistema é constituído por três estações, separadas por uma distância variável para criar uma pequena rede simples. Uma estação possui uma placa de comunicação para simular um nó da rede. Resumindo existem três estações ou três nós que estão interligados entre si. O funcionamento da rede *mesh* consiste no método de entrega de múltiplos saltos. Isto é, as mensagens chegam ao destino, utilizando nós intermediários pelo caminho.

Quando um nó quer enviar uma mensagem, este descobrirá automaticamente uma rota para o nó de destino. Se a rota não estiver disponível, uma nova rota será descoberta. Durante o percurso de uma mensagem, se um dos nós intermediários achar que não pode ser entregue ao próximo salto, este responde ao nó origem com uma mensagem de erro.

A mensagem de erro retornada (no caminho da volta para a origem) permite que os nós possam excluir a rota definida inicialmente, que está indisponível. Isso significa que, se uma rota para um destino se tornar inutilizável (seja porque um nó intermediário estar desligado, ou ficar fora de alcance) uma nova rota será estabelecida na próxima vez que uma mensagem for enviada.

#### 5.3.1 Hardware

As componentes eletrônicas são iguais aos do cenário 1 (subsecção 5.2.1), acrescentando mais uma estação à rede ponto a ponto. Assim em vez de duas estações (dois nós), existem três para testar as ligações. O autor atua sobre o sistema através de uma interface web, utilizando o *smartphone*. Na Figura 5.11 está representado o diagrama para esta rede simples.

Para este cenário, o diagrama apresenta dois caminhos para as mensagens enviadas da estação 1. A cor verde indica que existe campo de visão livre e o caminho não possui obstáculos. Este percurso livre permite que hajam comunicações entre as estações 1 e 2

e as estações 2 e 3. A seta indicada a cor vermelha intermitente indica que o caminho entre a estação 1 e a estação 3 está bloqueado. Isto é, ambos os nós não conseguem estabelecer comunicações.

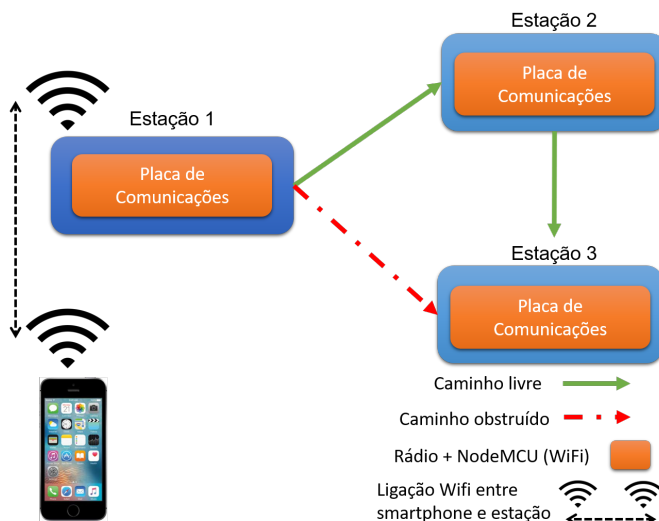


Figura 5.11: Diagrama da rede mesh simples.

### 5.3.2 Software

O método implica a utilização de um telemóvel com acesso à Internet e a aproximação deste à placa, onde o nó está inserido. O NodeMCU cria um ponto de acesso e possibilita o acesso à aplicação através do *smartphone*. Para isso, o *manager* RHMesh foi utilizado, porque a topologia da rede é fluida ou desconhecida, levando que os nós possam entrar e sair do serviço, mesmo sem aviso prévio.

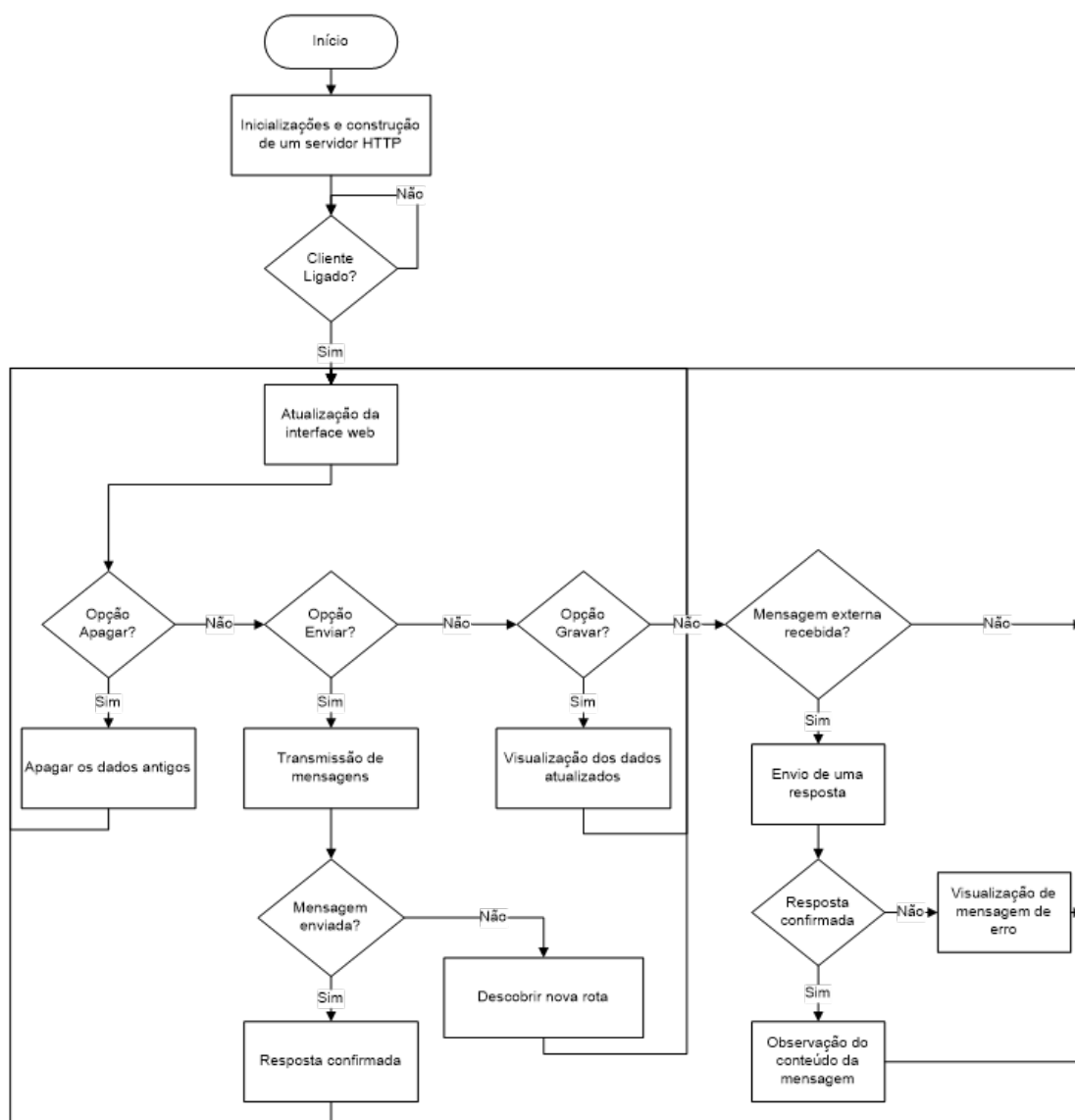
O programa desenvolvido é inserido na estação 1, para realização dos testes, enquanto que na estação 2 e 3 não se aplica a utilização de uma interface web. Nestas duas estações, o módulo rádio fica à escuta de mensagens que possam surgir de outras estações. Caso uma destas estações recebe uma mensagem, é revelado o seu pacote de dados, o endereço do seu remetente e atualizada a tabela de rotas.

Este tipo de topologia apresenta um panorama diferente para as ligações, porque a informação circula entre os nós. A partir da listagem, são descritos de uma forma

sequencial as etapas do funcionamento do programa.

- Inicializa a *driver* e o *manager* do rádio com as configurações iniciais (verificação de ligações, definição da frequência de operação, potência de transmissão, configuração do modo de operação);
- Configuração do módulo como *Access Point* para a criação de um servidor HTTP e atualização da página *web* criada;
- Verifica qual a opção pretendida para a aplicação, encaminhando a ligação para ativar o rádio através de uma página *web*;
- Se selecionada a opção Apagar ou Guardar, esta permite apaga os dados antigos e guardar as informações recolhidas dos testes, respetivamente;
- Se selecionada a opção Enviar, envia uma mensagem para o nó destino e espera por uma confirmação da sua receção dentro de um tempo limite;
- Se a rota for definida existe a representação da tabela atualizada de rotas;
- O módulo estando à escuta, quando recebe outra mensagem responde com uma confirmação;
- Se a rota estiver indefinida ou indisponível, existe a descoberta automática de uma nova rota, com o envio de uma mensagem de *broadcast* para a rede.
- Repetir o ciclo novamente com a atualização da interface.

O fluxograma representado na Figura 5.12 descreve o processo de recolha de dados da rede *mesh* simples. O ciclo encontra-se simplificado com a disposição de caixas de texto retangulares, que significam um processo e losangos que representam decisões.

Figura 5.12: Fluxograma referente à rede *mesh* simples.

No cenário 3 foi realizado a integração dos trabalhos, levando à implementação de uma rede *mesh* para analisar e transmitir dados de estações meteorológicas. O processo de recolha de dados é assegurado com a estação meteorológica (placa de sensores), colocando a rede de longo alcance ao serviço do utilizador.

No processo das inicializações existe a definição das variáveis e respetivos constru-

tores (Listagem 5.8) como no programa anterior (Ligação Ponto a Ponto). A diferença está na introdução do construtor da classe `RHMesh`, que permite a utilização de várias funções, introduzidas na secção 3.4. Este último utiliza o driver definido na linha 3 do código, e um endereço referenciado em `MESH1_ADDRESS`. No construtor da classe `ESP8266WebServer`, utiliza-se o valor da porta padrão 80 e no construtor `RH_RF95` os parâmetros correspondem aos pinos (valores dos pinos do NodeMCU) SS e DIO0, respetivamente.

