

Nelson Filipe Duarte Pinto

**SISTEMA DE MONITORIZAÇÃO E CONTROLO REMOTO EM
REDES ENERGÉTICAS INTELIGENTES**



UNIVERSIDADE DO ALGARVE

Instituto Superior de Engenharia

2018

Nelson Filipe Duarte Pinto

**SISTEMA DE MONITORIZAÇÃO E CONTROLO REMOTO EM
REDES ENERGÉTICAS INTELIGENTES**

Mestrado em Engenharia Elétrica e Eletrónica

Especialização em Tecnologias de Informação e

Telecomunicações

Trabalho efetuado sob a orientação de:

Professor Doutor Jânio Miguel E. Ferreira Monteiro

Professor Doutor Jorge Filipe Leal Costa Semião



UNIVERSIDADE DO ALGARVE

Instituto Superior de Engenharia

2018

Sistema de monitorização e controlo remoto em redes energéticas inteligente

Declaração de autoria de trabalho

Declaro ser o autor deste trabalho, que é original e inédito. Autores e trabalhos consultados estão devidamente citados no texto e constam da listagem de referências incluída.

Nelson Filipe Duarte Pinto

Copyright © 2018. Todos os direitos reservados em nome de Nelson Filipe Duarte Pinto. A Universidade do Algarve tem o direito, perpétuo e sem limites geográficos, de arquivar e publicitar este trabalho através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, de o divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Dedicado aos meus pais, namorada e filho.

AGRADECIMENTO

Gostaria de agradecer ao Professor Doutor Jânio Miguel Evangelista Ferreira Monteiro a grande disponibilidade em me acompanhar e orientar, ao longo desta tese. As suas ideias, comentários, conhecimentos, críticas e opiniões foram significativas para a conclusão de todo este trabalho. Um muito obrigado por todo o apoio.

Um agradecimento aos Professores Jorge Semião e Cristiano Cabrita, pelo acompanhamento e troca de ideias. Um agradecimento especial ao Prof. Luís Oliveira, por auxiliar e facultar material para alguns dos ensaios.

Ao colega Dario Cruz, pelos debates e *brain storming* diários.

A todos os outros docentes e funcionários da Área Departamental de Engenharia Elétrica e Eletrónica, que me apoiaram de uma forma ou de outra, no decorrer das aulas e destes trabalhos.

Agradeço a todos os colegas de curso, com os quais se trocaram bastantes ideias e horas de estudo. A destacar o meu amigo de longa data, David Oliveira, presente ao longo de todo o meu percurso académico e com o qual se passaram infindáveis horas de trabalho e “gandulice”.

Aos meus pais e família, pelo apoio constante.

À minha namorada Liliana, pela paciência e compreensão infinita, dos meses de trabalho que se passaram.

Ao meu filho Simão, pelo apoio na escrita da tese, tantas vezes a escrever por cima do que eu escrevia. Um amor.

Aos meus amigos, que “sofreram” com a minha ausência.

A todos os que fizeram parte da minha vida académica, direta e indiretamente, contribuíram para chegar a este ponto.

Um muito obrigado a todos, fico-vos eternamente grato.

Nelson Pinto, Faro, 26 de setembro, 2018

RESUMO

A necessidade do aumento da eficiência energética, forçada por políticas a nível mundial, promove a busca de novas soluções de produção e gestão do consumo elétrico. A adoção generalizada de fontes de energia renováveis, em larga e pequena escala, introduz novos paradigmas na rede elétrica tradicional, que passam pela geração distribuída e pela gestão do consumo. Assim, e uma vez que as fontes de energia renováveis dependem de fatores ambientais não controláveis, deve passar-se a controlar o consumo, para ajustá-lo à produção. Para gerir corretamente a demanda, é necessário conhecer o estado atual da rede e das fontes de geração, assim como prever os níveis de consumo e de produção energética futuros. A previsão assenta por sua vez em algoritmos de aprendizagem que requerem dados históricos das variáveis envolvidas. O ajuste manual da demanda é uma tarefa complicada, pelo que há a necessidade de criar sistemas que possibilitem a automação desse ajuste, com o mínimo de intervenção humana.

Neste contexto, o surgimento de conceitos como a Internet das Coisas, abre um novo leque de possibilidades em termos de obtenção de dados. A utilização destas tecnologias, permite monitorizar variados parâmetros elétricos e ambientais, obtendo os dados fundamentais para tomar decisões sobre o uso otimizado da mesma e posteriormente atuar nos equipamentos a controlar.

Neste âmbito, no presente trabalho foi desenvolvido um sistema de monitorização elétrica e ambiental, utilizando a tecnologia WSN de longo alcance LoRaWAN, que permite uma elevada dispersão geográfica dos sensores. O trabalho está integrado no projeto AGERAR. O sistema encarrega-se da obtenção, transmissão e visualização dos dados, recolhidos em vários pontos de uma micro rede. Foram desenvolvidos alguns módulos de medição elétrica, para aplicação numa micro rede piloto, existente num dos parceiros do projeto.

PALAVRAS-CHAVE: Internet das Coisas, Redes de sensores sem fios, Monitorização, LoraWAN, Consumo energético, Micro redes

ABSTRACT

The need to increase energy efficiency, influenced by global policies, is driving the search for new solutions in the production and demand management of electricity. The adoption of renewable energy sources, on both a large and small scales, is introducing new paradigms in the traditional electrical grid, that include distributed generation and consumption management. In fact, since the renewable energy sources depend on uncontrolled environmental factors, energy consumption must be adjusted in response to the generation levels. In order to manage demand, we need to know the current state of the electrical grid and of the renewable energy sources, as well as to predict the consumption levels and future energy production. Prediction is based on learning algorithms that require historical data from all the involved systems. The manual adjustment of demand is a very complex task, so there's a need to create automation systems that perform this adjustment, with a minimum of human interaction.

In this context, concepts as the Internet of Things support a new range of opportunities, reflected in data measurement. The use of these technologies allows the monitoring of several electric and environmental parameters, obtaining the fundamental data to make decisions about the optimized use of energy, and to act properly in the equipment to be controlled.

In this context, the present work describes an electrical and environmental monitoring system that was developed, using the long range WSN technology LoraWAN, which will allow a high geographic dispersion of the sensors. This work is a part of European funded project AGERAR and is responsible to obtain, transmit and display data obtained at several points of an electrical *microgrid*. Several electric measurement modules were developed, that are going to be installed in a pilot *microgrid* of one of the project partners.

KEYWORDS: Internet of Things, Wireless Sensor Networks, Monitoring, LoraWAN,
Energy consumption, Microgrid

ÍNDICE

1.	Introdução.....	1
1.1	Contextos.....	1
1.1.1	Contexto energético – Diretivas e Metas Propostas	1
1.1.2	Contexto das Redes Elétricas	2
1.1.3	Contexto dos Dispositivos Internet.....	5
1.2	Problema.....	6
1.3	Objetivos da Dissertação	7
1.4	Estrutura da Tese.....	10
2.	Estado da Arte	11
2.1	Breve Introdução às Redes de Sensores sem Fios	11
2.2	Análise das Tecnologias WSN Existentes.....	12
2.2.1	WSN para curtas distâncias.....	13
2.2.2	WSN para médias distâncias	15
2.2.3	WSN para longas distâncias	16
2.3	Análise Comparativa das Diferentes Tecnologias WSN.....	19
2.4	Considerações Finais.....	21
3.	Lora e LoraWAN	23
3.1	Camada Física PHY Lora.....	23
3.1.1	Conceitos Base	23
3.1.2	Modulação Lora.....	26
3.1.3	Formato da Trama e Normas Aplicáveis	31
3.1.4	Link Budget e Sensibilidade.....	34
3.2	Camada de Acesso ao Meio MAC LoraWAN	37
3.2.1	Introdução.....	37
3.2.2	Arquitetura e Componentes de Rede	38

3.2.3	Classes de Dispositivos	40
3.2.4	Características da rede	43
3.2.5	Trama LoraWAN	46
3.2.6	Ativação dos Nós	47
3.2.7	Segurança	49
3.3	Comunicação para Redes Backhaul	50
3.3.1	Gateway – Servidor de Rede	51
3.3.2	Servidor de Rede – Outros Servidores	53
4.	Desenvolvimento do Trabalho	55
4.1	Micro Rede INTA	55
4.2	Arquitetura da Rede WSN	57
4.2.1	Escolha de material LoraWAN	57
4.2.2	Esquema da rede de sensores	58
4.3	Componentes da Rede WSN	60
4.3.1	Nós Sensores	60
4.3.2	Gateway	63
4.3.3	Servidor de Rede NS	70
4.3.4	Servidor de Aplicações Thingsboard	70
4.3.5	Rede Backhaul	71
4.3.6	Estação Meteorológica	72
4.3.7	Servidor de Escalonamentos SS	74
4.3.8	Comunicação e Protocolos Entre os Diferentes Componentes	75
4.3.9	Formatos de Dados no Payload dos Nós LoraWAN	77
4.4	Módulos de Medição	78
4.4.1	Módulos na Micro Rede INTA	79
4.4.2	Módulo Emissor (Lopy Power Board)	81
4.4.3	Equipamentos RS485 (Modbus) Para Medição de Tensão e Corrente AC ..	84

4.4.4	Transdutores para medições DC.....	85
4.4.5	Módulo de Medição de 4 Correntes (DC_4i).....	87
4.4.6	Módulo de Medição de Tensão DC (DC_1V).....	88
4.4.7	Módulo de Medição de Tensão e Corrente DC (DC_1V_1i).....	89
4.4.8	Módulo de Medição de Tensão e Três Correntes DC (DC_1V_3i)	90
4.5	Software dos Módulos de Medição.....	91
4.5.1	Lógica de execução do código fonte.....	91
4.5.2	Conversão de variáveis no transdutor HO 10-P	93
4.5.3	Conversão de variáveis no transdutor Tamura	94
4.5.4	Conversão de variáveis no transdutor LV 25-P.....	95
5.	Implementação Prática da Arquitetura Proposta	97
5.1	Ensaio Iniciais	97
5.1.1	Ensaio de cobertura INTA.....	97
5.1.2	Ensaio de perda de pacotes LoraWAN	101
5.1.3	Ensaio dos módulos de medição com HO 10-P	106
5.1.4	Ensaio dos módulos de medição com LV 25-P.....	107
5.2	Setup para implementação inicial nas instalações INTA.....	109
6.	Conclusões e Trabalho Futuro.....	113
6.1	Conclusões.....	113
6.2	Trabalho Futuro	115
6.3	Publicações	116
	Bibliografia e Referências	117
	Apêndices	127
A.	Características de Redes WSN	127
A.1	Arquiteturas de redes WSN	127
A.2	Considerações WSN.....	128
A.3	Categorização de redes sem fios	129

A.4	Constituição de um nó.....	131
A.5	Aplicações WSN.....	133
A.6	Aplicações WSN em Smart Grids.....	134
B.	Tabela Comparativa Tecnologias WSN.....	137
C.	Material LoraWAN.....	138
C.1	Módulos LoraWAN	138
C.1.1	STM32L072Z-LRWAN1.....	138
C.1.2	Adafruit Feather M0	139
C.1.3	Pycom Lopy	139
C.1.4	Libelium Waspote LoRaWAN	139
C.1.5	Microchip DV164140-1	140
C.1.6	Multitech MultiConnect mDot	140
C.1.7	Seedstudio Seeeduino LoRaWAN.....	140
C.1.8	Comparação de características técnicas dos módulos.....	141
C.2	Gateway.....	141
C.2.1	LoraGo PORT	142
C.2.2	Seedstudio Lora Hat.....	142
C.2.3	IMST iC880A.....	142
C.2.4	Laird Sentrius RG186	142
C.2.5	Multitech Conduit GATEWAY IP67 BASE STATION.....	142
C.2.6	Kerlink Wirnet Station.....	143
C.2.7	Comparação de características técnicas das gateways.....	143
C.3	Servidores LoraWAN NS.....	144
C.3.1	Loraserver.io.....	144
C.3.2	Lorawan-server.....	147
C.4	Servidores de Aplicação AS	148
C.4.1	AS Offline	148

C.4.2	AS Online	149
D.	Instalação e configuração inicial do RaspberryPi3	149
E.	Reset do Concentrador iC880.....	150
F.	Configuração da Gateway	151
G.	Configuração do Servidor de Rede NS	153
H.	Servidor de Aplicações AS.....	154
I.	Script de Leitura da Estação Metereológica.....	155
J.	MQTT	163
K.	Envio de mensagens para nós LoraWAN (downlink)	166
L.	Pontos de leitura na rede INTA	168
M.	Componentes dos Módulos DC.....	169
M.1	Sensores de corrente.....	169
M.2	Tipos de transdutores de corrente	170
M.3	Conversor analógico digital ADS1115	171
M.4	Dimensionamento do transdutor de corrente LEM HO 10-P.....	172
M.5	Dimensionamento do transdutor de corrente TAMURA L31S050S05FS	175
M.6	Dimensionamento do transdutor de tensão LEM LV-25P	180
N.	Tabela comparativa de transdutores de corrente	185
O.	Esquemático da placa Lopy Power Board V1	186
P.	Esquemático da placa Lopy Power Board V2.....	187
Q.	Esquemático da placa DC_4i_V01	188
R.	Esquemático da placa DC_1V_V01	189
S.	Esquemático da placa DC_1V_1I_V01	190
T.	Esquemático da placa DC_1V_3I_V01	191
U.	Configurações utilizadas no NS e AS	192
U.1	Setup de rede e configurações utilizadas.....	192
U.2	Configurações de rede.....	195

U.3	Configurações de gateway.....	195
U.4	Configurações de Handlers.....	196
U.5	Configurações de Connectors.....	197
U.6	Configurações de perfis de dispositivos.....	199
U.7	Configurações dos Nós no NS e AS	201
V.	Valores Obtidos nas leituras dos módulos de tensão	202
V.1	Li-Ion DC_1V_3i.....	203
V.2	DC BUS DC_1V.....	203
V.3	DC_1V_1i.....	204
V.4	PV2 DC_1V_1i.....	205
V.5	PV3 DC_1V_1i.....	206

LISTA DE FIGURAS

Figura 1.1 – Microgrid DC integrada na rede AC.....	3
Figura 1.2 - Possíveis áreas de controlo de um edifício	6
Figura 1.3 - Demonstração de aplicação de uma rede WSN a uma micro rede. Nesta podem-se ver os barramentos AC (a vermelho), DC (a azul), os dispositivos de adaptação AC-DC, os diversos sensores WSN e respetiva representação de uma rede backhaul. As setas representam os sentidos da corrente.	9
Figura 2.1 - Comparação do alcance de diferentes tecnologias	13
Figura 3.1 - Chirp ascendente de 0 a 50 Hz e respetivo espectrograma onde se consegue ver a subida linear da frequência com o tempo (criado em Matlab)	24
Figura 3.2 - Esquema de funcionamento DSSS (esquerda) e FHSS (direita).....	25
Figura 3.3 - Largura de banda dos canais LoRa.....	26
Figura 3.4 - Chirp	27
Figura 3.5 - Espectrograma de diferentes SF Lora.....	27
Figura 3.6 – Exemplo de chip Lora, onde se mostram os 64 pontos onde pode existir o salto de frequência.....	29
Figura 3.7 - Modulação LoRa do valor binário 001010b (a vermelho). A azul, chirp base ...	30
Figura 3.8 - Excerto de uma captura de trama LoRa [93]	31
Figura 3.9 - Exemplo de duty cycle a 20%, em três canais	32
Figura 3.10- Composição da trama física LoRa [75]	32
Figura 3.11 - Excerto de captura de trama LoRa, onde se pode ver o preâmbulo e parte do payload com os dados modulados [72]	33
Figura 3.12 - ToA em função de payload e SF	34
Figura 3.13 - Esquema de link budget (adaptado de [99]).....	35
Figura 3.14 - Esquema representativo entre as diversas variáveis de um sistema LoRa (DR representa os códigos LoRaWAN)	37
Figura 3.15 - Arquitetura LoraWAN (adaptada de [108]).....	39
Figura 3.16 - Classe A LoraWAN.....	41
Figura 3.17 - Classe B LoRaWAN com exemplo de 8 ping slots entre beacons e 2 receções ping.....	42
Figura 3.18 - Esquema de transmissão e receção classe C LoRaWAN	43
Figura 3.19 - Número máximo de transmissões por hora consoante o valor de DR e tamanho do payload	46

Figura 3.20 - Estrutura da trama e composição do campo PHYPayload.....	47
Figura 3.21 - Chaves de segurança LoraWAN e aplicação das mesmas na trama	50
Figura 3.22 - Mensagens keepalive GW-NS	51
Figura 3.23 - Mensagens de troca de dados (lado esquerdo) e formatos das mensagens GWMP (lado direito)	52
Figura 4.1 - Esquema da micro rede INTA.....	55
Figura 4.2 - Arquitetura WSN proposta	59
Figura 4.3 - Consola REPL, integrada no editor de texto Atom.....	61
Figura 4.4 - Placa de expansão (esquerda), placa GPS/Acelerómetro (centro) e Lopy (direita)	62
Figura 4.5 – Antena e pigtail Lora (lado esquerdo), antena WiFi embutida (lado direito)	63
Figura 4.6 - Ligações SPI.....	64
Figura 4.7 - Placa IMST iC880A-SPI	65
Figura 4.8 - Ligações RPi3 iC880A-SPI	66
Figura 4.9 – Placa perfurada inicial (A), esquema kicad da pcb (B), placa finalizada (C)	66
Figura 4.10 - Salto no RSSI após troca de antena (A) e antena instalada no exterior do lab. (B)	160
Figura 4.11 - Gateway e NS (com antena inicial)	68
Figura 4.12 - Canais loraWAN alocados (excerto do ficheiro de configuração).....	69
Figura 4.13 - Estação meteorológica do ISE (A) e página de dados html da estação (B).....	73
Figura 4.14 – Arquitetura protocolar do sistema.....	75
Figura 4.15 - Esquema 3D (lado esquerdo) e placa protótipo da primeira versão (lado direito)	82
Figura 4.16 - Lopy Power Board Versão 2. Esboço 3D (lado esquerdo), placa finalizada sem módulo MODBUS (lado esquerdo)	83
Figura 4.17 - Exemplo de barramento MODBUS.....	85
Figura 4.18 – Medidor de energia SDM230 (A). Medidor de energia SDM530 ligado a uma Lopy Power Board V1 (B)	85
Figura 4.19 - Pré visualização 3D (A) e placa protótipo de leitura de 4 correntes (B)	88
Figura 4.20 - Pré visualização 3D (A) e placa protótipo de leitura de 1 tensão (B)	89
Figura 4.21 - Pré visualização 3D	89
Figura 4.22 - Pré visualização 3D da placa protótipo de leitura de 1 tensão e 3 correntes bidirecionais.....	90
Figura 4.23 - Flowchart genérico de funcionamento dos módulos emissores	91

Figura 5.1 - Simulação de potência recebida, na área da INTA. Resultado da simulação em CloudRF (A) e mesma simulação em Google Earth (B).....	98
Figura 5.2 - Site survey em redor do edifício principal.....	99
Figura 5.3 - Estado de receção à distância aproximada de 250 metros com DR5	99
Figura 5.4 – Site survey ao recinto (A), potência recebida simulada no local extremo do site survey (B) e perfil de transmissão do sinal simulado (C)	100
Figura 5.5 - Setup de 4 Lopy (A) e local de colocação do mesmo (B)	101
Figura 5.6 - Gráfico de mensagens recebidas por canal (esquerda) e pacotes perdidos por dia da semana (direita)	102
Figura 5.7 - LSNR e RSSI médias por dias	103
Figura 5.8 - Perda de pacotes por dia e hora, para os vários DR	104
Figura 5.9 - Perda de pacotes e RSSI/LSNR, separados por configuração e por horário, a partir das 00:00 de dia 15.....	105
Figura 5.10 - Bancada de ensaio para os módulos de tensão	107
Figura 5.11 - Todos os componentes do sistema, pronto a implementar na microrede	110
Figura 5.12 - Pormenor da instalação na micro rede INTA. Transdutores de corrente a tracejado amarelo e caixa com todos os conversores ADC (A). Módulo emissor Lopy Power Board, por cima do quadro elétrico, para conectar via Modbus a medidores de energia no interior do quadro elétrico (B)	111
Figura A.1 - Arquitetura de rede em single hop (em cima), em malha (em baixo, esquerda) e adhoc (em baixo, direita).....	127
Figura A.2 Redes homogénea com 1 salto (À direita) e heterogénea com vários saltos (esquerda)	131
Figura A.3 Constituição de nó	132
Figura C.1 - Arquitetura Loraserver.io	145
Figura I.1 - Excerto de conteúdo da base de dados Sqlite	155
Figura J.1 - Arquitetura típica MQTT	164
Figura J.2 - Publicação e subscrição de dados em “/node1/dados”.....	164
Figura J.3 - Exemplo de estrutura de tópicos, com várias hierarquias	165
Figura M.1 - Esquema de sensor de efeito Hall	169
Figura M.2 – Esquema de transdutor de efeito de hall em loop aberto (lado esquerdo) [174] e em loop fechado (lado direito) [177]	171
Figura M.3 - Gráfico de tensão de saída em função de corrente primária[174]	171
Figura M.4 - Transdutor LEM HO 10-P montado em pcb	173

Figura M.5 - Gráfico de não linearidade.....	173
Figura M.6 - Transdutor Tamura L31S050S05FS	175
Figura M.7 - Ampop em montagem diferencial.....	176
Figura M.8 - Simulação de circuito de condicionamento, 32 A. Circuito utilizado (em cima) e gráfico de tensão de saída em função da tensão de entrada (em baixo).....	178
Figura M.9 – Simulação de circuito de condicionamento, 120 A. Esquema (topo) e gráfico de tensão de saída em função de entrada (baixo)	180
Figura M.10 - Transdutor LEM LV 25-P	181
Figura M.11 - Circuito interno e montagem aconselhada do LV 25-P	181
Figura U.1 - Página inicial do servidor de rede	192
Figura U.2 - Exemplo de configurações de rede no NS	193
Figura U.3 - Vista de configuração do token, na plataforma Thingsboard	195

LISTA DE TABELAS

Tabela 3.1 - Débitos em função de SF.....	28
Tabela 3.2 – Potência e duty cycles utilizados na banda dos 868 MHz.....	31
Tabela 3.3 - SNR mínima no recetor em função do SF.....	36
Tabela 3.4 - Sensibilidade em função de SF e Bw.....	36
Tabela 3.5 - Códigos de débito DR e potência de emissão TXPower.....	44
Tabela 4.1 – Canais configurados	69
Tabela 4.2 - Lista de módulos para aplicação na micro rede INTA.....	80
Tabela 4.3 – Formato de byte de estado do nó.....	93
Tabela 5.1 - Valores totais obtidos	102
Tabela 5.2 - Valores obtidos em módulo de leitura de 4 correntes.....	106
Tabela 5.3 - Resultados dos ensaios dos módulos de medição de tensão.....	108
Tabela A.1 - Exemplos de possíveis utilizações IoT em smart grids.....	134
Tabela B.1 - Tabela Comparativa de tecnologias WSN	137
Tabela C.1 - Tabela comparativa de módulos emissores LoRaWAN.....	141
Tabela C.2 - Tabela comparativa de placas concentradoras	143
Tabela C.3 - Tabela comparativa de gateways comerciais completas	144
Tabela J.1 - Exemplo de wildcards e tópicos MQTT.....	165
Tabela L.1 - Pontos de leitura DC.....	168
Tabela L.2 - Pontos de leitura AC	168
Tabela M.1 - Valores exemplo de conversão do ADS1115.....	172
Tabela M.2 - Correntes máximas e mínimas, em função de tensão de referência utilizada [180]	174
Tabela M.3 - Resistências teóricas para diversos valores de tensão de entrada	182
Tabela M.4 - Dimensionamento de resistências do primário, em função de tensão de entrada	183
Tabela M.5 - Corrente e tensão no secundário, em função da tensão primária	184
Tabela N.1 - Tabela comparativa de transdutores de corrente.....	185
Tabela V.1 - Valores obtidos no módulo DC_1V_3i, antes da calibração	203
Tabela V.2 - Valores obtidos no módulo DC_1V_3i, após calibração	203
Tabela V.3 - Valores obtidos no módulo DC_1V, antes da calibração.....	203
Tabela V.4 - Valores obtidos no módulo DC_1V, após calibração	204

Tabela V.5 - Valores obtidos no módulo DC_1V_1i PV1, antes da calibração	204
Tabela V.6 - Valores obtidos no módulo DC_1V_1i PV1, após calibração	204
Tabela V.7 - Valores obtidos no módulo DC_1V_1i PV2, antes da calibração	205
Tabela V.8 - Valores obtidos no módulo DC_1V_1i PV2, após calibração	205
Tabela V.9 - Valores obtidos no módulo DC_1V_1i PV3, antes da calibração	206
Tabela V.10 - Valores obtidos no módulo DC_1V_1i PV3, após calibração.....	206

LISTA DE ABREVIATURAS E ACRÓNIMOS

3GPP	<i>3rd Generation Partnership Project</i>
ABP	<i>Activation By Personalization</i>
AC	<i>Alternating current (corrente alternada)</i>
ADR	<i>Adaptative Data Rate</i>
AES	<i>Advanced Encryption Standart</i>
AES128	<i>Advanced Encryption Standard 128 bits</i>
AFA	<i>Agile Frequency Adaptation</i>
AMI	<i>Advanced Metering Infrastructures</i>
AP	<i>Access Point</i>
AS	<i>Application Server</i>
BAN	<i>Body Area Network</i>
BLE	<i>Bluetooth Low Energy</i>
BPS	<i>Bits Por Segundo</i>
BPSK	<i>Binary Phase Shift Keying</i>
BR-EDR	<i>Basic Rate/Enhanced Data Rate</i>
BS	<i>Base Station</i>
CHIRP	<i>Compressed High Intensity Radar Pulse</i>
COAP	<i>COstrained Application Protocol</i>
CPS	<i>Cyber/Physical Systems</i>
CSL	<i>Coordinated Sampled Listening</i>
CSMA-CA	<i>Carrier Sense Multiple Access, Collision Avoidance</i>
CSS	<i>Chirp Spread Spectrum</i>
DBPSK	<i>Differential Binary Phase Shift Keying</i>
DC	<i>Direct Current (corrente continua)</i>
DD	<i>Decimal Degrees</i>
DER	<i>Distributed Energy Resources</i>
DG	<i>Distributed Generation</i>
DPSK	<i>Differential Phase Shift Keying</i>
DR	<i>Data Rate</i>

DS	<i>Distributed Storage</i>
DSSS	<i>Direct Sequence Spread Spectrum</i>
EC-GSM-IOT	<i>Extended Coverage Gsm lot</i>
ECC	<i>Electronic Communication Committee</i>
ERP	<i>Efective Radiated Power</i>
EV	<i>Electric Vehicles</i>
FEC	<i>Forward Error Correction</i>
FHSS	<i>Frequency Hopping Spread Spectrum</i>
FSCM	<i>Frequency Shift Chirp Modulation</i>
FSK	<i>Frequency Shift Keying</i>
FSK	<i>Frequency Shift Keying</i>
FTP	<i>File Transfer Protocol</i>
G2V	<i>Grid To Vehicle</i>
GFSK	<i>Gaussian Frequency Shift Keying</i>
GMSK	<i>Gaussian Minimum Shift Keying</i>
GSM	<i>Global System For Mobile Communications</i>
HAL	<i>Hardware Abstraction Layer</i>
HAN	<i>Home Area Network</i>
HART	<i>Highway Addressable Transducer Protocol</i>
IOT	<i>Internet Of Things</i>
IP	<i>Internet Protocol</i>
ISM	<i>Industrial, Scientific And Medical</i>
JS	<i>Join Server</i>
JSON	<i>Javascrrips Object Notation</i>
LAN	<i>Local Area Network</i>
LBT	<i>Listen Before Talk</i>
LPWAN	<i>Low Power Wide Area Networks</i>
LSNR	<i>Relação Sinal Ruído Lora</i>
LTE	<i>Long Term Evolution</i>
LTE-M	<i>Long Term Evolution/Machine</i>
LTE-NB	<i>Long Term Evolution Narrowband</i>

M2M	<i>Machine To Machine</i>
MAC	<i>Medium Access Control</i>
MAN	<i>Metropolitan Area Network</i>
MBPS	<i>Mega Bits Por Segundo</i>
MEMS	<i>Sistemas Micro Eletromec�nicos</i>
MFSK	<i>M-Ary Frequency Shift Keying</i>
MISO	<i>Master Input Slave Output</i>
MOSI	<i>Master Output Slave Input</i>
MQTT	<i>Message Queieuing Telemetry Transport</i>
MWSN	<i>Multimedia Wireless Sensor Networks</i>
NB-IOT	<i>Narrowband lot</i>
NQAM	<i>N Quadrature Amplitude Modulation</i>
NS	<i>Network Servers</i>
NSN	<i>Nanoscale Sensor Networks</i>
NTP	<i>Network Time Protocol</i>
NVS	<i>Non Volatile Storage</i>
OFDMA	<i>Orthogonal Frequency Division Multiple Access</i>
OQPSK	<i>Offset Quadrature Phase Shift Keying</i>
OTAA	<i>Over The Air Activated</i>
PAN	<i>Personal Area Network</i>
PLC	<i>Power Line Communication</i>
QAM	<i>Quadrature Amplitude Modulation</i>
QPSK	<i>Quadrature Phase Shift Keying</i>
RFID	<i>Radio Frequency Identification</i>
RIT	<i>Receiver/Initiated Transmission</i>
RPMA	<i>Random Phase Multiple Access</i>
RP-SMA	<i>Reverse Polarity SubMiniature version A</i>
RSSI	<i>Received Signal Strenght Indication</i>
SC-FDMA	<i>Single Carrier Frequency Division Multiple Access</i>
SCADA	<i>Supervisory Control and Data Acquisition</i>
SCLK	<i>Serial Clock</i>

SDR	<i>Software Defined Radio</i>
SF	<i>Spreading Factor</i>
SG	<i>Smart Grid</i>
SNR	<i>Relação Sinal Ruído, Signal To Noise Ratio</i>
SPI	<i>Serial Peripheral Interface</i>
SS	<i>Slave Select</i>
TDMA	<i>Time Division Multiple Access</i>
TOA	<i>Time On Air</i>
TWSN	<i>Terrestrial Wireless Sensor Networks</i>
UGWSN	<i>Underground Wireless Sensor Networks</i>
UNB	<i>Ultra Narrow Band</i>
UWSN	<i>Underwater Wireless Sensor Networks</i>
V2G	<i>vehicle to grid</i>
VSWR	<i>Voltage Standing Wave Ratio</i>
WAN	<i>Wide Area Networks</i>
WSN	<i>Wireless Sensor Networks</i>

1. INTRODUÇÃO

1.1 CONTEXTOS

De forma a que se entenda a necessidade em criar sistemas de monitorização energética, devemos primeiro apresentar algumas das situações que promovem estes desenvolvimentos. Os subcapítulos seguintes, resumem, sem aprofundar, os diferentes cenários que permitem enquadrar a realidade política e energética atual e que levam à busca de soluções de monitorização.

1.1.1 CONTEXTO ENERGÉTICO – DIRETIVAS E METAS PROPOSTAS

No final do século passado, a crescente preocupação ambiental levou a que se fizesse uma aposta em fontes renováveis de produção de eletricidade. O estabelecimento de metas como uma redução de 80% (em relação aos níveis de 1990) nas emissões de CO₂ em 2050 para a Europa, levou a que países, empresas e entidades públicas tivessem de procurar soluções que possibilitem essa redução. Neste contexto, a energia solar, eólica ou de biomassa sendo importantes fontes limpas de energia, que permitem emissões de CO₂ nulas, foram gradualmente adotadas por todo o mundo, potenciadas por variadas imposições políticas.

Uma vez que as cidades consomem 75% da energia gerada e produzem 80% do CO₂, assume grande importância atuar na melhoria da respetiva eficiência, em vários dos seus setores [1], em particular nas componentes de edifícios, de transportes e de produção elétrica [2]. De forma a promover a substituição gradual de equipamentos antiquados e já pouco eficientes, a legislação europeia *Energy Efficiency Directive* (2012/27/EU) definiu que até 2020 as poupanças de energia deveriam atingir a meta de 20%, face ao estimado para essa mesma data se nenhuma medida forem tomadas. Ao mesmo tempo, veio também promover medidas para forçar entidades públicas e privadas a adotar produtos e serviços eficientes assim como reduzir os consumos nos edifícios.

Nos países mediterrânicos, em particular, os quais dispõem de muitas horas de sol anuais, pode-se aproveitar este facto para ajudar a atingir os objetivos de redução de CO₂, através da produção fotovoltaica. Especificamente em Portugal o número médio de horas de sol por ano

varia entre as 2200 e as 3000, consoante a região [3]. Apesar disso, apenas 1,6% da energia elétrica produzida em Portugal provem de fonte solar [4]. Ao mesmo tempo, a localização geográfica junto ao mar, faz com que exista vento com alguma regularidade e intensidade. Consegue-se assim obter um rácio de 21,6%, sendo esta a percentagem do total da produção elétrica nacional a partir de fontes renováveis [5][4]. Conseguimos inclusive produzir o suficiente para alimentar o país durante dias, utilizando apenas fontes renováveis [6]. No entanto, e uma vez que a grande maioria da energia produzida é consumida no momento da sua geração, há períodos da noite em Portugal em que a energia produzida por fontes eólicas supera a energia consumida, o que conduz a desperdícios. Neste contexto importa implementar métodos que permitam ajustar o consumo aos ciclos de produção.

1.1.2 CONTEXTO DAS REDES ELÉTRICAS

Tradicionalmente a geração e posterior distribuição elétrica suporta-se numa arquitetura composta por centrais de produção térmica e hídrica, linhas de transmissão em alta/média tensão e posterior distribuição em baixa tensão ao consumidor final.

Mais recentemente, a introdução das novas fontes de energia renovável, em particular a eólica e a fotovoltaica, veio criar novos desafios que decorrem, em parte, da sua natureza intermitente, mas também do facto da sua curva de geração não ser controlável, nem sempre se adequando com as curvas de consumo. Para compensar essas duas características, para além das centrais clássicas, passou a recorrer-se também a centrais de bombagem, que permitem armazenar o excesso de produção sobre a forma de energia potencial, sempre que a produção excede o consumo, injetando depois essa energia na rede sempre que o consumo excede a produção renovável.

Se por um lado a um nível macro o problema ficou parcialmente resolvido, o mesmo não se pode dizer relativamente à introdução de fontes de energia renovável que se veio a verificar posteriormente, dita de geração distribuída (*distributed generation*, DG) [7]. Neste contexto, a legislação nacional e internacional veio permitir a dita microprodução, em que os anteriores consumidores passaram a poder também produzir, injetando a energia por eles gerada na rede elétrica convencional.

Surgem aqui os conceitos de *microgrid* e *nanogrid*, sendo estas consideradas pequenas redes de baixa tensão/potência (isoladas ou não da rede principal) que podem ser geridas de forma independente [2]. De uma forma geral as micro redes podem abranger um ou vários

edifícios, numa área geográfica limitada. Nestas redes, apesar da distribuição ser normalmente feita em corrente alternada, estão em estudo outras soluções que passam pela criação de circuitos em corrente contínua, ou mesmo um misto das duas, tal como apresentado na Figura 1.1.

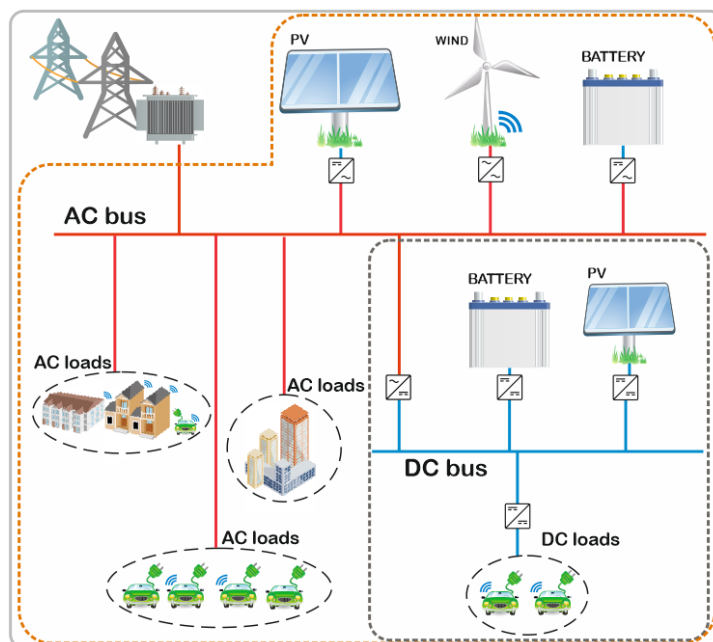


Figura 1.1 – Microgrid DC integrada na rede AC

Se por um lado a integração distribuída de novas fontes renováveis é importante, o carácter intermitente dessas fontes traz algumas complicações ao normal funcionamento das redes elétricas, uma vez que, em termos instantâneos é necessário manter o equilíbrio entre os níveis de produção e de consumo, o que por sua vez se repercute nos níveis de tensão. A rede tradicional apenas foi projetada considerando o fluir de corrente num sentido, mas esta nova realidade faz com que em alguns troços a corrente possa passar a fluir nos dois sentidos. Tal pode trazer problemas de qualidade (como sobre e subtensões), para além de segurança no fornecimento de energia, devido à intermitência de todas estas fontes [2]. Uma forma de minimizar estes problemas passa pela introdução de sistemas de armazenamento, que para além de permitirem acumular energia para uma utilização posterior, garantem também o correto balanceamento da rede ao fornecer energia à rede que se encontra a montante, quando necessário.