Listagem 5.8: Construtores das classes `ESP8266WebServer`, `RH_RF95` e `RHMesh`

```
1 ESP8266WebServer server(80);
2
3 RH_RF95 driver(15,4);
4
5 RHMesh manager(driver, MESH1_ADDRESS);
```

A construção do servidor HTTP utiliza o mesmo processo de funcionamento que o utilizado na Listagem 5.2 da secção 5.2.2. Algumas alterações são introduzidas na função `setup`, ao serem acrescentadas mais opções para aceder à função `handleRoot`. Uma nova opção `"/config=1"` serve para o utilizador configurar alguns parâmetros de entrada dentro de um formulário. Outra opção acrescentada `"/form"` serve para a criação de um pequeno formulário, que servirá para recolher o endereço de destino e o tempo de escuta pretendido pelo utilizador.

Após a criação do servidor, este passa a escutar os pedidos HTTP dos clientes. Quando o utilizador conecta-se à rede WiFi, este comporta-se como cliente, existindo um fluxo de pedidos HTTP e respostas do servidor.

Na Listagem 5.9 está representada a transmissão de uma mensagem rádio, quando existe a passagem do parâmetro `"dest"` na hiperligação URL. Este parâmetro aparece no final do caminho do URL e tipicamente é separado por um ponto de interrogação. Cada parâmetro de consulta (*query*) é especificado no formato "NomeParam = ValorParam". Se existirem vários parâmetros que necessitem de ser passados, eles devem ser separados

por um &.

Listagem 5.9: Envio da mensagem por transmissão rádio - Rede Mesh Simples.

```

1  (...)
2  if(server.argName(0) == "dest")
3  {
4      addr = server.arg(0).toInt();
5      manager.setTimeout(server.arg(1).toInt());
6      Serial.print("Send message to mesh ");
7      Serial.println(addr);
8      Serial.print("Retrans_Timeout:");
9      Serial.println(server.arg(1).toInt());
10
11     if(manager.sendtoWait(data, sizeof(data), addr) == ...
12         RH_ROUTER_ERROR_NONE)
13     {
14         txGood++;
15         // It has been reliably delivered to the next node.
16         // Now wait for a reply from the ultimate router
17         memset(buf, 0, len);
18         if(manager.recvfromAckTimeout(buf, &len, 3000, &from))
19         {
20             Serial.print("got reply from : 0x");
21             Serial.print(from, HEX);
22             Serial.print(": ");
23             Serial.println((char*)buf);
24             Serial.print("RSSI: ");
25             Serial.println(driver.lastRssi(), DEC);
26             Serial.println("****RoutingTable****");
27             manager.printRoutingTable();
28             rxGood++;
29             (...)
30         } (...)

```

Na linha 2, o parâmetro "dest" é retirado do URL com recurso à função *argName*, que

procura pelo nome do primeiro parâmetro (índice 0). Na linha 4, o *arg(0)* aponta para o primeiro parâmetro URL retirando o valor do parâmetro para uma string, enquanto que *toInt* converte-o num valor inteiro. O conteúdo deste parâmetro corresponde ao valor do endereço de destino (*addr*), que foi retirado recorrendo à função *toInt*.

Para além deste parâmetro, o utilizador recolhe também o valor do tempo de escuta (*Timeout*) com recurso ao mesmo conjunto de métodos, mas procurando pelo segundo parâmetro (índice 1 - *arg(1)*). Este valor é introduzido na função *setTimeout* que estabelece o tempo de escuta por mensagens de resposta (ACK).

Na Listagem 5.10 está representada a construção da página HTML através da função *snprintf*. O processo utilizado é o mesmo que o da Listagem 5.5, com algumas alterações nas variáveis e na estrutura da página. Esta função guarda na variável *html*, uma string com o conteúdo apresentado no terceiro argumento. Para conseguir introduzir o conteúdo das variáveis dentro da string, os valores são passados recorrendo ao formato (“%d”, *txGood*) como o utilizado na função *printf*.

A estrutura da página segue o enquadramento normal com o cabeçalho (HEADER), corpo (BODY) e os estilos (STYLE) da página. Cada parágrafo de texto ou hiperligação pode ter um tamanho ou cor diferente, com um alinhamento e enquadramento específico dentro das tags `<style>`. Para manter a estrutura dentro das formatações, foram realizados vários testes de ajuste ao texto ou ao formato dos objetos.

O código da linha 8 até à linha 10 corresponde às variáveis que o utilizador procura analisar através dos testes.

Na linha 12 existe a introdução de um formulário com duas variáveis de entrada, para registar o endereço de destino e o tempo de espera pretendido. Assim o utilizador consegue inserir os dados de entrada para o envio da mensagem. Depois da página estar criada, é enviada a string com o conteúdo HTML para o cliente.

Na linha 16 existe uma hiperligação para atualizar a página *web* sempre que o utilizador necessite de desbloquear a página ou atualizar os dados.

Listagem 5.10: Construção da página HTML - Rede Mesh Simples

```

1 snprintf ( html, 2000,
2 "<html>\ <head>\
3   <meta http-equiv='refresh' content='1'/'>\ <title>WifiAP</title>\<style>\
4   (...) </style>\</head>
5 <body>\
6   <h1>Radio Wifi ACESS Point</h1>\
7   <p><b> Uptime: </b>%02d:%02d:%02d</p>\
8   <p><b>RSSI:</b> %d dBm   <b>SNR:</b> %d dB</p>\
9   <p><b>TX Good:</b> %d   <b>ACK Good:</b> %d <b>PDR:</b> %d % </p>\
10  <p><b>RX Good:</b> %d <b>Got Message From:</b> %d </p>\
11  <p><form action="/form"> </p>\
12  <p> <b>Destino:</b><input type="text" name="dest"> <b>Timeout:</b> ...
13     <input type="text" name="timeout" ><input type="submit" ...
14     value="Enviar"></form> </p>\
15  <p>%s  %s  %s</p>\
16  <p><table> %s </table></p>\
17  <p><table> %s </table></p>\
18  <p><a href="javascript:window.location.reload();">!Refresh!</a> </p>\
19 </body>\
20 </html>",
    hr, min % 60, sec % 60, driver.lastRssi(), driver.lastSNR(), ...
    txGood, rxGood, pdr, msgGood, msgfrom, ledText2, ledText3, ledText4, ...
    tablerotas, table10);
    server.send (200, "text/html", html ); (...)
```

### 5.3.3 Resultados

Este caso comporta-se como um sistema com três módulos, cada um conectado a um microcontrolador ESP8266 NodeMCU 1.0, e este alimentado por uma fonte externa de 5 Volts. Como é possível verificar na Figura 5.13, os módulos possuem uma antena *RubberDucky* sintonizada para a frequência 433 Mhz.

Todos os módulos estão configurados de igual forma para evitar erros na parametrização inicial das tramas. Esta parametrização está detalhada na folha do fabricante do

SX1278 e especificada no *firmware* utilizado para este caso. O esquema de configuração apresenta os seguintes parâmetros: Potência de Transmissão = 10 dBm, Largura de faixa = 125 kHz, *Coding Rate* = 4/8, *Spreading Factor* = 4096 Chips/Symbol, CRC Ligado.

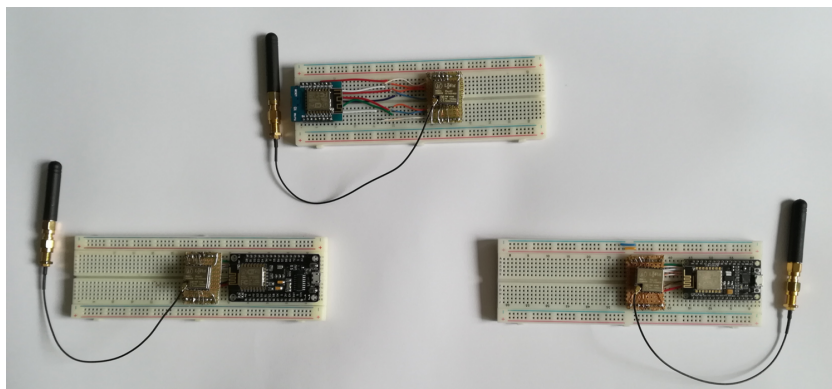


Figura 5.13: Representação física da rede simples.

Segundo as especificações, esta configuração é a mais indicada para distâncias maiores, portanto o autor optou por essa para implementar este exemplo. A realização dos testes envolveu estabelecer dois pontos fixos distanciados entre si, dentro do perímetro de uma habitação, com uma estação numa divisão e outra na divisão mais afastada da casa, junto a uma janela.

As distâncias compreendidas entre estas estações, seriam de cerca de 5 metros, com obstáculos entre eles (paredes que os separam). A estação 1 consistiu numa posição móvel para existir transmissão e receção de dados. Este último permitiria estabelecer rotas diferentes entre os restantes, permitindo que houvesse um contato com a rede, mesmo em situações onde o contato direto não existia.

Assim sendo foi possível verificar dois tipos de situações, onde num caso existe um contato direto com a origem e o destino, e noutra existe um intermediário a servir de ponte de ligação entre as partes. A primeira situação está representada na Figura 5.14, com a mensagem a ser enviada diretamente do nó 1 (origem) para o nó 2 (destino).

Aqui verifica-se também as rotas planeadas para a mensagem principal, com as tabelas dos nós 1 e 2 a serem preenchidas com uma rota com o destino e o próximo salto do

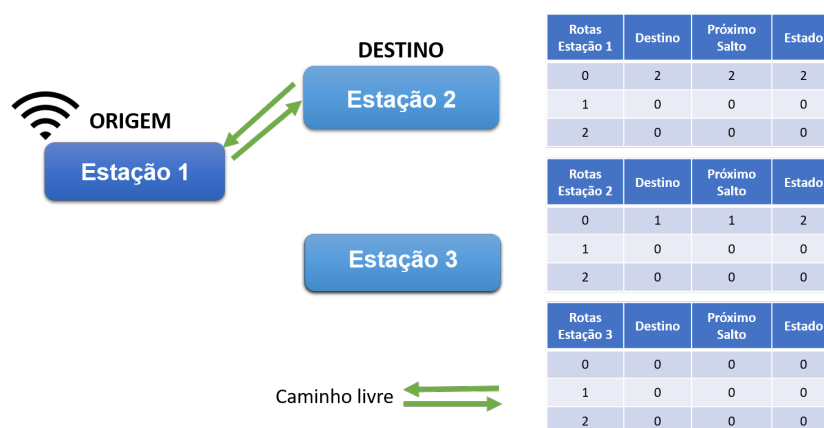


Figura 5.14: Encaminhamento direto da mensagem e respetiva tabelas de rotas.

seu vizinho. O estado é um valor definido no programa para validação da rota, o valor 2 significa que a rota é válida e pode ser utilizada para encaminhamento de mensagens. Nas restantes rotas, os valores dos parâmetros que estão preenchidos com zero, correspondem a rotas que não estão definidas.

Os resultados obtidos para esta situação de contato direto, comportam-se como uma situação de comunicação entre dois pontos fixos, exemplificados no parágrafo anterior. A distância ideal para verificar experimentalmente este teste, está exemplificada na comunicação simples de ponto a ponto.

Na Figura 5.15 está representada a situação de contato indireto, com a utilização de um nó intermédio para ajudar a encaminhar a mensagem da origem para o destino. Quando existe uma falha de comunicação entre dois nós (nó 1=origem; nó2=destino), o nó de origem envia uma mensagem de *broadcast* para os restantes elementos da rede.

Estes por sua vez, retransmitem esta mensagem, depois de se adicionarem à lista de nós visitados por este pedido. Assim é possível atualizar a tabela de rotas dos elementos (com a criação de uma nova rota), permitindo que a mensagem original seja encaminhada até ao destino.

Quando existe o envio de tramas por vias alternativas, os módulos reencaminham a mensagem de *broadcast* para os restantes elementos, levando a haver uma atualização no mapeamento das rotas. Como é possível observar pela Figura 5.15, existe apenas uma

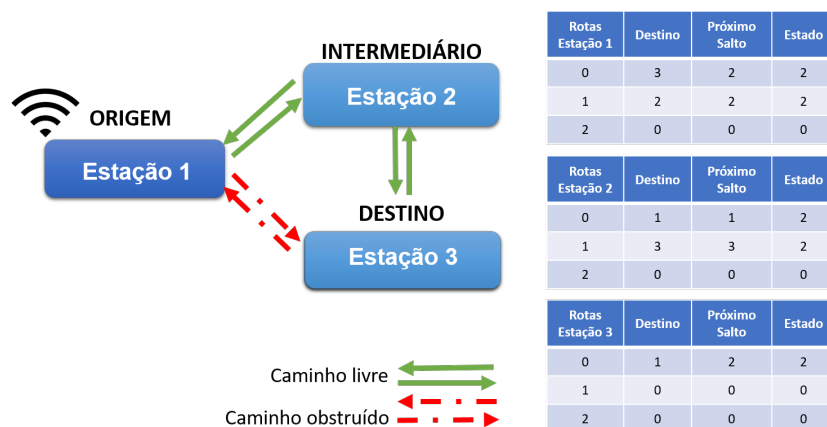


Figura 5.15: Encaminhamento alternativo da mensagem e respetiva tabelas de rotas.

rota na tabela do módulo 3, com o destino sendo o módulo 1 e o próximo salto, o módulo 2. Nas rotas do módulo 2, existe um contato direto entre os elementos, logo as rotas estão definidas apenas para eles. No módulo 1 existem duas rotas: uma que o liga ao módulo 2 e outra que encaminha a mensagem de retorno para o módulo 3, depois de esta passar pelo módulo 2.

Na Figura 5.16 estão representados esses dois cenários com a apresentação da interface criada para interagir com o rádio. No ponto a) existe o encaminhamento direto, enquanto que no ponto b) existe o encaminhamento alternativo das mensagens com as tabelas de rotas da estação 1.

Dependendo da estação, esta irá transmitir mensagens para um destino que será um número representativo de outra estação. No campo *Timeout* é enviado para controlar o tempo que a estação espera pelo envio de uma mensagem. Os campos *TxGood* e *ACKGood* mostram o número de mensagens transmitidas e o número de mensagens de retorno recebidas, respetivamente. O campo *RxGood* representa o número de mensagens recebidas por outras estações e seu endereço (*Got Message From*). A taxa de entrega de mensagens (PDR) apresenta em percentagem o rácio entre o número de mensagens bem transmitidas pelo número total de mensagens transmitidas.

A interface permite analisar o envio de onze mensagens que corresponde a um ciclo

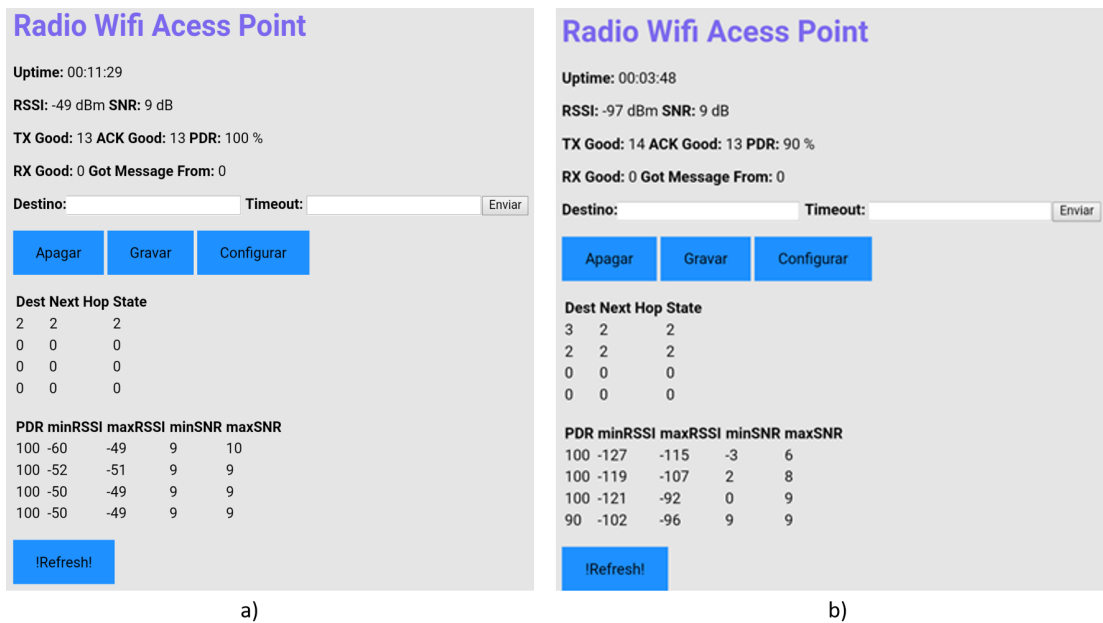


Figura 5.16: a) Encaminhamento direto da mensagem b) Encaminhamento alternativo.

de teste, retornando os valores mínimos e máximos de SNR e RSSI por ciclo. Quando é selecionado o botão Enviar as mensagens são transmitidas para a estação "Destino", se o número de mensagens ultrapassar as onze de um ciclo, o utilizador pode não guardar esses dados, utilizando assim a opção Apagar. Se este pretender guardar um ciclo de teste, após enviar onze mensagens, o utilizador guarda através da opção Gravar. A opção Configurar serve para introduzir dados nos campos Destino e *Timeout*.

## 5.4 Rede Mesh Integrada

Nesta secção serão abordados os resultados do projeto com a apresentação de imagens relacionadas com o *debugging* do processo. O sistema é constituído por três estações, uma principal e duas secundárias, e encontram-se representadas na Figura 5.17.

Uma estação possui um conjunto de leds para permitir ao utilizador identificar o que está a acontecer mais rapidamente. Desta forma como as estações estarão desligadas do computador, o acesso ao *debugging* é facilitado com estes indicadores luminosos.

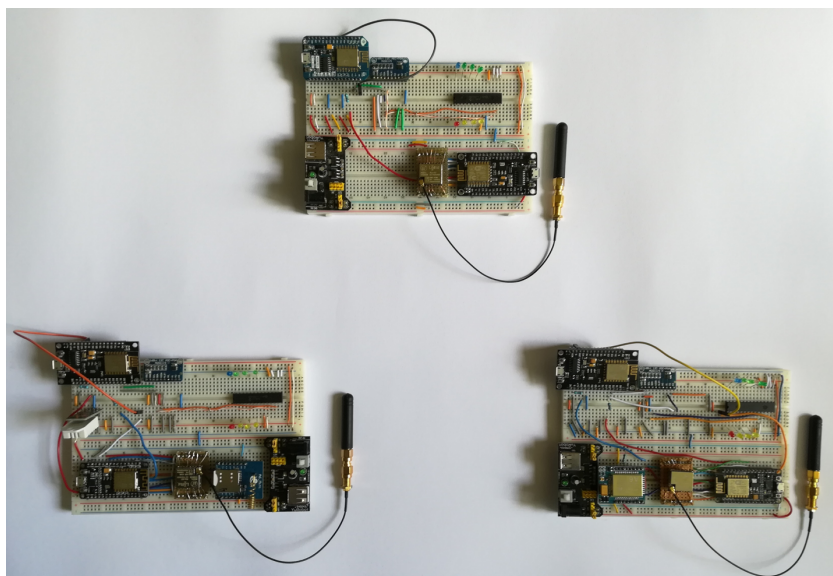


Figura 5.17: Representação física da rede mesh integrada.