Neste âmbito, os sistemas de armazenamento baseados em baterias são maioritariamente considerados como uma solução que poderá vir a ser utilizada em *micro* redes (não tanto na rede geral, embora já tenham surgido algumas soluções para esta). Os sistemas mais comuns de armazenamento passam pelas baterias eletroquímicas (de ácido ou iões de lítio), mas existem também propostas de baterias de fluxo ou sistemas baseados em ar comprimido. No

seu conjunto, à semelhança das fontes renováveis, esses sistemas de acumulação podem estar espalhados pela rede, num esquema de armazenamento distribuído (*distributed storage*, DS). Neste contexto pode afirmar-se que, quer a geração, quer o armazenamento distribuído, transformaram a rede elétrica num enorme sistema de energia distribuída (*distributed energy resources*, DER) [7][8]. Para além das baterias, e uma vez que as mesmas têm sempre um tempo de vida limitado, importa encontrar soluções que permitam coordenar produção e consumo.

Ao nível das micro redes, não sendo possível controlar a geração, têm vindo a considerar-se soluções que permitam adaptar os níveis de consumo em função da geração, naquilo que é denominado de *demand response*. A aquisição dos dados necessários para otimizar a procura de uma forma lata, em toda a rede elétrica, é um processo complicado, devido a que cada elemento participante na DER pode ser gerido por entidades independentes, não existindo troca de comunicações entre estas. Ao mesmo tempo, existe uma diversidade de elementos que torna a rede geral pouco homogénea a nível de equipamentos e arquitetura. Já dentro de uma micro rede o processo é mais simples e faz todo o sentido implementar estes mecanismos, seja por motivos económicos, seja por garantir a funcionalidade correta da mesma.

A modernização da infraestrutura elétrica existente, de forma a considerar todas estas novas realidades, tem vindo a fazer uso de novas tecnologias de comunicação, com o objetivo de tornar a rede mais segura, fiável, eficiente e dotando-a de alguma “inteligência”. Esta é a base das denominadas *Smart Grids* (SG), um conceito que visa integrar as tecnologias de informação e comunicação na rede de produção e distribuição elétrica [9]. Apesar do conceito *Smart Grid* ser de fácil perceção, a sua definição não é muito exata. No entanto espera-se que a evolução das *Smart Grids* possa resultar em [8][10]:

- Novos produtos, serviços e mercados, diferenciando distintos perfis de consumo, como por exemplo tarifas dinâmicas;
- Acomodar novas soluções de geração e armazenamento distribuídas (no contexto dos DER);
- Tornar os consumidores como parte integrante da rede, permitindo que estes ajustem o seu perfil de utilização consoante a necessidade da rede, permitindo por exemplo agendamento e otimização do uso de energia;
- Permitir diferentes tipos de energia (contínua, alternada, diferentes níveis de tensão) na mesma rede, consoante as necessidades dos diferentes utilizadores;
- Automatização das operações de rede;

- Operar de forma mais resiliente em situações de perturbações externas na rede (desastres naturais por exemplo), permitindo a deteção e recuperação das mesmas;
- Otimizar a produção e as operações de manutenção, assim como o balancear de carga, diminuindo perdas no sistema.

A concretização das *Smart Grids* depende de alguns conceitos em fase de desenvolvimento, os quais analisaremos em seguida.

1.1.3 CONTEXTO DOS DISPOSITIVOS INTERNET

As *Smart Grids* desenvolveram-se num contexto de massificação do acesso à Internet e do surgimento de conceitos como a Internet das Coisas (*Internet of things*, IoT). O IoT refere-se à possibilidade de qualquer dispositivo poder estar conectado à Internet (ou a uma rede *Internet Protocol*, IP) e, por essa via, comunicar com pessoas e/ou máquinas. O baixo custo de fabrico desses dispositivos, associado a uma grande ubiquidade no acesso à Internet garantem hoje o desenvolvimento destas tecnologias, que têm vindo a englobar as mais variadas áreas de aplicação. As evoluções verificadas nos serviços de armazenamento, análise de dados e nas plataformas de apresentação desses dados aos utilizadores, contribuíram também para o sucesso do IoT. A sua adoção em redes elétricas tem sido, no entanto, lenta, apesar de ser um dos elementos chave das *Smart Grids* [1].

Neste âmbito, uma das aplicações de dispositivos IoT resulta da combinação entre equipamentos de medição remota e em tempo real (*advanced metering infrastructures*, AMI), com algoritmos que analisam esses dados, na Internet. Assim, a deteção de anomalias na rede, a medição de desgastes e performance dos equipamentos ou a estimação de falhas de equipamento baseado em medições regulares, constituem exemplos de aplicações do IoT em *Smart Grids*. No entanto, a deteção de consumos anómalos requer medir outras variáveis que contribuem para esse consumo como a temperatura ambiente, a intensidade solar ou ocupação. O leque de dispositivos IoT existentes permitem hoje efetuar essas medições.

Para além do consumo, muitos equipamentos de automação de iluminação, ventilação e aquecimento ou ar condicionado, já possibilitam hoje ligação à Internet. Essa ligação suporta não só a monitorização, mas também a respetiva atuação remota, assim como a integração com plataformas de *software* na *cloud*.

Por último, a junção de todos estes elementos (apresentados na Figura 1.2), quando combinada com os algoritmos de otimização e de *machine learning*, vem hoje conferir uma

maior inteligência aos edifícios, que de outra forma eram passivos, fazendo com que estes tomem decisões autónomas, sem intervenção humana, de forma a maximizar a sua eficiência.

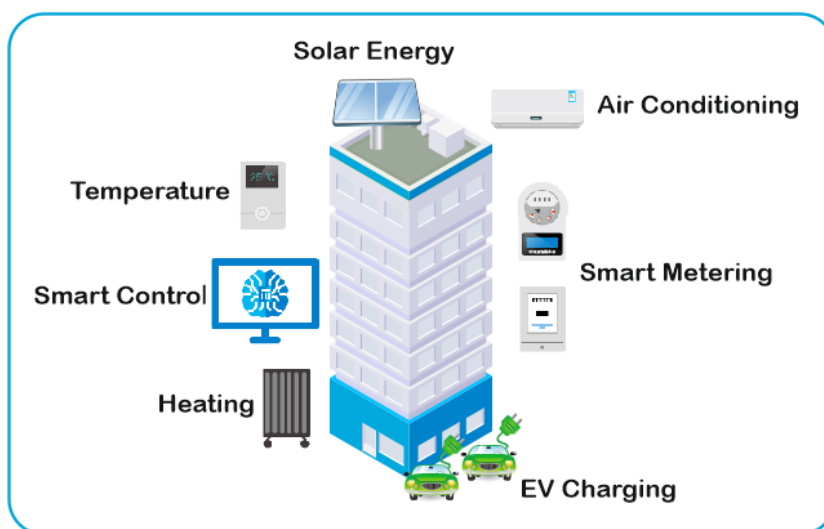


Figura 1.2 - Possíveis áreas de controlo de um edifício

1.2 PROBLEMA

Gerir de forma eficiente os recursos elétricos é uma tarefa difícil, que requer a integração de conhecimentos e tecnologias de diversos domínios, que passam pelo IoT, as redes e equipamentos elétricos incluindo veículos elétricos, as fontes de energia renovável e a aprendizagem máquina, combinada com a otimização.

Se por um lado se pretende aumentar a produção recorrendo a fontes de energia renovável solar e eólica, também é verdade que a geração oriunda dessas fontes é considerada intermitente. Estabilizar esta intermitência passa pela utilização de sistemas de armazenamento em baterias, mas também pela correta gestão de cargas e consumos, com o objetivo de utilizar eficientemente a energia gerada, maximizando os investimentos em fontes renováveis e minimizando os custos. Para poder tomar decisões de gestão da rede nomeadamente sobre o armazenamento, consumo ou exportação energética, é necessário adquirir dados da rede, monitorizar alguns parâmetros ambientais, estruturais e da própria rede elétrica. Esta decisão tem de considerar que a produção e consumo não mantêm a mesma potência ao longo do tempo, podendo ser necessário ajustar a utilização da rede em termos de consumos. Este ajuste à carga de rede, quando feito em tempo real e baseado apenas no estado atual de produção e consumo, pode levar a potenciais falhas do serviço, algo que nunca é

desejável numa rede elétrica. Uma solução consiste em fazer previsão futura do consumo e produção, baseada em dados atuais e aprendendo com os dados históricos, permitindo assim antecipar necessidades e atuar de forma rápida e eficiente, onde necessário.

De forma a que todos estes mecanismos funcionem, é necessário obter informação periodicamente, sobre os diversos componentes da rede elétrica. Esta obtenção de dados pode assentar numa rede de sensores sem fios (*wireless sensor network*, WSN), espalhados pelos pontos chave de medição. Se na produção, transmissão e distribuição elétrica existe um problema de interferência eletromagnética na utilização de WSN, em micro redes elétricas ou edifícios pode-se aplicar esta solução sem problemas [9]. Em relação às áreas a cobrir, muitas das soluções existentes de redes de sensores sem fios não suportam grandes alcances entre dispositivos, sendo necessário por isso um dimensionamento adequado da localização dos pontos de medição. Nestas, é necessário garantir a cobertura da rede, sendo que a localização do dispositivo dependerá essencialmente desta cobertura. Por vezes pode ocorrer que estes pontos se encontrem geograficamente afastados ou sejam móveis, tornando-se ainda mais importante garantir este alcance. No entanto, a maioria das tecnologias, apenas permite umas dezenas, ou poucas centenas de metros de alcance, limitando assim a utilização dos dispositivos.

1.3 OBJETIVOS DA DISSERTAÇÃO

Esta dissertação apresenta o desenvolvimento de um sistema de monitorização de variáveis elétricas e ambientais, utilizando as mais recentes tecnologias de redes de sensores sem fios de longo alcance e baixa potência (*Low Power Wide Area Networks*, LPWAN). O sistema desenvolvido servirá para a posterior criação de uma micro rede “inteligente”.

A funcionalidade base do sistema consistirá na medição de correntes e tensões contínuas ou alternadas, provenientes de geração fotovoltaica ou eólica, assim como a medição do consumo e estado de armazenamento da energia. Para estas medições, serão desenvolvidos alguns sensores de corrente e tensão. Adicionalmente, o sistema deverá permitir obter dados meteorológicos do local de geração, assim como outros dados relevantes do local de implementação.

Para além da obtenção de dados, o sistema deverá garantir a sua transmissão, processamento, armazenamento e apresentação aos utilizadores, tudo isto em tempo quase

real. Espera-se também que o sistema permita efetuar alguma atuação remota nos nós, sendo esta desencadeada de forma autónoma por algum tipo de algoritmos de escalonamento, previsão e deteção de anomalias, os quais não serão desenvolvidos nesta tese, deixando apenas o sistema pronto para este mecanismo. Desta forma, será possível dotar o sistema de alguma “inteligência”, e atuar sobre algumas cargas da rede, de forma autónoma. Uma das aplicações pretendidas será o agendamento de carregamentos de veículos elétricos.

Esta tese de mestrado enquadra-se no âmbito do projeto AGERAR (Armazenamento e gestão de energias renováveis em aplicações comerciais e residenciais, 0076_AGERAR_6_E), financiado pela União Europeia, através do programa FEDER (Fundo Europeu de Desenvolvimento Regional), no âmbito do INTERREG V-A Espanha-Portugal (POCTEP). Para além da Universidade do Algarve, este projeto inclui diversos parceiros ibéricos nomeadamente a Universidade de Évora, Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial (INEGI), Instituto Tecnológico de Galícia (ITG), Agencia Andaluza de la Energia, Agencia Regional de Energia e Ambiente do Algarve, Instituto Nacional de Técnica Aeroespacial (INTA) e a Universidade de Sevilha na qualidade de parceiro líder. O objetivo deste projeto consiste em desenvolver e avaliar soluções técnicas para promover a eficiência energética em micro redes elétricas residenciais e comerciais, de forma a otimizar o usos de energias renováveis e sistemas de armazenamento de energia. Parte das soluções encontradas serão implementadas numa micro rede experimental existente nas instalações da INTA. Devido às particularidades desta micro rede, a qual contém todos os elementos essenciais de uma DER e que será descrita ao longo da tese, estabeleceram-se algumas características que o sistema deverá cumprir, nomeadamente:

- Bom alcance, de forma a cobrir o recinto da instalação e permitir monitorizar os veículos automóveis elétricos e híbridos a hidrogénio existentes. O recinto apresenta um raio de sensivelmente 1 quilómetro;
- Custo acessível, de forma a que economicamente se torne viável a aquisição e utilização dos equipamentos;
- Modular, de forma a que se apliquem módulos para as diferentes situações. Cada sensor desenvolvido deverá ser aplicável noutra situação idêntica para a qual foi projetado (p. ex.: um sensor DC para medir a tensão e corrente proveniente de painéis solares, deverá funcionar em qualquer local, desde que dentro dos mesmos valores nominais para os quais foi projetado);
- *Open source*, que nos dará flexibilidade para alterações nos equipamentos;

- Privado, uma vez que as instalações INTA são de natureza militar, pretende-se que a rede não seja pública nem utilize operadores comerciais. Conseguimos ainda com esta solução reduzir o custo de manutenção;
- Seguro, sendo que os dados deverão ser encriptados;
- Escalável, permitindo a introdução de mais nós ou pontos de medida sem problemas.

A aplicação nas instalações INTA permitirá demonstrar o funcionamento de uma rede WSN LPWAN, para monitorizar todas as situações comuns de uma micro rede elétrica, que incluam geração solar e eólica, as interligações entre componentes de rede AC e DC, o armazenamento de energia e as cargas variáveis, conforme exemplifica a Figura 1.3. Esta figura representa um esquema genérico de uma micro rede, com componentes AC, DC, geração renovável e armazenamento distribuídos, e uma rede de sensores WSN implementada nos locais importantes a medir e com ligação a uma rede IP/Internet. Todos estes casos existem na micro rede INTA, o que permitirá simular variados cenários. No entanto, alguns dos componentes do nosso sistema, não serão utilizados nesta rede (estação meteorológica ou mecanismos de gestão de cargas), uma vez que esses componentes serão desenvolvidos por outros parceiros.

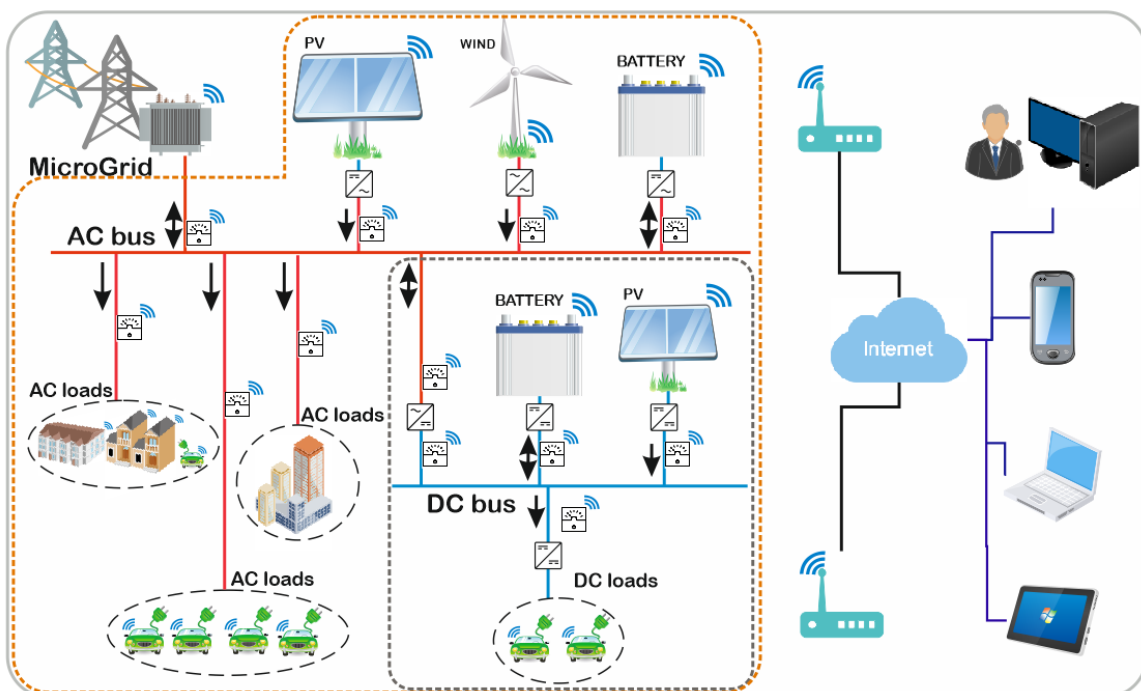


Figura 1.3 - Demonstração de aplicação de uma rede WSN a uma micro rede. Nesta podem-se ver os barramentos AC (a vermelho), DC (a azul), os dispositivos de adaptação AC-DC, os diversos sensores WSN e respetiva representação de uma rede backhaul. As setas representam os sentidos da corrente.

Por fim pretende-se que os sistemas desenvolvidos neste contexto possam ser replicáveis noutras micro redes.

1.4 ESTRUTURA DA TESE

Depois de neste capítulo se terem introduzido os contextos base e resumido o problema e a solução que se pretende implementar, a restante tese encontra-se estruturada da seguinte forma.

O capítulo 2 descreve o estado da arte, analisado no início do desenvolvimento da tese, refletindo também a disponibilidade de tecnologias de redes sem fios existentes. Apresenta-se também a comparação e seleção da tecnologia, com base nos requisitos impostos pela micro rede a monitorizar.

O capítulo 3 descreve a tecnologia WSN escolhida (LoRaWAN) com mais detalhe, explicando os conceitos base e o funcionamento da mesma, assim como as necessidades particulares desta.

O capítulo 4 descreve a conceção e o desenvolvimento do sistema e dos seus componentes. Descrevem-se também alguns dos componentes escolhidos para a criação da rede de sensores. Este desenvolvimento é baseado na micro rede INTA, pelo que se descreve com algum detalhe esta rede.

No capítulo 5 descrevem-se os ensaios efetuados aos diversos componentes. Mostra-se também um pouco do trabalho de instalação, a decorrer.

No capítulo 6 tiram-se as conclusões dos resultados obtidos, no decorrer da implementação prática do sistema, assim como, possíveis melhorias ou modificações para trabalho futuro. Referem-se também, algumas publicações desenvolvidas no decorrer dos trabalhos.

No final, encontram-se vários Apêndices que contém informações técnicas dos diversos componentes, assim como, códigos fonte relevantes de todo o trabalho.

2. ESTADO DA ARTE

Neste capítulo analisam-se várias tecnologias de rede de sensores sem fios disponíveis no mercado, algumas com bastantes anos e provas dadas, outras bem mais recentes. Ao longo deste capítulo serão apresentados os conceitos e tecnologias existentes assim como uma análise baseada nas necessidades do nosso sistema.

2.1 BREVE INTRODUÇÃO ÀS REDES DE SENSORES SEM FIOS

Uma rede de sensores sem fios (*wireless sensor networks* - WSN) consiste numa série de dispositivos autónomos, designados de nós sensores, distribuídos por uma determinada localização geográfica limitada e providos de capacidades de adquirir informação do meio ao seu redor, processar essa informação e enviá-la através de um canal sem fios [11]. Os desenvolvimentos dos últimos 20 anos na miniaturização da eletrónica e em tecnologias de informação, vieram permitir que se possa hoje encontrar *hardware* (sensores, microcontroladores e transdutores) capaz de ler determinadas variáveis ambientais ou elétricas (por exemplo temperatura, tensão elétrica, entre outros) executando *software* eficiente, capaz de processar a informação adquirida nestes pequenos dispositivos. Os avanços no domínio das telecomunicações permitiram a facilidade na troca de dados entre dispositivos e/ou o envio para servidores locais ou na *cloud*, através da Internet. O custo baixo e a alta disponibilidade de soluções, garantiu a elevada adoção destes sistemas e o sucesso das chamadas redes de sensores sem fios. A tal ponto que se estima que em 2020 existam mais de 50 biliões de dispositivos conectados e um mercado de 14,4 triliões de dólares [1].

Em seguida analisaremos as principais redes sem fios da atualidade, aplicáveis às redes energéticas inteligentes.

No Apêndice A encontra-se uma breve descrição de alguns conceitos chave que se utilizaram para efetuar a seguinte análise.

2.2 ANÁLISE DAS TECNOLOGIAS WSN EXISTENTES

Para analisarmos as características técnicas de cada tecnologia, necessitamos primeiro de definir quais as especificações que se pretendem cumprir. Existem algumas características a ter em conta nas WSN, conforme descrito no Apêndice A. Um dos elementos chave é o alcance. Por experiência de trabalhos anteriores, desenvolvidos na universidade, já era do conhecimento que muitas das tecnologias apresentam problemas de alcance, mesmo dentro de um único edifício. Para a implementação na INTA queríamos garantir comunicação com dispositivos móveis e alcances de pelo menos 1 quilómetro. Ao mesmo tempo, sendo as instalações da INTA militares, precisaríamos de uma rede privada com os dados encriptados para garantir a proteção dos mesmos. Queríamos também uma solução com um custo aceitável, mas que fosse programável e configurável de forma modular, escalável, preferencialmente *open source* e bidirecional.

Com isto em mente analisaram-se as tecnologias mais comuns no mercado, de forma a perceber as características de cada uma e o respetivo estado da arte. Existem um grande número de soluções, concorrendo entre si e cada uma apresentando diferentes vantagens, adaptadas a diferentes necessidades. Muitas das tecnologias podem funcionar bem num contexto e mal noutra, sendo que não há uma solução global para todos os casos. As necessidades da implementação ditarão a tecnologia a escolher. Em seguida não se irá aprofundar tecnicamente cada uma das soluções, mas sim descrever parâmetros técnicos que as permitam comparar, como: (1) Alcance teórico em linha de vista e em ambiente urbano; (2) Débitos binários máximos no *uplink* e *downlink* e capacidade de tráfego nos dois sentidos; (3) Topologias e arquiteturas físicas e/ou protocolares do sistema; (4) Tamanho máximo do *payload* de dados permitido; (5) Número de dispositivos permitidos na rede; (6) Duração de baterias e eventual consumo energético do mesmo; (7) Banda de frequências e canais utilizados; (8) Segurança das transmissões; (9) Disponibilidade no mercado, custo de aquisição e de funcionamento; (10) Outras características particulares da tecnologia, que possam ser significativas na performance da mesma.

Tendo como base o alcance, a Figura 2.1 classifica as tecnologias consoante o seu alcance teórico previsto, considerando curtas distâncias como um alcance até 250 metros, distância média entre 250 a 1000 metros e longa distância acima de 1 km. Embora se necessite de um alcance máximo de 1000 metros para os dispositivos móveis, grande parte da rede ficará fixa num raio de 50 metros, sendo por isso necessária a análise das soluções de alcances mais

curtos. De seguida resume-se a análise efetuada a cada tecnologia, focando nos parâmetros anteriores.

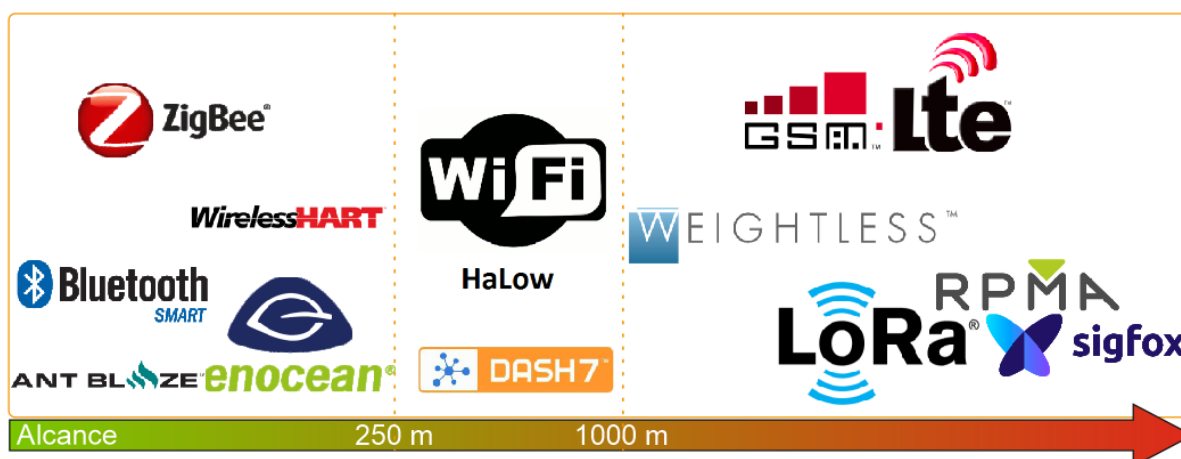


Figura 2.1 - Comparação do alcance de diferentes tecnologias

2.2.1 WSN PARA CURTAS DISTÂNCIAS

Uma das tecnologias WSN mais conhecidas é o Bluetooth, existente desde os anos 90, sendo por isso uma tecnologia bastante desenvolvida, disponível e de baixos custos. Existe em duas variantes, consoante a aplicação seja para dispositivos de áudio (*Basic Rate/Enhanced Data Rate*, BR/EDR) ou para dispositivos de baixo consumo energético (*Bluetooth low energy*, BLE). A versão mais atual é a 5, desenvolvida especialmente para baixos consumos energéticos e aplicações IoT. Nesta versão foram aplicadas técnicas de deteção e correção de erros, assim como se diminuíram os tempos de transmissão em relação à versão 4, permitindo desta forma débitos de até 2 Mbps (Mega *bits* por segundo) [12][13]. Os alcances teóricos variam entre 200 metros em linha de vista e 40 metros em ambiente urbano, com uma potência de emissão máxima de 20 dBm. O tamanho dos *payloads* varia entre 31 e 255 *bytes*, com possibilidade de encriptação AES-128, sendo as ligações bidirecionais em modo *master/slave* mas apenas entre dois dispositivos. Opera na banda livre dos 2,4 GHz ISM (*Industrial, scientific and medical*, ISM) com 79 canais de 1 MHz (ou 20 de 2 MHz) utilizando uma técnica de espalhamento de espectro por saltos de frequência (*frequency hopping spread spectrum*, FHSS) com modulações *differential phase shift keying* (DPSK) [14][15][16][17][18]. A disponibilidade de material é grande e o custo baixo. Foi publicada em 2017 uma especificação que define uma arquitetura em malha (*Bluetooth mesh*) utilizando esta tecnologia. Esta permite maiores alcances, à custa de maior gasto energético. No entanto, é apenas definida em

software, sendo que irá funcionar em qualquer *hardware* de versão superior à 4 [19][20][21][22].

A tecnologia ANT, proprietária da empresa Dynastream, foi desenvolvida inicialmente para aplicações desportivas, tendo evoluído para uma solução IoT (ANT Blaze). Esta última garante redes fiáveis em ambientes que exijam facilidade de instalação e manutenção, permitindo assim aplicações de controlo e monitorização. Consegue alcances de 30 metros e permite ligar até 500 nós em diversas topologias (*Peer to Peer*, estrela ou malha) com débitos de até 60 kbps. O ANT trabalha na banda 2,4 GHz ISM, com 79 canais de 1 MHz modulados em *Gaussian frequency shift keying* (GFSK). Garante funcionalidades de *broadcast*, *acknowledgement* e funciona sem problemas onde existam redes Wi-Fi e Bluetooth, a trabalhar nas mesmas frequências. A segurança dos dados é garantida com encriptação AES-128 (*Advanced encryption standart*) [23].

O ZigBee é uma tecnologia de baixo consumo suportada na norma IEEE802.15.4, que define uma *stack* protocolar completa. Geralmente usa-se numa rede em malha, sendo que um pacote tem de passar por vários dispositivos até atingir a *gateway*, exigindo assim a utilização de algoritmos de *routing*. Funciona nas bandas 2,4 GHz ou 868 MHz ISM, usando modulações *offset quadrature phase shift keying* (OQPSK) ou *binary phase shift keying* (BPSK) em espalhamento de espectro direto (DSSS). Permite 16 canais de 2MHz onde aplica técnicas de *Carrier Sense Multiple Access* com *Collision Avoidance* (CSMA-CA), de forma a detetar e minimizar colisões e interferências neste espectro tão utilizado. Consegue alcances de 150 metros e débitos até 250 kbps (2,4GHz) ou 20 kbps (868 MHz), numa potência de emissão até 20 dBm. Existe um grande número de módulos no mercado oriundos de dezenas de fabricantes, com preços a partir dos 15 euros, sendo uma tecnologia amplamente implementada [12][24][25][26][27].

O WirelessHART é uma tecnologia proprietária baseada no protocolo de automação industrial *Highway addressable transducer protocol* (HART) e criada essencialmente para ambientes industriais. Adaptou o HART para utilizar o canal rádio em vez de um meio cablado [26][28]. Garante alcances de até 250 metros em campo aberto, com débitos de 250 kbps e pacotes de 127 bytes, permitindo a implementação de topologias malha, estrela ou estrela-malha. Os dados estão encriptados utilizando AES de 128 bits. A camada física é baseada na norma IEEE 802.15.4 e utiliza a banda 2,4 GHz ISM. Define um esquema de controlo de acesso do tipo *Time Division Multiple Access* (TDMA), o que exige grande sincronismo temporal entre nós, assim como *Frequency Hopping Spread Spectrum* (FHSS), dividindo a

banda em 15 canais de 2 MHz e modulando os sinais em OQPSK. A maioria das soluções existentes são para indústrias, com custo elevado e proprietárias, mas estando preparadas para cumprir tarefas específicas com a máxima fiabilidade [29][30][31].

O EnOcean, tecnologia proprietária, foca-se em soluções de domótica e aproveitamento de energia, convertendo energia mecânica, térmica ou eletromagnética em energia elétrica. Desta forma dispensa o uso de baterias nos seus sensores. Está normalizada (ISO/IEC 14543-3-10) e a *stack* protocolar está definida pela EnOcean Alliance [32]. Os débitos são de 125 kbps com pacotes típicos de 14 *bytes*, mas que podem chegar aos 255. Opera na banda livre ISM dos 868 MHz (sujeita ao regulamento da mesma), mais especificamente na frequência 868,3 MHz, e apenas num único canal, modulando os sinais em *Frequency Shift Keying* (FSK). As topologias utilizadas são estrela, malha e ponto a ponto, devendo-se utilizar *gateways* com alimentação externa e que permitem a intercomunicação com outros protocolos utilizados em domótica [33][34][35]. Existem vários módulos comerciais para captura de energia, *transceivers* para comunicações e sensores utilizando a tecnologia, e a preços acessíveis.

2.2.2 WSN PARA MÉDIAS DISTÂNCIAS

Para distâncias superiores a 250 metros e que alcancem até 1000 metros temos as soluções Wi-Fi Halow e Dash7, apresentadas de seguida.

De forma a aumentar o alcance e diminuir o consumo da rede Wi-Fi, o IEEE propôs uma alteração técnica desta, o que resultou no Wi-Fi Halow (802.11ah). Embora ainda não exista material comercialmente disponível, estima-se que os alcances rondem os 1000 metros, com débitos entre 150 kbps e 346,6 Mbps e pacotes que podem conter até 7991 *bytes*. Como no Wi-Fi tradicional, cada dispositivo liga-se a um ponto de acesso (*access point*, AP), o qual permite ligar até 8191 dispositivos. A banda de frequências utilizadas varia mundialmente, embora esteja baseada nos 900 MHz, com canais de 1, 2, 4, 8 ou 16 MHz, modulados em BPSK, *quadrature phase shift keying* (QPSK) ou *n-quadrature amplitude modulation* (n-QAM) [36][37][38][39][40].

O DASH7 é um protocolo aberto que pretende garantir comunicações bidirecionais utilizando tecnologia de identificação por rádio frequência (*radio frequency identification*, RFID) ativa, em bandas sub-GHz. Define toda a comunicação *end-to-end*, num esquema pergunta/resposta, e utiliza uma *stack open source* OSS-7 [12][41]. Os débitos variam entre 9, 6, 55.5 ou 166.6 kbps, com baixas latências e alcances normais de 1 km, mas que podem

chegar aos 5km. Utiliza a banda dos 868 MHz ISM, em canais de 25 ou 200 kHz modulados com 2GFSK [42][43][44]. O material disponível é muito limitado, mas é possível implementar a tecnologia em vários *transceivers* rádios disponíveis [44].

2.2.3 WSN PARA LONGAS DISTÂNCIAS

Para distâncias superiores a 1000 metros existem bastantes soluções no mercado, algumas baseadas na rede móvel tradicional.

Para potencializar aplicações IoT, a 3GPP (*3rd Generation Partnership Project*) adaptou a estrutura da rede celular 4G *Long Term Evolution* (LTE), já amplamente disponível, de forma a permitir a utilização de dispositivos terminais de baixos débitos, grande autonomia e custo reduzido. Esta traduziu-se apenas numa alteração de *software* na rede, e ficou conhecida como LTE-M (*Long Term Evolution-Machine*) [45][46]. O débito é de 1 Mbps bidirecional, numa topologia idêntica à da rede LTE normal, exigindo assim a associação de um operador. A poupança energética resulta da redução de potência de emissão e receção descontinua de dados, permitindo assim que os dispositivos terminais possam ter autonomias de até 10 anos. Permite o *roaming* e tem latências de milissegundos [12][47][48][49][50]. Já existem operadores de telecomunicações e módulos de comunicação de diversos fabricantes, com custos acessíveis.

As alterações efetuadas na LTE-M não foram suficientes, pelo que a 3GPP adaptou ainda mais a estrutura LTE, criando a tecnologia NarrowBand-iot (NB-IoT). Esta foi especificada na *Release* 13 e ficou conhecida como LTE-NB (*Long term evolution narrowband*). O principal ponto de desenvolvimento consistiu na redução energética, simplificando a estrutura dos modems de forma a minimizar a energia consumida, e tornar o dispositivo mais barato. Mais uma vez, a nível de rede, basta uma atualização de *software* na rede LTE [13][45][46]. Com esta tecnologia conseguem-se alcances até 22 km, débitos máximos de 250 kbps em *downlink*, 170 kbps em *uplink*, e 200000 subscritores por portadora utilizada. Irá funcionar nas mesmas bandas do LTE, num canal de 200 kHz modulado em QPSK e BPSK. Este canal é alocado dentro da banda LTE, na banda de guarda ou numa portadora independente. A topologia é a mesma de uma rede móvel normal, exigindo a existência de um prestador de serviço associado [48][51][52][53][54]. Não permite *roaming* sendo por isso ideal para dispositivos fixos. A latência será na ordem dos segundos [55]. Já existem bastantes módulos de comunicações e operadores a fornecer o serviço.

Apesar de mundialmente muitos operadores já terem adotado o LTE, existem zonas onde apenas existe a anterior rede *Global System for mobile Communications* (GSM). De forma a permitir uma globalização de aplicações IoT, a *Release 13* da 3GPP veio incluir a tecnologia *Extended Coverage GSM IoT* (EC-GSM-IoT). A ideia foi simplificar a rede de forma a ganhar capacidade e autonomia dos dispositivos, sem alterar as infraestruturas já existente. Desta forma permite o funcionamento paralelamente ao GSM tradicional, bastando um *upgrade* de *software*. Foram introduzidos novos esquemas de codificação, novos canais lógicos de controlo e dados, a possibilidade de manter os terminais mais tempo desligados, e a possibilidade de envio de dados em rajada [13][45][56]. As bandas de frequência utilizadas serão as dos 900 MHz do GSM, utilizando um canal de 200 kHz. Os débitos serão de 70 ou 240 kbps, consoante a modulação seja GSMK ou 8PSK. As potências máximas de emissão serão de 23 dBm [47][57]. Neste momento não há grande desenvolvimento neste tipo de solução, pelo que ainda não se encontram equipamentos no mercado.

O SigFox é um sistema desenvolvido pela empresa francesa com o mesmo nome, com uma *stack* proprietária toda desenvolvida pela empresa e *hardware* desenvolvido apenas pelos parceiros da mesma. A Sigfox criou parcerias com operadoras de telecomunicações tradicionais, adicionando nas células destas, as suas *base stations* (BS), com vista a criar uma rede global. Tem uma arquitetura que obriga os dados a passar pelos seus servidores, onde são depois disponibilizados às aplicações de cliente, exigindo uma subscrição de serviço [12][13][58][59]. Os alcances podem chegar a 50 km em campo aberto ou 10 km em ambiente urbano, com um débito de apenas 100 bps. O tamanho dos pacotes é de apenas 12 bytes em *uplink* e 8 em *downlink*. O tempo de transmissão de um pacote é na ordem de segundos, limitando o máximo de mensagens por dia a 140 em *uplink* e 4 em *downlink*. Esta limitação advém da regulamentação europeia que rege os canais ISM e que limita potência de emissão e tempo de ocupação do espectro, sendo que esta tecnologia opera nos 868 MHz. Usa um canal de 192 kHz, dividido em múltiplos sub canais de apenas 100 Hz, designados de *ultra narrow band* (UNB), e onde se transmitem os dados. São modulados no *uplink* com BPSK e no *downlink* com GFSK. O sistema aplica um esquema de saltos de frequência para minimizar as colisões dentro do canal [60][61][62]. A diminuição do tamanho dos dados e simplificação dos cabeçalhos das tramas, tornou todo o protocolo leve de forma a não exigir grande processamento nos dispositivos, contribuindo assim para autonomias de até 20 anos. Do mesmo modo o custo dos dispositivos é relativamente baixo devido a esta simplicidade técnica. Não há um limite definido para o número de nós na rede, permitindo um máximo na ordem dos milhões [61][63][64][65][66].