A placa de sensores tem quatro leds para identificar as etapas do ciclo do programa. Inicia com a intermitência de um led azul para verificar a ligação à rede WiFi, depois sincroniza com o tempo do servidor NTP. Após estas etapas, o processo continua, recolhendo os dados e enviando para a placa de comunicações. Cada etapa passada significa que um led será ligado. Quando este ciclo estiver concluído, passamos para os leds da placa de comunicações, que acendem em conformidade com a operação executada.

Assim sendo, quando chegamos a esta fase, o led vermelho ligado indica que existe a transmissão da mensagem para a estação principal. Dependendo da resposta, os leds irão acender em conformidade com o sucesso ou não da operação. Como é possível verificar na Figura 5.18, se o envio da mensagem for bem sucedido, e existir uma mensagem de confirmação, dois leds acendem e o último fica apagado (item a) - estados 4 e A).

Se a mensagem não chegar a estação principal, a sequência de leds amarelos fica desligada (item b) - estados 4 e C). Na Figura 5.18 esta sequência identifica também se existiu alguma mensagem a ser recebida, através da ativação do quarto led (item c) - estados 4 e D). Após este processo, o led vermelho fica intermitente, esperando receber alguma mensagem durante o tempo que lhe resta. Nos últimos dez segundos do programa

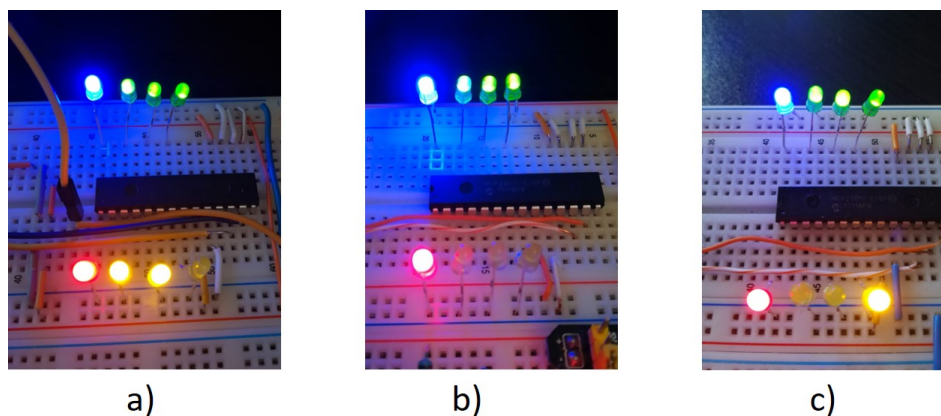


Figura 5.18: Envio e receção da mensagem com sucesso a) sem sucesso b) Resposta recebida c).

existe a intermitência dos leds azul e vermelho para indicar que a estação irá adormecer brevemente.

O programa ao ser executado responde a estes estados que pretendem simplificar a visualização dos processos sem aceder ao computador. Na Figura 5.19 é possível verificar as mensagens da placa de sensores de uma estação.

```

30521,0,31608,0 - Timeout while setting time
30521,0,31608,0 - Clock set from NTP.
30521,0,31608,0 - Setting time failed again. Error code: -1
19700101000030000
DHT22 Humidade: 66.40 % Temperatura: 18.50 *C
Falha na leitura do sensor DS18B20

Velocidade Vento: 0.00km/h

Tensao obtida no cata-vento: 0.48V
Direcao do vento: 2.00 graus
Pluviosidade: 0.00 mm
Pluviosidade/h: 0.00mm/h
Pluviosidade/24h: 0.00mm/24h

Tensao obtida humidade solo: 0.56
Humidade Solo: 100.00%
http://smarthelix.cloudapp.net/prototype-web/api/push/SHCustom:sh0d5ff02?ts=1970010100003
[Dados dos sensores enviados!]
[Receber Info das Comunicacoes]
[Data: A

```

Figura 5.19: Visualização do *debugging* do processo da placa de sensores.

Como é possível verificar existe um processo a correr para a sincronização das horas, mas não teve sucesso. Após várias tentativas, a estação continua, recolhe os dados e envia a hiperligação para a placa de comunicações. Depois espera para receber informação sobre

o estado das comunicações e acende a sequência de leds respetiva. As falhas na leitura e os valores nulos recebidos devem-se à existência de apenas um sensor (DHT22). A aplicação de todos os sensores traria complicações no manuseamento das estações, visto só estar a ser testado a viabilidade das comunicações.

Na Figura 5.20 estão apresentadas as janelas de visualização da placa de comunicações da estação principal e secundária. No envio e receção de uma mensagem bem sucedida, a estação secundária ao enviar uma mensagem para a estação principal recebe uma mensagem de retorno ("Ack\_from\_node1\_time:1522941827") com o tempo atualizado em segundos.

COM3 Estação principal	COM4 Estação secundária
[Receber mensagens.....31163 :: 15:23:42	.
[Receber mensagens.....32164 :: 15:23:43	.
[Receber mensagens.....33164 :: 15:23:44	.
[Receber mensagens.....34164 :: 15:23:45	[Wifi not connected!]
[Receber mensagens.....35241 :: 15:23:46	[Receber url dos sensores..]
got message from : 0x3: http://smarthelix.cloudapp.net/p	[Data: http://smarthelix.cloudapp.net/prototype-web/api/push/SI
****RoutingTable****	[Transmitir radio]
0 Dest: 3 Next Hop: 3 State: 2	[Send message to mesh 1]
1 Dest: 0 Next Hop: 0 State: 0	[Sending SUCESS...]
2 Dest: 0 Next Hop: 0 State: 0	[got reply from : 0x1: Ack_from_node1_time:1522941827
3 Dest: 0 Next Hop: 0 State: 0	RSSI: -62
4 Dest: 0 Next Hop: 0 State: 0	****RoutingTable****
5 Dest: 0 Next Hop: 0 State: 0	0 Dest: 1 Next Hop: 1 State: 2
6 Dest: 0 Next Hop: 0 State: 0	1 Dest: 0 Next Hop: 0 State: 0
7 Dest: 0 Next Hop: 0 State: 0	2 Dest: 0 Next Hop: 0 State: 0
8 Dest: 0 Next Hop: 0 State: 0	3 Dest: 0 Next Hop: 0 State: 0
9 Dest: 0 Next Hop: 0 State: 0	4 Dest: 0 Next Hop: 0 State: 0
Ack_from_node1_time:1522941827	5 Dest: 0 Next Hop: 0 State: 0
[HTTP] begin...	6 Dest: 0 Next Hop: 0 State: 0
[HTTP] GET...	7 Dest: 0 Next Hop: 0 State: 0
[HTTP] GET... code: 200	8 Dest: 0 Next Hop: 0 State: 0
OK	9 Dest: 0 Next Hop: 0 State: 0
Success send: 1	[Enviar SMS.....]
D]	[Sending SMS]
[Receber mensagens.....37926 :: 15:23:48	'Ack_from_node1_time' '1522941827'
[Receber mensagens.....38926 :: 15:23:49	[Estado A]
[Receber mensagens.....39927 :: 15:23:50	[Estado E]
[Receber mensagens.....40927 :: 15:23:51	[Receber mensagens.....40396 :: 15:23:52
	[Estado E]
	[Receber mensagens.....41397 :: 15:23:53
	[Estado E]
	[Receber mensagens.....42397 :: 15:23:54
	[Estado E]
	[Receber mensagens.....43398 :: 15:23:55

Figura 5.20: Visualização do *debugging* do processo (Estado A).

A partir deste tempo, o relógio da estação secundária pode ser atualizado e sincroni-

zado com a hora global. De assinalar que surge sempre um pequeno atraso de segundos entre a hora global e a hora dos relógios. Na estação principal existe a receção de uma mensagem, logo o estado é o D, enquanto que na estação secundária o estado é o A. Depois tanto na estação principal como na secundária, existe um período de receção por mensagens que possam ser reencaminhadas (estado E). Passado um tempo definido para cada estação, este adormece até ao próximo ciclo.

Na Figura 5.21 está representado o processo do estado C, ou seja, o processo que ocorre quando a estação secundária não consegue enviar uma mensagem, não havendo assim mensagem de retorno. O programa repete o envio alguns ciclos, senão este passa para o modo de receção e adormece passado o tempo definido.

COM3 Estação principal	COM4 Estação secundária
[Receber mensagens.....110226 :: 18:56:40	.
[Receber mensagens.....111226 :: 18:56:41	.
[Receber mensagens.....112227 :: 18:56:42	[Wifi not connected!]
[Receber mensagens.....113227 :: 18:56:43	[Receber url dos sensores..]
[Receber mensagens.....114227 :: 18:56:44	[Data: http://smarthelix.cloudapp.net/prototype-web/api/push/
[Receber mensagens.....115228 :: 18:56:45	[Transmitir radio]
[Receber mensagens.....116228 :: 18:56:46	[Send message to mesh 1]
[Receber mensagens.....117229 :: 18:56:47	sendtoWait failed. Are the intermediate meshes running?
[Receber mensagens.....118229 :: 18:56:48	****RoutingTable****
[Receber mensagens.....119230 :: 18:56:49	0 Dest: 0 Next Hop: 0 State: 0
[Receber mensagens.....120230 :: 18:56:50	1 Dest: 0 Next Hop: 0 State: 0
[Receber mensagens.....121230 :: 18:56:51	2 Dest: 0 Next Hop: 0 State: 0
[Receber mensagens.....122231 :: 18:56:52	3 Dest: 0 Next Hop: 0 State: 0
[Receber mensagens.....123231 :: 18:56:53	4 Dest: 0 Next Hop: 0 State: 0
[Receber mensagens.....124232 :: 18:56:54	5 Dest: 0 Next Hop: 0 State: 0
[Receber mensagens.....125232 :: 18:56:55	6 Dest: 0 Next Hop: 0 State: 0
[Receber mensagens.....126233 :: 18:56:56	7 Dest: 0 Next Hop: 0 State: 0
[Receber mensagens.....127233 :: 18:56:57	8 Dest: 0 Next Hop: 0 State: 0
[Receber mensagens.....128233 :: 18:56:58	9 Dest: 0 Next Hop: 0 State: 0
[Receber mensagens.....129234 :: 18:56:59	[Estado C]
130234ms :: 18:57:0	[Receber url dos sensores..]
18:57:0	http://smarthelix.cloudapp.net/prototype-web/api/push/SHCusto
DORMIR AGORA!!	[Transmitir radio]
	[Send message to mesh 1]
	sendtoWait failed. Are the intermediate meshes running?
	****RoutingTable****
	0 Dest: 0 Next Hop: 0 State: 0
	1 Dest: 0 Next Hop: 0 State: 0
	2 Dest: 0 Next Hop: 0 State: 0
	3 Dest: 0 Next Hop: 0 State: 0
	4 Dest: 0 Next Hop: 0 State: 0
	5 Dest: 0 Next Hop: 0 State: 0
	6 Dest: 0 Next Hop: 0 State: 0
	7 Dest: 0 Next Hop: 0 State: 0
	8 Dest: 0 Next Hop: 0 State: 0
	9 Dest: 0 Next Hop: 0 State: 0

Figura 5.21: Visualização do *debugging* do processo (Estado C).