O Weightless é uma tecnologia do tipo celular aberta, desenvolvida para comunicações M2M e gerida pela Weightless SIG. Foi desenvolvida por forma a ser de baixo custo, fiável, com boa cobertura e autonomia, segura e que permita milhões de dispositivos ligados. A respetiva arquitetura protocolar define o funcionamento das camadas Física, MAC e de Rede, sendo que as aplicações são definidas pelo utilizador. Existe em 3 variantes, desenvolvidas por diferentes empresas [12][13][58][60][67]. A Weightless-W foi desenvolvida inicialmente pela NEUL [60] para aproveitar a largura de banda resultante da libertação dos canais analógicos de TV e rádio. Tem débitos entre 1 kbps e 10 Mbps, em canais de 5MHz, com pacotes a partir de 10 *bytes* e alcance urbano de até 5 km. Estima-se uma autonomia dos dispositivos de até 5 anos, sendo esta variante a que mais consome. A Weightless-N, utilizada pela empresa nWave [58], funciona apenas para transmissões unidirecionais no *uplink*, com débitos de 100 bps e pacotes de até 20 *bytes*. Utiliza canais muito estreitos de 200 Hz, modulados em *differential binary phase shift keying* (DBPSK) e tem um alcance estimado de 3 km com autonomia máxima de 10 anos. A Weightless-P foi desenvolvida pela M2 Communications [58] para transmissões bidirecionais fiáveis. Os débitos variam entre os 200 bps e 100 kbps com alcances estimados de 2 km em ambiente urbano. Usa canais de 12,5 kHz modulados em OQPSK, sendo flexível no tamanho dos pacotes permitidos. Permite débitos e potências adaptáveis, mecanismos de acesso ao meio síncronos baseados em tempo e frequência (TDMA, FDMA) ou assíncronos (estilo ALOHA), roaming entre células, bidirecionalidade e mecanismos de *acknowledgement*, com um baixo custo dos equipamentos. Esta solução é a mais adotada atualmente, com autonomias que podem atingir os 8 anos. A arquitetura Weightless utiliza, para além dos nós, uma *base station*, um *network manager* que controla a rede e uma base de dados com informações de utilizadores ou frequências disponíveis. Todas as variantes implementam encriptação AES-128 nos dados, autenticação na rede e técnicas de salto em frequência FHSS para minimizar interferências. A disponibilidade de material é limitada, pelo que o custo de aquisição e implementação é elevado.

A *random phase multiple access* (RPMA) é uma tecnologia proprietária desenvolvida de raiz pela empresa Ingenu para aplicações M2M. Toda a *stack* foi desenvolvida pela empresa de forma a garantir as especificações pretendidas, evitando as restrições das tecnologias LoRa ou Sigfox. Similarmente à Sigfox, segue um modelo de negócio onde vende os seus chips e cobra pela utilização da rede, rede essa que pretende implementar de forma global [12][60][63]. A empresa garante alcances de 10 km em ambiente urbano e 200 em linha de vista. Os débitos máximos são de 624 kbps no *uplink* e 156 kbps no *downlink*, com pacotes

entre 6 a 10 kBytes. Cada ponto de acesso permite ligar até 380000 nós, garantindo autenticação dos nós, encriptação de 256 *bits* e *roaming*. Utiliza a banda dos 2,4 GHz, com canais de 1 MHz (existem 40) modulados em *frequency shift keying* (FSK), e utilizando espalhamento de espectro direto (DSSS). Devido à alta interferência nestas bandas, foram aplicadas técnicas de diversidade espacial, otimização da potência de emissão e acesso ao meio em *time slots*. A segurança é garantida pela privacidade da rede [68][69][70]. Os módulos não se encontram em venda ao público, apenas sendo disponibilizados a parceiros ou fabricantes de dispositivos. O custo de implementação é elevado, não pelos nós mas pela infraestrutura de rede, ascendendo aos milhares de euros [71].

O LoRa é uma tecnologia rádio de camada física, proprietária da empresa Semtech, a qual fabrica *chips* e licencia a tecnologia a outros fabricantes. Para dar suporte a esta camada, foi criada pela LoRa Alliance uma *stack* de acesso ao meio *open source* (LoRaWAN) e que permite criar uma rede completa com esta tecnologia. Uma das grandes vantagens LoRaWAN é ser possível de implementar uma rede privada de longo alcance, bastando para tal adquirir o material, não dependendo de operadores e por isso sem esses custos adicionais. Apesar disso, há operadores de telecomunicações a fornecer acesso a esta rede [59][72][73][74][75]. Os alcances podem chegar a 30 km em campo aberto e 10 km em ambiente urbano, com débitos entre 300 bps e os 50 kbps, assim como o tamanho dos dados entre 8 e 250 bytes. Funciona na banda livre ISM dos 868 MHz (sujeita portanto às regulamentações da mesma), com canais de 125 ou 250 kHz, modulados em *chirp spread spectrum* (CSS). A segurança dos dados é assegurada através de várias camadas de encriptação AES-128, seja nos dados de utilizador seja nos dados de rede. A camada física garante uma boa cobertura e penetração do sinal, sendo robusto contra ruídos, efeitos *doppler* e multi-percurso. Ao mesmo tempo, o baixo consumo garante dispositivos que podem durar até 10 anos [66][73][74][75][76][77][78][79][80]. Existe grande disponibilidade de material, sendo por isso fácil e económico implementar uma rede completa nesta tecnologia.

2.3 ANÁLISE COMPARATIVA DAS DIFERENTES TECNOLOGIAS WSN

Após a descrição das principais tecnologias de redes sem fios, podemos verificar que existem diversas opções que se adaptam a diferentes tipos de cenários de utilização. A tabela no Apêndice B resume as principais características das tecnologias analisadas.

Das tecnologias de curto alcance existentes, o Bluetooth destaca-se pela sua capacidade de bidirecionalidade, débitos altos, grande disponibilidade no mercado e custo baixo de aquisição. Apesar de ter curto alcance, a versão *mesh* irá permitir ampliar os alcances, permitindo assim implementar redes que adicionalmente à monitorização, permitam efetuar controlo com latências aceitáveis. Zigbee também é uma tecnologia com provas dadas no mercado, embora com um custo de aquisição elevado. A escalabilidade e respetiva configuração da rede é mais complexa. O EnOcean apresenta-se como uma solução interessante do ponto de vista energético, estando mais pensado para domótica. O tamanho dos dados que permite transmitir, 14 *bytes*, é limitado o que obriga a enviar vários pacotes quando se pretende enviar mais informação. No entanto, como está em banda livre ISM, sofre das limitações impostas pela regulamentação. O WirelessHART é uma solução bastante fiável, mas desenvolvida para ambiente industrial, sendo por isso o seu custo de implementação muito elevado.

A tecnologia de médio alcance DASH7 apresenta-se como uma proposta que se enquadra bem nos requisitos impostos, com bons débitos e um alcance médio. Tem boas características de segurança e funciona de forma bidirecional sem limitações. Apesar disso, a disponibilidade no mercado é baixa e o custo elevado, o que limita a sua escolha. No caso do WiFi HaLow ainda não se identificou material disponível.

As redes celulares oferecem uma boa opção em termos de cobertura, bastando um operador disponibilizar o serviço para cobrirmos todo um país. Apesar disso a aplicação desta tecnologia a zonas rurais mais remotas pode não ser possível devido ao fraco sinal nesses locais. Oferecem bons débitos e alcances, assim como a possibilidade de *roaming* nos dispositivos. No entanto têm um custo recorrente associado ao operador, o que os pode tornar desvantajosos a longo prazo. Na altura desta análise, as soluções NB-IoT ou EC-GSM IoT ainda não se encontravam disponíveis ao público em Portugal. As redes Weightless apresentam características como bidirecionalidade com débitos altos, segurança e boa escalabilidade, mas apresentam uma fraca disponibilidade no mercado. Requerem também um operador. Tal torna o custo mais elevado a longo prazo. As redes Sigfox e RPMA apesar de terem um bom alcance, têm também um custo fixo de operador. O Sigfox, está amplamente disponível, mas impõe uma limitação de 140 mensagens diárias de 12 bytes, o que é insuficiente para uma monitorização elétrica com algum detalhe. A RPMA oferece bons débitos, segurança e bidirecionalidade mas todo o material é proprietário, de custo elevado e não está disponível ao público. A tecnologia LoRaWAN suporta um bom alcance, com débitos razoáveis e permite a implementação de uma rede totalmente privada e encriptada, sem

necessidade de operadores. O custo fica assim mais reduzido em relação às outras soluções, embora necessite de alguma configuração nos dispositivos. Tem que respeitar as normas da banda ISM e a bidirecionalidade é limitada, factos que têm que ser considerados na implementação.

2.4 CONSIDERAÇÕES FINAIS

Devido à nossa necessidade base de um alcance superior em rede privada, a escolha das soluções de redes sem fios recai sobre o LoRaWAN para monitorização e algum controlo, em que não se imponha requisitos de tempo real. Esta, para além do elevado alcance, tem um custo reduzido e é bastante escalável, cumprindo assim as nossas necessidades base. Permite a criação de uma rede totalmente privada e tem vários mecanismos de encriptação dos dados, uma das imposições da nossa rede, pois a proteção e segurança dos mesmos é fundamental. Os pontos mais fracos assentam na obrigatoriedade em cumprir os requisitos legais ISM, como o tempo de ocupação de espectro e potência de emissão máxima, e a possível latência no *downlink*. No caso de se imporem requisitos de tempo real em comunicações bidirecionais, a opção será utilizar comunicação via WiFi, sendo que nestes casos apenas se poderá funcionar em distâncias mais curtas. As duas tecnologias deverão coexistir. A rede de sensores será assim utilizada para efetuar diversas medições elétricas e ambientais, numa *microgrid*, em pontos distintos e separados entre eles. Permitirá adquirir informação em tempo quase real sobre o estado da *microgrid* assim como apresentá-la aos utilizadores autorizados.

3. LORA E LORAWAN

Neste capítulo serão explicadas com mais detalhe as tecnologias Lora e LoraWAN, assim como o material necessário à implementação de uma rede LoraWAN. Os primeiros dois subcapítulos resumem as especificações da norma, definida pelos fabricantes (Semtech e LoraWAN Alliance) para a tecnologia. O terceiro subcapítulo resume algumas características das comunicações na rede IP, para a rede *backhaul*, necessária ao funcionamento LoraWAN.

3.1 CAMADA FÍSICA PHY LORA

O termo LoRa refere-se a uma tecnologia de rádio frequência proprietária e patenteada da empresa Semtech [83], desenhada para aplicações de redes de baixa potência e longo alcance *low power wide area networks* (LPWAN). Esta tecnologia permite a aplicação em soluções onde sejam requeridas áreas alargadas de cobertura (na ordem dos km) e em simultâneo, grande autonomia das baterias do equipamento terminal (na ordem de anos). Define apenas uma camada física de transmissão rádio. Sendo uma tecnologia proprietária, pouco conhecimento oficial se tem do seu funcionamento para além do que é expresso nas patentes, no Semtech *application notes* AN1200.13, 17 e 22 [75][76][77] e do trabalho desenvolvido em *DecodingLora* [84] e *Reversing Lora* [85]. Neste subcapítulo faz-se uma análise das diversas fontes de informação que descrevem a camada física, de forma a tornar perceptível o que supostamente acontece aquando de uma transmissão.

3.1.1 CONCEITOS BASE

No LoRa, a modulação utilizada consiste numa técnica de espalhamento espectral onde a frequência do sinal vai sendo alterada linearmente, ao longo do tempo de duração do sinal[72], sendo este sinal designado de *chirp* (*Compressed High Intensity Radar Pulse*) e a modulação de *Chirp Spread Spectrum* (CSS). Baseia-se numa técnica inicialmente desenvolvida para aplicações de radar e sonar, que permite uma boa robustez em canais rádio com muito ruído [86], imunidade ao efeito Doppler [72][87] e boa tolerância a efeitos causados por multipercurso do sinal [87]. É tolerante a pequenos desvios de frequência, tornando os equipamentos mais económicos pois não necessitam de osciladores muito precisos [72]. A

Figura 3.1 mostra a forma de onda de um *chirp*, onde se pode ver o aumento da frequência ao longo do tempo, e o respetivo espectrograma.

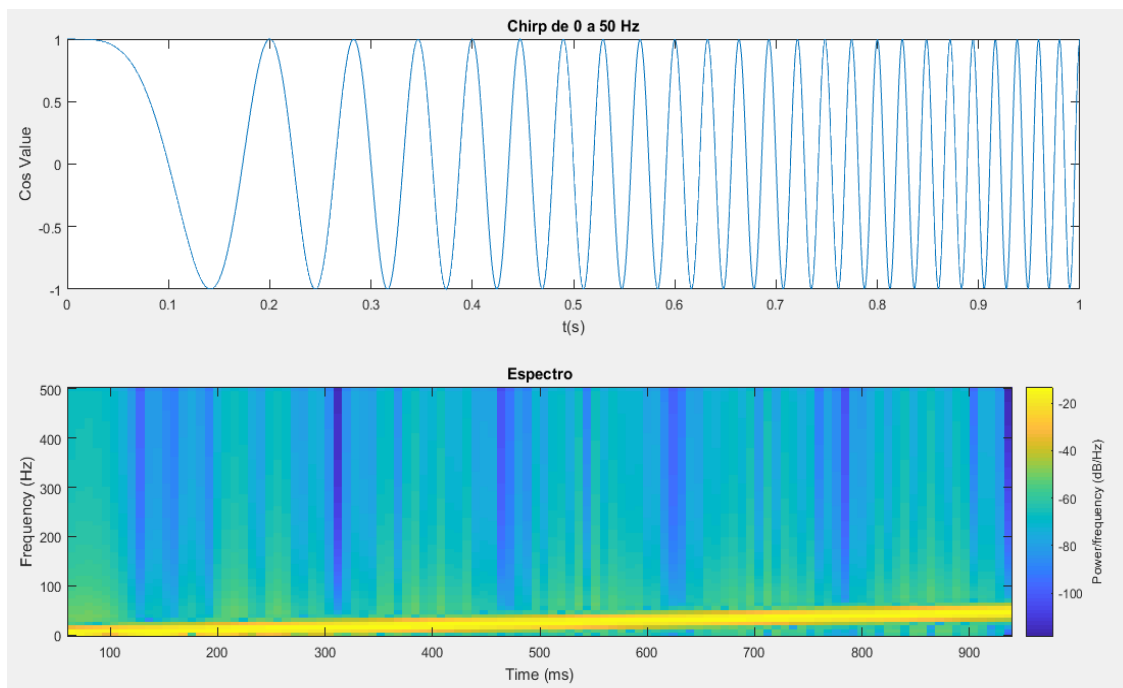


Figura 3.1 - Chirp ascendente de 0 a 50 Hz e respetivo espectrograma onde se consegue ver a subida linear da frequência com o tempo (criado em Matlab)

A Semtech aproveitou esta técnica, aplicando este tipo de sinal numa largura de banda elevada (125, 250 ou 500 kHz) quando comparada com os débitos binários permitidos (250 bps a 50 kbps). Ao mesmo tempo a eficiência da modulação utilizada e esquemas de correção de erros FEC, suportam baixos valores de relação sinal ruído (*signal to noise ratio*, SNR). A SNR traduz a relação entre a potência do sinal e o ruído existente no canal. Quanto menor for este valor, mais baixa a potência do sinal se encontra em relação ao nível de ruído. Um recetor com elevada sensibilidade terá capacidade de decodificar sinais com SNR negativos.

O teorema Shannon-Hartley [72][88] que relaciona o débito binário sem erros de um canal, com a largura de banda B e relação sinal ruído $\left(\frac{S}{R}\right)$ (expresso em função de unidades lineares, e não em dB), é dado pela fórmula seguinte[77]:

$$C = B * \log_2 \left(1 + \frac{S}{R} \right) \Leftrightarrow \frac{C}{B} = \log_2 \left(1 + \frac{S}{R} \right) \Leftrightarrow \frac{S}{R} = 2^{C/B} - 1 \quad (1)$$

Assim, sabemos que, mantendo a largura de banda fixa, quanto maior for o débito C pretendido maior terá que ser a SNR. Nas redes convencionais pretende-se ter débitos elevados o que acarreta SNRs elevados. Já no contexto das LPWAN pretendem-se ter alcances elevados e reduzida interferência. A reduzida potência de emissão e os alcances pretendidos têm como

consequência a obtenção de valores de SNR na recepção o que é contraposto com a utilização de larguras de banda muito superiores à necessária. Impõe também que elevados níveis de sensibilidade do recetor. Assim, a conjugação de baixos débitos, típicos de aplicações IoT, em canais largos, suportam os alcances elevados.

Para dados de utilizador que apenas precisem de uma curta largura de banda para serem transmitidos sem erro, podem-se aplicar técnicas que espalham essa curta largura de banda numa banda muito superior [89]. Espalhar um sinal permite-nos ter o chamado ganho de processamento, obtido pela relação entre a largura de banda do sinal espalhado B_{ss} e a largura de banda mínima necessária para enviar a informação B_w [90]:

$$G_p = \frac{B_{ss}}{B_w} \quad (2)$$

Este ganho permite que a SNR seja negativa, com sinais abaixo do chamado *noise floor* (soma de todos os ruídos existente no meio), mas que o recetor ainda consiga recuperar. Em LoRa conseguem-se decodificar sinais com -20 dB de SNR.

Existem diversas técnicas para espalhar o espectro sendo as mais comuns a *direct sequence spread spectrum* (DSSS) e a *frequency hopping spread spectrum* (FHSS) [89]. O DSSS multiplica os dados de utilizador por um código binário especial, de frequência muito superior à dos dados. Se os dados fossem transmitidos na largura de banda mínima, teriam um canal estreito de potência mais elevada. Ao aplicar o espalhamento, os dados passam a estar codificados com uma frequência muito superior à sua inicial, sendo que na transmissão a potência do sinal será inferior, mas ocupando uma largura de banda maior (ver na Figura 3.2, lado esquerdo). Já o FHSS, envia os dados variando continuamente a portadora, seleccionada de entre diversas frequências (canais) disponíveis, sendo que estes saltos seguem um padrão conhecido entre emissor e recetor, como se demonstra na Figura 3.2, lado direito.

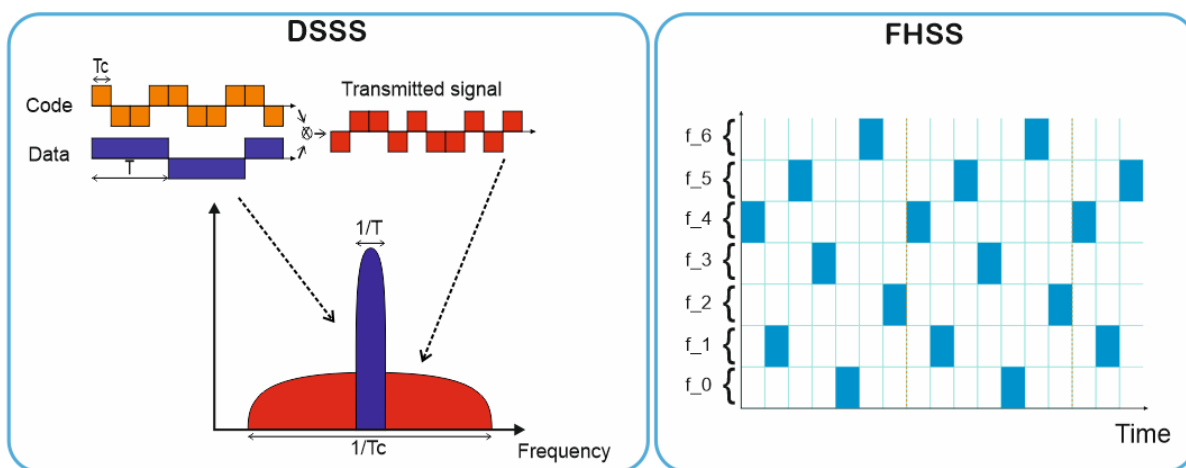


Figura 3.2 - Esquema de funcionamento DSSS e FHSS

3.1.2 MODULAÇÃO LORA

O LoRa não segue estes esquemas tradicionais, pois espalha os sinais ao longo da largura de banda do canal escolhido. A modulação altera a frequência inicial do *chirp*, que é cíclico, podendo ser descrita como *frequency shift chirp modulation* (FSCM) [91] ou como uma *m-ary frequency shift keying* (mFSK) [71].

Para explicar o funcionamento da modulação vamos definir vários conceitos chave:

- Largura de banda (Bw)
- Símbolo
- Fator de espalhamento (SF)
- *Rate code*

A largura de banda (Bw) de um canal corresponde à gama de frequências por onde os sinais que contêm os dados estão a ser transmitidos. Em LoRa os canais podem ter larguras de banda de 125, 250 ou 500 kHz. Existe uma frequência central f_c , correspondente à frequência onde os canais se vão centrar, conforme representado na Figura 3.3. Na Europa, a banda de frequências utilizadas será nos 868 MHz, enquanto nos Estados Unidos será nos 915 MHz. Noutras zonas do globo serão outras frequências.

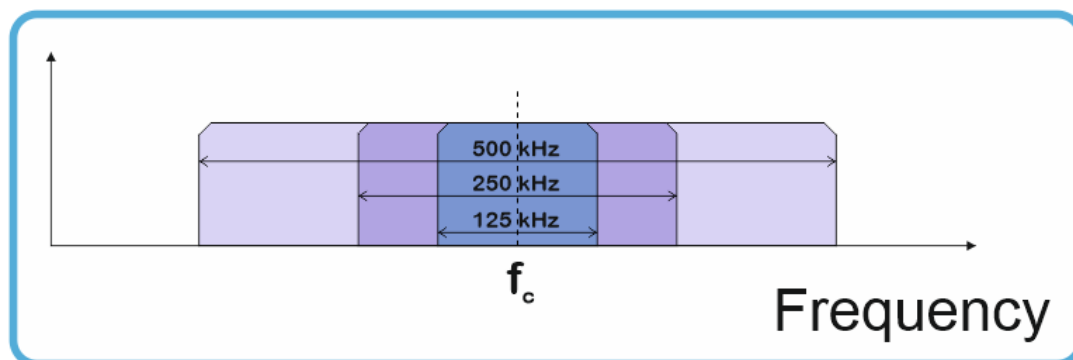


Figura 3.3 - Largura de banda dos canais LoRa

Um símbolo parte do sinal físico onde os dados estão codificados e corresponde a um *chirp* que varia entre duas frequências, uma mínima (f_{min}) e outra máxima (f_{max}). Estas frequências resultam da largura de banda (Bw) e da frequência central (f_c) de acordo com:

$$f_{min} = f_c - \frac{Bw}{2} \quad [Hz] \quad (3)$$

$$f_{max} = f_c + \frac{Bw}{2} \quad [Hz] \quad (4)$$

Um *chirp* irá variar linearmente e de forma cíclica entre a frequência mínima e a frequência máxima durante T_s segundos, conforme apresentado na Figura 3.4. Esta duração, denominada

de período do símbolo (T_s), é definida em função da largura de banda (Bw) e do fator de espalhamento (SF) pela seguinte fórmula [77]:

$$T_s = \frac{2^{SF}}{Bw} \quad [s] \quad (5)$$

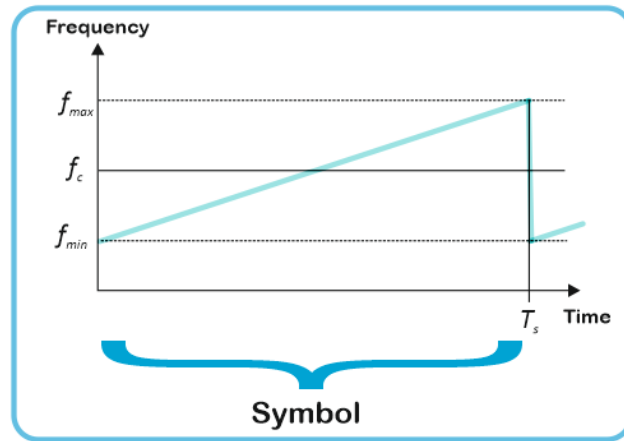


Figura 3.4 - Chirp

Por fim, o fator de espalhamento (*spreading factor*, SF), determina quantos *bits* de informação se podem codificar por símbolo. Em LoRa, podemos transmitir por cada símbolo entre 6 e 12 *bits*. Assim um SF=7 indica que um símbolo LoRa terá codificados 7 *bits*. O SF determina também a duração do símbolo, pois conforme a fórmula anterior diferentes SFs levam a diferentes durações de T_s . A Figura 3.5 exemplifica um espectrograma para os SF utilizados em LoRa, a variar entre 7 e 12 *bits* por símbolo. Este espalhamento permite ao recetor que diferentes sinais com diferentes SFs no mesmo canal possam ser decodificados corretamente, pois garantem ortogonalidade entre eles. No entanto, quando temos SFs e canais diferentes, podem existir combinações que não são ortogonais e, como tal, não devem ser utilizadas [92].

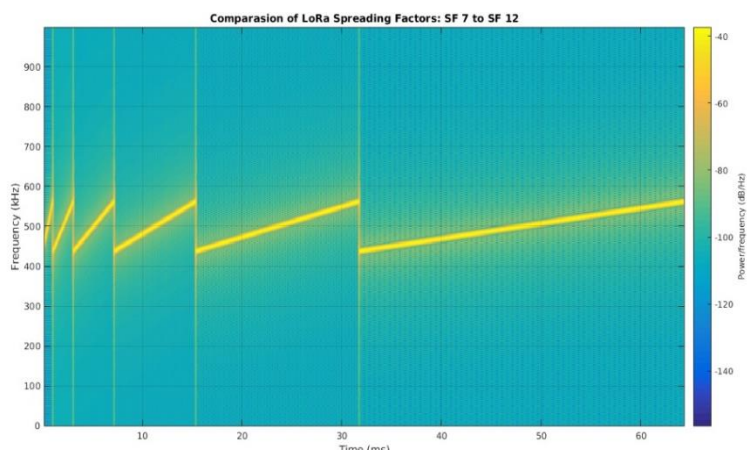


Figura 3.5 - Espectrograma de diferentes SF Lora

O LoRa implementa técnicas de detecção e correção de erros (*forward error correction* FEC), adicionando *bits* de proteção aos dados do utilizador. Por cada 4 *bits* de dados podem-se utilizar entre 1 e 4 *bits* adicionais. Esses *bits* adicionais permitem: (1) detetar erros, quando se utilizam 1 ou 2 *bits* adicionais, (2) corrigir 1 erro, com 3 bits adicionais, ou (3) corrigir 1 erro e detetar um erro adicional com 4 *bits* adicionais [85]. Neste âmbito, o *rate code* é utilizado para quantificar a redução de débito de dados, em função do número de *bits* adicionais utilizado (este último designado de *CR*), através da expressão:

$$Rate\ Code = \frac{4}{4 + CR} \quad (6)$$

Com base nestes parâmetros, podemos analisar os débitos binários que o esquema permite. Ao saber o período do símbolo (T_s), podemos saber diretamente quantos símbolos por segundo (taxa de símbolos por segundo, R_s) temos, fazendo o inverso:

$$T_s = \frac{2^{SF}}{B_w} \Leftrightarrow \frac{1}{T_s} = \frac{B_w}{2^{SF}} \Leftrightarrow R_s = \frac{B_w}{2^{SF}} \quad [baud] \quad (7)$$

Como cada símbolo corresponde a SF bits, basta multiplicar o débito em (símbolos por segundo) por SF (*bits* por símbolo) e teremos o débito binário bruto $R_{b_{raw}}$:

$$R_{b_{raw}} = SF * R_s \quad [bps] \quad (8)$$

No entanto, como o FEC introduz um *overhead*, o débito binário útil R_b é obtido através do *rate code* utilizando [77]:

$$R_b = SF * R_s * Rate\ Code \quad [bps] \quad (9)$$

Os débitos obtidos, em função de SF e B_w são apresentados na Tabela 3.1 (nota: o modo FSK, utiliza todo o canal de 500 kHz sem nenhum espalhamento de espectro, para maximizar

Configuração (SF – Bw)	Débito binário [bps]	Fator de espalhamento SF	Chips por símbolo
LoRa: SF12/125 kHz	250	6	64
LoRa: SF11/125 kHz	440	7	128
LoRa: SF10/125 kHz	980	8	256
LoRa: SF9/125 kHz	1760	9	512
LoRa: SF8/125 kHz	3125	10	1024
LoRa: SF7/125 kHz	5470	11	2048
LoRa: SF6/250 kHz	11000	12	4096
FSK	50000		

Tabela 3.1 - Débitos em função de SF

o débito).

Para analisar como os dados são modulados voltemos ao SF. Como referido, a escolha do SF determina o número de *bits* por símbolo, mas também define o número de divisões temporais que iremos fazer num símbolo como sendo 2^{SF} . Estas separações são designadas por *chips* (não confundir com *chirps*), por analogia feita à designação utilizada em DSSS [77] e tem valores que variam entre 64 e 4096 (Tabela 3.1). Utilizando por exemplo um SF igual a 6 *bits*/símbolo, iremos ter $2^6 = 64$ *chips* por símbolo, como mostra a Figura 3.6. Assim, com 6 *bits* podemos representar 64 valores binários possíveis, onde numa conversão direta com o sistema de numeração decimal correspondem aos valores entre 0 e 63.

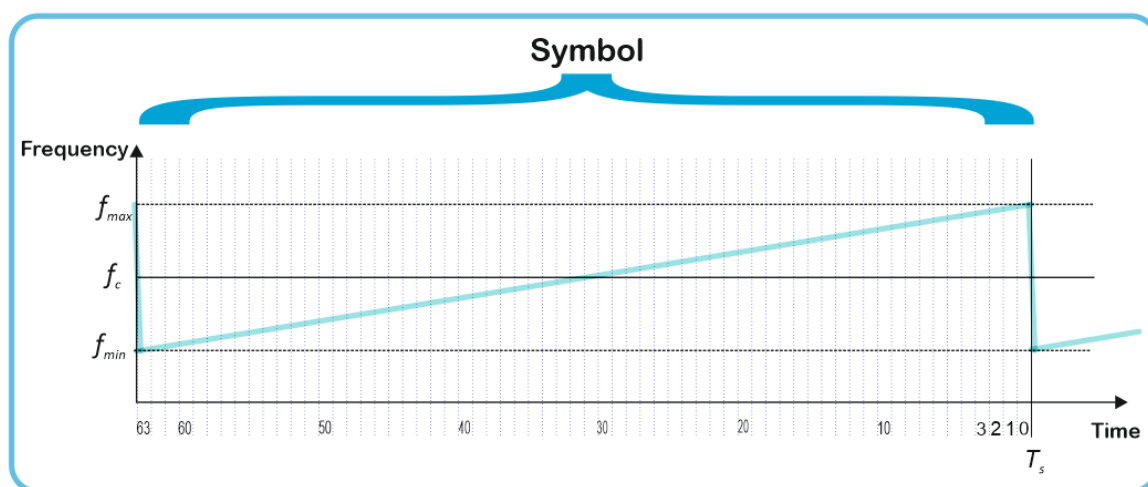


Figura 3.6 – Exemplo de chip Lora, onde se mostram os 64 pontos onde pode existir o salta de frequência

Como dito anteriormente, os símbolos LoRa são cíclicos, variando sempre entre a frequência mínima e a frequência máxima. A forma como em LoRa se transmite um valor passa por alterar o momento dentro do tempo de símbolo T_s , onde se dá o salto entre as frequências máximas e mínimas. Ou seja, este local corresponde ao valor que queremos transmitir. Do mesmo modo, esta alteração faz com que a frequência inicial seja alterada para um valor diferente da frequência mínima, superior a esta e inferior à frequência máxima. Toda a descrição matemática pode ser vista em [91], onde utilizaram a sigla FSCM devido exatamente a este salto na frequência inicial dependente do valor a modular.

Vamos exemplificar com valores, de modo a tornar mais perceptível o funcionamento. Suponhamos que temos um canal de 125 kHz, na banda 868,1 MHz e um SF=6 *bits*/símbolo. Este SF corresponde a 64 *chips* e a frequências mínima e máxima de 868,0375 MHz e 868,1625 MHz, respetivamente. O período do símbolo terá a duração de $T_s = \frac{2^6}{125000} = 512 \mu s$. Admitamos que se pretende transmitir um valor binário de 6 *bits* igual a 001010_b,

correspondente ao valor decimal 10_d . O que a modulação vai fazer é alterar o ponto de salto do *chirp* para a posição 10 (correspondente ao valor a modular), conforme representado pela linha a vermelho da Figura 3.7. A frequência inicial do *chirp* passa a ser dada por:

$$f_0 = f_{min} + \frac{Bw}{2^{SF}} * Data_{decimal} \Rightarrow \tag{10}$$

$$f_0 = 868\,037,5 + \frac{125}{64} * 10 = 868\,074,606 \text{ kHz}$$

Do mesmo modo, assumindo um tempo inicial $T_0 = 0$, o ponto onde ocorre o salto corresponde a $80 \mu s$ após o início do período, de acordo com:

$$t_c = T_0 + \frac{T_s}{2^{SF}} * Data_{decimal} [s] \Rightarrow t_c = 0 + \frac{512 \mu s}{64} * 10 = 80 \mu s \tag{11}$$

A Figura 3.7 demonstra o *chirp* (a vermelho) representando o valor modulado. A mesma solução é aplicada a dados de maiores dimensões, que serão divididos em grupos de SF bits e modulados com esta mesma técnica. A Figura 3.8 mostra um excerto de uma trama LoRa, capturado com recurso a *software defined radios* (SDR), onde se podem ver os saltos em diferentes zonas que representam os valores modulados. A vermelho estão representados os períodos de cada *chirp*.

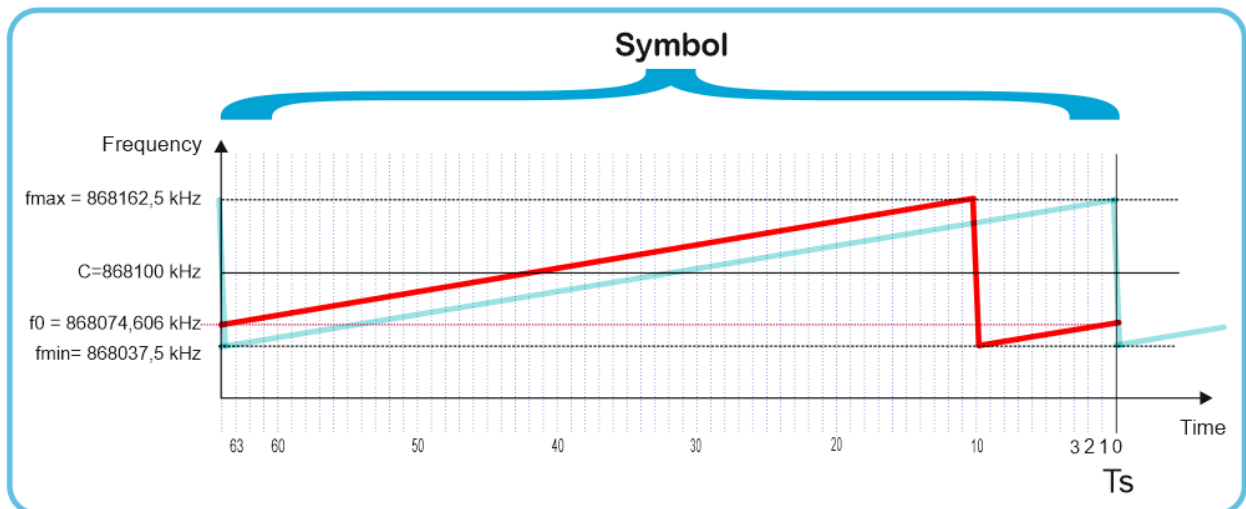


Figura 3.7 - Modulação LoRa do valor binário 001010b (a vermelho). A azul, chirp base

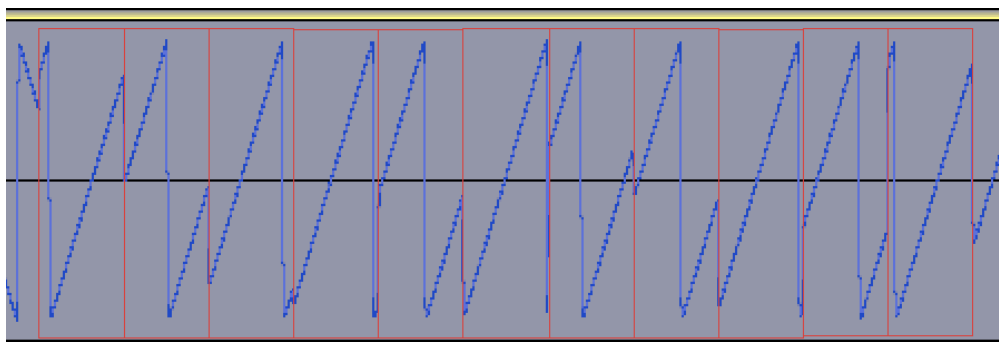


Figura 3.8 - Excerto de uma captura de trama LoRa [93]

3.1.3 FORMATO DA TRAMA E NORMAS APLICÁVEIS

A utilização das bandas livres ISM 868MHz, traz a vantagem do maior alcance quando comparado com as soluções dos 2,4 GHz, uma vez que frequências mais baixas sofrem de menos perdas de propagação no meio. Ainda assim, é exigido o cumprir de determinadas regras estipuladas por entidades reguladoras, de forma a que os equipamentos não ocupem o espectro a tempo inteiro, evitando a saturação do canal. No caso dos 868 MHz, a *Electronic Communication Committee* (ECC) impõe restrições à potência radiada efetiva (*effective radiated power*, ERP) e ao tempo de ocupação do canal (*duty cycle*), conforme mostra a Tabela 3.2 [94]. Para que não se tenha que cumprir o requisito do *duty cycle* é necessário integrar técnicas de escuta ao meio (*listen before talk*, LBT) juntamente com a possibilidade de mudar de canais de forma integrada (*agile frequency adaptation*, AFA), mantendo-se apenas a limitação na potência de emissão. Apesar disso, estas técnicas não são aplicadas nas soluções de domínio livre.

Frequências	Potência máxima	Duty Cycle obrigatório
868 – 868,6 MHz	25 mW e.r.p.	Duty cycle \leq 1 % ou LBT+AFA
868,7 – 869,2 MHz	25 mW e.r.p.	Duty cycle \leq 0,1 % ou LBT+AFA
869,4 – 869,65 MHz	500 mW e.r.p.	Duty cycle \leq 10 % ou LBT+AFA

Tabela 3.2 – Potência e duty cycles utilizados na banda dos 868 MHz

Na secção 5.2.1 da norma ETSI EN300.220 [95] a ERP está definida como a potência radiada na direção do lóbulo mais potente, utilizando uma antena dipolo de meia onda, limitando a potência emitida em cada transmissão.

Em relação ao *duty cycle* (DC_{banda}) o mesmo é definido como sendo a relação entre a soma do tempo de transmissões num determinado intervalo de tempo, para uma respetiva

largura de banda (ver secção 5.4.1 do EN300.220). Define também o tempo de observação como sendo 1 hora.

$$DC_{banda} = \frac{T_{transmissao_acumulado}}{T_{observação}} * 100 \quad [\%] \quad (12)$$

Ou seja, esta limitação determina o tempo máximo que se pode transmitir em determinada banda. A Figura 3.9 mostra um exemplo de 20 % *duty cycle* em cada canal, onde cada um dos 3 canais utiliza o tempo máximo de transmissão permitido, 12 minutos por hora. No caso LoRa, que não implementa técnicas LBT+AFA, podem existir *duty cycles* de 0,1%, 1% ou 10 %, consoante os canais escolhidos. No entanto vão depender do módulo rádio e da capacidade deste em sintonizar esses mesmos canais.

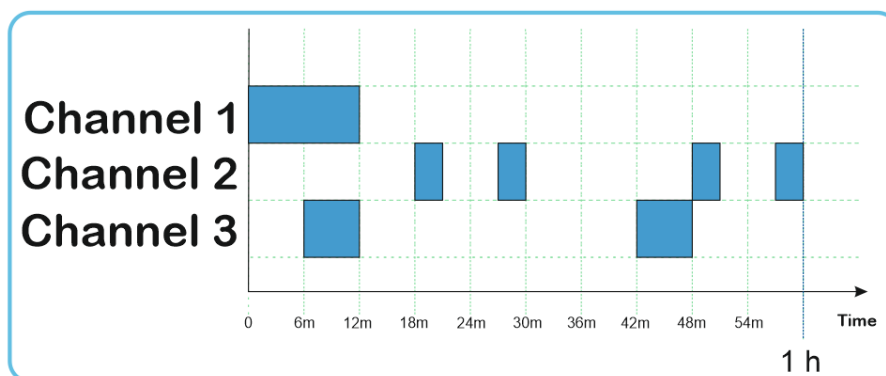


Figura 3.9 - Exemplo de duty cycle a 20%, em três canais

Tendo esta restrição em mente, torna-se importante saber a duração de cada transmissão, o *Time-On-Air* (ToA), que será o tempo ocupado no canal para aquela transmissão. O ToA depende por sua vez de vários fatores entre os quais está a estrutura da trama física LoRa, que segue o formato da Figura 3.10. Esta está dividida em 4 partes, preâmbulo, cabeçalho, *payload* e CRC do *payload*, descritas de seguida [96].

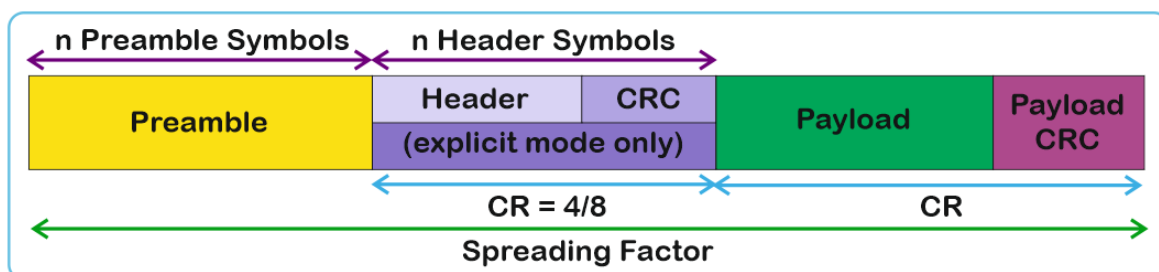


Figura 3.10- Composição da trama física LoRa [75]

O preâmbulo tem uma duração mínima de 4,25 símbolos, mas é normalmente programado para um valor superior, por norma 8. Serve para deteção da trama por parte dos recetores e sincronização temporal. No preâmbulo, existem uma série de *chirps* ascendentes após os quais estão 2 *chirps* descendentes que contêm codificada uma palavra de sincronismo (ver Figura 3.11).

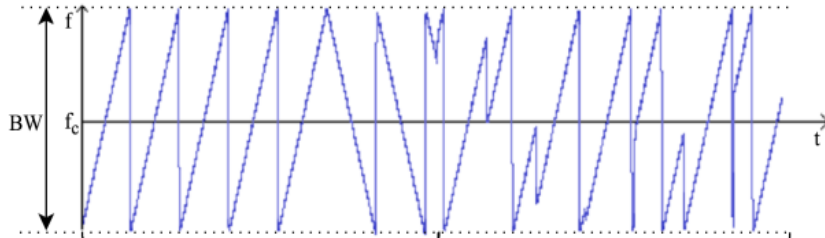


Figura 3.11 - Excerto de captura de trama LoRa, onde se pode ver o preâmbulo e parte do payload com os dados modulados [72]

O cabeçalho é opcional e permite conter informações como o tamanho de *payload*, existência de verificações de erros (*cyclic redundancy check*, CRC) e *code rate* (CR, i.e. número de *bits* para FEC referidos anteriormente) do *payload*. O CR do cabeçalho é fixo e a duração é de aproximadamente 8,4 símbolos.

O *payload* contém os dados de utilizador. A fórmula que define o tamanho do *payload* assegura que independente dos dados do utilizador, este terá uma duração mínima de 8 símbolos. O *payload* CRC serve para verificar a integridade dos dados. A duração típica é de 8 símbolos. O tempo de símbolo obtém-se a partir da já referida fórmula (5). A duração do preâmbulo (T_{pre}), depende do número de símbolos que utilizamos para sincronismo $n_{preSmbL}$ de acordo com:

$$T_{pre} = (n_{preSmbL} + 4,25) * T_{symb} [s] \quad (13)$$

A duração da restante trama é obtida a partir da equação (14) onde: *Pay* representa os *bytes* do *payload*, *Head* (do tipo booleano) representa a existência ou não de cabeçalho, e *DE* (do tipo booleano) traduz a utilização ou não de uma otimização para baixos débitos. O *DE* é utilizado apenas para SFs de 11 ou 12, e em canais de 125 kHz, para compensar deslocamentos que possam ocorrer nas frequências de oscilação dos cristais, decorrentes de alterações de temperatura ou posição [84].

$$T_{frame} = T_{symb} * \left[8 + \max \left(\text{ceil} \left(\frac{8 * Pay - 4 * SF + 28 + 16 - 20 * Head}{4(SF - 2 * DE)} \right) * (CR + 4), 0 \right) \right] \quad (14)$$

A duração total da trama (dito *Time on Air*, *ToA*) será então obtida pela soma dos dois componentes anteriores e corresponde ao tempo de transmissão:

$$T_{oA} = T_{pre} + T_{frame} \quad [s] \quad (15)$$

A Semtech disponibiliza uma ferramenta que permite calcular estes valores (*LoRa Modem Calculator Tool*) [97]. A TheThingsNetwork (rede LoRaWAN pública de acesso gratuito) tem folhas de cálculo online que permitem fazer este mesmo cálculo, assim como quantas transmissões se podem fazer por hora [98].

Após esta análise é fácil perceber que alterações no SF e largura de banda do canal acarretam variações no tempo de cada transmissão. A Figura 3.12 exemplifica a variação do ToA para pacotes de diferentes tamanhos consoante o SF. Um incremento de 1 no SF provoca aproximadamente a duplicação no tempo de transmissão de cada símbolo. Por exemplo, podem-se ver que para os SF 11 e 12, *payloads* superiores a 190 e 70 bytes respetivamente, ultrapassam um *duty cycle* de 1 % (3,6 segundos).

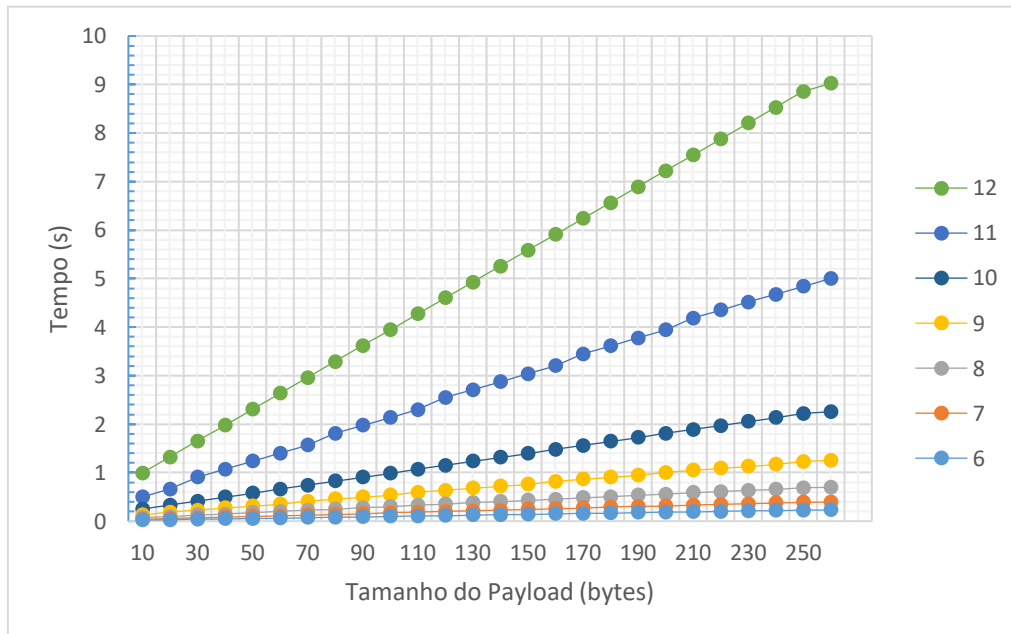


Figura 3.12 - ToA em função de payload e SF

3.1.4 LINK BUDGET E SENSIBILIDADE

O Link Budget permite relacionar as potências ao longo de uma transmissão entre um emissor e um recetor, considerando todas as perdas e ganhos existentes no percurso, conforme esquematizado na Figura 3.13. No caso de emissões rádio, podemos utilizar a expressão seguinte que traduz a potência que chega ao recetor P_{rx} , resultando do cálculo do seguinte *link budget* [99]:

$$P_{rx} = P_{tx} + G_{tx} - L_{tx} - L_{fs} - L_m + G_{rx} - L_{rx} \quad [dBm] \quad (16)$$

Esta expressão é composta pela potência do emissor (P_{tx}), os ganhos das antenas de emissão (G_{tx}) e recepção (G_{rx}), as perdas nos conectores e cabos existentes em cada extremo da ligação (L_{tx}, L_{rx}), as perdas de propagação no meio (L_{fs}) e as perdas envolvidas nos fenómenos de multi percurso ou *fading* (L_m).

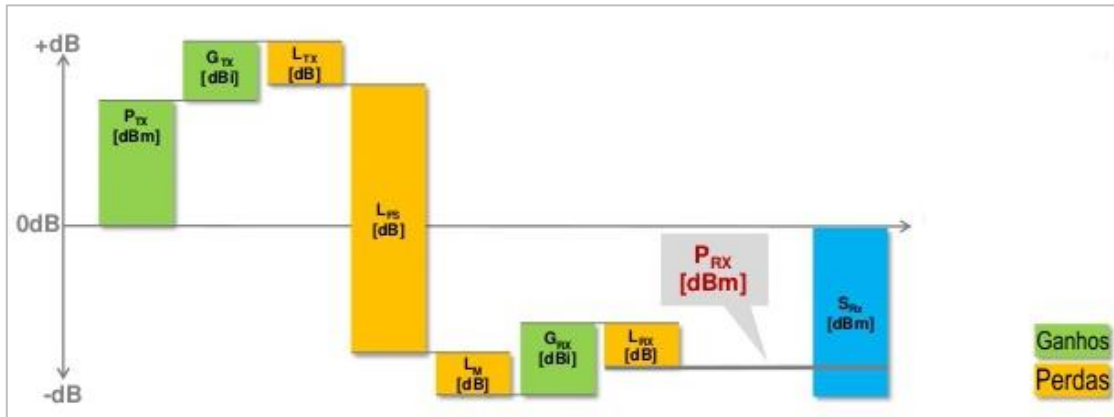


Figura 3.13 - Esquema de link budget (adaptado de [99])

Um determinado recetor tem uma sensibilidade S_{rx} que define a potência mínima de sinal que este precisa receber, de forma a conseguir decodificar corretamente o sinal. Assim, a potência recebida P_{rx} tem de ser superior à sensibilidade do dispositivo:

$$P_{rx} \geq S_{rx} \quad (17)$$

Por sua vez, a sensibilidade do recetor LoRa é determinada pela fórmula:

$$S_{rx} = -174 + 10 \log_{10} Bw + NF + SNR \quad [dBm] \quad (18)$$

em que: (1) NF define o *noise figure* do recetor (é fixo em cada implementação de *hardware*), (2) o SNR traduz a relação sinal ruído mínima permitida pelo recetor, a qual depende da modulação utilizada e (3) o valor -174 deriva do ruído térmico e é fixo. A SNR por sua vez varia de acordo com SF tal como apresentado na Tabela 3.3.

Valores mais baixos de S_{rx} traduzem uma melhor capacidade de decodificar sinais recebidos com potências mais débeis. Neste âmbito, também quanto mais baixos forem Bw e SNR , tanto melhor para a capacidade de decodificar o sinal recebido [100].

Fatores de espalhamento maiores têm mais ganho de processamento, levando a que os valores mínimos de SNR aceites pelo receptor possam ser inferiores. SFs menores levam a maiores débitos binários, mas tem como consequência a necessidade de uma SNR superior, o que diminui a sensibilidade do recetor. Similarmente, maiores larguras de banda permitem

maiores débitos binários, mas elevam o valor de S_{rx} (i.e. diminuem a capacidade de descodificar o sinal recebido).

A sensibilidade LoRa depende de 2 variáveis que podemos ajustar, o fator de espalhamento SF e Bw . A Tabela 3.4 dá-nos valores típicos de sensibilidades em função de SF e Bw , onde se podem constatar esses factos. Estes valores foram obtidos em [97]. Os maiores alcances são obviamente obtidos em canais mais estreitos e com maior espalhamento.

SF	SNR [dB]
12	-20,0
11	-17,5
10	-15,0
9	-12,5
8	-10,0
7	-7,5

Tabela 3.3 - SNR mínima no recetor em função do SF

Sensibilidade (dBm)		Fator de espalhamento					
		7	8	9	10	11	12
Largura de banda	125 kHz	-123	-126	-129	-132	-134,5	-137
	250 kHz	-120	-123	-126	-129	-131,5	-134
	500 kHz	-117	-119	-122	-125	-128,5	-131

Tabela 3.4 - Sensibilidade em função de SF e Bw

Para efeito de cálculo de débito, para além da Largura de banda Bw e do fator de espalhamento SF, há ainda a considerar o *code rate* CR.

Assim, a alteração destes três parâmetros provoca mudanças na sensibilidade do recetor, alcance, débito binário e tempo de transmissão. A escolha deve recair sobre uma combinação que garanta as especificações da instalação (alcances, débitos necessários, número de dispositivos, etc.), após uma avaliação das condições de utilização no local (tipo de ambiente, interferências, etc). A Figura 3.14 demonstra de forma visual estas relações [101].

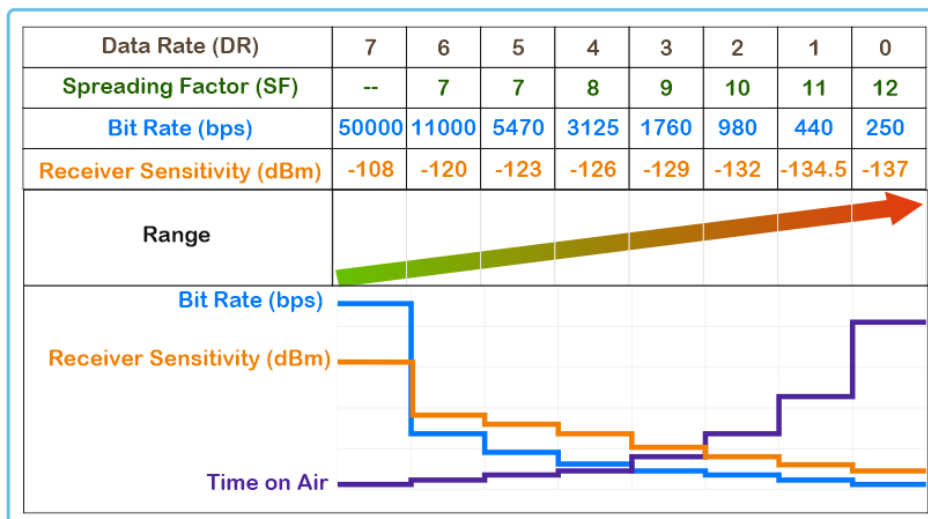


Figura 3.14 - Esquema representativo entre as diversas variáveis de um sistema LoRa (DR representa os códigos LoRaWAN)

3.2 CAMADA DE ACESSO AO MEIO MAC LORAWAN

3.2.1 INTRODUÇÃO

O termo LoRaWAN refere-se ao protocolo de acesso ao meio (*medium access control*, MAC) e arquitetura de rede criada pela LoRaWAN Alliance de forma a dar suporte à camada física LoRa [102]. Esta foi desenvolvida de raiz para redes de sensores sem fios de longo alcance, utilizando a tecnologia rádio LoRa, para transmissões pouco frequentes e de *payload* pequeno entre sensores e servidor [72]. Este desenho, com aplicações IoT em mente, garantiu um máximo de poupança de energia e um baixo custo de fabrico e implementação, o que a tornou numa das soluções mais populares do mercado [103]. A especificação 1.0 foi lançada em janeiro de 2015, sendo a referência para a maioria das implementações existentes no mercado. Em outubro de 2017 saiu a versão 1.1, que veio trazer clarificações nalguns aspetos e novas funcionalidades. A maior alteração foi a da definição de funcionamento do *roaming*, que veio adicionar novos elementos na arquitetura da rede. Ao mesmo tempo foram melhorados os mecanismos de segurança, onde existiam algumas vulnerabilidades que permitiam ataques à rede. Sobre esta última não se vai entrar em grandes detalhes, uma vez que o material que iremos utilizar será da norma 1.0.1.

3.2.2 ARQUITETURA E COMPONENTES DE REDE

A topologia de rede utilizada é designada como estrela-de-estrelas [104], e definiu diversos tipos de dispositivos obrigatórios na arquitetura, conforme se pode ver na Figura 3.15:

- Os dispositivos terminais, nós finais, que por norma são sensores de baixa potência que monitorizam determinada variável. Comunicam diretamente com as *gateways* existentes, não existindo nenhum tipo de associação a estas. Desta forma apenas existe um salto (*single hop*) [105], mantendo uma latência uniforme e a simplicidade de não ser necessário nenhum encaminhamento de dados. Estes nós transmitem normalmente de forma assíncrona, apenas quando têm dados a enviar, não necessitando de nenhum sincronismo regular com a rede e permitindo manter o dispositivo adormecido durante mais tempo. Apesar de existirem mecanismos de sincronização para transmissões mais regulares, são pouco utilizados (ver classe B, seção 3.2.3.2). A sua camada de aplicação está ligada a um servidor de aplicação na *cloud*, o qual é o único capaz de receber os seus dados.
- As *gateways*, dispositivos intermédios que encaminham os pacotes entre terminais e servidor de rede, também chamados de encaminhadores de pacotes (*packet forwarder*). A comunicação entre as *gateways* e o servidor de rede, designada de *backhaul*, pode ser feita por qualquer tipo de interface IP de débito mais elevado, nomeadamente WiFi, LTE ou Ethernet. A trama rádio é transformada num objeto JSON e enviada via UDP/IP ao servidor pela *backhaul* [106]. Assim, a função da *gateway* é apenas adaptar os dois meios (rádio LoRa <> *Backhaul* IP) de forma transparente entre extremos e adicionar na mensagem de *uplink* informação sobre a qualidade de receção [72]. Ao não existir mecanismos de associação entre os dispositivos terminais e *gateways*, faz com que seja possível a várias *gateways* receber o mesmo pacote, reencaminhando-o sempre para o servidor de rede.
- Os servidores de rede (*network servers*, **NS**) gerem toda a rede. As funções deste passam por permitir a ligação de nós à rede, verificar mensagens duplicadas recebidas de várias *gateways*, decodificar e fazer verificações de segurança nas mensagens, para enviar os dados aos servidores de aplicação respetivos, encaminhar respostas de aplicações ou *acknowledgments* aos nós finais escolhendo a melhor *gateway*, gerir os débitos binários (*adaptive data rate*, ADR) dos nós e responder aos pedidos MAC dos nós.
- As aplicações utilizam os dados recebidos do servidor de rede processam-nos para o utilizador. Estas têm de descriptar as mensagens recebidas de forma a poder utilizar e

processar os dados dos nós [107]. Estão normalmente na *cloud* num servidor independente do NS, o chamado servidor de aplicações (*application server, AS*). Esta parte não está definida na especificação, deixando a cargo do utilizador a sua implementação [106]. Podem estar ligadas a vários NS, recebendo os dados de nós em redes distintas, que sejam do mesmo utilizador ou aplicação.

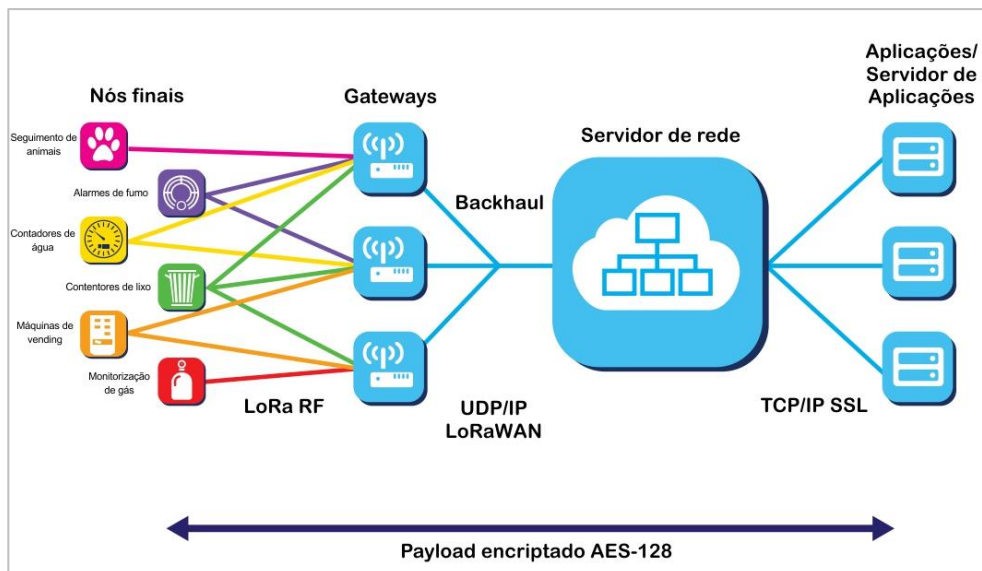


Figura 3.15 - Arquitetura LoraWAN (adaptada de [108])

A arquitetura de rede em estrela-de-estrelas foi pensada de forma a que exista apenas um salto entre o nó final e uma *gateway*. Arquiteturas em malha exigem protocolos de encaminhamento para gerir os percursos dos pacotes, permitindo maiores alcances à custa de um processamento adicional em cada nó. As complexidades da rede em malha juntamente com este processamento contribuem para um maior consumo energético. Ao mesmo tempo em LoRaWAN não existe associação entre nós e *gateways*, permitindo a que qualquer *gateway* que receba um pacote o encaminhe ao servidor, podendo este receber mais que uma vez o mesmo pacote [102]. A não existência de associação simplifica o procedimento de envio de dados, bastando ao nó transmitir (nota: não confundir associação com ativação).

Os dispositivos terminais podem comunicar de duas formas, quando têm dados para enviar (ex: houve alguma alteração nos seus sensores) ou com agendamento (ex: envio periódico de leituras de sensores). Nunca precisam de acordar para escutar a rede, permitindo estar adormecidos entre transmissões. Desta forma consegue-se minimizar o consumo, pois é nestas fases de escuta ou sincronismo que o gasto energético é superior [107]. Estas características levam a dispositivos mais simples, baratos e eficientes. Não é permitida a comunicação direta

entre dispositivos terminais sendo sempre necessário passar os dados pelo servidor de rede [72].

3.2.3 CLASSES DE DISPOSITIVOS

A especificação LoRaWAN classifica os dispositivos em 3 classes distintas (A, B ou C), dependente do seu consumo energético, às quais correspondem diferentes perfis de transmissão e recepção de dados.

3.2.3.1 Classe A

Os dispositivos de classe A, foram desenvolvidos maioritariamente para transmissões *uplink*. Estes podem enviar dados quando têm necessidade (se existem dados para enviar) ou de forma periódica, mas sem agendamento. Utilizam um mecanismo de acesso ao meio estilo ALOHA no *uplink* [109]. Embora permitam comunicações bidirecionais, apenas podem receber dados após um envio em *uplink* de dados, onde se abrem duas pequenas janelas de recepção (*receiver-initiated transmission*, RIT) [110]. Na primeira janela, o nó escuta no mesmo canal e utilizando o mesmo SF da última transmissão *uplink*, denominada de modo *Rx1*. Na segunda janela, o nó escuta num SF mais alto (p.ex. SF = 12, $F_{rx} = 869,525$ MHz), denominada de modo *Rx2*, que permite aumentar a probabilidade de recepção e evitar problemas no canal rádio [111]. A primeira janela é obrigatória, mas a segunda depende da recepção ou não de um *downlink* na primeira. Em caso afirmativo, o dispositivo não precisa de abrir a segunda janela, mas em caso negativo tem que abrir [112]. A Figura 3.16 mostra o esquema de *timings* utilizado. Os valores de *RxDelay1* e *RxDelay2*, traduzem o início de cada uma das janelas, i.e. o tempo depois da transmissão em *uplink* em que o servidor de rede pode enviar dados ao nó terminal. *RxDelay1* tem o valor por omissão de 1 segundo. Já o *RxDelay2* obtém-se através de $RxDelay2 = RxDelay1 + 1s$, sendo por isso de 2 segundos. No caso de ser necessária uma transmissão *downlink* (oriunda do servidor de rede), este tem de esperar até que exista uma transmissão *uplink* do nó, para depois poder então enviar os seus dados. Tal traduz-se numa elevada latência das comunicações em *downlink*. No caso do nó precisar de fazer um *uplink* de 2 mensagens, tem de esperar que as janelas de transmissão do primeiro envio passem, para poder enviar a segunda mensagem. Esta é a classe de dispositivos mais

eficientes, pois não utilizam nenhum mecanismo de escuta do meio, estando adormecidos se não houver dados a transmitir [72]. A especificação exige que todos os dispositivos implementem esta classe. A aplicação ideal é em sensores de monitorização que tenham um tráfego maioritariamente de *uplink* e não necessitem de receber muitos dados do servidor [111].

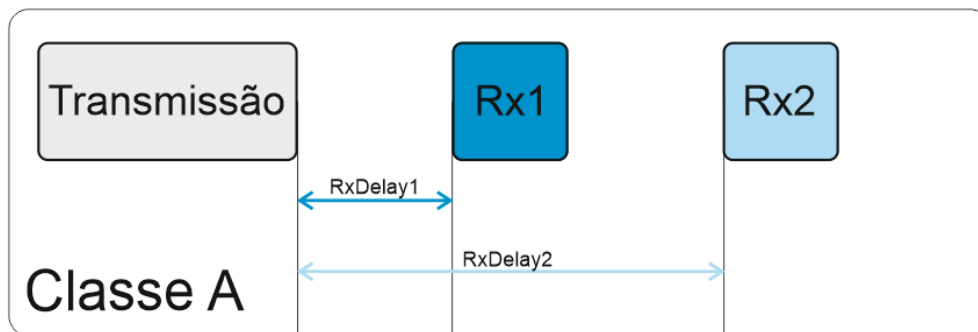


Figura 3.16 - Classe A LoraWAN

3.2.3.2 Classe B

Os dispositivos de classe B, permitem transmissões bidirecionais com janelas de *downlink* agendadas, garantindo um valor médio de latência no *downlink* (*coordinated sampled listening*, CSL). O *uplink* processa-se de forma semelhante à classe A. O consumo é intermédio quando comparado com os dispositivos de classes A e C. O funcionamento passa pelo envio periódico por parte da *gateway*, de um sinal de *beacon* (sinal com SF=9 em canal de 125 kHz, por omissão na frequência 869,525 MHz), que permite sincronizar os relógios dos nós de classe B. Após a recepção do *beacon* os nós abrem pequenas janelas de receção (chamadas de *ping slot*) que permitem ao servidor transmitir-lhes dados (sendo estas transmissões designadas de *ping*). Desta forma não é necessária uma transmissão *uplink* para abrir janelas de receção [113]. Este modo de funcionamento não é estático, podendo ser aplicado consoante a necessidade a um dispositivo de classe A. O nó ativa-se na rede como classe A sendo a camada de aplicação deste a decidir a mudança de modo, quando necessário. Assim, o servidor de rede tem de ser informado da periodicidade, DR e canal dos *ping slots*, e foram introduzidos novos comandos MAC na versão 1.1, para permitir esta troca de informação. Esses comandos permitem mensagens *multicast*. Dispositivos desta classe estão pouco implementados em *hardware*, devido à falta de clarificação da especificação na sua versão 1.0. A versão 1.1 veio clarificar alguns aspetos, pelo que se espera que este modo seja implementado futuramente. A Figura 3.17 mostra um exemplo (os tempos não são reais, apenas demonstrativos) onde se

podem ver os *beacons* recebidos periodicamente da *gateway* (a amarelo) e 8 *ping slots* abertas e disponíveis para enviar dados aos nós. Nesta imagem, um nó tem abertas para recepção as *slots* número 2 e 6, onde pode receber mensagens *downlink* do servidor de rede. Na realidade, o período entre *beacons* tem uma duração de 128 segundos, divididos numa janela útil de 122,88 segundos mais a duração fixa do *beacon* de 2,12 segundos e um intervalo de guarda de 3 segundos. A janela por sua vez é dividida em 4096 *ping slots* de 30 ms de duração, indexados de 0 a 4095. A cada período do *beacon* são escolhidas de forma pseudoaleatória, as várias *ping slots* de escuta nas quais o nó poderá receber dados. Este processo é efetuado de forma idêntica no nó e NS, utilizando o mesmo algoritmo, de forma a que os valores gerados sejam semelhantes. Com as *ping slots* atribuídas, o recetor deve ligar o seu rádio nesses precisos momentos, de forma a escutar se existe algum *downlink* para si. Se o NS tiver mensagens *downlink* para o nó, deverá enviá-las apenas nos *ping slots* atribuídos.

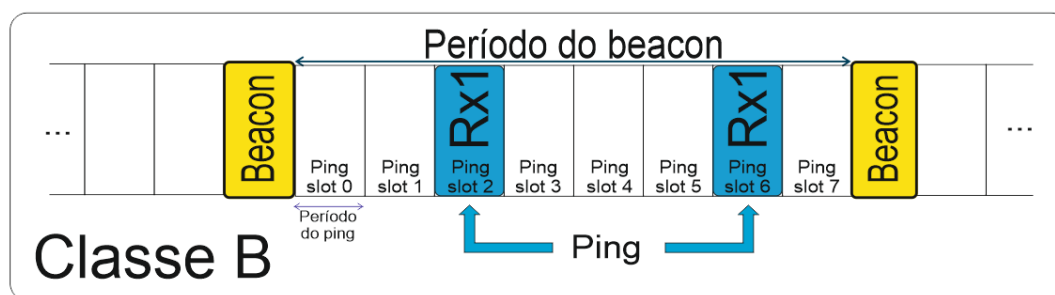


Figura 3.17 - Classe B LoRaWAN com exemplo de 8 ping slots entre beacons e 2 receções ping

3.2.3.3 Classe C

Os dispositivos de classe C são bidirecionais e estão continuamente a escutar o meio, exceto quando transmitem dados (*continuous listening*). A latência é baixa no *downlink* mas o consumo de energia é elevado, sendo que apenas devem ser implementados em locais onde não existam constrangimentos energéticos [109]. Nesta classe o nó deve escutar continuamente no modo *Rx2*. De forma similar à classe A, no fim de fazer um *uplink*, o nó abre imediatamente uma janela de recepção com as configurações do modo *Rx2*. Após um determinado atraso (*RxDelay1*) muda para uma janela de recepção com as mesmas configurações utilizadas na última mensagem *uplink* (modo *Rx1*). Quando esta termina, muda novamente para as configurações de *Rx2*, onde se mantém até existir nova mensagem *uplink*,

conforme a Figura 3.18. Tal como na Classe A, $RxDelay1$ tem o valor por omissão de 1 segundo e $RxDelay2$ de 2 segundos.

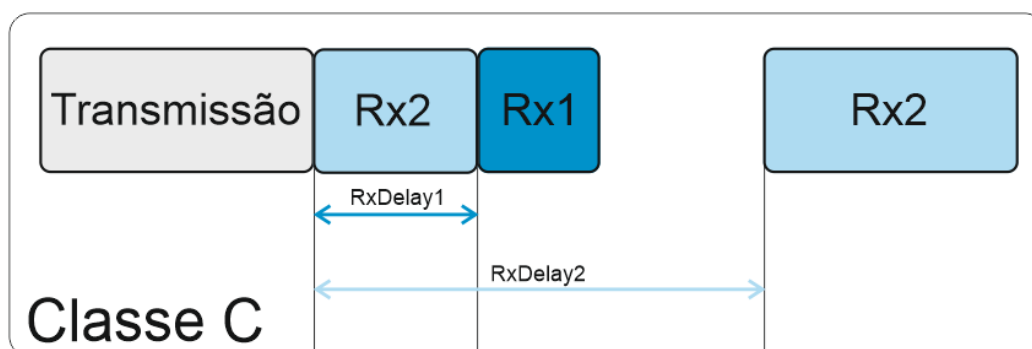


Figura 3.18 - Esquema de transmissão e recepção classe C LoRaWAN

Nos exemplos anteriores não se refere, mas as janelas de recepção $Rx1$ (e $Rx2$ para classe A) estão apenas abertas durante um curto espaço de tempo, onde os rádios escutam pelo preâmbulo das tramas físicas. Ou seja, um recetor não está à escuta durante 1 segundo em $Rx1$, mas sim alguns milissegundos após $RxDelay1$. Este necessita de pelo menos 5 símbolos de preâmbulo para fazer a deteção da transmissão, sendo que por norma este tem 8 ou mais. Assim, o recetor deve abrir uma janela que permita escutar pelo menos 5 símbolos, sendo este valor configurado nos rádios LoRa e sem possibilidade de ajuste via LoRaWAN. Como cada dispositivo tem o seu relógio, basta uma variação temporal maior e deixa de ser possível detetar um preâmbulo [114].