Em ambiente de teste, as distâncias entre estações são reduzidas, portanto para forçar a indisponibilidade na comunicação, colocou-se a estação principal a adormecer. Assim a estação secundária tenta enviar e nunca irá conseguir transmitir corretamente.

Desta forma as mensagens de erro retornam que há falhas na transmissão, portanto não existe maneira de descobrir um novo caminho para a estação principal. Isto é visível pela apresentação da tabela de rotas que não apresenta nenhum percurso alternativo.

## 5.5 Análise do consumo de corrente

O controlo eficiente dos custos energéticos começa com a medição e monitorização das fontes de energia. Para tal foi efetuado um estudo sobre o consumo energético do sistema, com a recolha da corrente consumida ao longo do tempo. Na Figura 5.22 são apresentados dois componentes principais: o sensor de corrente INA219 e o NodeMCU. O circuito analisado representa a carga que é alimentada por uma fonte de alimentação (ex: uma bateria).

Este sensor funciona como um amperímetro, porque as medições são feitas em série com o circuito aberto, possuindo uma resistência *shunt* muito pequena de 0.1 Ohms. A tensão de alimentação pode ser 5 ou 3.3 V e as medições podem detetar tensões até 26 V e 3.2 A com uma resolução de 0.8 mA. Se a configuração for alterada para uma deteção máxima de 400 mA, a resolução é igual a 0.1 mA.

A ligação estabelecida permite verificar a corrente que passa na carga em direção à massa. Estas medições têm que ser efetuadas entre a terminal negativo da carga e a massa real. O que acontece é que a queda de tensão na resistência é proporcional à variação de corrente. Uma referência de massa instável pode trazer problemas para muitos circuitos, resultando em medições incorretas.

O autor utilizou apenas um sensor para realizar os testes, no entanto é possível acoplar mais do que um, através do endereçamento de comunicação I2C. Para conseguir um endereço diferente, o utilizador necessita de soldar os pontos de contato A0 ou A1. Neste teste foi efetuada uma recolha da corrente consumida durante cerca de 90 segundos,

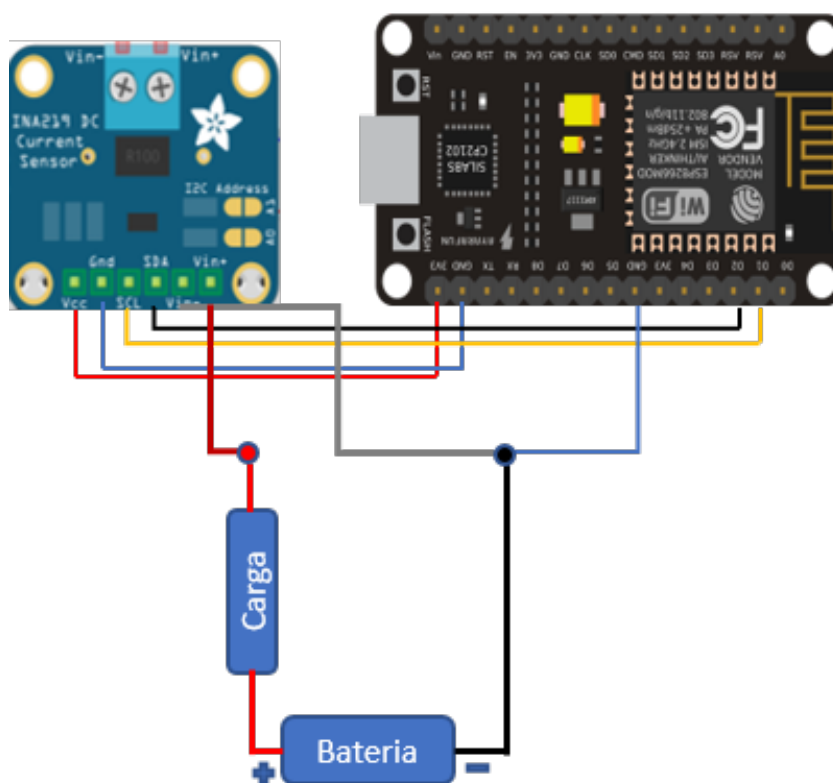


Figura 5.22: Esquema de ligações para a análise do consumo.

com um período ativo de 70 segundos e de descanso a durar 15 segundos. Na Figura 5.23 foi analisado separadamente a corrente consumida para o módulo GSM e para o módulo LoRa. Nos períodos de descanso a corrente do módulo GSM chega aos 4.8 mA enquanto que no módulo LoRa aos 0.8 mA. Quando passam os 15 segundos iniciais, os módulos acordam do descanso e surgem picos de corrente causados pelo despertar dos aparelhos.

A linha laranja representa a corrente consumida do módulo GSM, que apresenta um consumo com algumas flutuações. Após o despertar, existem picos de corrente porque o módulo tenta registar-se à rede GSM, estando pronto para iniciar comunicações. Perto dos 50 segundos, o módulo envia uma mensagem SMS que irá aumentar o consumo da corrente (máx: 117.2 mA), levando a uma diminuição para 35 mA até este adormecer.

A linha azul corresponde à corrente consumida do módulo LoRa, que se mantém abaixo da linha de consumo do módulo GSM. Inicialmente existe um pico de corrente

para o módulo LoRa que inicializa e permanece num estado de *standby* consumindo 2.7 mA. Quando chega aos 50 segundos, o módulo entra em transmissão de dados e a corrente consumida atinge os 29.8 mA. Depois desce novamente, atingindo o período de receção aos 60 segundos. Nesta fase a corrente consumida chega aos 11.7 mA, descendo novamente no período de descanso.

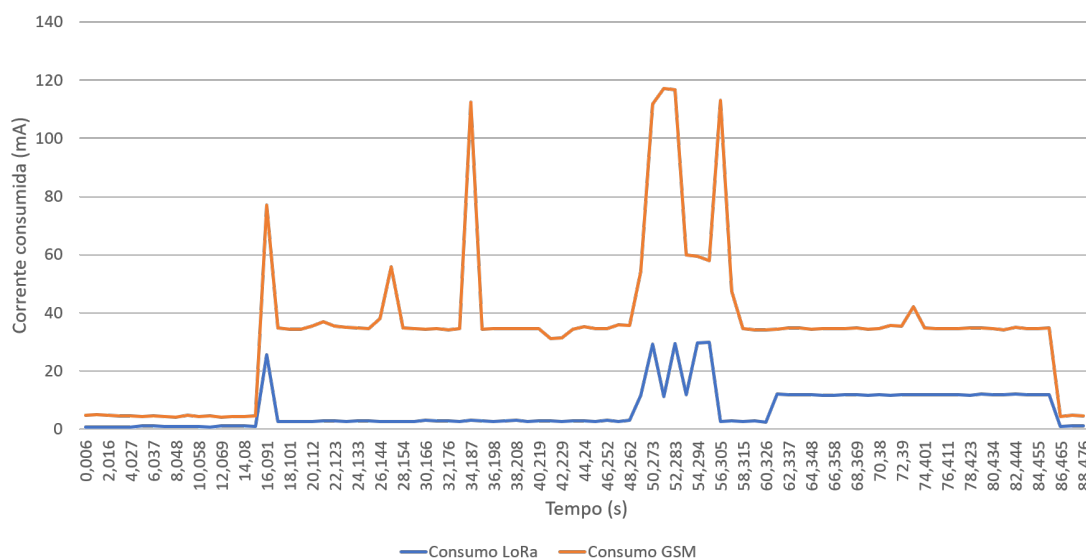


Figura 5.23: Consumo de corrente dos módulos rádio.

O consumo da placa de sensores resultou numa linha em função do tempo, que representa o consumo do NodeMCU com o sensor DHT22 e a interface de leds (através do MCP23017). Este consumo de corrente não inclui todos os sensores porque o autor estaria apenas a validar as comunicações, facilitando no manuseamento das estações. Todos os restantes dados são retirados, mas sem a presença da totalidade dos sensores.

Na Figura 5.24 é possível verificar os consumos de corrente parciais de uma estação secundária. Os testes efetuaram-se separadamente para compreender a influência de cada parte no consumo final. Durante 1 minuto e meio efetuou-se um 1 ciclo de descanso e um de trabalho, com um período de 15 segundos para descanso e 70 segundos para trabalho.

Na linha azul está representada a curva do consumo para a placa de sensores, a linha laranja representa o consumo da placa de comunicações. A curva azul apresenta

um consumo a rondar os 63 mA nos ciclos de trabalho e cerca de 3.6 mA nos ciclos de descanso. Alguns dos testes revelaram que podem surgir picos de corrente entre 100 a 120 mA durante o período ativo de trabalho.