3.2.4 CARACTERÍSTICAS DA REDE

Em variada literatura é referida a elevada capacidade da rede, sendo que esta deriva da utilização de *transceivers* multicanal nas *gateways* e da técnica *adaptive data rate* (ADR). Fatores como a frequência de transmissão (quantidade de vezes que o nó transmite), débitos binários (e respetivo *time on air*), tamanho do *payload* e canais concorrentes afetam esta capacidade. Os *transceivers* multicanal, permitem escutar e receber dados de diversos canais em simultâneo. Uma *gateway* LoRaWAN pode processar diversos canais em simultâneo (diferentes canais e SFs), permitindo assim a receção simultânea de vários pacotes [111]

Os débitos são definidos na especificação LoRaWAN como uma combinação de larguras de banda e SFs, resultando nos chamados códigos Data Rate (**DR**). O mesmo acontece em

termos de potência de emissão, gerando o código *TXPower*, conforme se pode ver na Tabela 3.5.

DataRate (DR)	Configuração	Débito binário(bps)	TXPower	Potência (dBm) (1.0.2)	Potência (dBm) (1.1)
0	LoRa: SF12/125 kHz	250	0	20	Max EIRP
1	LoRa: SF11/125 kHz	440	1	14	Max EIRP – 2 dB
2	LoRa: SF10/125 kHz	980	2	11	Max EIRP – 4 dB
3	LoRa: SF9/125 kHz	1760	3	8	Max EIRP -6 dB
4	LoRa: SF8/125 kHz	3125	4	5	Max EIRP -8 dB
5	LoRa: SF7/125 kHz	5470	5	2	Max EIRP – 10 dB
6	LoRa: SF7/250 kHz	11000	6	-	Max EIRP -12 dB
7	FSK / 500kHz	50000	7	-	Max EIRP – 14 dB
8 ... 14	RFU				
15	Utilizado no comando MAC LinkADRReq		15	Utilizado no comando MAC LinkADRReq	

Tabela 3.5 - Códigos de débito DR e potência de emissão TXPower

Em relação a TXPower, a anterior especificação 1.0.2 fixava o valor desta potência, mas na nova especificação 1.1 este valor é dependente das regulamentações do canal. Na Europa, as bandas da regulamentação ETSI M e N têm uma potência máxima de 14 dBm enquanto a banda P tem 27 dBm. Desta forma, a alocação dos canais pela rede pode alterar a potência de transmissão. Obviamente tem de haver capacidade por parte do *hardware* em utilizar essa potência, sendo que a maioria dos equipamentos disponíveis na Europa não ultrapassa os 14 dBm de potência máxima de emissão. Estes códigos limitam as combinações de largura de banda e SF possíveis, sendo que não é possível utilizar SFs superiores a 7 em canais de 250 kHz, por exemplo. Na Europa apenas se utiliza os DR de 0 a 5.

A técnica ADR (*adaptive data rate*) consiste em alterar de forma automática os débitos binários dos nós, de forma a garantir que cada nó utiliza o SF mais adaptado ao seu canal. O servidor de rede é, normalmente, responsável por ordenar esta alteração, sendo necessário garantir o equilíbrio entre a robustez da ligação e os débitos utilizados [91][102].

No entanto em casos onde o canal rádio sofra alterações rápidas de atenuação, por exemplo num nó móvel, a norma define que o próprio nó deve ajustar os seus débitos, tentando sempre minimizar o **ToA** [115]. Se um nó transmite num DR baixo e a receção na *gateway* é boa, avaliado pela qualidade do sinal recebido (utilizando a relação sinal-ruído Lora *LSNR*), pode-

se aumentar o DR baixando o *time on air* e aumentando o débito. Desta forma liberta-se o canal mais rapidamente para outros nós transmitirem dados. Claro que nesse canal, outros nós podem transmitir simultaneamente desde que utilizem um DR diferente. Esta gestão parte do servidor de rede, que após receção de um pacote com boa qualidade, deve enviar um pedido ao nó para alterar o seu DR na próxima transmissão. Mais uma vez, minimizando o tempo necessário para transmitir os dados permite-nos poupar bateria e ganhar autonomia no nó, ao mesmo tempo que ganhamos capacidade na rede. Do mesmo modo, se um pacote foi recebido com uma potência *received signal strenght indication (RSSI)* alta, significa que podemos baixar a potência de emissão do nó, de forma a poupar alguma energia na transmissão. Ao mesmo tempo minimiza-se a ocorrência do problema *near-far*. Esta gestão é feita de forma individual para cada nó pelo NS [107]. Estas alterações são pedidas através de comandos MAC, responsáveis por modificar e pedir configurações do nó. Apenas se aconselha a utilização da técnica ADR em nós fixos, sendo que em nós móveis não se deve utilizar, definindo nestes casos valores de DR estáticos.

Uma rede LoraWAN tem de implementar 3 canais obrigatórios de 125 kHz, nas frequências 868.1, 868.3 e 868.5 MHz, onde as *gateways* estarão à escuta. Os 125 kHz apenas permitem utilizar os DR de 0 a 5. Assim, as mensagens de pedido de ativação na rede (chamadas *join-request*) são sempre transmitidas num destes três canais. No entanto a rede pode ter mais 5 canais adicionais, em frequências escolhidas pelo utilizador e nas quais se podem utilizar canais de maior largura de banda, e por consequência os DR 6 e 7 [104]. Dentro dos canais disponíveis, os dispositivos devem escolher o canal para a transmissão de cada mensagem de forma pseudoaleatória, implementando assim a diversidade de utilização de frequências, tornando a ligação mais robusta, ao mesmo tempo que diminuem o *duty cycle* do nó.

O tempo de transmissão é limitado pela regulamentação ETSI. Para garantir o respeito da norma. A cada transmissão LoRaWAN num canal, não se pode voltar a transmitir nesse mesmo canal durante um determinado tempo calculado pela fórmula:

$$T_{off\ banda} = \frac{ToA}{DC_{banda}} - ToA \quad [s] \quad (19)$$

Nesta fórmula DC_{banda} representa o *duty cycle* para a banda em questão e ToA o tempo de transmissão. Assim, é preciso considerar que consoante os DR e tamanho dos *payloads* que se pretendam transmitir, existirá sempre um tempo mínimo durante o qual não se poderá voltar a transmitir, nesse mesmo canal. Existirá também um número máximo de mensagens que se podem enviar por hora [104]. A Figura 3.19 mostra a quantidade possível de transmissões por

hora, num canal, consoante o DR escolhido (de 0 a 6, o DR7 não é LoRa mas sim FSK) e o tamanho do *payload*. É também importante referir que na Europa não se utilizam valores de DR superiores ao 5. É fácil perceber que deveremos limitar ao mínimo essencial, o tamanho do *payload*, assim como a taxa de envios, para nos garantir mais margem na capacidade de rede. No caso de existirem muitos nós, torna-se de extrema importância dimensionar corretamente estes valores, para minimizar a taxa de erros e colisões. Se houver necessidade de aumentar a capacidade de rede podemos aumentar o número de canais, desde que se respeitem as normas e o recetor tenha a capacidade de os descodificar.

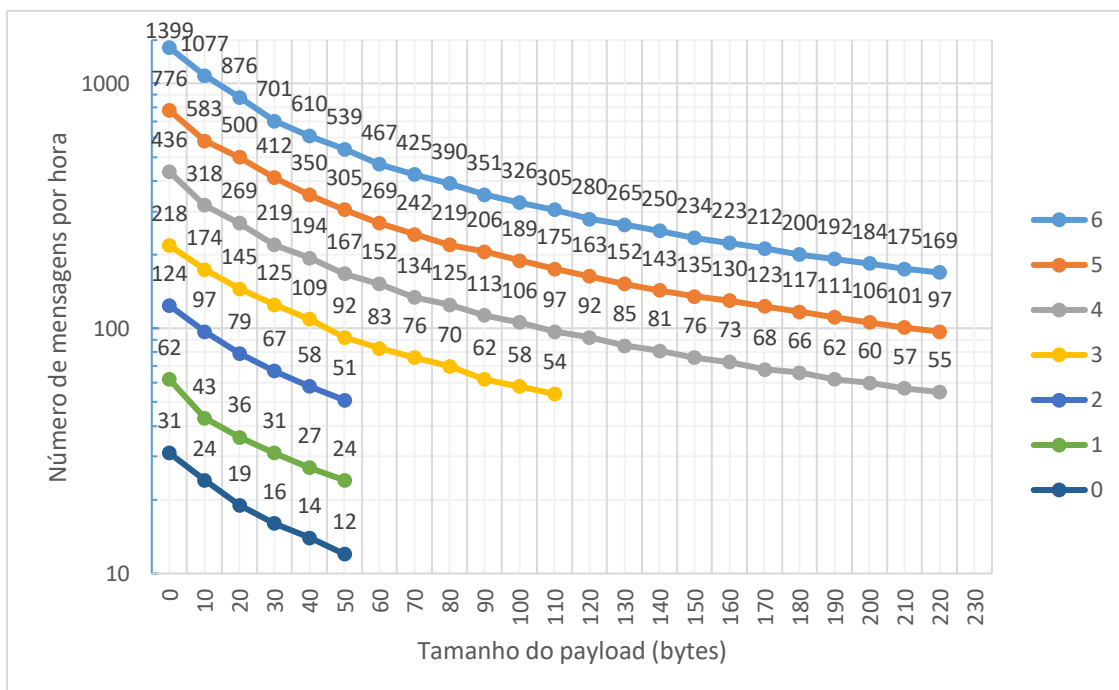


Figura 3.19 - Número máximo de transmissões por hora consoante o valor de DR e tamanho do payload

3.2.5 TRAMA LORAWAN

Os dados e comandos de gestão de rede LoraWAN são colocados no *payload* da camada física (designado de **PHYPayload** em LoraWAN, ver capítulo 3.1.3), sendo o formato deste último, variável consoante o tipo de mensagem. Existem três tipos de mensagens consoante sejam mensagens que transportem dados do utilizador ou de rede (dados no campo **MACPayload**) ou mensagens para a autenticação dos nós na rede (mensagens *join request/accept*, ver 3.2.6.2). Nas mensagens de dados, o **PHYPayload** é composto por 3 campos: 1) cabeçalho MAC com informações de rede (*message header*, **MHDR**), 2) *payload*

de dados de rede e/ou aplicação (**MACPayload**) e 3) código de verificação de integridade (*message integrity code*, **MIC**). As mensagens de *join*, têm um formato distinto de mensagens de dados, consoante sejam um pedido de *join-request* ou uma confirmação do mesmo, *join-accept*. A Figura 3.20 mostra a estrutura da trama física e onde ficam as mensagens da camada LoraWAN [104]. Não iremos aprofundar a descrição da trama MAC, ficando apenas esta descrição da estrutura, útil para se perceber onde são aplicados os mecanismos de segurança.

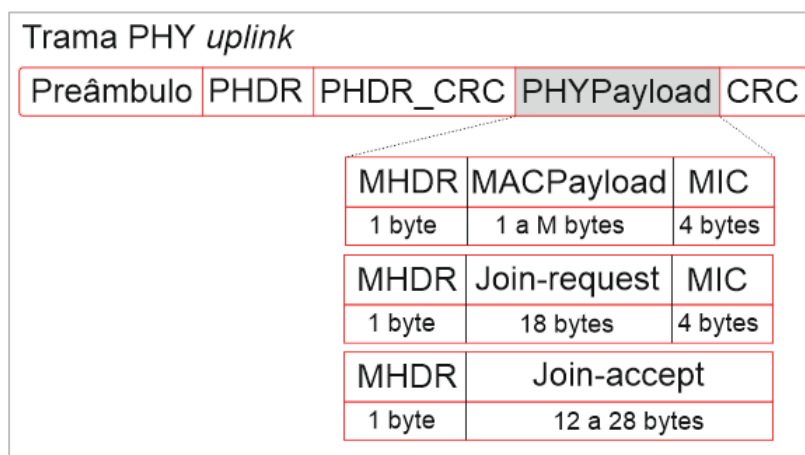


Figura 3.20 - Estrutura da trama e composição do campo PHYPayload

3.2.6 ATIVAÇÃO DOS NÓS

Para que nós finais se possam ligar a uma rede LoraWAN, num processo chamado de ativação, existem vários identificadores e chaves necessários a este procedimento. As chaves são utilizadas para encriptar dados ou criar as mensagens de verificação de integridade **MIC**. Quando um nó se encontra configurado num NS, com as suas chaves de segurança, mas nunca se ativou na rede, diz-se que o nó está comissionado. Após a primeira ligação, o nó encontra-se ativo na rede.

Em seguida iremos explicar apenas a versão 1.0, pois esta é a que utilizaremos. Algumas chaves e mecanismos foram alterados na nova versão, mas não os iremos abordar.

3.2.6.1 Activation by personalization, ABP

No método de ativação *activation by personalization* (ABS), o NS, AS e nó têm de ser previamente configurados com as chaves de segurança de forma a permitir a encriptação e

descriptação das mensagens. Adicionalmente, o gestor da rede deve atribuir um endereço ao nó. Estas 3 informações configuram-se nos nós, NS e AS, mantendo-se ao longo da vida do dispositivo. Descrevem-se de seguida as mesmas [72][101][104]:

- **DevAddr**, *device address*. Este é um valor de 32 bits, que permite identificar o nó na rede. É composto por 2 partes, 25 bits que identificam o dispositivo naquela rede e têm de ser únicos, mais 7 bits que permitem identificar a rede. Estes últimos utilizam-se em redes públicas, sendo que em redes privadas podemos utilizar os 32 bits para identificar o nó.
- **NwkSKey**, *network session key*. Chave de rede com 128 bits, utilizada pelo nó e NS para verificar a integridade das mensagens transmitidas. Apenas se utiliza no nó e NS, onde estes a utilizam para calcular uma mensagem de verificação de integridade dos dados (**MIC**).
- **AppSKey**, *application session key*. Chave utilizada para encriptar e descriptar os dados do nó, garantindo a confidencialidade dos mesmos. Apenas se utiliza no nó e AS, garantindo a segurança *end to end* do *payload* e que ninguém consegue descriptar os dados.

Com estes dados configurados, os nós podem enviar dados imediatamente. De notar que neste processo a rede deve ser desenhado pelo utilizador, de forma a garantir os endereços para cada dispositivo. Do mesmo modo, sendo as chaves comuns a todos os dispositivos, é importante garantir que ninguém externo à rede as conhece, ou tornaria a rede vulnerável. Por este motivo, é aconselhado a utilização do método OTAA.

3.2.6.2 Over the air activation, OTAA

No método *over the air activation* (OTAA) é também necessário configurar inicialmente nos nós, NS e AS, algumas chaves de segurança. No entanto, estas são distintas das utilizadas em ABP, sendo descritas de seguida [72][101][104]:

- **DevEUI**, *device extended unique identifier*. Este identificador de 64 bits, similar a um endereço MAC de redes *ethernet*, é único para cada dispositivo e definido no fabrico do mesmo.
- **AppEUI**, *application extended unique identifier*. Este código identifica a aplicação responsável por processar os dados daquele nó.

- **AppKey**, *application key*. Chave de 128 *bits* utilizada para encriptar os dados do utilizador pela aplicação.

Quando um dispositivo se quer ligar a uma rede, existe um processo de *join* onde o nó envia um pedido *join-request*, contendo a **DevEUI**, **AppEUI** e um valor aleatório **DevNonce**, que identifica aquele pedido. O NS deverá confirmar se aquele **DevEUI** é permitido na rede, respondendo em caso afirmativo com um *join-accept*, o qual contém informações como: 1) identificador de rede (**NetID**), 2) endereço do nó (**DevAddr**), 3) valor aleatório para gerar chaves de rede (**AppNonce**), 4) dados sobre o canal rádio como atrasos dos *downlink* (**RxDelay**), débitos (**DLSettings**) e frequências dos canais (**CFList**). Quando o nó recebe estes dados, utiliza-os para gerar 2 chaves de encriptação, a **AppSKey** e **NwkSKey**, de funções idênticas ao explicado para ABP. A partir deste momento, os nós podem enviar dados. Como se pode constatar, as chaves de encriptação são geradas aleatoriamente e para cada nó. Desta forma, mesmo que um nó seja comprometido, o resto da rede mantém-se segura.

3.2.7 SEGURANÇA

As chaves de segurança faladas no capítulo anterior, servem para garantir dois níveis de encriptação distintos. Uma trama LoraWAN pode conter comandos de gestão de rede MAC e/ou dados do utilizador. Assim, de forma a isolar os dois componentes, utilizam-se as duas chaves **AppSKey** e **NwkSKey**, aplicadas aos distintos campos da trama. A Figura 3.21 seguinte mostra uma trama LoraWAN, onde as diferentes encriptações são aplicadas. A chave **AppSKey** é aplicada diretamente aos dados do utilizador (no campo *frame payload* **FRMPayload**), por um algoritmo AES128 em modo contador (AES-CTR). Apenas a aplicação com a mesma chave os poderá desencriptar. A chave **NwkSKey** é utilizada juntamente com informações do nó (**DevAddr**, valores de contadores de mensagens), num algoritmo AES-CMAC, o qual é aplicado a toda a trama. Este processo gera o **MIC**, 4 *bytes* únicos para aquela mensagem, e que garantem ao NS que a mensagem não foi alterada no envio [104].

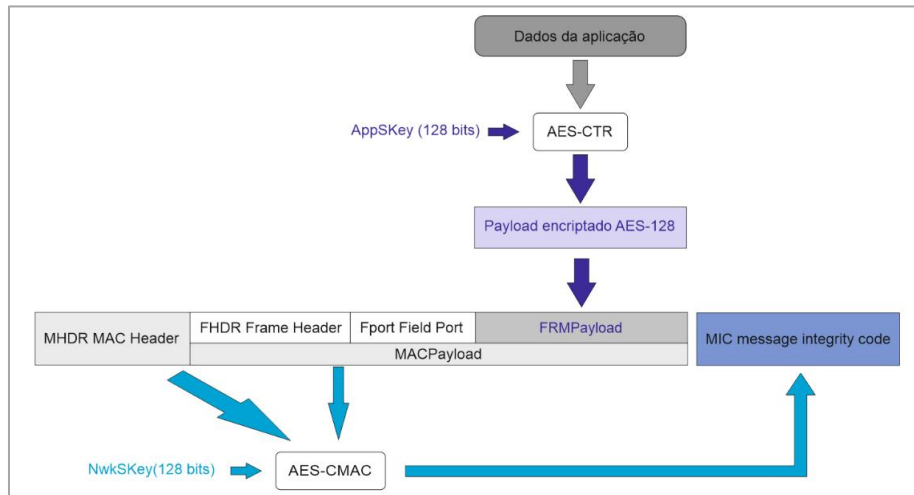


Figura 3.21 - Chaves de segurança LoraWAN e aplicação das mesmas na trama

Um outro mecanismo de segurança importante são os contadores de mensagens existentes em cada nó e NS, designados de *frame counters* (**FCnt**). Cada vez que uma mensagem é enviada pelo nó, o contador de *uplink* **FCntUp** é incrementado. Do mesmo modo, a cada mensagem *downlink* enviada pelo NS, um contador **FCntDown** é incrementado. Estes são utilizados no processo de encriptação, e permitem saber se existiu perda de pacotes de ambos os lados, uma vez que são transmitidos na trama LoraWAN.

3.3 COMUNICAÇÃO PARA REDES BACKHAUL

Estas comunicações *backhaul* são especificadas na implementação de referência, e definem os protocolos a utilizar nas ligações entre as *gateways* e o servidor de rede, ou entre o servidor de rede e as aplicações. Correspondem à camada de transporte do modelo OSI. As ligações de rede e de nível inferior podem suportar-se em qualquer tecnologia (*ethernet*, LTE, 3G, fibra...).

3.3.1 GATEWAY – SERVIDOR DE REDE

3.3.1.1 Datagramas UDP

A comunicação entre a *gateway* e o servidor de rede tem uma implementação de referência definida pela Semtech, que é feita sobre o protocolo UDP, utilizando um porto fixo no NS e arbitrário na *gateway* [116][117]. Podem, no entanto, ser utilizados outros modelos que contenham implementações de segurança ou utilizem ligações TCP, ficando este desenvolvimento a cargo do utilizador. Neste domínio são utilizados dois tipos de mensagens, os *PUSH* e os *PULL*, definidos num protocolo simples designado de *gateway message protocol (GWMP)*. Estas mensagens seguem no campo de dados dos datagramas UDP. Existem mensagens de transporte de dados (*PUSH_DATA* e *PULL_RESP*), de confirmação de entrega (*PUSH_ACK*, *PULL_ACK* e *TX_ACK*) e de informação (*PULL_DATA*). Os dados das aplicações seguem num campo de *payload* das mensagens de *PUSH_DATA* e *PULL_RESP* (*downlinks* e *uplinks* respetivamente), em formato JSON. Não tem nenhum mecanismo de autenticação ou retransmissões de mensagens.

As mensagens *PULL_DATA*, são enviadas regularmente (*keepalive*) e servem para informar o NS do porto UDP que a *gateway* tem à escuta, permitindo manter aberta alguma *firewall* existente no NS. Um NS que receba uma mensagem destas deve responder imediatamente com um *PULL_ACK* ao mesmo porto da *gateway* confirmando desta forma a receção. A Figura 3.22 esquematiza este tipo de transmissões. Na mensagem *PULL_ACK* é devolvido um *token* que foi previamente enviado na mensagem de *PULL_DATA*. O *token* representa um valor arbitrário, criado e enviado previamente pela *gateway*. A *gateway* envia sempre o seu identificador único *gatewayEUI* para o NS saber a quem pertence aquela mensagem.

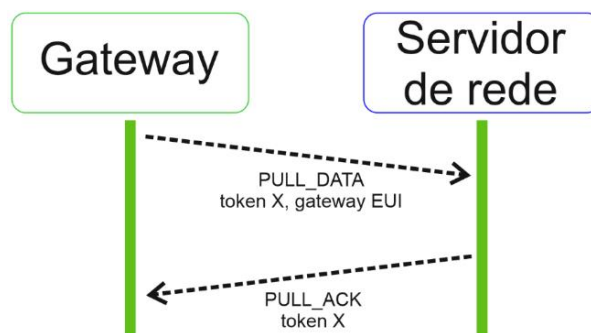


Figura 3.22 - Mensagens keepalive GW-NS

Quando uma *gateway* tem dados provenientes dos nós para enviar ao NS, utiliza uma mensagem *PUSH_DATA*, enviada a partir de um porto aleatório, e contendo os dados do nó em formato JSON. O NS deve confirmar a receção para o mesmo porto, numa mensagem *PUSH_ACK*.

Se o NS tiver dados do AS para um nó, deve enviar uma mensagem *PULL_RESP* contendo os dados em formato JSON à *gateway*. Esta deverá confirmar a receção com uma mensagem *TX_ACK*, caso se utilize a versão 2 do protocolo GWMP. Caso seja a versão 1 não se dá esta resposta. Na versão 2, caso exista algum erro na transmissão, este pode opcionalmente ser transmitido ao NS. A Figura 3.23 esquematiza as duas situações e mostra os formatos das mensagens GWMP trocadas por UDP.

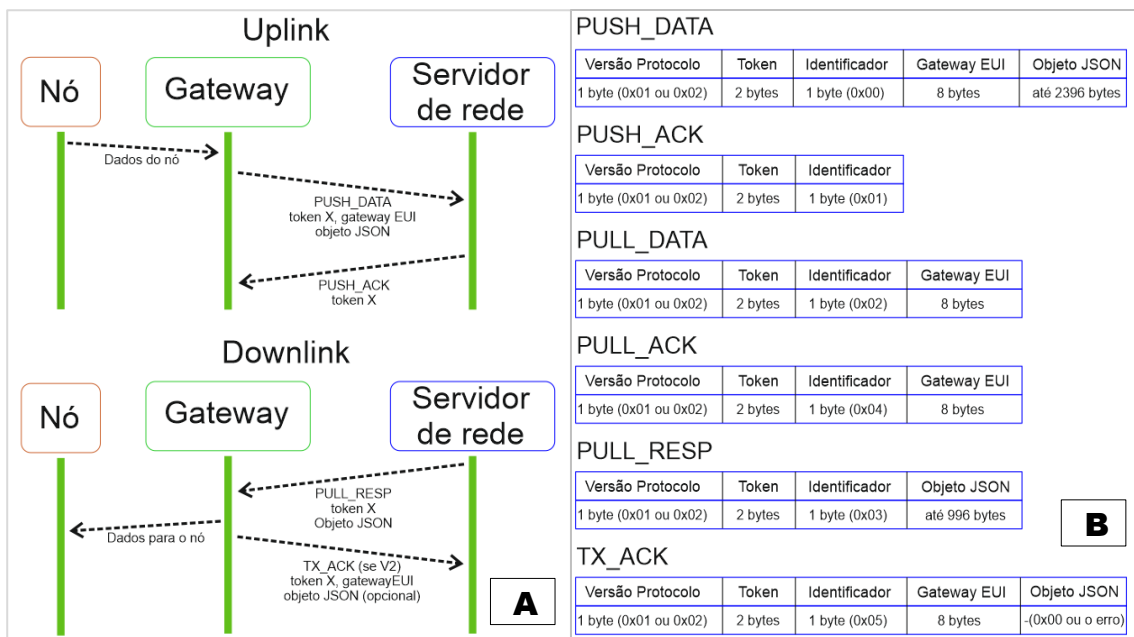


Figura 3.23 - Mensagens de troca de dados em up e downlink (A) e formatos das mensagens GWMP (B)

3.3.1.2 Payloads JavaScripts Object Notation (JSON)

Como referido, os dados transmitidos no payload têm uma formatação *JavaScripts Object Notation* (JSON). A JSON é uma formatação simples utilizada para troca de dados, que utiliza texto legível (caracteres ASCII) sendo por isso independente de qualquer linguagem de programação [118]. Um objecto é simplesmente um conjunto “nome:valor”, colocado entre chavetas e podendo um nome conter vários valores. Por exemplo, o texto seguinte contém dois

objectos *time* e *chan* onde estão data e horas em caracteres (dentro de aspas) e o valor numérico “2”. Não podem existir espaços em branco nem outros separadores:

```
{“time”：“20171110154410”,“chan”：2}
```

Os objectos podem estar organizados numa hierarquia com vários níveis, contendo diferentes nomes em cada nível. Em LoraWAN o nível superior contém diferentes tipos de objectos consoante sejam mensagens de *uplink* ou *downlink*. No *uplink* entre a *gateway* e o NS, têm de existir zero ou mais tipos de objetos *rxpk*, *stat*, *error* ou outros. Estes representam informações do pacote recebido ou estatísticas e informações de erros de transmissão da *gateway*, respectivamente.

```
{“rxpk”：{...},“rxpk”：[{...},{...},...,{...}],“stat”：{...},“other”：{...}}
```

No *downlink* apenas existem objectos tipo *txpk* correspondendo a funções e dados para a transmissão *gateway*-nó. Num segundo nível existem mais informações respectivas a cada objecto do nível superior. Por exemplo, o objecto *rxpk* contém um objecto *freq* onde está a informação da frequência onde a trama foi recebida.

```
{“rxpk”：{“freq”：869.1}}
```

Podem ser agrupadas várias tramas dentro de um mesmo objecto, formando um *array* de objetos. Estes devem ser colocados entre parênteses retos e separados por vírgulas conforme o exemplo seguinte de um *rxpk*:

```
{“rxpk”：[{“time”：“20171110...”},...,{“time”：“20171111...”},...,{“time”：“20171112...”},...]}
```

Os objetos existentes no *downstream* e *upstream* são descritos em [117], onde está a descrição de cada objeto, o tipo de dados que contém e a obrigatoriedade ou não do objeto. O exemplo seguinte mostra uma mensagem recebida, contendo uma trama LoRa de 32 bytes com SF=7 e BW=125 kHz, recebida sem erros na frequência 866.349812 MHz com RSSI=-35dBm e SNR=5.1dB.

```
{“rxpk”：{“time”：“2017-11-01T16:21:17.528002Z”,“tmst”：3512348611,“chan”：2,“rfch”：0,“freq”：866.349812,“stat”：1,“modu”：“LORA”,“datr”：“SF7BW125”,“codr”：“4/6”,“rssi”：-35,“lsnr”：5.1,“size”：32,“data”：“-DS4CGaDCdG+48eJNM3Vai-zDpsR71Pn9CPA9uCON84”}}
```

3.3.2 SERVIDOR DE REDE – OUTROS SERVIDORES

Na comunicação entre servidor de rede e servidor de aplicações tem de ser utilizado o protocolo TCP de forma a garantir ligações fiáveis. Os dados devem seguir no *payload* TCP formatados em JSON, sem encriptação adicional à encriptação LoRaWAN. Existem bastantes

objetos descritos em [117], cobrindo todos os tipos de informação que o protocolo permite enviar. A versão 1.1 veio estabelecer que todos os elementos da rede devem implementar soluções de segurança para esta comunicação (por exemplo, ligações *hyper text transfer protocol secure*, HTTPS), ficando a cargo de quem está a desenvolver a rede a escolha da melhor solução, desde que respeite a utilização TCP e JSON [119].

4. DESENVOLVIMENTO DO TRABALHO

No capítulo anterior foi analisado o essencial sobre o funcionamento de LoraWAN e os componentes necessários para implementar uma rede nesta tecnologia. Neste capítulo iremos explicar a proposta de rede a implementar, com a análise do material escolhido assim como descrever os componentes desenvolvidos e utilizados. Analisaremos também a micro rede onde o sistema será aplicado, de forma a especificar as características que devemos assegurar.

4.1 MICRO REDE INTA

Para definirmos as características da rede WSN a desenvolver analisaremos em seguida as particularidades da micro rede da INTA, um dos locais onde será aplicado o sistema. Esta micro rede contém diversos componentes, estando identificados os pontos chave de medição de variáveis. Nesta rede não se pretende efetuar nenhuma atuação, apenas medição. A Figura 4.1 apresenta a estrutura da rede, assim como as respetivas identificações dos pontos de medida de variáveis elétricas. Esses pontos são identificados pelas letras “u” ou “i”, traduzindo respetivamente pontos de leitura de tensão ou de corrente. A imagem representa os circuitos em corrente contínua (DC, a cor azul) e a rede em corrente alternada (AC, a vermelho tracejado), assim como os diferentes componentes como inversores, geração (solar, eólica e

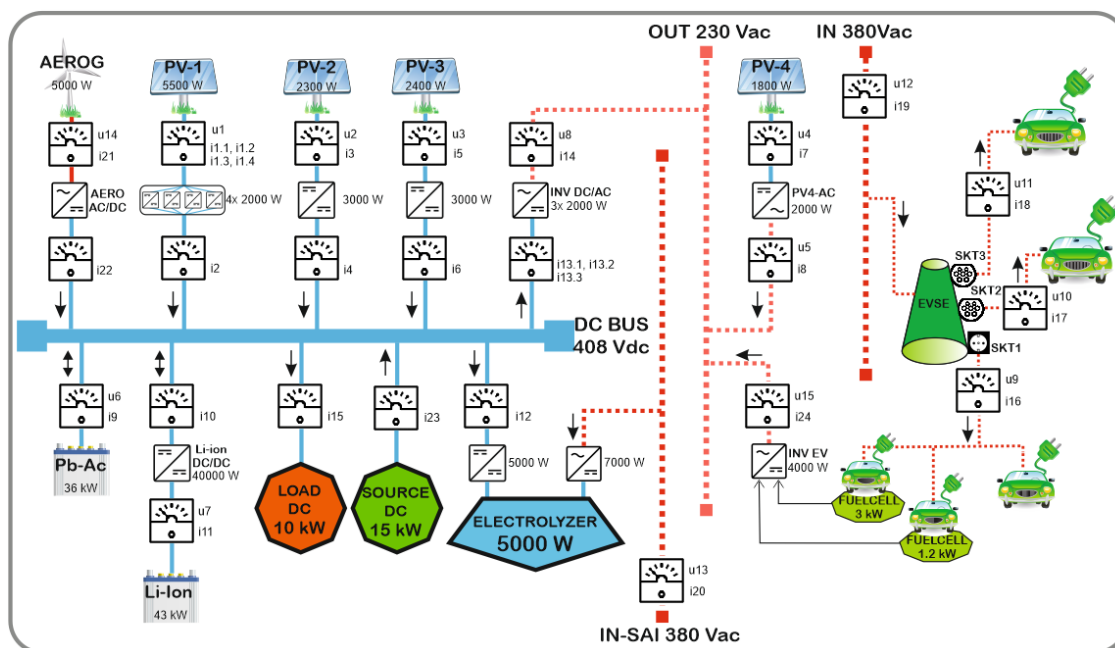


Figura 4.1 - Esquema da micro rede INTA

hidrogénio), armazenamento (baterias Pb-Ac, Li-Ion ou *fuelcell* nos veículos elétricos movidos a hidrogénio) e cargas (carregamento de veículos elétricos, o eletrolisador de hidrogénio, uma carga controlável).

A rede interna é composta por um barramento DC e três barramentos AC, 2 trifásicos (IN e IN-SAI) e 1 monofásico (OUT). O barramento DC alimenta o barramento AC monofásico, através de um inversor (INV DC/AC). Existem 4 campos fotovoltaicos (PV-1, 2, 3 e 4) de diferentes potências, sendo que cada um inclui um inversor para adaptar as tensões elétricas da geração às do barramento onde esteja ligado. É assim necessário medir as correntes de entrada e saída destes, assim como as tensões, de forma a conhecer o estado dos mesmos. Existe um aerogerador AC (AEROG), o qual injeta corrente no barramento DC através de um conversor (AEROG AC/DC). O armazenamento é feito em 2 bancos de baterias de tecnologias de chumbo (Pb-Ac) ou iões de lítio (Li-Ion). As primeiras ligam diretamente ao barramento DC, enquanto as segundas necessitam de um conversor para adaptar as tensões distintas (Li-Ion DC/DC). Estas baterias permitem armazenar ou fornecer energia ao barramento, em determinadas condições definidas pela INTA e fabricante. Existe uma fonte externa, permitindo alimentar o barramento DC (*Source DC*), assim como uma carga regulável que permite simular perfis de carga na rede (*Load DC*). O barramento DC juntamente com um trifásico, alimentam um eletrolisador de hidrogénio, onde se gera este elemento, permitindo alimentar 2 veículos elétricos híbridos com célula de combustível. Estes podem adicionalmente fornecer energia à rede monofásica AC, gerando energia a partir do hidrogénio e injetando-a através de um inversor (INV EV). Existe também uma estação de carregamento de veículos elétricos (EVSE) com 3 tomadas, duas delas trifásicas IEC62196 do tipo 2 (SKT 2 e 3) e uma monofásica Schuko (SCK1).

Esta micro rede está toda implementada em locais relativamente próximos, não sendo a distância um fator significativo. No entanto, pretende-se futuramente monitorizar a tensão da bateria (e eventual *state of charge*, SOC) dos veículos a pilha de combustível. O alcance elevado pretendido, advém exatamente da distância máxima que estes veículos podem percorrer no recinto. Durante o executar desta tese não foi possível avançar com esta monitorização, pelo que não existirá nenhum módulo para este fim. No entanto, os módulos da rede sem fios devem considerar as distâncias que os veículos podem percorrer dentro da INTA. Desta forma os módulos DC desenhados para leitura de tensão, funcionarão sem problemas, bastando redimensionar as resistências. Esta integração nos veículos, será um dos passos futuros do projeto.

4.2 ARQUITETURA DA REDE WSN

Ao desenhar a arquitetura WSN, foram considerados todos os componentes necessários e a possível integração entre eles. Se criar uma rede LoraWAN de forma autónoma seria fácil, integrar os dados desta para outro tipo de rede apresentaria algumas particularidades. Toda a estrutura foi pensada de forma a permitir a utilização de material diverso, não ficando dependentes de nenhum fabricante nem tecnologia. De seguida descrevemos, resumidamente, o material LoraWAN existente no mercado e que analisamos para definir a arquitetura da rede de sensores.

4.2.1 ESCOLHA DE MATERIAL LORAWAN

Para escolher que material utilizar para implementar a rede WSN, foram analisados vários componentes existentes no mercado no final de 2017. No Apêndice C encontra-se a descrição mais pormenorizada e análise do material, com as características técnicas de cada um. Atendendo às necessidades da rede, os nós deveriam ser de baixo custo e *open source*. A capacidade de ser classe A e C LoraWAN, seria uma mais valia. Assim, acabou-se por escolher os módulos Lopy da Pycom. O baixo custo (38 euros com antena), assim como a versatilidade de entradas, saídas e *interfaces*, LoraWAN A e C, a quantidade significativa de memória (comparativamente aos outros módulos) e a programação em MicroPython foram pontos fortes. Adicionalmente, a existência de Wi-Fi e Bluetooth, permitiria uma versatilidade futura nas ligações (ver Apêndice C.1).

Para a *gateway* a opção acabou por ser a placa concentradora IMST iC880, ligada a um Raspberry Pi3. Foram analisadas as soluções descritas no Apêndice C.2. O facto de deter 8 canais, ser *open source* e o baixo custo foram decisivos na sua escolha. Adicionalmente, esta solução permitiria integrar o servidor de rede, num mesmo equipamento. Manteve-se a opção de utilizar a *gateway* Laird Sentrius RG186 para a implementação na INTA, de forma a ter um equipamento mais fiável a longo prazo, visto que esta também apresenta um baixo custo.

Em relação ao servidor de rede, foram instalados e ensaiados os dois descritos no Apêndice C.3, acabando por se verificar que o Lorawan-Server seria melhor escolha pela versatilidade das ligações externas e por ser computacionalmente leve, permitindo ser executado no Raspberry Pi 3.

Para a visualização e armazenamento de dados, a escolha recaiu sobre a plataforma Thingboard, devido às funcionalidades mais avançadas que esta apresenta por comparação com a Freeboard. Entretanto, a plataforma Freeboard inicialmente gratuita, passou a ser paga, não nos deixando por isso mais soluções. Para além disso, a Thingboard é uma solução *open source* que pode ser instalada num qualquer computador ou RPi, o que permitir-nos-ia ter uma plataforma de visualização local, numa rede privada e não dependendo assim de serviços externos ou com custos acrescidos. A possibilidade de criação de utilizadores, a definição de zonas, os alarmes ou as *dashboards* de visualização interativas, foram pontos considerados importantes. A análise deste material encontra-se no Apêndice C.4.

4.2.2 ESQUEMA DA REDE DE SENSORES

A utilização dos módulos Lopy permite ter no mesmo aparelho WiFi, BLE e LoraWAN, sendo por isso bastante versátil. Do mesmo modo a utilização de componentes e protocolos de código aberto torna-se uma mais valia em termos de adaptabilidade das soluções. A nível de componentes necessários, consideram-se os elementos obrigatórios de uma rede LoraWAN (nós, *gateway*, servidores de rede e aplicações) juntamente com nós ou componentes que não são LoraWAN. Nestes últimos englobam-se a rede *backhaul* e respetivos dispositivos, assim como uma estação meteorológica que nos permite conhecer as condições meteorológicas no local e em tempo real. Adicionalmente, incluiu-se a possibilidade de integrar nós WiFi. Na Figura 4.2 estão representados os componentes da rede e a estrutura pensada para a mesma, assim como os protocolos e sentidos de comunicação utilizados entre os diferentes componentes. Dentro do quadrado tracejado cor de laranja encontra-se representada a rede de sensores e *gateway* LoraWAN, onde todas as comunicações assentam na tecnologia LoRa. Estão representados dentro deste quadrado os nós sensores e respetivos pontos de medida, de qualquer tipo (tensão, temperatura, etc). Os nós efetuam o envio de dados a cada 20 segundos, contendo uma média das leituras dos diversos sensores, leituras estas efetuadas a cada 2 segundos. A caixa verde representa a *gateway*/servidor de rede, obrigatório numa rede LoraWAN. Neste caso, juntamos tudo num componente, um RaspberryPi com uma placa concentradora LoraWAN. Este liga a um servidor de aplicações distinto, implementado num outro RaspberryPi e representado pelo quadrado azul. Esta ligação, representada pela seta de cor preta, assenta numa comunicação via rede IP. Adicionalmente, o servidor de rede permite receber dados vindos de um servidor de agendamentos, representado dentro do quadrado

castanho, o qual será implementado numa máquina independente e ligado por rede IP. O objetivo deste é enviar pedidos de atuação aos nós sensores. Este servidor e os mecanismos de agendamento não serão implementados no trabalho desta tese, ficando apenas o sistema pronto para receber estes dados e atuar na rede LoraWAN. Podemos também ter dispositivos ligados ao AS via rede IP, representados dentro de quadrados creme. De forma a garantir acesso remoto aos dados, o AS será ligado à Internet, disponibilizando assim os dados em qualquer lugar e a qualquer utilizador que tenha credenciais de acesso.

Para cada um destes componentes foram analisadas as opções de comunicação existentes. Se entre nós, *gateway* e servidor de rede iríamos utilizar a comunicação definida na norma LoRaWAN, entre servidor de rede e aplicações a escolha dependeria do que estes componentes permitissem. Assim, na análise dos mesmos, verificamos a disponibilidade comum dos protocolos *hyper text transfer protocol* (HTTP) e *message queieuing telemetry transport* (MQTT). Optamos pela utilização simultânea destes dois protocolos, uma vez que, por exemplo, a estação meteorológica apenas permitia a utilização do HTTP. No entanto, sempre que possível, iríamos utilizar o MQTT pois é um protocolo mais eficiente. O funcionamento deste encontra-se descrito resumidamente no Apêndice J. Tivemos de incluir um componente adicional no servidor de rede, o chamado *broker* MQTT, necessário para encaminhar os dados do servidor de agendamentos. Na figura encontram-se descritos os protocolos utilizados entre cada componente.

Para aplicação nas instalações da INTA e no âmbito final do projeto Agerar, considerar-se-á outra estação meteorológica e outro AS, de outros parceiros do projeto. Assim estes

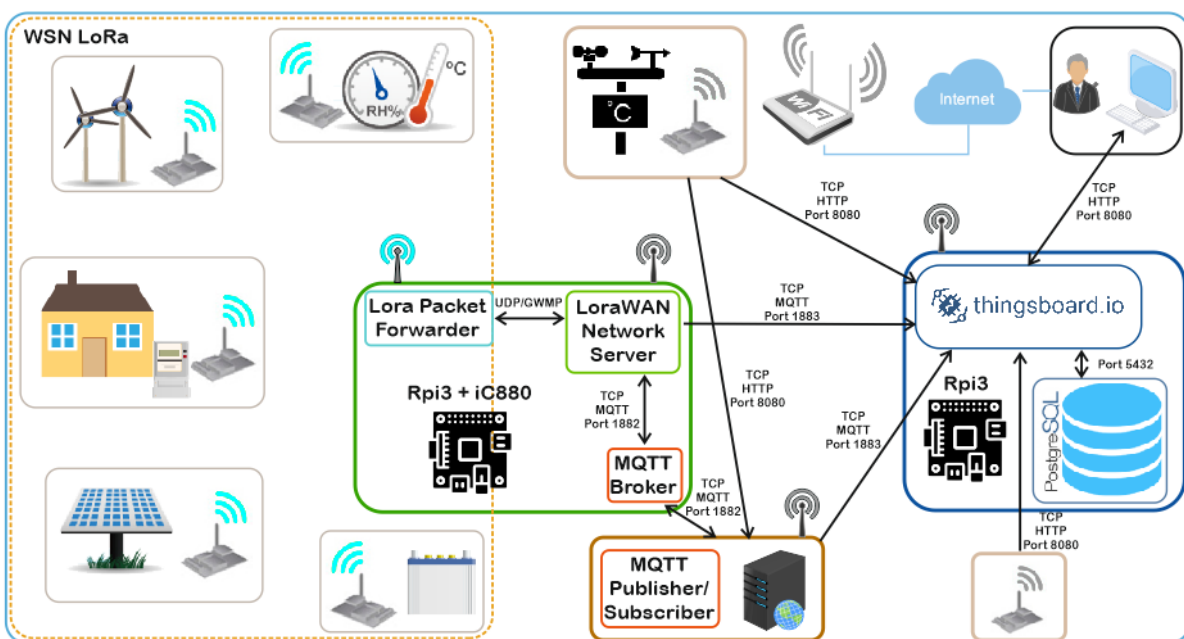


Figura 4.2 - Arquitetura WSN proposta

elementos constam aqui para que se possa testar localmente uma solução completa, antes da final, permitindo a realização de ensaios durante a implementação. No capítulo seguinte descrevem-se os diversos componentes e a interligação entre eles.

4.3 COMPONENTES DA REDE WSN

Nesta proposta de arquitetura consideraram-se os seguintes componentes:

- Nós sensores (LoraWAN ou rede IP);
- *Gateways* (LoraWAN);
- Servidor de rede (NS, LoraWAN);
- Servidor de aplicações (AS, rede IP);
- *Router* para *backhaul* (rede IP);
- Servidor de escalonamentos (rede IP).

Ir-se-á utilizar um único RPI3 que agrega as funções de *gateway* e NS. Por ser uma rede de pequena dimensão com poucos nós e ao necessitar de uma única *gateway* não foi necessário utilizarem-se máquinas distintas. Ao mesmo tempo a utilização de pequenas máquinas garante portabilidade dos equipamentos para testes em várias localizações. No entanto, a solução pode manter a escalabilidade, adicionando mais nós e *gateways*, e onde os NS e AS podem correr em máquinas com mais capacidade de processamento. No âmbito proposto os nós serão colocados maioritariamente em locais com disponibilidade de energia, não aproveitando nem requerendo a capacidade *low power* da tecnologia. No entanto esta poderá ser introduzida sem qualquer alteração ou configuração adicional na rede, bastando apenas utilizar nós com essa capacidade.

4.3.1 NÓS SENSORES

Decidiu-se separar os nós em 2 tipos, consoante as suas ligações de rede. Assim poderemos ter nós dos seguintes tipos:

- Nós LoraWAN, compostos pelos módulos Lopy;
- Nós WiFi, os quais estão ligados apenas a esta rede sem fios. Estes podem ser módulos Lopy, Esp8266, Arduinos com *shield* WiFi ou quaisquer outros módulos;

Aqui não incluímos a descrição da funcionalidade de cada nó, uma vez que esta depende dos transdutores ligados ao nó. Serão criados nós para diferentes medições, em quaisquer destes 2 tipos. Por exemplo, um sensor que esteja a ler dados de uma linha Modbus pode enviar por LoraWAN ou WiFi, consoante as suas capacidades de rede e o local de implementação. Na micro rede INTA apenas iremos utilizar nós LoraWAN.

4.3.1.1 Lopy

Tal como descrito anteriormente optou-se por utilizar os microcontroladores Lopy nos nós. O Apêndice A.C.1.3 contém mais informação destes dispositivos. A utilização dos Lopys permitir-nos-á utilizar LoraWAN e WiFi, assim como Bluetooth, caso exista alguma necessidade adicional. A programação é feita escrevendo os *scripts* no *software* Atom, e enviando-os ao nó, utilizando o *plugin* Pymakr. Adicionalmente poder-se-ia optar por enviá-los por FTP, no entanto o método anterior permite avaliar os comandos, numa linha de comandos REPL (*Read, Evaluate, Print, Loop*), conforme se pode ver na Figura 4.3. Esta, similar a uma linha de comandos, encontra-se integrada no Atom, e permite interagir com o módulo em tempo real, facilitando desta forma o teste de códigos. Obviamente isto é possível, pois a linguagem MicroPython não é compilada.

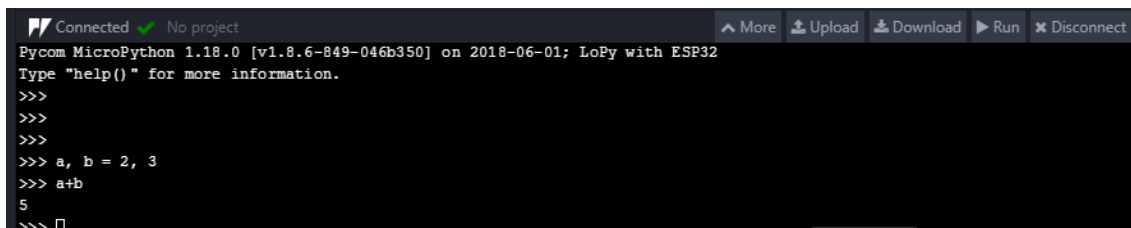
A screenshot of the Atom editor's REPL console. The window title is "Connected No project". The console shows the following text: "Pycom MicroPython 1.18.0 [v1.8.6-849-046b350] on 2018-06-01; LoPy with ESP32", "Type 'help()' for more information.", and a series of prompts ">>>" followed by user input and output: "a, b = 2, 3", "a+b", and "5". The console also shows a "More" menu with options for "Upload", "Download", "Run", and "Disconnect".

Figura 4.3 - Consola REPL, integrada no editor de texto Atom

Nos nós LoraWAN, os formatos dos *payloads* e tipo de dados, deveriam ser bem definidos para a aplicação em questão. Para posterior filtragem de dados no NS, este conhecimento torna-se relevante, não sendo nenhuma imposição da tecnologia. Nos nós WiFi não se coloca esta restrição, pois enviam os dados diretamente do nó para a plataforma de visualização. Apesar disso, estes têm de ser também formatados de forma correta, para que o AS os reconheça.

Adicionalmente, os nós LoraWAN podem receber mensagens vindas do NS. Estas mensagens *downlink*, que podem ser enviadas a qualquer dispositivo ativo na rede, devem ser minimizadas de forma a não ultrapassar o *duty cycle* da *gateway*. Se pretendermos minimizar

a latência e alcançar uma atuação mais rápida, deveremos configurar os respetivos nós como classe C. Como a maioria dos nossos sensores não utilizam baterias, não será um problema mantê-los à escuta. No caso de se pretender ter nós alimentados a baterias devemos configurá-los apenas como classe A. De qualquer forma, é sempre boa prática minimizar o tráfego de *downlink* em LoraWAN. A nível de nós WiFi esta restrição não se aplica.

Em relação ao microcontrolador, a forma mais fácil de o programar é através de uma placa de expansão. Estas existem com várias funcionalidades, nomeadamente com a possibilidade de disponibilizar todos os pinos do módulo para o exterior, e/ou de permitir ligar um cartão SD e uma bateria externa. Existe também uma placa com GPS e acelerómetro, que não disponibiliza de origem o acesso aos pinos. A Figura 4.4 mostra estes dispositivos.



Figura 4.4 - Placa de expansão (esquerda), placa GPS/Acelerómetro (centro) e Lopy (direita)

4.3.1.2 Antenas

Nos nós LoraWAN utilizamos as antenas oficiais dos módulos Lopy. Estas são antenas de exterior monopólo tipo *whip* específicas para 868 MHz, com 2.2 dBi de ganho e um conector RP-SMA (*Reverse Polarity SubMiniature version A*) macho. Para a ligação ao Lopy precisávamos de um pequeno adaptador RP-SMA fêmea para u.FL, designado de *pigtail*. Este vem incluído no conjunto, mas não permite a utilização no exterior. Nos módulos Lopy, a antena WiFi está embutida no dispositivo. A Figura 4.5 mostra as duas antenas.

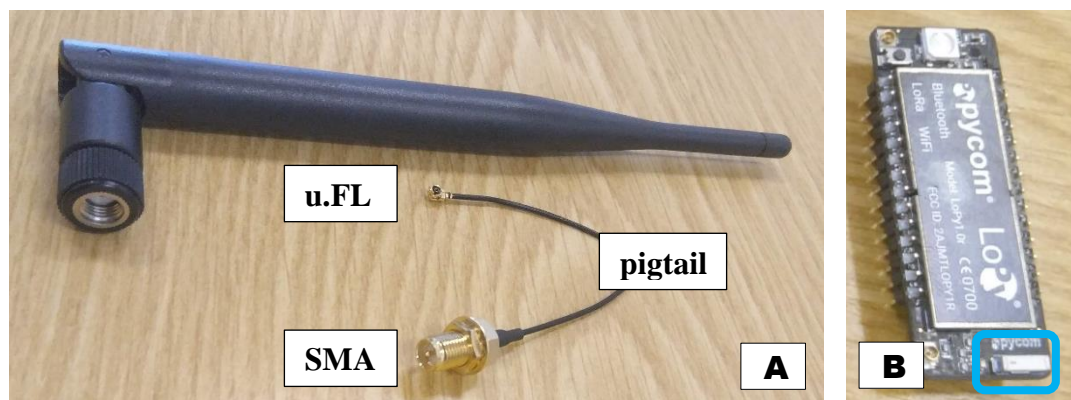


Figura 4.5 – Antena e pigtail Lora (A), antena WiFi embutida (B)

4.3.2 GATEWAY

Para a *gateway*, foi escolhido o método do concentrador Lora ligado a um Raspberry Pi3, pela facilidade de configuração e possibilidade de integrar o servidor de rede NS no mesmo equipamento. Uma vez que nesta fase inicial apenas se pretendia utilizar uma única *gateway*, esta solução seria mais prática que dispor de uma máquina autónoma a fazer de NS. Mesmo no caso de várias *gateways*, seria possível correr o servidor no RPI, mas à custa de uma sobrecarga do mesmo, que poderia resultar numa falha de rede, em casos de tráfego elevado. Descrevemos resumidamente de seguida, os diversos componentes e a configuração dos mesmos.

4.3.2.1 Raspberry Pi3

Este microcomputador foi a escolha para correr a *gateway* devido à sua disponibilidade, baixo custo e facilidade de utilização. Não se pretende descrever este componente, deixando apenas a nota que se instalou o sistema operativo Raspbian Stretch. Após esta instalação, foram apagados todos os programas que não iriam ser necessários, mantendo o ambiente gráfico ativo e o acesso SSH. Foi ativada a *interface* SPI e definido um IP estático nesta máquina. A necessidade de intercomunicação entre NS-AS, leva a que seja necessário garantir os IPs nestas duas máquinas.

O Apêndice D descreve pormenorizadamente a instalação e configuração do mesmo.

4.3.2.2 IMST iC880A-SPI

Este componente apresenta-se numa pequena placa baseada em dois recetores rádios Semtech SX1257, ligados a um processador de banda base Semtech SX1301. Garantem uma sensibilidade de receção de -138 dBm e potência de emissão de até 20 dBm [120]. Os chips SX1257 são responsáveis pela receção dos sinais vindos da antena e transformação destes num formato de dois componentes, representando a fase e quadratura do sinal recebido (In-phase/Quadrature I/Q) [121]. Estes são enviados ao SX1301, o qual tem de desmodular os diversos componentes e guardar os dados obtidos numa fila, para posterior envio ao *host* ao qual está ligado. Na transmissão segue-se o processo inverso. Este envio ao *host* é feito via *Serial Peripheral Interface* (SPI), num protocolo *master-slave* síncrono e *full duplex* que utiliza 4 fios para comunicação. Neste protocolo existe sempre um dispositivo *master* e um ou vários *slaves*. O sincronismo é garantido por uma linha de relógio *serial clock* SCLK comum a todos os dispositivos e cada *slave* tem de ser habilitado por uma linha *slave select* SS. Os dados são trocados nas linhas *master output slave input* (MOSI) e *master input slave output* (MISO), num esquema como o da Figura 4.6 [122].

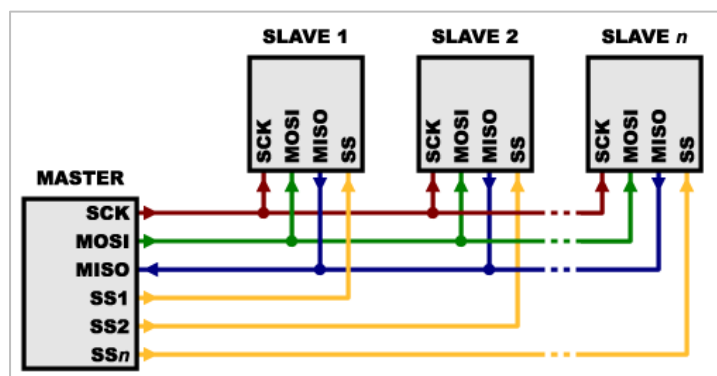


Figura 4.6 - Ligações SPI

O SX1301 permite a receção de 10 canais em simultâneo, no entanto apenas 8 são utilizáveis por LoraWAN. A utilização destes 8 canais, possibilita a deteção e respetiva desmodulação simultânea de 8 pacotes de dados, independentemente do DR utilizado. Os outros 2 canais utilizam larguras de banda ou modelação não utilizadas em LoraWAN (ex, 500 KHz ou FSK) [123]. A escolha destes *chips* permite a utilização do código fonte de referência *hardware abstraction layer* HAL da própria Semtech, que lida com as comunicações e controlo da placa pelo *host*. É designado de *lora_gateway* e está disponível em [124]. Inclui, para além da biblioteca base *libloragw*, uma série de programas de teste do concentrador como testes de comunicação SPI, transmissão ou receção de dados.

O concentrador, apresentado mostra a Figura 4.7, deve ser alimentado com 5 V e necessita de uma corrente máxima de 700 mA (embora o consumo anunciado seja inferior). Pode ser soldada uma ficha para alimentação externa ou alimentada pelos pinos 21 e 22, da placa. Inclui 7 leds de indicação do estado de funcionamento, conectores u.FL para antena Lora e GPS, pontos de solda para módulo GPS ou suportes adicionais de antena (não incluídos). Do lado inferior existem 26 pinos com diversas funções, como a *interface* SPI.

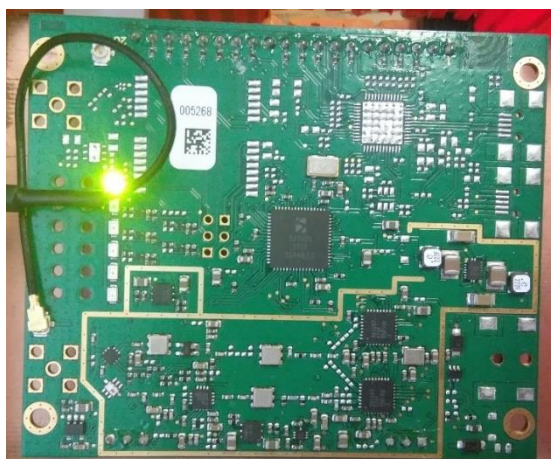


Figura 4.7 - Placa IMST iC880A-SPI

4.3.2.3 Shield Raspberry Pi

Uma vez que o concentrador não inclui a placa de ligação ao Raspberry Pi, esta foi criada, seguindo as ligações de referência do fabricante. Essas indicações especificam que juntamente com as linhas SPI devemos ligar uma linha de *reset*, que serve para reiniciar corretamente o módulo. Adicionalmente a alimentação foi obtida nos pinos de 5 V do Raspberry. Consultando a documentação do RPI3 sabemos que o mesmo consome uma corrente de aproximadamente 750 mA. Como estamos a utilizar uma fonte de 2,5 A, sobra-nos corrente suficiente para alimentar os 700 mA do IC880, sendo esta a escolha para alimentação. A Figura 4.8 mostra o esquema de ligações utilizado.

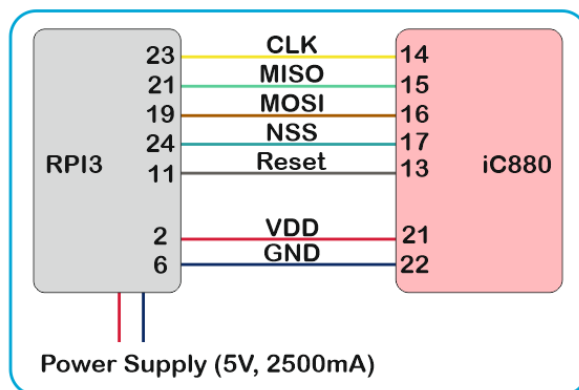


Figura 4.8 - Ligações RPi3 iC880A-SPI

Foi criado um protótipo inicial com uma placa perfurada que serviu para os primeiros testes. Posteriormente, criou-se uma pcb protótipo com as devidas ligações entre o RPi3 e o concentrador, conforme as imagens da Figura 4.9.

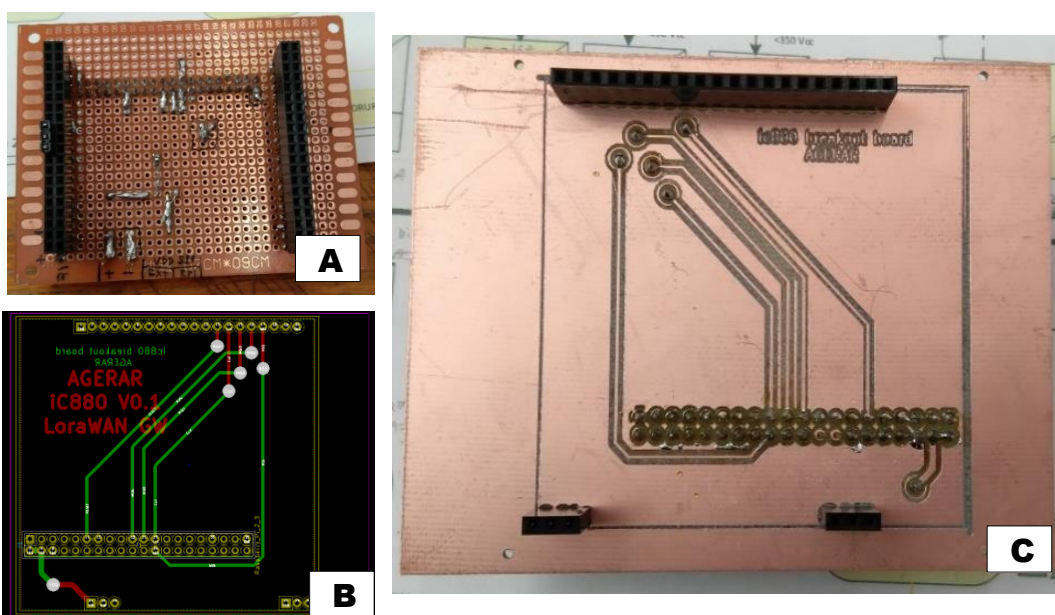


Figura 4.9 – Placa perfurada inicial (A), esquema kicad da pcb (B), placa finalizada (C)

4.3.2.4 Antena

Numa fase inicial utilizaram-se as mesmas antenas dos nós Lopy. Posteriormente foi adquirida uma antena exterior de marca Linx ANT-868-ID para avaliar a performance por comparação com a dos Lopy. Nos testes de cobertura iniciais, verificou-se que a cobertura não era a melhor, tendo-se optado por adquirir uma antena de qualidade superior. Esta antena é do

tipo dipólo de meia onda, com um ganho de 0,6 dBi e uma *Voltage Standing Wave Ratio* (VSWR) inferior a 1,9:1, numa largura de banda de 50 MHz. Esta medida indica o quanto de um sinal enviado pelo emissor à antena, é refletido de volta pela mesma. É sempre apresentado numa relação N:1, sendo que quanto mais próximo da unidade este valor N, melhor a antena. Na frequência central 868 MHz é indicado um VSWR de 1.087. A antena inclui um cabo de 2 metros com conector SMA, braçadeira de fixação e é aprovada na categoria IP67 [125].

Quando se procedeu à ligação da nova antena e por comparação com a anterior verificou-se um aumento de aproximadamente 10 dBm no RSSI. A Figura 4.10 mostra o salto existente na potência do sinal recebido (RSSI), a partir de um nó fixo colocado num laboratório do ISE. No lado direito mostra-se a antena instalada no exterior da janela da sala 160.

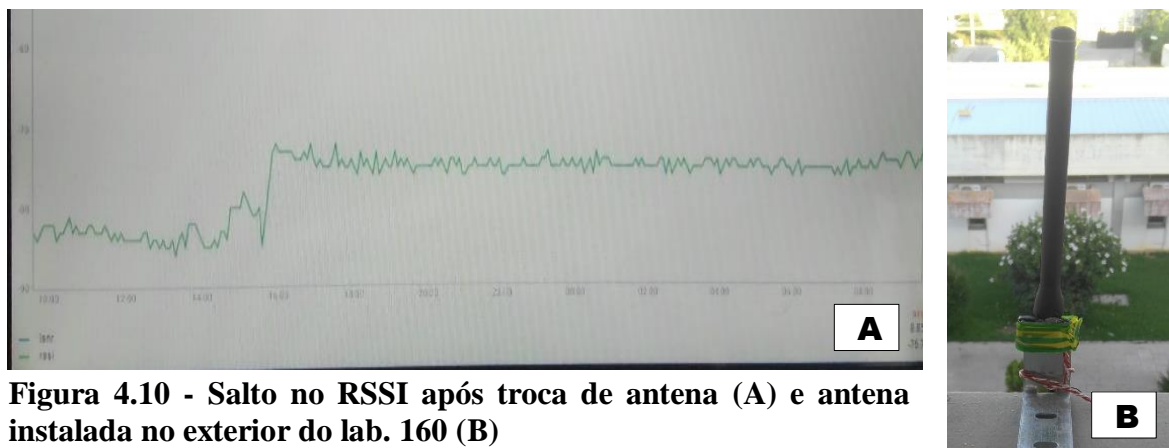


Figura 4.10 - Salto no RSSI após troca de antena (A) e antena instalada no exterior do lab. 160 (B)

4.3.2.5 Montagem e Configuração Inicial

A utilização da *shield* permite que por simples encaixe dos componentes, se possa ligar o RPI e configurar o *software* encaminhador de pacotes, ficando uma *gateway* pronta a funcionar, como mostra a Figura 4.11. O passo inicial passa por garantir um *reset* do concentrador no arranque do RPI. Para tal acrescenta-se um pequeno *script*, programado para correr cada vez que o sistema operativo inicia, e que força o pino do RPI ligado ao pino de *reset* do iC880 a nível alto, durante alguns segundos. Este passo garante o reinício limpo do concentrador e é descrito no Apêndice E.

Com este processo a funcionar, procedeu-se à instalação da biblioteca HAL *lora_gateway* [124] e o *software packet forwarder* [126] de referência. O processo consiste em descarregar os ficheiros para o RPI, configurar os modos de *debug* num ficheiro de configuração e compilar todos os pacotes. Após este passo devemos verificar o funcionamento da biblioteca,

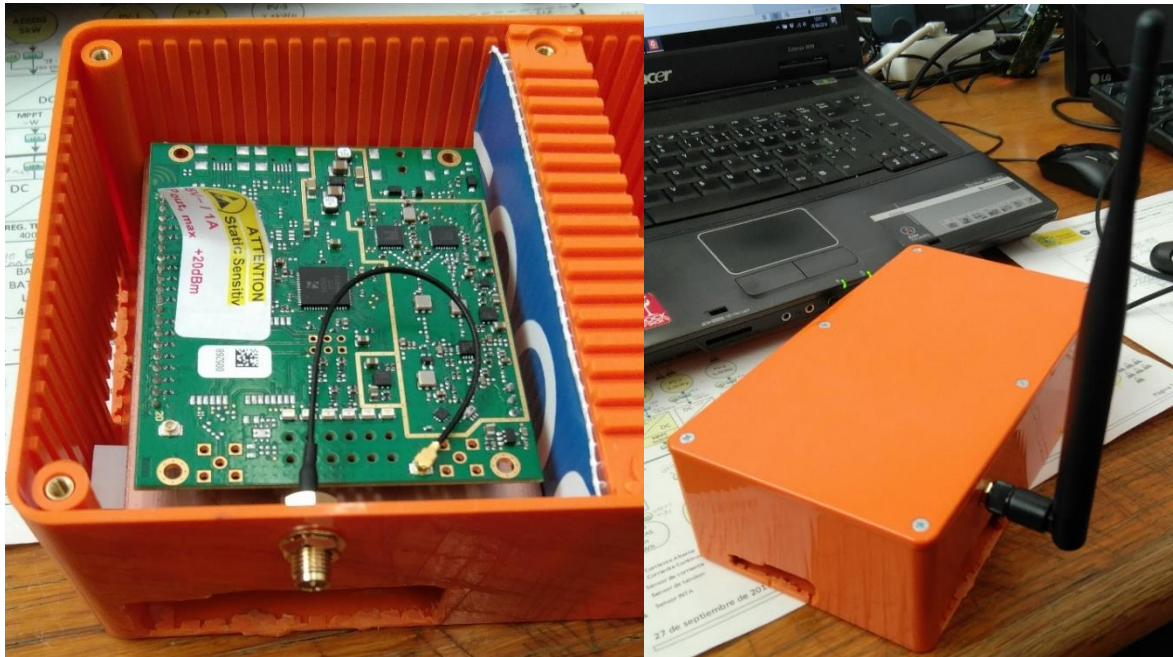


Figura 4.11 - Gateway e NS (com antena inicial)

utilizando um dos programas de teste, que correm a partir da linha de comandos. Nesta fase inicial é bom ativar o *debug* SPI, REG e HAL, de forma a detetar erros. No meu caso, ao correr o *util_tx_test* deparei-me com o erro HAL de falha de início do concentrador. O *debug* REG permitiu-me ver que um registo devolvia um valor errado, indicando que não conseguia bloquear a *phase lock loop* (PLL) do concentrador, ou que este não calibrava. O problema residia na diferença de frequência do SPI nos dois dispositivos, que provocava falhas de funcionamento. A correção passou por ajustar a frequência no código fonte para 1 MHz. Após esta edição, bastou recompilar os programas.

O passo seguinte consistiu em configurar o encaminhador de pacotes. Este tem dois ficheiros de configuração locais e globais, em formato JSON. Nas globais configuram-se as frequências dos 8 canais, potências de emissão e parâmetros de comunicação com o NS. É assim importante estarem decididas as frequências que a rede vai utilizar assim como o IP:Porto do NS. No caso concreto implementado o endereço seria *'localhost'* com o porto por omissão do *packet forwarder* (1680), uma vez que tudo corre na mesma máquina e a comunicação é interna. No caso de utilizar um NS noutra máquina, configuram-se os endereços de rede deste. O identificador da *gateway*, um código hexadecimal de 16 caracteres que a identifica na rede (endereço MAC), deve ser definido nestas configurações e nas locais. No caso utilizamos o "AABBCDEF01600001", representando a *gateway* 1 da sala 160. Posteriormente, este também deverá ser configurado no NS. De seguida podemos correr o *lora_pkt_fwd* manualmente, procurando verificar via linha de comandos, que existem

mensagens PUSH/PULL entre o *packet forwarder* e o *host*, indicando que existe ligação entre os componentes.

A Tabela 4.1 resume as frequências utilizadas e as respectivas configurações de rede. Estas frequências correspondem às bandas g e g1 da ETSI, com 1% de *duty cycle*. A nomenclatura utilizada no ficheiro de configuração global é a seguinte: os rádios dos SX1257 (identificados como “*radio_0*” e “*radio_1*”) utilizam uma frequência base (“*freq*”) definida em Hertz, sendo os canais (“*chan_multiSF_0*” a “*chan_multiSF_7*”) ajustados como *offsets* (“*if*”) em relação à mesma. Para cada canal devemos selecionar ainda o identificador de rádio (“*radio*”) que queremos utilizar consoante a frequências escolhidas (ver Figura 4.12). Por exemplo, numa frequência 868,5 MHz (“*freq*”: 868500000), para definirmos o canal 1 (“*chan_multiSF_1*”) nos 868,3 MHz ajustamos um *offset* de -200000 hertz (“*if*”: -200000).

Após realizar estas configurações basta colocar o programa de *reset* e o *packet forwarder* a correr no arranque do RPI. Juntamente com isso, cria-se um ficheiro de *log* para registar o funcionamento do encaminhador, para algum *debug* necessário. De forma a minimizar este ficheiro, após verificar que tudo funciona corretamente (inclusive com o NS) deve-se recompilar a biblioteca *libloragw* e o programa *lora_pkt_fwd* sem os *debugs* ativos e implementar um *logrotate* sobre a localização do ficheiro. Este último, funcionalidade presente em diversas distribuições Linux, verifica o tamanho dos ficheiros de *log* armazenando-os ou apagando-os em determinadas condições, estabelecidas por quem as configura. Neste ponto temos uma *gateway* LoraWAN pronta a funcionar. Para o ensaio do

Canal	Frequência	radio	if
Canal 0	868.1	1	-400000
Canal 1	868.3	1	-200000
Canal 2	868.5	1	0
Canal 3	867.1	0	-400000
Canal 4	867.3	0	-200000
Canal 5	867.5	0	0
Canal 6	867.7	0	200000
Canal 7	867.9	0	400000

Tabela 4.1 – Canais configurados

```

"radio_0": {
  "enable": true,
  "type": "SX1257",
  "freq": 867500000,
  "rssi_offset": -166.0,
  "tx_enable": true,
  "tx_notch_freq": 129000, /* [126..250] kHz */
  "tx_freq_min": 863000000,
  "tx_freq_max": 870000000
},
"radio_1": {
  "enable": true,
  "type": "SX1257",
  "freq": 868500000,
  "rssi_offset": -166.0,
  "tx_enable": false
},
"chan_multiSF_0": {
  /* Lora MAC channel, 125kHz, a11 SF, 868.1 MHz */
  "enable": true,
  "radio": 1,
  "if": -400000
},
"chan_multiSF_1": {
  /* Lora MAC channel, 125kHz, a11 SF, 868.3 MHz */
  "enable": true,
  "radio": 1,
  "if": -200000
},

```

Figura 4.12 - Canais loraWAN alocados (excerto do ficheiro de configuração)

sistema montou-se tudo numa pequena caixa existente, estando previsto numa fase futura criar uma caixa à medida, impressa numa impressora 3D. O Apêndice F descreve todo este procedimento com mais detalhe.

4.3.3 SERVIDOR DE REDE NS

A instalação do servidor de rede é simples, bastando descarregar o ficheiro *debian* de instalação e correr o *dpkg* do mesmo, no RPI da *gateway* anteriormente apresentado. Após este passo basta configurar o NS: (1) para que utilize a porta do *packet forwarder* (1680 por omissão), (2) com os certificados SSL caso se utilizem e (3) com as credenciais de acesso (*{http_admin_credentials, {<<"user">>, <<"pass">>}*) à página web de administração do servidor. Esta configuração é feita num ficheiro de formato JSON. Com a conclusão deste passo pode-se aceder à página de administração, a partir do *browser* do Rpi3, no endereço 127.0.0.1:8080 e utilizar as credenciais respetivas para acesso à mesma. É nesta *interface web* que serão configurados os nós, *gateways* e o tratamento a dar aos pacotes recebidos, conforme explicaremos posteriormente. Para mais detalhes sobre este processo de instalação deverá ser consultado o Apêndice G.