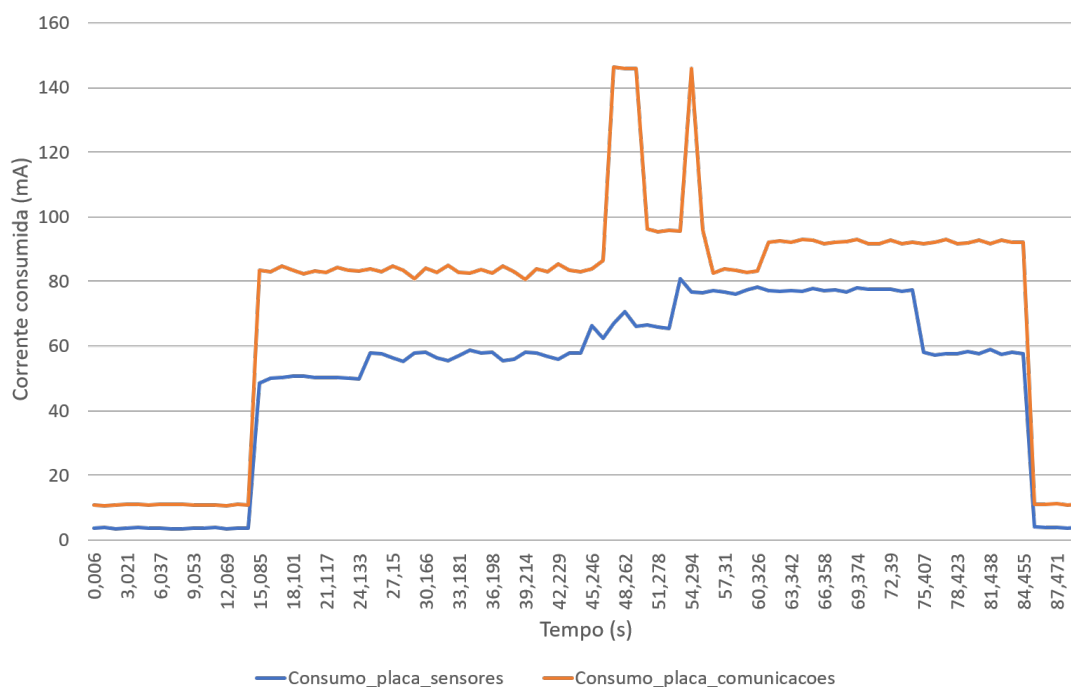


Figura 5.24: Consumo de corrente de uma estação.

Como na estação secundária não há ligação WiFi, a placa de sensores recolhe os dados dos sensores e procura enviar a string resultante para a placa de comunicações. Após a alteração do código, os períodos de sono profundo podem ser introduzidos até um máximo de 71 minutos.

O autor analisou inicialmente o consumo do módulo LoRa e GSM, e só depois analisou a placa de comunicações como um todo. Na linha laranja está representado este consumo para um ciclo, ou seja, adormece durante cerca de 15 segundos e trabalha durante 70 segundos.

De acordo com os resultados, existe um conjunto de picos de corrente que são influenciados pelos processos que ocorrem nos módulos rádio e NodeMCU. No início da execução existe um período constante, em que o NodeMCU tenta conectar-se à rede WiFi, mas

sem sucesso. Depois existe o NodeMCU que tenta estabelecer uma ligação com a placa de sensores, para receber esses dados. Com o sucesso desta etapa, o NodeMCU utiliza o módulo LoRa para transmitir esta informação para outro módulo mais próximo, que responde com uma mensagem de confirmação. Durante o processo de envio é transmitida uma mensagem SMS através do módulo GSM, para informar o utilizador da transmissão efetuada.

Os picos visíveis representam estas etapas, levando o sistema a atingir máximos entre 120 a 250 mA. No período de sono profundo o módulo GSM consome cerca de 4.5 mA e o conjunto NodeMCU e módulo LoRa consumiu cerca de 1 mA, logo somado vai dar uma consumo de 5.5 mA. Apesar de ser definida a menor resolução possível (0.1 mA), na prática verificou-se que mesmo sem algo para medir, o sensor não deteta correntes mais pequenas que 0.8 mA.

A validação das linhas de consumo obrigou à execução de muitos testes, porque existia alguma instabilidade na alimentação dos circuitos de carga, provocando alguns erros de leitura. A análise do consumo de corrente é importante para poder estimar a duração de baterias que sejam instaladas nas estações secundárias.

Assim sendo, a autonomia de uma bateria pode ser calculada através da seguinte equação [45]:

$$\textit{Autonomia da Bateria} = \frac{\textit{Capacidade}}{\textit{Corrente média}} \quad (5.1)$$

Este valor representa a duração da bateria expressa em horas, que pode ser convertida em dias ou meses. A capacidade é expressa em mAh e a corrente média ( $I_{av}$ ) representa o valor de corrente consumida do sistema. Para isso é necessário a corrente ativa ( $I_{ativa}$ ) e a corrente de descanso ( $I_{descanso}$ ), bem com a percentagem desses duas parcelas num dia. Desta forma com estes valores, a corrente média será [45]:

$$I_{av} = I_{Activo} \times \%Activa + I_{Descanso} \times \%Descanso \quad (5.2)$$

Se for considerado uma duração de 1 minuto de período ativo num dia, ou seja, corresponde a 0.017 h (1 min / 60 min = 0.017 h) ou 0.071 % de um dia (0.017/24 =

0.00071). Se considerarmos 200 mA para a corrente no período ativo e 4 mA para a corrente no descanso, a corrente média consumida por dia será 4.14 mA.

Com este valor, assumindo uma bateria com capacidade de 4000 mAh teremos uma autonomia de 40 dias (966 horas), se for uma bateria de 10000 mAh durará até cerca de 101 dias (2415 horas). Para conseguir uma duração maior, a opção passa por reduzir mais a corrente de descanso ou optar por ciclos de descanso de maior duração.

Se for considerado uma duração de 15 minutos de período ativo num dia, ou seja, corresponde a 0.25 h ( $15 \text{ min} / 60 \text{ min} = 0.25 \text{ h}$ ) ou 1.04 % de um dia ( $0.25/24 = 0.0104$ ). Se considerarmos 200 mA para a corrente no período ativo e 4 mA para a corrente no descanso, a corrente média consumida por dia será 6.04 mA:

Assumindo uma bateria com capacidade de 4000 mAh teremos uma autonomia de 28 dias (662 horas), se for uma bateria de 10000 mAh durará até cerca de 69 dias (1656 horas). Tendo isto em conta se for reduzida a corrente de descanso e o período ativo num dia, a autonomia da bateria aumenta, mesmo que o sistema ligado consuma muita corrente.

## 5.6 Análise económica do sistema

Os custos de implementação são importantes na criação de um sistema de comunicação, porque a escolha dos componentes depende destes valores. Tendo em conta as alternativas, a solução passa sempre por escolher aquela que sirva no funcionamento do sistema e esteja dentro do orçamento estipulado.

Para isso foi realizado um levantamento dos componentes constituintes de uma estação e o preço do custo destes equipamentos. Na Tabela 5.8 é possível verificar os componentes eletrónicos e o preço por quantidade de material adquirido. Na rede *mesh* integrada foi incorporado o módulo GSM para possibilitar a sua instalação na necessidade de utilizar alguns destes módulos como último recurso do sistema.

A sua implementação em todas as estações é facultativa, porque o custo associado aos cartões SIM obriga a uma subscrição mensal por unidade. Este pagamento é neces-

Tabela 5.8: Componentes e custo da estação.

Quantidade	Material	Custo (€)
1	Módulo Rádio SX1278	3,04
1	Módulo GSM (opcional)	3,91
2	Microcontrolador NodeMCU	5,2
1	MCP23017	0,87
8	Leds	0,16
2	Resistências	0,04
1	Kit Sensores	53,06
2	Adaptadores	2,38
1	Antena e Conetor SMA	1,59
1	ADS1115	1,74
1	Cabos	1,8
	<b>TOTAL</b>	<b>73,79</b>

sário, para que possam ser enviadas mensagens de texto ou chamadas telefónicas para o utilizador deste serviço.

O conjunto kit sensores engloba todos os sensores, que são pequenos aparelhos baratos, comprados em lojas de fabricantes chineses. Os sensores mais caros foram o pluviómetro, anemómetro e sensor de chuva, que necessitam de adaptadores RJ-11 para recolher estas informações. Os restantes componentes também foram adquiridos em distribuidores chineses em lojas online com preços reduzidos em comparação com os produtos originais.

A Tabela 5.9 resume o custo calculado para três soluções para o módulo rádio. Dependendo da quantidade (Qtd) ou número de estações, o custo total de uma estação ou dez estações varia para módulos diferentes. Se não existir módulo rádio instalado, o preço da estação será de 65,25 euros.

Se uma estação possuir um módulo GSM, o utilizador tem que contabilizar o custo do cartão e respetivo saldo inicial oferecido. Estes valores corresponde a 9.90 euros com 5 euros de saldo incluídos, respetivamente. Para o caso em questão optou-se por um cartão SIM com um tarifário base simples. Este tarifário permite o envio diário de 200

Tabela 5.9: Tabela comparativa de preços de uma estação com diferentes rádios.

Qtd	Estação com			C.Qtd (€)	Custo Subscrição (€)		Total (€)
	1 GSM	1 Xbee	1 LoRa		Um Ano	Três Anos	
1	Não	Não	Não	0	0	0	65,25
1	Sim	Não	Não	13,81	15	55	149,06
1	Não	Não	Sim	3,04	0	0	68,29
1	Não	Sim	Não	18,98	0	0	84,23
1	Sim	Não	Sim	16,85	15	55	152,1
10	Não	Não	Não	0	0	0	652,5
10	Sim	Não	Não	138,1	150	550	1490,6
10	Não	Não	Sim	30,4	0	0	682,9
10	Não	Sim	Não	189,8	0	0	842,3
10	Sim	Não	Sim	168,5	150	550	1521

SMS ou 200 minutos de chamadas telefónicas para números da própria rede. Supondo que não há gastos extra, apenas as recargas, a rede cobra 1,5 € por mês. Totalizando no primeiro ano, um cartão possui 5 euros para três meses, ou seja, são necessários mais 15 euros para cobrar os gastos. Após o primeiro ano, o custo da subscrição será de 20 euros por ano, logo em três anos o encargo corresponderá a 55 euros (15+20+20).

Se for assumido um conjunto de 10 estações com 10 módulos GSM, o custo total aumenta exponencialmente devido ao custo de subscrição anual dos carregamentos. O custo total de 10 estações com 10 módulos LoRa corresponde a metade do custo com os módulos GSM ( $682,9/1490,6 = 0,46$ ). Como é possível verificar, se for instalado um módulo GSM e um módulo LoRa por estação, o custo mantém-se perto do custo apenas com GSM.

A instalação do módulo Xbee é um pouco mais reduzida, porque não estamos a considerar a utilização de um aparelho *Gateway* que tem um preço fixo de 163,18 euros. Uma rede constituída por 10 estações com módulos Xbee e *Gateway* terá um custo de 1005,48 € (163,18+842,3).

Tendo isto em conta, o utilizador verifica que a rede integrada poderá ser utilizada apenas com módulos LoRa, comportando-se como o sistema mais económico comparativamente com rádios diferentes.

## Capítulo 6

# Considerações finais

Neste capítulo será abordado as considerações finais deste projeto, através da realização de uma apreciação global. Mais concretamente identificar as dificuldades que surgiram e apontar as alterações que melhorariam o projeto no futuro.

### 6.1 Dificuldades

O projeto possuiu algumas dificuldades no estudo do módulo rádio mais indicado para o trabalho. Isto porque existe no mercado muitas soluções para as comunicações a longa distância, contudo filtrar estas alternativas em termos de qualidade, funcionalidade e preço foi uma das primeiras dificuldades. Numa fase muito primordial a escolha por determinados modelos de módulos rádio revelou-se não ser a mais acertada por não encaixarem nos propósitos do trabalho. Depois de existir um conjunto de módulos rádio para teste, começou o processo de teste e validação dos módulos a nível técnico. Cada modelo possuía circuitos integrados, *softwares* e configurações diferentes, levando a um maior esforço para decifrar o seu funcionamento. Os obstáculos apareciam quando surgiam erros de programas ou outras complicações que só se conseguiam resolver após muitas tentativas. As discrepâncias entre os resultados da simulação e os resultados práticos também dificultaram porque tiveram que ser feitos ajustes para acertar o sistema.

## 6.2 Apreciação final

Este projeto surgiu de uma proposta da empresa SmartHelix, para o desenvolvimento de um sistema de comunicação longo alcance. Basicamente o sistema tinha de estar organizado numa rede *mesh* que responda em zonas de grande alcance e consiga comunicar entre estações.

Neste projeto optou-se por testar os modelos de rádio em cenários distintos, desenvolvendo a estrutura da rede integrada com a solução mais adequada. Este processo tornou-se mais demorado porque cada módulo obrigou a um profundo trabalho de pesquisa e análise das opções mais fiáveis. Apesar desses contratemplos, no final verificou-se que é possível implementar um sistema de comunicação de longo alcance para uso industrial ou agrícola.

Tendo em conta os múltiplos testes para cada módulo e configuração, os obstáculos causados pelos erros dos programas e ajuste de *hardware*, foram alguns dos principais problemas. A quantidade de pinos do microcontrolador limitavam a utilização de muitos periféricos, obrigando à separação da estação em duas placas. Na placa de sensores sobravam poucos pinos para poder acrescentar os módulos rádio e GSM, logo optou-se por acrescentar outro NodeMCU que fizesse esse trabalho.

No entanto, este projeto cumpriu na totalidade os objetivos a que foi proposto, porque todo o sistema de comunicação funciona, permitindo assim transmitir e receber dados a longas distâncias. Dentro de uma estação secundária, a informação recolhida dos sensores é encapsulada numa mensagem para ser transferida para a placa de comunicações. Neste local o módulo rádio LoRa envia essa mensagem para o módulo LoRa da estação principal. Depois este simplesmente retorna uma mensagem de confirmação e transmite essa mensagem para a Internet.

Deste modo, face às características do projeto, o sistema pode funcionar em diferentes locais de uma área agrícola ou industrial, para comunicar os dados ambientais do local para uma central. Esta atuação permite controlar sistemas de irrigação, informando os seus utilizadores do estado das suas plantações. Os processos químicos e ambientais po-

dem assim ser monitorizados após a instalação do sistema em qualquer local, promovendo assim a gestão inteligente sem necessidade de intervenientes a tempo inteiro.

Após a instalação do sistema, os dados ambientais podem ser monitorizados com um baixo custo económico, mesmo com pontos situados a quilómetros da estação principal. Desta forma promove-se não só a portabilidade do sistema, mas também uma gestão inteligente sem necessidade de técnicos a tempo inteiro.

### 6.3 Trabalho futuro

No futuro pode fazer-se uma análise mais intensiva da influência das obstruções nas comunicações longo alcance. Mais concretamente um estudo sobre a penetração do sinal em edifícios, florestas e campo aberto. A ideia passaria por aperfeiçoar o sistema existente com diferentes antenas ou módulos mais avançados. Também seria interessante aumentar o número de estações secundárias para verificar a interoperabilidade dos nós, tanto em ambiente florestal como em ambiente citadino.

O dimensionamento da estação numa placa PCB, de forma a reduzir o tamanho, para ser aplicado num recipiente estanque e facilitar assim o manuseamento do material entre cada instalação. A sincronização do tempo também poderia ser melhorada, com recurso a um relógio digital externo para manter a precisão do tempo dentro das estações.



# Referências Documentais

- [1] R. C. Dorf, *The Electrical Engineering Handbook Series*, second ed. (CRC Press, 2003)
- [2] C. Laster, *The Beginner's Handbook of Amateur Radio*, fourth ed. (McGraw-Hill Company, 2001)
- [3] MiniPhysics, *The Electromagnetic Spectrum | Mini Physics - Learn Physics Online* (2017),  
URL: <https://www.miniphysics.com/electromagnetic-spectrum{}25.html>
- [4] A. V. Raisanen and A. Lehto, *Radio Engineering for Wireless Communication and Sensor Applications* (Artech House, 2003)
- [5] 4Gon Solutions, *Fresnel Zones* (2017),  
URL: <http://www.4gon.co.uk/solutions/technical{}fresnel{}zones.php>
- [6] B. Schweber, *Antenna Principles, Part 1 | DigiKey* (2017),  
URL: <https://www.digikey.com/en/articles/techzone/2017/apr/understanding-antenna-specifications-and-operation>
- [7] Q. Liu and Q. Liu, *A study on topology in computer network*, Proceedings - 7th International Conference on Intelligent Computation Technology and Automation, ICICTA 2014 (2015), pp. 45–48
- [8] D. M. Hernandez, G. Peralta, L. Manero, R. Gomez, J. Bilbao and C. Zubia, *Energy and coverage study of LPWAN schemes for Industry 4.0*, 2017 IEEE International

- Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM) (2017), pp. 1–6
- [9] LoRa Alliance, *Lora Alliance* (2017),  
URL: <https://www.lora-alliance.org/>
- [10] Htdvisser, *LoRaWAN - The Things Network Wiki* (2017),  
URL: <https://www.thethingsnetwork.org/wiki/LoRaWAN/Home>
- [11] H. Charles, *NRF24L01 real life range test – Charles’s Blog* (2013),  
URL: <http://hallard.me/nrf24l01-real-life-range-test/>
- [12] Nordic Semiconductor, *nRF24L01+ Datasheet*, Nordic Semiconductor (2008), p. 75,  
URL: <https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss{ }Preliminary{ }Product{ }Specification{ }v1{ }0.pdf>
- [13] Grusin Mike, *RFM69HCW Hookup Guide* (2016),  
URL: <https://learn.sparkfun.com/tutorials/rfm69hwc-hookup-guide?{ }ga=2.260021130.1651276792.1510677922-160417562.1507647718>
- [14] Hope Microelectronics, *RFM69HCW Datasheet* (2006), pp. 1–79
- [15] Semtech Corporation, *SX1276/77/78/79 Datasheet* (2015) (March), p. 132,  
URL: <http://www.semtech.com/images/datasheet/sx1276{ }77{ }78{ }79.pdf>
- [16] Coudé Roger, *Radio Mobile VE2DBE* (2017),  
URL: <http://www.ve2dbe.com/english1.html>
- [17] B. J. Henderson, *Radio Propagation and Radio Coverage Computer Simulation Program* (2013)
- [18] ESP8266 Datasheet, *ESP8266EX Datasheet*, Espressif Systems (2015), pp. 1–31,  
URL: <https://www.adafruit.com/images/product-files/2471/0A-ESP8266{ }{ }Datasheet{ }{ }EN{ }v4.3.pdf>

- [19] A. Yadi, *Fullstack IoT Development* (2016),  
 URL: [https://www.slideshare.net/andriy\\_yadi/fullstack-iot-development](https://www.slideshare.net/andriy_yadi/fullstack-iot-development)
- [20] Wim3d, *Mobile Intercom With A6 Module and Arduino* (2017),  
 URL: <http://www.instructables.com/id/Mobile-Intercom-With-A6-Module/-and-Arduino/>
- [21] M. F. Mira, *Desenvolvimento de um circuito eletrônico com sensores para uma estação meteorológica* (2017), p. 85
- [22] ANACOM, *Quadro Nacional de atribuição de frequências 2009/2010* (2010), pp. 1–262,  
 URL: [http://www.anacom.pt/streaming/qnaf20092010{}\\_07042010.pdf?contentId=1019281{&}field=ATTACHED{}\\_FILE](http://www.anacom.pt/streaming/qnaf20092010{}_07042010.pdf?contentId=1019281{&}field=ATTACHED{}_FILE)
- [23] E. C. Committee, *ERC Recommendation 70-03*, September (2015)
- [24] A. Kumbhar, *Overview of ISM Bands and Software-Defined Radio Experimentation*, *Wireless Personal Communications* (2017) (June), pp. 1–14
- [25] K. D.R and D. SheeHan, *Deep Dive: Which LPWAN Technology is Right for IoT?* (2016),  
 URL: <https://pt.slideshare.net/infiswift/lpwan-for-iot-62591541>
- [26] R. Blake and L. Chartrand, *Wireless Communication Technology*, first ed. (Delmar Cengage Learning, 2000)
- [27] I. Poole, *Electromagnetic Waves and Radio Propagation :: Radio-Electronics.Com* (2017),  
 URL: [http://www.radio-electronics.com/info/propagation/em{}\\_waves/electromagnetic{}\\_waves.php](http://www.radio-electronics.com/info/propagation/em{}_waves/electromagnetic{}_waves.php)