Os dados recebidos dos nós LoraWAN são encaminhados para o AS Thingsboard, utilizando o protocolo MQTT.

4.3.4 SERVIDOR DE APLICAÇÕES THINGSBOARD

De início, o Servidor de Aplicações Thingsboard foi testado num computador existente no laboratório 160, mas por portabilidade, decidiu-se instalar o mesmo num RPI independente. A instalação inicial foi idêntica à da *gateway*, limpando programas desnecessários e alterando os utilizadores de origem, exceto que aqui não ativamos o SPI. Foi configurado um IP fixo para utilização na nossa rede privada. Este pode guardar os dados em vários tipos de bases de dados SQL ou NoSQL, algo que deve ser decidido antes o que utilizar. Optei por uma base de dados PostgreSQL, a qual o serviço suporta e que corre fluentemente no RPI. Esta deve ser instalada antes da Thingsboard, sendo necessário proceder à criação manual da base de dados “*thingsboard*”, e dos respetivos utilizadores, utilizando a ferramenta *psql* de gestão do PostgreSQL. Após este passo pode-se instalar a Thingsboard. O instalador é disponibilizado

num ficheiro *Debian*. Concluída a instalação, devemos configurar no mesmo a utilização da base de dados PostgreSQL com os dados de utilizador criados inicialmente. Depois de arrancar o serviço é necessário aguardar um momento para que tudo carregue, ficando depois a página de administração disponível. Para aceder à mesma basta, a partir do *browser* do RPi3, navegar para o endereço 127.0.0.1:8080 e introduzir as credenciais de administrador (Login – “sysadmin@thingsboard.org”; Password – “sysadmin”), ficando com acesso à criação de *tenants* (grupos de utilizadores que partilham acesso ao servidor) e utilizadores. Foi criado um *tenant* (*Agerar*) e um utilizador nesse *tenant* (*Lab160*, com email nfpinto@ualg.pt), o qual deverá ser ativado pelo email associado ou *link* gerado. Com isto confirma-se que o servidor de aplicações está funcional.

Como se pretende que esta plataforma possa mostrar os dados num ecrã *touchscreen*, configurou-se o RPI para arrancar a página *web* das *dashboards* em modo quiosque. Desta forma fica-se com um pequeno quiosque pronto a mostrar os dados dos nós. Sempre que reinicia, o quiosque abre a página de *dashboards* do utilizador escolhido. Como era para uma utilização privada, não era preocupante este facto, mas em caso de necessidade, a página a abrir poderá ser uma *dashboard* pública, sem acesso aos dados dos nós e à administração do *tenant*. O processo de instalação encontra-se descrito no Apêndice H.

4.3.5 REDE BACKHAUL

Para o nosso *setup*, o *backhaul* ideal passa por utilizar a rede *intranet* da universidade. Para nos ligarmos por Wi-Fi é-nos exigida a autenticação dependente do email de utilizador. Na ligação por cabo este problema não se coloca, mas restringe-nos a mobilidade. Apesar da rede interna nos permitir interligar os dispositivos, não conseguimos de forma fácil configurar IPs estáticos, necessários nos dois servidores, uma vez que estas definições apenas são acessíveis aos serviços informáticos da universidade. Apesar de existir uma rede interna para estes dispositivos, nesta fase optou-se por utilizar um *router* Wi-Fi do projeto, garantindo assim uma melhor gestão de IPs, portabilidade de todo o *setup*, e melhor controlo. Adicionalmente, qualquer outro dispositivo Wi-Fi se podia ligar a esta mesma rede. Foram atribuídos IP’s estáticos ao AS e NS assim como foi também colocado na zona desmilitarizada (*demilitarized zone*, DMZ) do *router*, o IP do AS, garantindo o acesso ao mesmo de dentro da *intranet* da universidade, bastando saber o IP atribuído ao *router*. Para saber esse IP é necessário entrar

na página de configuração do *router*, para verificar qual o IP externo atribuído a este pelos equipamentos da universidade.

4.3.6 ESTAÇÃO METEOROLÓGICA

Para que se efetue uma correta gestão e escalonamento de cargas elétricas, deve existir um bom conhecimento das variáveis de consumo e geração elétrica. Ambas dependem sempre das condições ambientais, tornando-se por isso fundamental conhecer o estado das mesmas, no local de produção. OS processos de comunicação com as estações meteorológicas dependem dos protocolos suportados pelas mesmas. Importa por isso que se façam testes com uma dessas estações, sabendo que posteriormente terão que ser criadas soluções específicas para as diversas versões que possam vir a ser integradas no projeto. Para começar utilizou-se uma estação meteorológica existente no topo do edifício principal do ISE (ver Figura 4.13, lado esquerdo), que mede a temperatura, humidade, radiação e intensidade do vento, entre outros parâmetros. A estação meteorológica está ligada a um *datalogger* da DataTaker modelo DT85, alimentado pela rede elétrica e com uma bateria de *backup*. A partir do *datalogger* podemos aceder aos vários dados da estação meteorológica, estando o mesmo ligado por cabo Ethernet à rede interna da universidade. Um pequeno servidor *web* permite apresentar os valores registados com uma determinada periodicidade, bastando para tal aceder via HTTP ao seu IP.

O *web server* tem duas interfaces, utilizando *flash* ou *html* simples. Na programação do *datalogger* são definidas várias medições, efetuadas em diferentes canais (nome dado à medição pela marca). A cada canal está também associada uma variável de periodicidade de captura de dados denominada de *Schedule* (a um *Schedule* do tipo A corresponde a registos de 10 em 10 segundos, B de minuto a minuto, C de hora a hora, e D de semana a semana). O *datalogger* disponibiliza uma tabela com todos os valores registados (Figura 4.13). Para nosso registo, a periodicidade de minuto a minuto chega, sendo por isso retirados os valores desta parte da tabela. Exceção-se a tensão da bateria que se encontra na classe “C”. A página *web* atualiza os dados de forma automática a cada 30 segundos, embora estes sejam atualizados no agendamento “A” a cada 10 segundos. A Figura 4.13 mostra uma captura da página *web*. Do “*Schedule B*” interessam-nos os valores de temperatura média (**Tmd**), humidade relativa média (**Hmd**), radiação média (**RadMed**), velocidade do vento média (**VVmd**), direção do vento (**DV**), pluviosidade média (**PR**), pressão atmosférica (**Pvs**) e a hora de captura (Hora).

Do “Schedule C” interessa-nos a tensão da bateria para enviar avisos de baixa tensão da mesma.

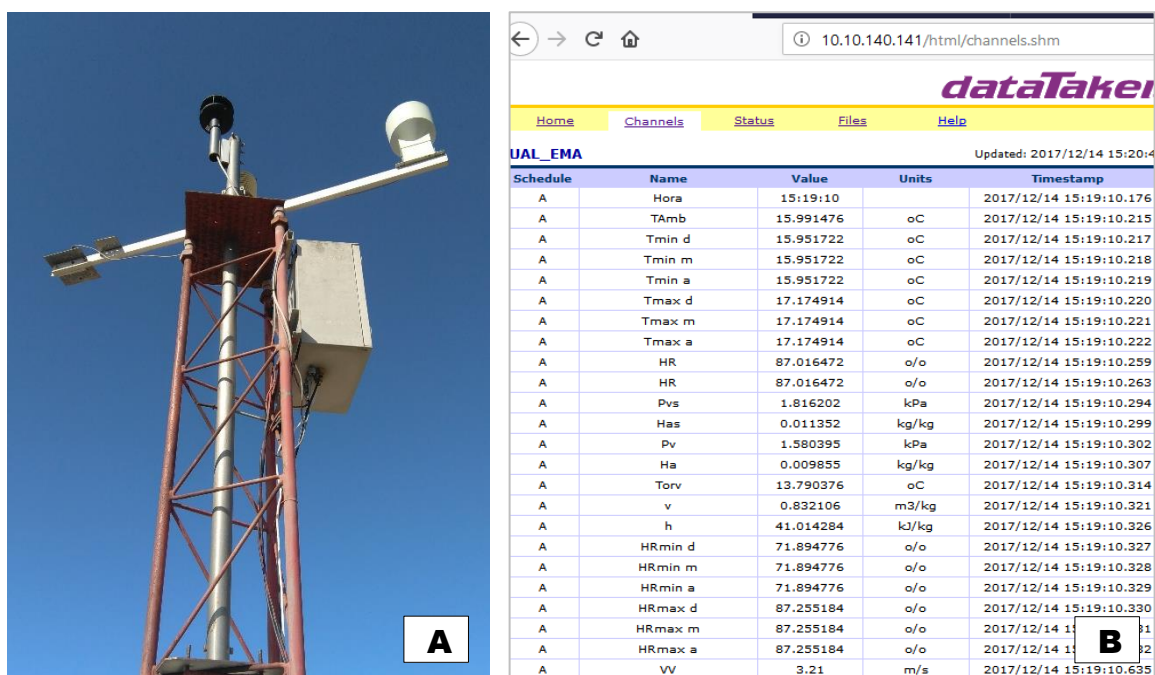


Figura 4.13 - Estação meteorológica do ISE (A) e página de dados html da estação (B)

Uma vez que o *datalogger* não tem como enviar os dados para outro dispositivo, apenas consegue apresentar nas suas páginas *web*, a obtenção destes passará por uma obtenção dos mesmos a partir da página *html* do mesmo (denominado de *parsing*). A página *html* é mais leve em termos de código, o que obviamente gera um tráfego de dados inferior, em comparação com a versão *flash*. O equipamento tem outros métodos de acesso, que utilizam o software próprio da marca, mas que não permitem uma extração automatizada dos dados, ou que geram um tráfego adicional desnecessário (por acesso *ftp*). A escolha da leitura da página *html* deveu-se a estes factores, uma vez que seria lida a cada minuto.

Para extração e envio de dados para a Thingsboard foram criados *scripts* em linguagem Python os quais seriam executados de minuto a minuto a partir de um servidor Linux existente na sala 160. O *script* criado efetua uma leitura da página acima referida, extraindo os valores que precisávamos nomeadamente a temperatura, humidade, radiação, velocidade e direção do vento, pluviosidade, pressão atmosférica e a tensão na bateria. Outros valores presentes na tabela podem ser adquiridos, como valores médios e máximos, mas não são guardados neste momento uma vez que podem ser obtidos a partir dos guardados. De futuro, se necessário, basta acrescentá-los ao *script*.

De forma a verificar o bom funcionamento da estação, foi implementado no código um mecanismo que envia *emails* de alerta, sempre que: (1) falhe a ligação ao dispositivo, (2) se obtenham leituras inválidas ou (3) quando o nível de bateria atinga os 8 V. O código fonte e descrição do mesmo está no Apêndice I.

4.3.7 SERVIDOR DE ESCALONAMENTOS SS

No âmbito do projeto AGERAR, pretende-se utilizar os dados da estação meteorológica de forma a efetuar o agendamento de cargas elétricas. Estes agendamentos serão baseados na produção renovável solar estimada para o dia e com base nos históricos do passado. Utilizando estes dados juntamente com alguma informação de previsão obtida *online* podem-se utilizar algoritmos de previsão e efetuar o escalonamento das cargas, com base nestas informações. Estas funcionalidades não fazem parte desta tese, mas cabe-nos garantir que a arquitetura de nós e dispositivos sensores já permite tratar e encaminhar corretamente estes dados.

Uma das principais aplicações consistirá no agendamento do carregamento de veículos elétricos, minimizando o custo de aquisição de energia ao operador de distribuição, enquanto procura maximizar a utilização de produção local e energia renovável (maioritariamente solar). Espera-se também que este sistema permita a gestão da corrente de carga dos veículos, de forma dinâmica, refletindo as preferências dos utilizadores, e aprendendo com os hábitos dos mesmos. Em qualquer dos casos, existirá um servidor que cria e gere escalonamentos (*Scheduling Server, SS*) e os envia aos respetivos dispositivos, seja via LoraWAN, seja via WiFi. Como tal, os dados meteorológicos, devem ser encaminhados para este servidor. Os agendamentos LoraWAN deverão ser criados para um nó, identificado pelo seu DevEUI. Uma vez que o NS recebe estes dados via MQTT, o servidor de escalonamentos deverá incluir um cliente MQTT, publicando os agendamentos ao NS e identificando o nó de destino. No Apêndice K está a descrição do processo de envio de mensagens. Do mesmo modo, deverá enviar os agendamentos ao AS, permitindo aos utilizadores saber o que foi decidido por este.

4.3.8 COMUNICAÇÃO E PROTOCOLOS ENTRE OS DIFERENTES COMPONENTES

Nas comunicações entre dispositivos, foram definidos vários protocolos de comunicação, cuja utilização depende das tecnologias a serem utilizadas. Esses protocolos não foram abordados anteriormente pois interessa apenas analisar em detalhe aqueles que se relacionam com as tecnologias utilizadas neste trabalho. A Figura 4.14 apresenta a arquitetura protocolar nos diversos pontos do sistema desenvolvido. Em seguida são descritos os protocolos utilizados em cada um dos pontos da arquitetura.

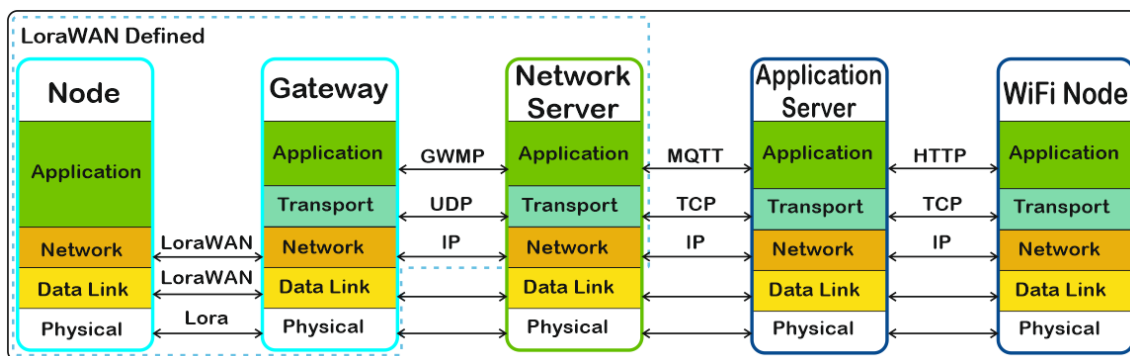


Figura 4.14 – Arquitetura protocolar do sistema

4.3.8.1 Comunicação entre Nós Lora e Gateway

A comunicação entre nós Lora e *gateway* passa obrigatoriamente pelo protocolo LoraWAN, conforme descrito no capítulo 3. Este ocupa as camadas física, ligação de dados e de rede do modelo OSI.

4.3.8.2 Comunicação entre Gateway e NS (dados LoraWAN)

Embora estejam a correr na mesma máquina, a comunicação entre Gateway e NS segue a implementação de referência da Semtech e utilizam o protocolo *GWMP*, conforme explicado no capítulo 3.3.1. Os dados são enviados via UDP em formato JSON, sendo que todo o processo acontece internamente no RPI, nas camadas de transporte e aplicação.

4.3.8.3 Comunicação entre NS e AS Thingsboard MQTT (dados LoraWAN)

O envio de dados entre o NS e o AS é feito utilizando MQTT (ver Apêndice J). Este protocolo está implementado nos dois servidores e embora não seja a única opção de troca de mensagens é a mais simples e leve. No caso Thingsboard, este atua como *broker* recebendo dados na porta 1883, enquanto o NS atua apenas como cliente MQTT. Os dados de cada nó são processados no NS, que os publica em formato JSON para o tópico *v1/devices/me/telemetry*. Todos os nós do NS enviam para este tópico, sendo que a distinção entre os mesmos é feita pela inclusão de um *token* de identificação único, configurado para cada nó no AS.

4.3.8.4 Envio de dados do SS para os Nós

De forma a permitir o envio de dados aos nós, a partir do servidor de escalonamentos, é utilizado um *broker* MQTT adicional, instalado no RPI NS. A adoção desta solução suporta-se na possibilidade de o NS receber os *downlink* por via MQTT. O *broker* escolhido foi o Mosquitto. O NS subscreve um tópico neste *broker* para o qual as mensagens devem ser enviadas pelo SS. Os nós são identificados pelo seu DevEUI, permitindo ao NS encaminhar corretamente os dados. Uma vez que o AS Thingsboard já utiliza a porta MQTT definida por omissão (1883), este *broker* tem de utilizar outra porta, sendo utilizada a 1882. A composição das mensagens e o método de envio são descritos no Apêndice K. A nível LoRaWAN, utiliza-se a banda g3 (frequência 869,525 MHz) para os *downlinks*, uma vez que esta tem 10 % de *duty cycle*, permitindo assim um maior envio de dados.

4.3.8.5 Comunicação entre Nós WiFi e AS Thingsboard

Para a comunicação entre os nós WiFi e o AS Thingsboard existem várias opções de troca de dados. O MQTT é uma delas, mas o AS permite receber dados via HTTP (como acontece com os dados recebidos da estação meteorológica), ou via CoAP (Constrained Application Protocol), um outro protocolo leve, comum em microcontroladores. Destes protocolos utilizaremos apenas o HTTP. Cada dispositivo que queira enviar dados por HTTP deve enviar um pedido *POST* ao endereço *http(s)://host:port/api/v1/\$ACCESS_TOKEN/telemetry* contendo os dados em formato JSON e o cabeçalho *"Content-Type:application/json"*.

Obviamente *\$ACCESS_TOKEN* corresponde ao *token* único do dispositivo, configurado na página de dispositivo do Thingsboard.

4.3.9 FORMATOS DE DADOS NO PAYLOAD DOS NÓS LORAWAN

Devido à necessidade de minimizar o tamanho das tramas, procuraram-se soluções eficientes de formatação das variáveis a enviar. Estes problemas não se colocam de forma tão importante em nós WiFi, uma vez que não existe um limite explícito na ocupação do canal. Assim, os dados podem ser enviados nos formatos originais.

Um dos aspetos tidos em conta passou pela análise da integração com o servidor NS LoraWAN, de forma a que os dados fossem facilmente filtrados por este. Nalguns casos seria possível diminuir o tamanho dos dados adotando formatos diferentes dos *standard* mas depois a recuperação destes seria mais complicada no NS. Neste âmbito, existem algumas soluções implementadas nomeadamente o Cayenne Low Power Payload [127] ou os IPSO *smart objects* [128]. A primeira é uma possível solução (se for utilizada a rede Cayenne) pois permite de forma simples e com poucos bytes identificar sensores e atuadores. A segunda é uma definição completa desenhada para comunicações M2M e dispositivos com recursos reduzidos. Foram analisadas as duas soluções, mas devido à utilização adicional de 1 *byte* para identificar cada canal, muitas das vezes os ganhos obtidos nos outros campos perdiam-se pela adição deste. Assim, para simplificar e dar uma maior flexibilidade, procedemos à definição dos tipos dos dados e formato do *payload* que iriam ser utilizados. Se na rede forem posteriormente adicionadas outras medições, podemos nessa altura utilizar as mesmas estratégias, cumprindo as regras descritas de seguida.

4.3.9.1 Tensão e Corrente

As correntes que iremos medir são maioritariamente inferiores a 63 A e fluem apenas num sentido. O tipo de dados *u_int16* permite uma representação entre 0 e 65535. Assim, este tipo de dados que ocupa apenas 2 *bytes*, é suficiente para enviar as correntes com precisão de miliampere, bastando para tal obter a nossa leitura em formato de virgula flutuante com 3 casas decimais e multiplicar por 1000. Por exemplo, uma leitura de corrente de 17,345 A será transformada no número inteiro 17345 e enviado apenas em 2 *bytes*. As tensões, sejam

alternadas sejam contínuas, nunca passam de 600 V, e a maioria dos equipamentos de medida apenas os apresenta com 2 casas decimais. Assim, o mesmo formato *u_int16* pode ser utilizado para medir estas tensões, multiplicando e dividindo por 100. Nos casos onde a corrente é bidirecional, o valor máximo expectável é de 120 A. Nestes casos optou-se por reduzir uma casa decimal e utilizar o tipo *int16*, o qual permite representações entre -32768 e 32767. Assim, para uma corrente de 119,54 A, este valor seria multiplicado por 100, enviando o valor 11954. No NS, este valor é posteriormente dividido por 100.

4.3.9.2 Coordenadas GPS

No decorrer do projeto AGERAR será necessário enviar as coordenadas GPS dos veículos híbridos a hidrogénio e elétricos da INTA. As coordenadas GPS são representadas em graus decimais (*decimal degrees*, DD) utilizando um valor entre -180,0 e 180,0 na latitude, e -90,0 e 90,0 na longitude. Neste sistema, o número de casas decimais utilizadas define a precisão do sistema, embora na prática esta precisão seja dependente do dispositivo recetor. Os recetores comerciais para aplicações genéricas não conseguem normalmente precisões superiores a 1 metro, sendo que no sistema DD bastam 5 casas decimais para se garantir esta resolução. Em aplicações gerais uma resolução na ordem dos 11 metros satisfaz as necessidades, sendo que 4 casas decimais chegam para o efeito. Refira-se uma solução proposta [129] onde bastariam apenas 3 *bytes* para as enviar, com uma precisão de 11 metros, mas que sofre do problema de compatibilidade, pois este não é um formato *standard*. Assim optou-se por utilizar variáveis do tipo *float32*, ocupando um pouco mais no *payload* LoraWAN, mas simplificando o processamento na receção. De referir que no decorrer do trabalho da tese, apenas se utilizaram coordenadas para medição de distâncias entre o nó e a *gateway*, nos ensaios *site survey*.

4.4 MÓDULOS DE MEDIÇÃO

Para além das comunicações na rede WSN, foi necessário desenvolver alguns módulos adicionais, responsáveis pela obtenção das variáveis elétricas. Assim, foram criados módulos de medição, com vista a serem aplicados na rede INTA, mas em simultâneo orientados a uma utilização generalista em outras micro redes. As primeiras versões dos módulos desenvolvidos constituem protótipos, desenhados no *software* KiCad para componentes *through hole* e

construídos através da fresadora CNC da universidade. A utilização deste método para criação de protótipos exige trabalhar com margens maiores no desenho, nomeadamente no dimensionamento de pistas. Uma vez que as pistas são isoladas no cobre da PCB, utilizando uma broca de gravação cônica, as características dessas pistas determinam a tolerância com que devemos trabalhar. No nosso caso utilizámos uma broca com base de 0,2 mm e ângulo de 60 graus, sendo que esta é bastante larga para componentes *surface mount device* (SMD). Espera-se, num trabalho futuro e após o teste de funcionamento, adaptar as placas para a utilização de tecnologia SMD, comprando as PCBs já feitas (pelos métodos de fotolitografia) e reduzindo assim o tamanho das placas, ao mesmo tempo que lhes são conferidos um aspeto mais profissional.

4.4.1 MÓDULOS NA MICRO REDE INTA

Para a aplicação na INTA, analisou-se ao detalhe a micro rede, de forma a escolher os tipos de transdutores que iríamos utilizar e que módulos se deveriam desenvolver. Com a diversidade de pontos de medição existentes, tentou-se agrupar os pontos de medida em função da sua proximidade física e tipo de medição elétrica, de forma a juntar diversas medições num único módulo emissor. Desta forma poderíamos criar módulos similares, replicando-os posteriormente onde necessário, em vez de desenvolver um módulo específico para cada local (são 64 pontos de medição). As tabelas em Apêndice L, apresentam as características elétricas de cada ponto da rede DC e AC que se pretende monitorizar.

Para as correntes DC analisaram-se os valores nominais (quando indicados) e máximos das correntes a monitorizar, assim como a seção do cabo existente. Para as tensões DC, apenas os valores nominais e máximos. Verificámos também os locais destes pontos de medida. No caso AC, optou-se por utilizar medidores comerciais (da marca Eastron, gama SDM [130]), com saídas Modbus e que deveriam ser colocados nos respetivos quadros elétricos. Muitos dos dispositivos de medição estão dentro de um armário com inversores de alta potência, pelo que se prevê que o ruído eletromagnético existente seja elevado. Assim optou-se por efetuar a transmissão Lora a partir do exterior do armário, para evitar interferências. Desta forma considerou-se a colocação de um emissor exterior, que é por sua vez ligado a módulos de medição que ficariam no interior do armário, nos pontos a monitorizar. A comunicação deveria passar para o exterior por cabo blindado, sendo a alimentação dos módulos feita por este mesmo cabo. Para locais fora deste armário, os módulos de leitura podem ficar junto ao

emissor. A Tabela 4.2, apresenta os pontos de medição agrupados, fazendo a correspondência com o módulo que se pretende desenvolver para a leitura, o respetivo equipamento de medida e uma placa emissora associada. Nesta tabela, definimos também o tamanho e formatação do *payload* Lora, úteis para o *parsing* que se realiza no NS. O maior *payload* tem 16 bytes, enviado 180 vezes por hora (a cada 20 segundos). Num único canal, conseguimos manter este envio em DR5 e DR4. Em 8 canais, podemos manter as transmissões até DR1. Com DR0, não será possível. Nem mesmo com o *payload* de 8 bytes. Os módulos e equipamentos de medida são descritos de seguida.

Tipo de medição	Nome	Identificação	Localização	Módulo	Equipamento de medição	Módulo Emissor	Formato Payload	Tamanho Payload
Corrente e Tensão AC	PV4-AC	u5, i8	Parque Veículos	RS485	SDM230	Módulo Lopy Power Board 1	u5, i8, u9, i16, u15, i24	12 bytes
	SCK1	u9, i16			SDM230			
	INV EV	u15, i24			SDM230			
	INV DC/AC	u8, i14	Armário Micro Rede	RS485	SDM230	Módulo Lopy Power Board 2	u8, i14, u14, i21	8 bytes
	AEROG	u14, i21			SDM530			
	IN	u12, i19	Quadro Elétrico	RS485	SDM530	Módulo Lopy Power Board 3	u12, i19, u13, i20	8 bytes
	IN-SAI	u13, i20			SDM530			
	SKT2	u10, i17	Parque Veículos	RS485	SDM530	Módulo Lopy Power Board 4	u4, i7, u10, i17, u11, i18	12 bytes
	SKT3	u11, i18			SDM530			
Corrente e Tensão DC	PV4	u4, i7	Parque Veículos	DC_1V_1L	LEM HO 10-P + LV25P	Módulo Lopy Power Board 5	u1, u2, i3, u3, i5, u7, i9, i10, i11	18 bytes
	PV1	u1		DC_1V	LEM LV-25P			
	PV2	u2, i3	Armário Micro Rede	DC_1V_1i	LEM HO 10-P + LV25P			
	PV3	u3, i5		DC_1V_1i	LEM HO 10-P + LV25P			
	Pb, Lilon	i9, u7, i10, i11		DC_1V_3i	LEM LV25P + Tamura L31			
Corrente DC	PV1	i1.1, i1.2, i1.3, i1.4	Armário Micro Rede	DC_4i	LEM HO 10-P	Módulo Lopy Power Board 6	i1.1, i1.2, i1.3, i1.4, i13.1, i13.2, i13.3	16 bytes
	DC/AC OUT	i13.1, i13.2, i13.3		DC_4i	LEM HO 10-P			
	PV1-DC/DC, PV2-DC/DC, PV3-DC/DC	i2, i4, i6		DC_4i	LEM HO 10-P	Módulo Lopy Power Board 7	u6, i2, i4, i6, i12, i15, i22, i23	16 bytes
	Eletrolyzer, DC Load, AERO AC-DC, DC Source	i12, i15, i22, i23		DC_4i	LEM HO 10-P			
Tensão DC	Pb-Ac	u6	DC_1V	LEM LV-25P				

Tabela 4.2 - Lista de módulos para aplicação na micro rede INTA

4.4.2 MÓDULO EMISSOR (LOPY POWER BOARD)

Para emissão de dados LoraWAN, foi desenvolvida uma placa onde se coloca o Lopy, e que permite alimentar e comunicar com módulos de medição remotos, instalados noutra local. Desta forma pode-se colocar a placa fora do armário da micro rede INTA. Este conta com um microcontrolador Lopy e fonte de alimentação independente (AC), um transdutor RS-485, 2 linhas de comunicação I2C com alimentação a 3,3 e 5V, cinco entradas a 3,3 V, quatro entradas a 5V e seis saídas genéricas a 3,3 V. Os pinos do microcontrolador ficam disponíveis para qualquer utilização adicional. Para comunicação entre módulos optou-se pelo protocolo I2C, pela facilidade de utilização e elevada disponibilidade de material. A necessidade de ler variáveis utilizando o protocolo Modbus, levou à inclusão de um transdutor, na mesma placa, tornando-a assim universal para todas as aplicações consideradas. Foi definido também que cada placa destas apenas se pode ligar a 4 dispositivos externos, sejam MODBUS ou I2C.

Na versão inicial, para alimentação, foi escolhido um conversor AC-DC de 5 W (5V, 1 A) modelo Vigortronix VTX-214-005-105, que permite alimentação AC (90-265V) ou DC (100-370 V), garante isolamento entre primário e secundário e tem proteção *hiccup* contra curto circuitos, de sobrecarga e sobretensão [131]. Esta proteção garante que o dispositivo desliga enquanto detetar a falha na saída, repondo o funcionamento assim que a falha seja resolvida. O consumo do Lopy atinge no máximo 300 mA, sobrando assim 700 mA para alimentação de dispositivos externos. Na entrada, foi adicionado um condensador para minimizar interferências eletromagnéticas (0.1 μ F, valor recomendado pela Vigortronix), um fusível de proteção (1 A, rápido) e um varistor MOV-14D431K, para proteção de picos de tensão. Na saída, colocaram-se dois condensadores (um eletrolítico de 100 μ F e um cerâmico de 100nF) para estabilizar o *ripple* e cortar algumas componentes de alta frequência, assim como um led para visualização que o dispositivo está ligado. Foi utilizado um módulo *transceiver* “RS485 to TTL” (SMAKN® SCM TTL to RS485), o qual é alimentado a 3,3 V e garante proteção de picos entre a linha RS485 e a TTL. Ao mesmo tempo, não precisa de linha de *enable* pois tem *automatic flow control*, o qual seleciona automaticamente o envio e receção, consoante os dados presentes na entrada TTL. Desta forma bastam duas linhas para comunicação com o microcontrolador. As 4 entradas a 5 V, utilizam um divisor de tensão, de forma a converter a escala de medição para os 3,3 V das entradas do Lopy. No Apêndice O encontra-se o esquemático do módulo. O custo aproximado de todo o material ronda os 63 euros. A Figura 4.15 mostra o módulo, na pré-visualização 3D do KiCAD e o protótipo já em fase de testes.

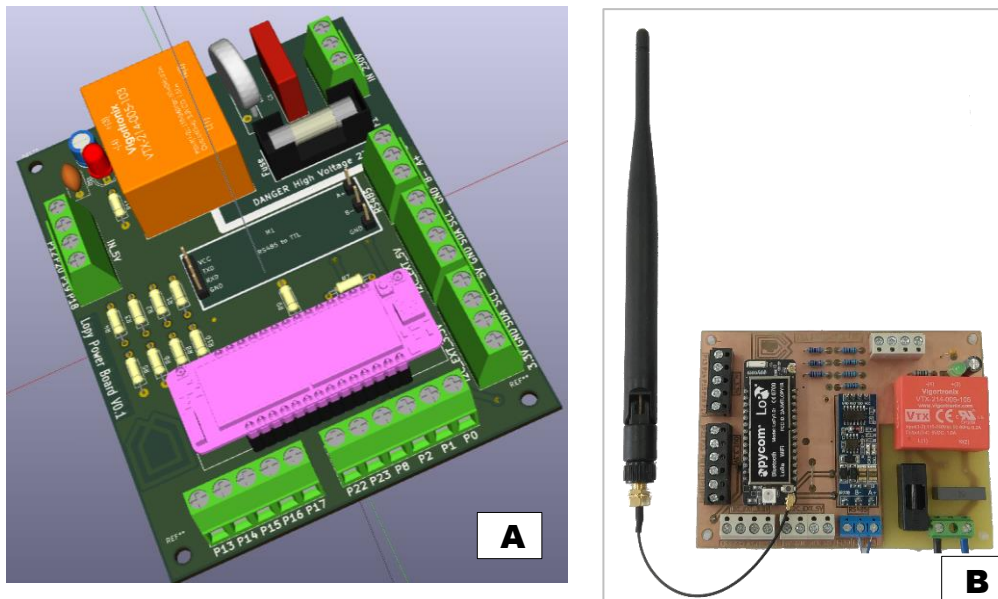


Figura 4.15 - Esquema 3D (A) e placa protótipo da primeira versão (B)

A nível de funcionamento de *software*, este irá variar consoante a aplicação do módulo e a funcionalidade que lhe desejarmos dar. Uma vez que os nossos módulos Lopy utilizados são os da versão mais antiga (com 512 kB RAM), deparámo-nos com problemas de memória ao tentar criar um código universal, que verificasse no arranque todos os componentes ligados e atuasse a partir daí. Assim, o código segue sempre a mesma lógica, mas é personalizado para cada ponto de aplicação, conforme descrito no capítulo 4.5.

Com a implementação prática do mesmo, deparamo-nos com algumas falhas de conceção, que levaram a algumas modificações numa segunda versão. Verificámos que o conversor AC/DC poderia ser insuficiente no fornecimento de corrente aos módulos externos. Os módulos de leitura de tensão podem exigir correntes até perto de 400 mA, cada um. Assim, trocou-se o conversor AC/DC por um RECOM RAC10-05SC, de 10 W e 5 V, permitindo o fornecimento de uma corrente até 2 A. A alimentação deste é apenas AC (85-305 V), mantendo as proteções *hiccup* de curto circuito, de sobretensão e sobrecarga [132]. Verificou-se também que não existiam resistências de *pull-up* nas linhas I2C, obrigatórias na especificação [132]. Em linhas de curta distância nunca foram verificados problemas, mas com linhas de maiores distâncias, estas passam a ser obrigatórias para garantir o correto funcionamento do protocolo. O valor mínimo destas é baseado nos valores da tensão de alimentação (V_{cc}), tensão de nível baixo (V_{ol}) e corrente em nível baixo (I_{ol}), segundo a fórmula [134]:

$$R_{pullupMin} = \frac{V_{cc} - V_{ol}}{I_{ol}} \text{ [ohm]} \quad (20)$$

Para uma alimentação de 3,3V, com tensão de nível baixo de 0 V e corrente típica a nível lógico baixo de 3 mA, o valor de $R_{pullupMin}$ deverá ser 1,1 k Ω . Para 5 V o valor será de 1,667 k Ω . O valor máximo dependerá de quantos dispositivos estão na linha, assim como a capacitância da mesma ($Cb_{sda-scl}[pF]$), e que afetam o tempo de subida das formas de onda ($t_{rise}[ns]$).

$$R_{pullupMax} = \frac{t_{rise}}{0.8473 * Cb_{sda-scl}} [ohm] \quad (21)$$

Optámos por uma resistência de 3.3 kOhm, superior ao valor mínimo de $R_{pullupMin}$ para 5 V. Devemos também respeitar que as linhas SCL e SDA, não podem ultrapassar os 400 pF de capacitância entre elas, sendo que no nosso caso, o *datasheet* do cabo a utilizar (AlphaWire 6300) nos dá 105 pF por metro. Para cada dispositivo na linha, devem-se considerar 15 pF adicionais [134]. Ao mesmo tempo, apercebemo-nos que a linha I2C de 5 V estava ligada diretamente aos pinos de 3,3 V do Lopy. Para garantir que isto não seria problemático, adicionou-se um conversor de níveis Sparkfun PCA9306 [135]. Adicionalmente, alterou-se a disposição dos pinos de saída, disponibilizando as duas saídas analógicas DAC do Lopy, nos pinos P21 e P22. A Figura 4.16 mostra o esboço e placa protótipo da segunda versão, encontrando-se o esquemático da mesma no apêndice P.