- [28] A. W. Graham, N. C. Kirkman and P. M. Paul, *Mobile Radio Network Design in the VHF and UHF Bands* (John Wiley & Sons, Ltd, 2006),  
URL: <http://doi.wiley.com/10.1002/9780470059128>
- [29] Atmel Company, *Range Calculation for 300MHz to 1000MHz Communication Systems* (2015),  
URL: <http://www.atmel.com/Images/Atmel-9144-Range-Calculation{ }/Application-Note.pdf>
- [30] I. Poole, *Free Space Path Loss FSPL | Formula Calculator | Radio-Electronics.com*,  
URL: <http://www.radio-electronics.com/info/propagation/path-loss/free-space-formula-equation.php>
- [31] I.-R. Recommendation, *Propagation data and prediction methods for the planning of indoor radiocommunication systems and radio local area networks in the frequency range 300 MHz to 100 GHz P Series Radiowave propagation* (2015), pp. 1238–8,  
URL: <https://www.itu.int/dms{ }pubrec/itu-r/rec/p/R-REC-P.1238-8-201507-S!!PDF-E.pdf>
- [32] S. Winder, J. Davies and J. J. Carr, *Radio and RF Engineering Pocket Book*, third ed. (Newnes, 2002),  
URL: <http://search.ebscohost.com/login.aspx?direct=true{ }db=e000tww{ }AN=195272{ }site=eds-live>
- [33] J. Burns, S. Kirtay and P. Marks, *Future use of Licence Exempt Radio Spectrum* **44** (2015) (July), p. 72
- [34] E. C. Committee, *The European table of frequency allocations and application in the frequency range 9 kHz to 3000 GHz*, Erc Report 25 (2011) (June), p. 275
- [35] H. Mazar, *International , Regional & National Regulation of Short Range Devices* **1** (2015)

- [36] B. Meador, *A Survey of Computer Network Topology and Analysis Examples* (2008), pp. 1–79,  
URL: <http://www.cs.wustl.edu/~jain/cse567-08/ftp/topology/index.html>
- [37] D. H. Kim and J. Y. Lim, *Using Wi-Fi and LoRa*, 2016 Ieee (2016), pp. 1–3
- [38] J. P. Bardyn, T. Melly, O. Seller and N. Sornin, *IoT: The era of LPWAN is starting now*, European Solid-State Circuits Conference **2016-October** (2016), pp. 25–30
- [39] U. Raza, P. Kulkarni and M. Sooriyabandara, *Low Power Wide Area Networks: An Overview*, IEEE Communications Surveys & Tutorials **19** (2017) (2), pp. 855–873
- [40] Weightless, *Weightless - Setting the Standard for IoT* (2017),  
URL: <http://www.weightless.org/>
- [41] Hope Microelectronics Co., *Datasheet: RFM95/96/97/98(W) v1.0* **98** (2014), p. 121,  
URL: [http://www.hoperf.com/rf{}\\_transceiver/lora/RFM95W.html](http://www.hoperf.com/rf{}_transceiver/lora/RFM95W.html)  
[http://www.hoperf.com/upload/rf/RFM95{}\\_96{}\\_97{}\\_98W.pdf](http://www.hoperf.com/upload/rf/RFM95{}_96{}_97{}_98W.pdf)
- [42] Ddan, *Using LoRa shield in Packet Mode | Open Electronics* (2016),  
URL: <https://www.open-electronics.org/using-lora-shield-in-packet-mode/>
- [43] Stor Marcel, *Comparison of ESP8266 NodeMCU development boards • my2cents* (2015),  
URL: <https://frightanic.com/iot/comparison-of-esp8266-nodemcu/-development-boards/>
- [44] Ai-Thinker, *A6/A7/A6C User Manual* (2016), p. 21,  
URL: [https://www.makerfabs.com/desfile/files/A6{}\\_A7{}\\_A6C{}\\_datasheet-EN.pdf](https://www.makerfabs.com/desfile/files/A6{}_A7{}_A6C{}_datasheet-EN.pdf)

- [45] Genevey Sebastien and Dominguez Christian, *Improving Battery-Powered Device Operation Time Thanks To Power Efficient Sleep Mode*,  
URL: [https://www.design-reuse.com/articles/42405/  
improving-battery/-powered/-device-operation-time-thanks/  
-to-power-efficient-sleep-mode.html](https://www.design-reuse.com/articles/42405/improving-battery/-powered/-device-operation-time-thanks/-to-power-efficient-sleep-mode.html)

# Anexo A

## A.1. Tabelas/Imagens informativas

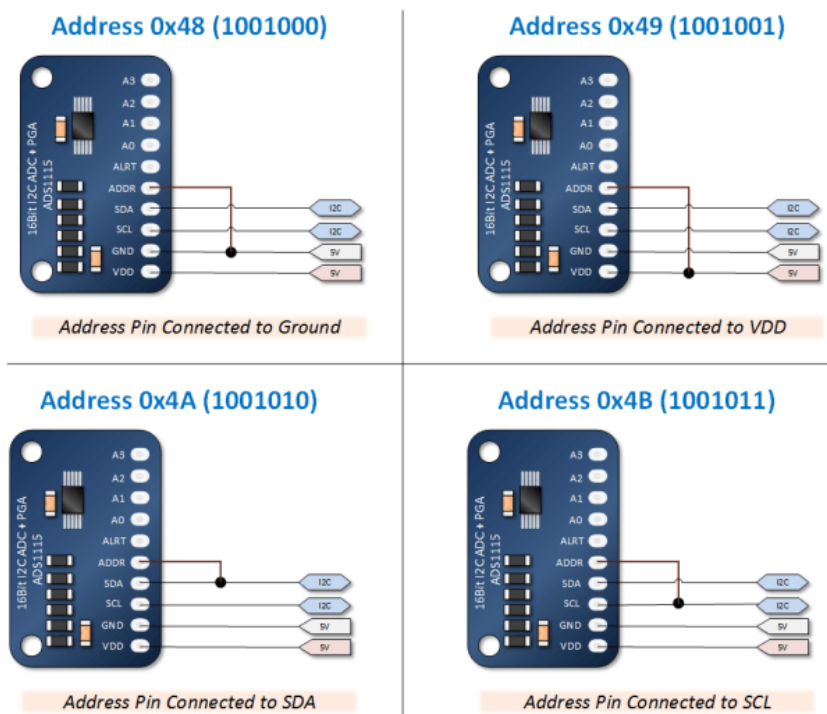


Figura 6.1: Conversor ADS1115.

Anexo A.

Radio Mobile

**➕ New Coverage**

From: T\_433 MHz\_11.14 dBm

Centre Site	<input type="text" value="Transmissor"/>	
Antenna Height (m above ground)	<input type="text" value="2"/>	6.56 ft
Antenna Type	<input type="text" value="Omni"/>	
Antenna Azimuth (°)	<input type="text" value="0"/>	
Antenna Tilt (°)	<input type="text" value="0"/>	
Antenna Gain (dBi)	<input type="text" value="6"/>	
Mobile Antenna Height (m)	<input type="text" value="2"/>	6.56 ft
Mobile Antenna Gain (dBi)	<input type="text" value="2"/>	
Description	<input type="text" value="T_433 MHz_11.14 dBm*"/>	
Frequency (MHz)	<input type="text" value="433"/>	
Tx power (Watts)	<input type="text" value="0.013"/>	11.14 dBm
Tx line loss (dB)	<input type="text" value="3"/>	
Rx line loss (dB)	<input type="text" value="0.5"/>	
Rx threshold (µV)	<input type="text" value="0.5"/>	-113.02 dBm
Required reliability (%)	<input type="text" value="70"/>	
Strong Signal Margin (dB)	<input type="text" value="10"/>	
Strong Signal Color	<input style="width: 100%;" type="color" value="#00ff00"/>	
Weak Signal Color	<input style="width: 100%;" type="color" value="#ffff00"/>	
Opacity (%)	<input type="text" value="50"/>	
Maximum range (km)	<input type="text" value="10"/>	6.2137 mi
Rendering	<input type="text" value="High resolution"/>	
Use land cover	<input checked="" type="checkbox"/>	
Use two rays	<input checked="" type="checkbox"/>	
<input type="button" value="Define as default values"/>		<input type="button" value="Restore original values"/>
<input type="button" value="Submit"/>		

Figura 6.2: Formulário de criação de uma nova cobertura [17].

Tabela 6.1: Tipos de aplicações para os dispositivos de pequena potência (SRD) [22]

#	Aplicações
1	Uso geral
2	Sistemas de detecção, seguimento e aquisição de dados
3	Sistemas de acesso sem fios / Redes locais via rádio (WAS/RLAN)
4	Aplicações em Caminhos de ferro
5	Sistemas telemáticos \\de transportes rodoviários - RTTT
6	Aplicações de Radiodeterminação
7	Alarmes
8	Controlo de modelos
9	Aplicações indutivas
10	Microfones emissores e equipamentos auxiliares auditivos
11	RFID - Sistemas de Identificação RF
12	Sistemas sem fios para aplicações médicas
13	Aplicações áudio sem fios
14	Sistemas de telecomando,telemedida,telealarmes e transmissão de dados