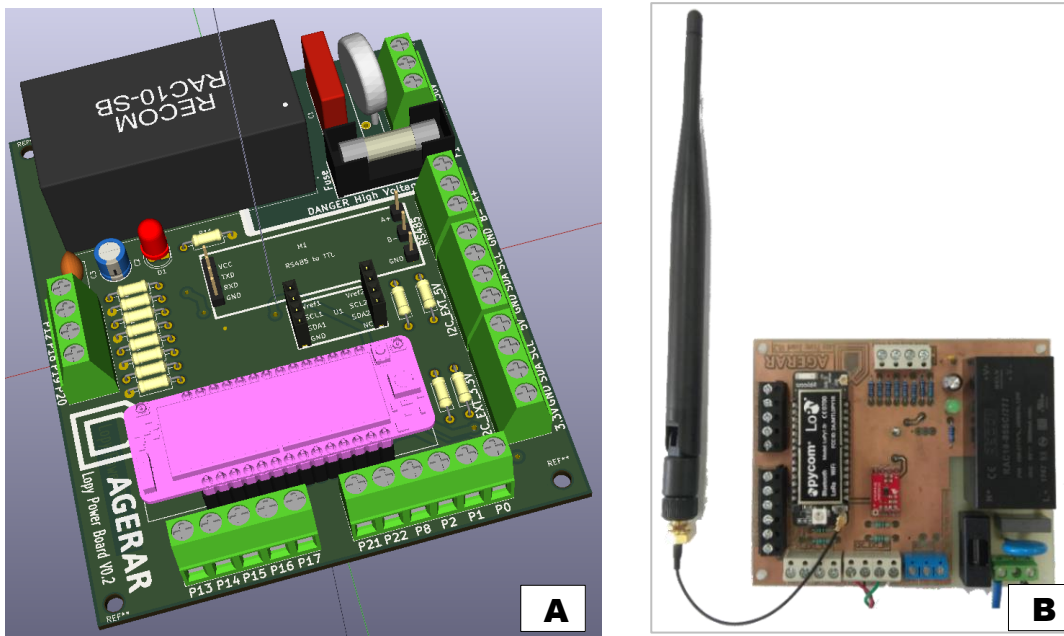


Figura 4.16 - Lopy Power Board Versão 2. Esboço 3D (A), placa finalizada sem módulo MODBUS (B)

Como já tínhamos alguns componentes adquiridos para a primeira versão, optou-se por utilizá-la, para os casos onde apenas se fizessem leituras MODBUS. Para leituras I2C, utiliza-se a segunda versão.

4.4.3 EQUIPAMENTOS RS485 (MODBUS) PARA MEDIÇÃO DE TENSÃO E CORRENTE AC

Para medições de correntes e tensões alternadas optou-se por utilizar equipamentos de medida comerciais, que disponibilizassem saídas MODBUS, para aplicação nas calhas DIN dos quadros elétricos. Devido à implementação na micro rede INTA, optamos por esta solução que nos daria mais fiabilidade e segurança nas leituras. A escolha dos equipamentos assentou principalmente na existência de saída MODBUS, no custo baixo e nas correntes máximas dos pontos de aplicação. As leituras que nos interessavam seriam a corrente e tensão em cada ponto do circuito AC. Qualquer medição adicional, não seria de grande importância para a aplicação que se pretende. Foram analisadas soluções da Schneider (iEM3250 e iEM3350), Legrand (série EMDX3), Iskra (EC1-80 e EC3-80) e Eastron (SDM230 e SDM560). No entanto os preços elevados das outras soluções fizeram-nos optar pelos equipamentos da Eastron, sendo escolhido o SDM230 para redes monofásicas e o SDM530 para trifásicas. Estes disponibilizam os dados em MODBUS RTU, sendo os seus registos de 16 *bits* (2 *bytes*). O formato de dados é de virgula flutuante IEEE754, de 4 *bytes*, sendo necessário o pedido de 2 registos consecutivos para obter um parâmetro. O primeiro registo lido é sempre o mais significativo. Permite também pedir até 80 registos por transação [136][137]. O modelo SDM230, permite ler correntes até 100 A, com uma precisão de 0,5%. Tem um ecrã retro iluminado, montagem em calha DIN e 2 saídas em pulsos, configuráveis. Permite medir tensão, corrente, potência, consumos totais e parciais, frequência, fator de potência, entre outros. O modelo SDM530, tem características similares, mas adaptado a linhas trifásicas.

Os dados seriam obtidos a partir de um módulo emissor (Lopy Power Board V1), ligado em modo *Master* às linhas A e B MODBUS do barramento RS-485. O módulo emissor iria estar ligado a vários medidores SDM em modo *Slave*, com diferentes endereços Modbus, diferenciando desta forma os equipamentos. Neste, eram efetuados pedidos de leitura de registos (*read holding registers*) aos respetivos endereços do equipamento, para obtenção dos valores que queríamos guardar. Por exemplo, utilizando o esquema da Figura 4.17, para obter a tensão na linha do medidor de endereço 10, devemos solicitar a este o valor presente no registo de endereço 30001. Esta solicitação é feita efetuando um pedido *read holding registers*. O medidor irá devolver então 4 *bytes*, contendo o valor 231,4. A Figura 4.18 mostra os dois equipamentos.

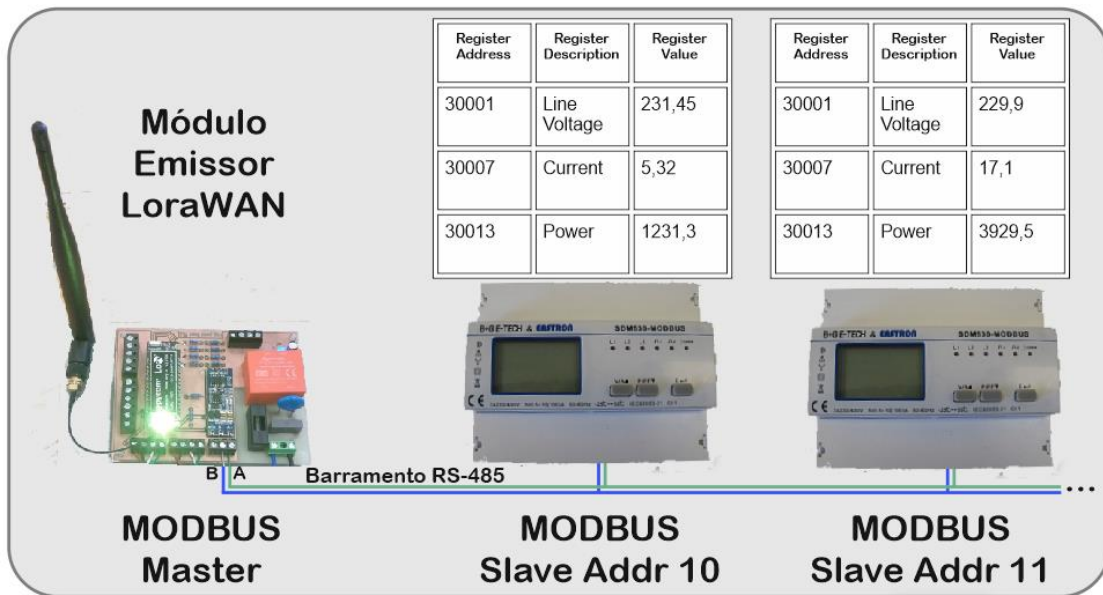


Figura 4.17 - Exemplo de barramento MODBUS



Figura 4.18 – Medidor de energia SDM230 (A). Medidor de energia SDM530 ligado a uma Lopy Power Board V1 (B)

4.4.4 TRANSDUTORES PARA MEDIÇÕES DC

Para as medições em corrente contínua, criámos vários módulos consoante a sua aplicação. Após a análise dos módulos necessários (capítulo 4.4.1), estavam definidos 4 módulos distintos para diferentes aplicações: 1) Módulo de leitura de quatro correntes, 2) módulo de leitura de uma tensão, 3) módulo de leitura de uma tensão e uma corrente e 4) módulo de leitura de três correntes e uma tensão. Nesta altura, foram analisadas diferentes tecnologias de transdutores de tensão e corrente, assim como o material existente. O Apêndice L descreve um pouco das tecnologias, assim como o material escolhido e o dimensionamento efetuado.

4.4.4.1 Transdutores de Corrente

Para escolher os transdutores de corrente, foi tido em consideração o cenário INTA. A maioria dos sensores a utilizar são para serem instalados em locais onde as correntes são inferiores a 32 Amperes, fluindo apenas numa direção (tabelas no Apêndice L). No caso das baterias de armazenamento, são necessários 3 sensores bidirecionais, para medição de correntes mais elevadas. Como existe disponível alimentação a 5 V nos módulos emissores, prefere-se utilizar esta tensão para alimentar os transdutores. Em termos físicos pretende-se também transdutores com um tamanho de abertura suficiente, de forma a permitir passar o condutor elétrico do respetivo circuito primário. Com estes valores em mente analisaram-se vários transdutores presentes no mercado considerando os seguintes fatores:

- Tensão de alimentação, unipolar ou simétrica e valor;
- Corrente nominal e máxima;
- Tecnologia utilizada;
- Preço;
- Precisão;
- Saída em tensão ou corrente e respetivos valores e condições de funcionamento;
- Dimensão das aberturas, por onde o primário atravessa o transdutor;

Com estes resultados foi construída a tabela do Apêndice N, que contém as características de cada modelo, de forma a escolher o transdutor mais correto para cada caso. Atendendo às características requeridas, a escolha recaiu no modelo HO 10-P da LEM para correntes até 32 A. Esta utiliza tecnologia de *loop* aberto, onde as limitações mais significativas são de correntes alternadas, não influenciando assim a nossa aplicação em DC. Estes módulos têm um custo reduzido, são pequenos e de fácil utilização, sem necessidade de circuito de condicionamento do sinal. A maior desvantagem dos mesmos prende-se com a precisão de 1,35% na corrente nominal e a necessidade de fornecer uma tensão de referência externa ao mesmo. Escolheu-se a versão alimentada a 5V e para fornecer a tensão de referência utilizou-se o *chip* MCP1700.

Para as correntes superiores ou bidirecionais, a escolha recaiu no Tamura L31S050S05FS, com uma corrente nominal de 50 A e máxima de 150 A, precisão de 0,5%, saída em tensão e montagem em painel. Mais uma vez, apresentam um custo reduzido e são de fácil utilização, sendo que estes têm uma abertura maior, ideal para os cabos de correntes superiores.

Adicionalmente, têm uma saída com a tensão de referência utilizada internamente, podendo assim utilizá-la como referência em circuitos de condicionamento do sinal.

4.4.4.2 Transdutores de Tensão

Para os transdutores de tensão escolhemos o LEM LV-25P de efeito de Hall. Na altura de comparar o material existente que garantisse isolamento galvânico, verificou-se a existência de soluções tradicionais com transformadores, óticas ou capacitivas. Estas necessitavam de garantir isolamento entre fontes de alimentação, pois muitos necessitam de alimentação independente no circuito de medição e no circuito de condicionamento. Embora o custo fosse inferior, tornavam-se menos práticas para a criação de módulos remotos, alimentados por uma única fonte e de tamanho mais compacto. Dentro dos transdutores de efeito de Hall, o material existente é bastante limitado, sendo a escolha deste transdutor unicamente pela facilidade de aquisição. Este garante-nos o isolamento galvânico entre os circuitos primário e secundário, e respetiva proteção do circuito de medição, não necessitando de duas fontes distintas. Necessita apenas de uma alimentação simétrica a ± 15 V, consumindo no máximo 45 mA (10 mA para funcionamento mais a corrente máxima no secundário de 35 mA), e tem uma precisão de 0,8%. Apesar da qualidade, o custo deste transdutor é elevado, rondando os 60 euros. O Apêndice M.6 descreve o dimensionamento deste transdutor para os módulos seguintes.

4.4.5 MÓDULO DE MEDIÇÃO DE 4 CORRENTES (DC_4I)

Para aplicação nos pontos que requerem a leitura de 4 correntes, desenhou-se uma placa contendo 4 transdutores LEM HO 10-P. A placa é alimentada a 5V e contém um *chip* MCP1700 de 1,2 V que gera a tensão de referência, a aplicar nos HO. Na saída deste é colocado um condensador de 1 uF, para garantir a correta estabilização da tensão [138]. O conversor ADC é o módulo ADS1115 (ver Apêndice M.3) da Adafruit, devido à necessidade de utilização de componentes *through hole*. Neste deixamos o pino de seleção de endereço ligado a um *header*, que com um *jumper* permite seleccionar o endereço 0x48 ou 0x49. Deixamos adicionalmente, um *header* para fazer *bypass* do quarto transdutor LEM e medir a tensão de referência, caso seja necessário saber o valor da mesma. Colocamos também, um led de indicação visual de alimentação. O consumo medido rondou os 80 mA, e o custo de

todo o material fica em 47 euros. O esquemático da mesma encontra-se em Apêndice Q. As imagens da Figura 4.19 mostram o esboço 3D e a placa protótipo.

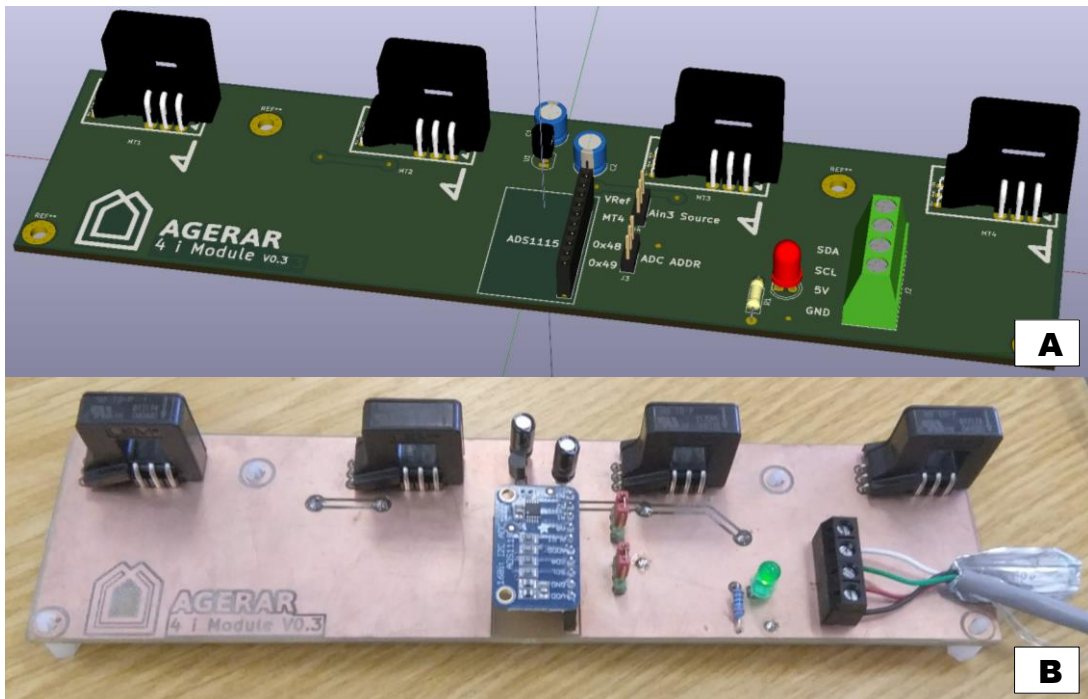


Figura 4.19 - Pré visualização 3D (A) e placa protótipo de leitura de 4 correntes (B)

4.4.6 MÓDULO DE MEDIÇÃO DE TENSÃO DC (DC_1V)

O módulo de medição de tensão DC, para ler apenas uma única tensão, é implementado com um transdutor LV25-P (ver Apêndice M.6). Como este necessita de uma alimentação simétrica de $\pm 15V$, foi utilizado um conversor DC-DC, elevando a tensão de 5 V para os $\pm 15V$. O consumo do LV25-P não passa dos 50 mA, então, optou-se pelo conversor RECOM RD-0515D/P [139]. Este debita 66 mA e tem proteção contra curto circuitos, além de não necessitar de componentes adicionais e ter custo reduzido. No entanto o seu consumo pode ser algo elevado, dependendo da carga do mesmo, sendo por isso necessário garantir 400 mA disponíveis para cada placa destas. Utilizamos um módulo ADS1115 da Adafruit e colocamos um *header* para seleção de endereço (48 ou 49), assim como um led para indicação de funcionamento. A resistência de carga é de 118 Ω , com 1% de tolerância (Apêndice M.6). Uma vez que este módulo só se utiliza num local (u6, Pb-Ac) com tensão nominal máxima de 408 V, as 4 resistências de 5 W teriam o valor de 10 k Ω . Estas estão colocadas com algum espaço entre elas, de forma a facilitar a dissipação. O consumo da placa em vazio rondou os

80 mA e com tensão nominal na entrada aproximou-se dos 200 mA. O custo é algo elevado, rondando os 85 euros, devido ao transdutor LV 25-P. O esquemático da mesma encontra-se no Apêndice R. As imagens seguintes mostram o esboço 3D e a placa protótipo desenvolvida.

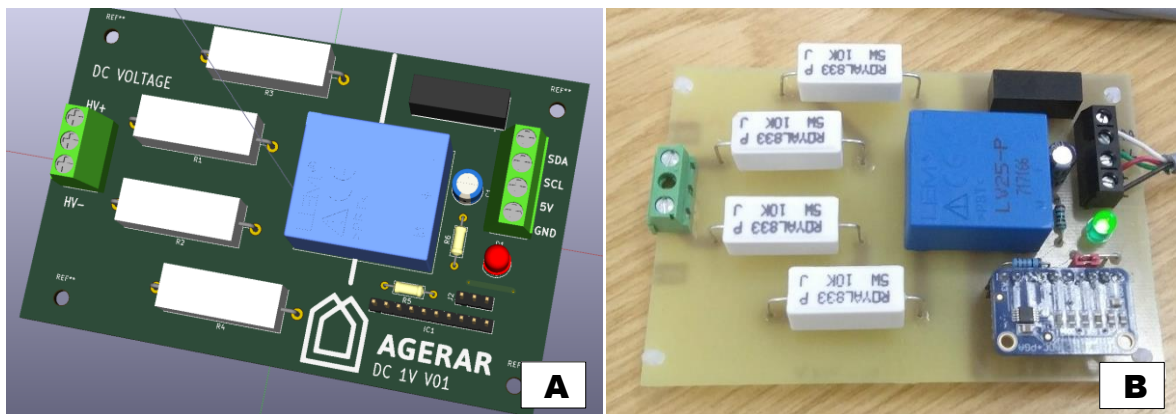


Figura 4.20 - Pré visualização 3D (A) e placa protótipo de leitura de 1 tensão (B)

4.4.7 MÓDULO DE MEDIÇÃO DE TENSÃO E CORRENTE DC (DC_1V_1i)

Para desenvolvimento do módulo de medição de uma corrente e de uma tensão DC (aplicável em PV2, PV3 e PV4), utilizou-se o mesmo material descrito anteriormente. A placa desenhada está representada na Figura 4.21, estando o esquemático no Apêndice S. Juntou-se um transdutor LV25-P e um LEM HO 10-P na mesma placa, em conjunto com os outros componentes necessários. Nesta placa apenas se introduziu um *header* adicional, de forma a poder utilizar os 4 endereços do ADS1115. Como nestes 4 locais considerámos a tensão

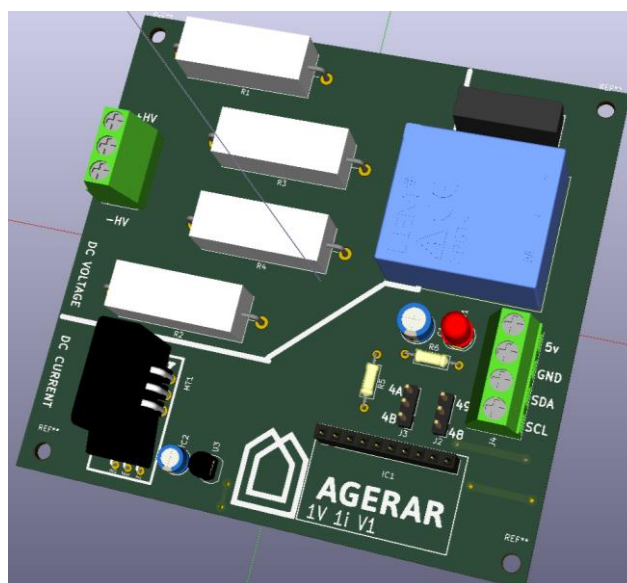


Figura 4.21 - Pré visualização 3D

nominal máxima dos PV, 320 V, as resistências de 5 W têm o valor de 8 k Ω . O custo desta placa fica a 93 euros.

4.4.8 MÓDULO DE MEDIÇÃO DE TENSÃO E TRÊS CORRENTES DC (DC_1V_3I)

O Módulo de medição de tensão e de três correntes DC irá ser aplicado unicamente nas baterias de armazenamento, pois é o único que permite a leitura de correntes bidirecionais (i9, i10, i11). Adicionalmente irá contar com um transdutor de tensão (u7), para leitura da tensão do banco de baterias Li-Ion. Foi dimensionado especificamente para a INTA e deverá permitir ligar os três transdutores Tamura, condicionar o sinal, e aplicá-lo ao ADC, conforme descrito no Apêndice M.5. Mais uma vez utilizaremos o ADS1115 para a leitura das saídas dos transdutores, sendo que nesta placa deixamos acessíveis os 4 possíveis endereços, através de 2 headers. Para o condicionamento do sinal (ver Apêndice M.5), foram utilizados dois *ampop* LM378, alimentados a ± 12 V por um conversor RECOM RD-0512D/P [139]. O LV25-P é alimentado pelo RECOM RD-0515D/P. As resistências do primário, são de 6,8 k Ω , para a tensão nominal máxima de 273 V. O custo do módulo fica em 95 euros, excluindo os transdutores. A Figura 4.22 mostra o esboço do mesmo, estando o esquemático no Apêndice T.

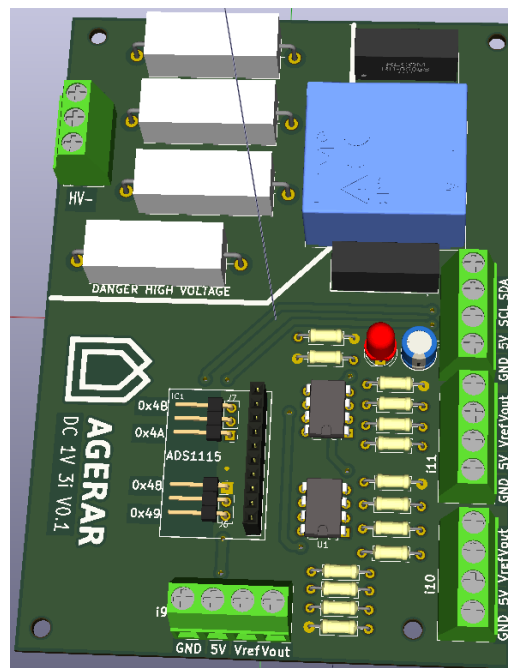


Figura 4.22 - Pré visualização 3D da placa protótipo de leitura de 1 tensão e 3 correntes bidirecionais

4.5 SOFTWARE DOS MÓDULOS DE MEDIÇÃO

4.5.1 LÓGICA DE EXECUÇÃO DO CÓDIGO FONTE

A estrutura lógica do *software*, implementado no módulo emissor LoPy foi idêntica em todos os módulos, variando neles apenas a obtenção (I2C ou MODBUS) e o processamento das leituras obtidas. A Figura 4.23 mostra o *flowchart* genérico dos módulos emissores.

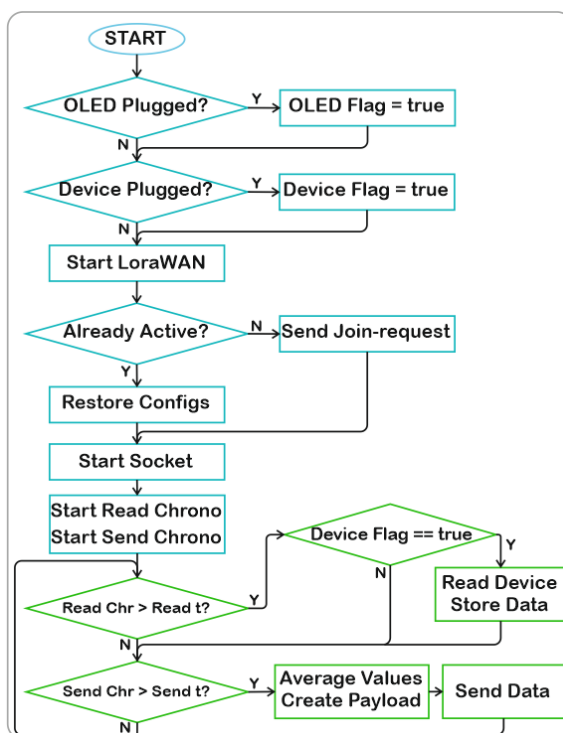


Figura 4.23 - Flowchart genérico de funcionamento dos módulos emissores

No arranque o módulo emissor verifica a presença de um ecrã OLED na linha I2C a 3,3 V, pesquisando se existe algum dispositivo com o endereço 0x3C. Se for detetado, é colocada a *true* uma *flag* que indica a presença do mesmo. Este ecrã é útil para *debug* e visualização dos dados do emissor, tendo sido para esse efeito foi utilizado um OLED SSD1603, de 128 por 64 linhas. De seguida é verificada a presença de dispositivos externos de medição nas linhas MODBUS e/ou I2C a 5 V. Para tal é feita uma solicitação de leitura de registos para os endereços configurados (seja MODBUS, seja I2C), e caso haja resposta, conclui-se que o dispositivo está presente na linha. É então colocada a *true* uma *flag* respeitante àquele dispositivo. Esta verificação só ocorre no arranque, sendo que é necessário ter os dispositivos externos ligados no arranque. Por exemplo, o emissor Lopy Power Board 4 lê dois dispositivos MODBUS (dois SDM530, com endereços 9 e 10) e um dispositivo I2C (DC_1V_1i, com

endereço I2C 0x49). No arranque, é solicitada a leitura do registo 30001 MODBUS, do *slave* com endereço 9. Se houver resposta, ativa-se a *flag* de presença do dispositivo 0. Repete-se o procedimento para o *slave* 10, com *flag* de dispositivo 1. No caso I2C, verifica-se apenas a presença de um dispositivo com endereço 0x49, sendo que em caso positivo se coloca a *true* a *flag* do dispositivo 2.

Após a verificação dos dispositivos externos presentes, iniciam-se as comunicações LoraWAN, verificando se já houve ativação na rede, através de uma variável guardada na memória persistente do microcontrolador (*non volatile storage*, NVS). Se for a primeira vez que o dispositivo liga, é feito o pedido de *join*, caso contrário apenas se recuperam as configurações já guardadas (por exemplo, contadores, canais, frequências e DR). Abre-se então a *socket* LoraWAN, onde se define o DR inicial e a utilização ou não do mecanismo ADR.

Iniciam-se aqui 2 cronómetros, um para leituras, outro para os envios, e entra-se no *loop* principal. Sempre que o cronómetro de leituras for superior ao tempo definido (2 segundos), é efetuado um pedido de leitura aos dispositivos presentes nas linhas, e reiniciado o cronómetro. É verificada a *flag* de dispositivo, sendo que caso seja *true*, é efetuado o pedido das leituras, sendo estas guardadas numa lista. Todas as *flags* de presença são verificadas, sendo que obviamente, só se efetuam pedidos de leitura quando esta é *true*, ignorando os casos *false*. Adicionalmente, se existir um OLED ligado, este apresenta os valores lidos. Nos casos MODBUS, para obter as leituras são efetuados pedidos a dois registos de 2 *bytes* consecutivos, sendo necessário juntá-los em 4 *bytes* de formato *float*. Este procedimento é feito logo após as leituras e antes de os guardar. No caso de leituras dos ADCs externos, os valores guardados são as leituras em bruto do mesmo, sendo por isso um valor inteiro.

Após as leituras, é verificado no cronómetro de envio, se já ultrapassou o tempo estipulado (20 segundos). Se for o caso, reinicia-se o cronómetro e efetuam-se as médias de todos os valores guardados das leituras. Processam-se estes valores, convertem-se no valor desejado e adaptam-se para o envio. Como dos ADCs temos os valores em bruto, é necessária a sua conversão em tensão ou corrente (ver capítulos 4.5.2, 4.5.3, 4.5.4). A adaptação destes para o envio, resulta do método descrito no capítulo 4.3.9, e consiste na transformação de *float* em *int*. Cria-se então o *payload*, agrupando os dados no formato descrito na Tabela 4.2, e procede-se ao envio. O ciclo volta ao início, verificando o cronómetro de leitura.

De referir, que várias outras funcionalidades estão implementadas no código. Um dos mecanismos para garantir que o módulo não fica parado se ocorrer algum erro, é a utilização de um *watchdog*. Este atua como um temporizador que força o microcontrolador a reiniciar,

se não for “alimentado” regularmente. No nosso caso, iniciamo-lo com 60 segundos, sendo este o tempo que temos para o “alimentar” (*feed*) antes de ele reiniciar. Sempre que se consiga efetuar uma leitura de um dispositivo externo, efetua-se um *feed* ao *watchdog*, reiniciando assim a contagem para mais 60 segundos.

Outro mecanismo de informação é a utilização de um *byte* de estado, enviado no *payload* LoraWAN. Neste, um *bit* a “1” representa quais dos 4 dispositivos externos estão desligados, ou se existiram erros de leitura nos dispositivos. O formato do *byte* é o da Tabela 4.3. Por exemplo, o *byte* 81h (10000001b) representa erros de leitura no dispositivo 4 e dispositivo 1 desligado no arranque. Outro exemplo, o *byte* 34h (00110100b) representa erros de leitura no dispositivo 1 e 3, estando o dispositivo 3 desligado no arranque.

Bit position	msb				lsb			
	7	6	5	4	3	2	1	0
Device	4	3	2	1	4	3	2	1
Status	Reading Errors				Not Conected			

Tabela 4.3 – Formato de byte de estado do nó

Adicionalmente, utiliza-se o led RGB existente no Lopy, para indicar visualmente o estado do mesmo. No arranque, um piscar vermelho de 0,5 segundos indica que se está a ativar na rede, sendo que após o processo estar concluído, pisca a cor verde por duas vezes. Um piscar rápido a cor amarela, indica que está no *loop* principal. A cada leitura com sucesso aos dispositivos, o led pisca a cor verde rapidamente e uma única vez. Uma falha nestas leituras faz piscar a vermelho 2 vezes, também rapidamente. Um envio LoraWAN com sucesso faz piscar de forma rápida a cor azul uma vez, enquanto uma falha no envio liga a cor vermelha durante 1 segundo.

4.5.2 CONVERSÃO DE VARIÁVEIS NO TRANSDUTOR HO 10-P

Para obter os valores de corrente ou tensão, é necessária a conversão do valor inteiro obtido. Este valor inteiro (ADC_{I_p}) varia entre 0 e 32767, sendo necessário efetuar a conversão na variável pretendida. Para o caso dos transdutores HO 10-P, pretende-se obter a corrente no primário (I_p). O primeiro valor a calcular é o correspondente de tensão (V_s) à entrada do ADC (saída do transdutor HO). Esta conversão tem em conta a escala do ADC (V_{FSR}) e o número de *bits* utilizados (N):

$$V_s = \frac{V_{FSR} * ADC_{I_p}}{2^{N_{bits}}} [V] \quad (22)$$

Posteriormente, e utilizando a fórmula 36 do Apêndice M.4, obtém-se a corrente no primário, subtraindo de V_s a tensão de referência (V_{ref}) e dividindo pela sensibilidade do transdutor (G).

$$I_p = \frac{V_s - V_{ref}}{G} [A] \quad (23)$$

Simplificando apenas numa fórmula:

$$I_p = \frac{V_{FSR} * ADC_{I_p} - V_{ref} * 2^{N_{bits}}}{G * 2^{N_{bits}}} [A] \quad (24)$$

A única variável é ADC_{I_p} , pois os outros valores são constantes e configurados no microcontrolador. Como só se usa metade da escala *full range* do ADC, V_{FSR} é 4,096 V com N igual a 15 *bits*, e uma tensão de referência de 1,2 V (V_{ref}), a equação 24 pode ser transformada em:

$$I_p = \frac{4,096 * ADC_{I_p} - 1,2 * 32678}{0,1 * 32678} = \frac{4,096 * ADC_{I_p}}{3267,8} - 12 [A] \quad (25)$$

Esta será a equação que nos permite obter a corrente no primário a partir do valor lido no ADC. O cálculo no microcontrolador utiliza os valores sem arredondamento. Adicionalmente, poderemos somar ou subtrair um *offset* ao valor lido pelo ADC, de forma a compensar alguma variação fixa do ponto zero, no transdutor.

4.5.3 CONVERSÃO DE VARIÁVEIS NO TRANSDUTOR TAMURA

No caso dos transdutores Tamura, obtém-se o valor inteiro do ADC em *full range*, o qual varia entre -32768 e 32767. O primeiro passo passa por obter a tensão correspondente à saída do condicionador, e posteriormente convertê-la na corrente primária. A obtenção da tensão (V_s) utiliza a fórmula 22, onde a escala do ADC (V_{FSR}) vale 8,192 V e o número de *bits* utilizados (N) é 16 (ver Apêndice M.5). Na conversão da corrente primária, os circuitos de 32 e 120 A têm sensibilidades diferentes, devido ao circuito condicionador. Para obter a corrente no primário (I_p), basta dividir a tensão obtida pelo ADC (V_s) pela sensibilidade do circuito (G).

$$I_p = \frac{V_s}{G} [A] \quad (26)$$

No caso de 32 A, a sensibilidade é 127 mV/A. Assim, a equação que nos dá a corrente no primário é:

$$I_p = \frac{V_{FSR} * ADC_{I_p}}{G * 2^{N_{bits}}} = \frac{8,192}{0.127 * 2^{16}} * ADC_{I_p} = \frac{8,192}{8323,072} * ADC_{I_p} \quad [A] \quad (27)$$

Para 120 A, a sensibilidade tem o valor de 33,75 mV/A. A equação aplicável é a seguinte:

$$I_p = \frac{V_{FSR} * ADC_{I_p}}{G * 2^{N_{bits}}} = \frac{8,192}{0.03375 * 2^{16}} * ADC_{I_p} = \frac{8,192}{2211,84} * ADC_{I_p} \quad [A] \quad (28)$$

4.5.4 CONVERSÃO DE VARIÁVEIS NO TRANSDUTOR LV 25-P

Os transdutores de tensão são dimensionados em função das tensões nominais do circuito primário (ver Apêndice M.6). Assim, para cada um dos 3 casos INTA teremos uma sensibilidade diferente, sendo necessário utilizar as tensões nominais do primário e secundário. Para obter a tensão no secundário (V_s), utilizamos novamente a fórmula (22), com V_{FSR} de 4,096 V e o número de *bits* (N) utilizados, 15. Esta tensão, está relacionada com a corrente I_s na resistência de carga R_m (118 ohm), pela seguinte expressão:

$$I_s = \frac{V_s}{R_m} = \frac{V_{FSR} * ADC_{V_s}}{118 * 2^{N_{bits}}} = \frac{4,096}{3\ 866\ 624} * ADC_{V_s} \quad [A] \quad (29)$$

Sabemos também que a corrente primária I_p se relaciona com a corrente secundária I_s por:

$$I_p = \frac{I_s}{2,5} = \frac{4,096}{9\ 666\ 650} * ADC_{V_s} \quad [A] \quad (30)$$

E que a tensão no primário V_p , é dada pela lei de Ohm, utilizando a corrente I_p e resistências R_1 e R_p :

$$V_p = I_p * (4 * R_1 + R_p) \quad [V] \quad (31)$$

Simplificando tudo numa única expressão:

$$V_p = \frac{4,096}{9\ 666\ 650} * ADC_{V_s} * (4 * R_1 + R_p) \quad [V] \quad (32)$$

Como cada tensão nominal tem um valor diferente de R_1 , esta assume 3 valores diferentes consoante o local de aplicação. Nas baterias de íões de lítio (u7), com tensão nominal de 273 V, foi escolhida uma resistência de 6,8 kΩ. Assim, no total, as quatro resistências no primário mais a resistência interna do transdutor valem 27428 ohm:

$$V_p = \frac{4,096 * 27428}{9\ 666\ 650} * ADC_{V_s} = \frac{112345,088}{9\ 666\ 650} * ADC_{V_s} [V] \quad (33)$$

No caso dos parques fotovoltaicos (u1, u2, u3), as resistências têm o valor de 8 kΩ, resultando num total de 32228 ohm. A expressão será então:

$$V_p = \frac{4,096 * 32228}{9\ 666\ 650} * ADC_{V_s} = \frac{132005,888}{9\ 666\ 650} * ADC_{V_s} [V] \quad (34)$$

Para o barramento DC (u6), o valor escolhido foi de 10 kΩ, resultando num total de 40228 ohm e na expressão:

$$V_p = \frac{4,096 * 40228}{9\ 666\ 650} * ADC_{V_s} = \frac{164773,888}{9\ 666\ 650} * ADC_{V_s} [V] \quad (35)$$

5. IMPLEMENTAÇÃO PRÁTICA DA ARQUITETURA PROPOSTA

Neste capítulo descrevem-se os primeiros ensaios efetuados aos módulos criados e à rede LoraWAN. Faz-se também uma primeira análise dos resultados obtidos em cada teste. No final do capítulo mostra-se o sistema completo, em fase de instalação na micro rede.

5.1 ENSAIOS INICIAIS

Antes de toda a rede ser instalada, foram testados de forma individual, diversos componentes. Desta forma pode ser avaliado o comportamento destes, antes da instalação final.

5.1.1 ENSAIO DE COBERTURA INTA

Por forma a verificar o comportamento da rede LoraWAN, incluindo os seus diversos módulos e algum do material definido no contexto real onde irá ser aplicada, foi feita uma visita às instalações da INTA.

Antes da visita, o local de instalação foi simulado, em termos de propagação do sinal e potência recebida, utilizando o *software* CloudRF, considerando um modelo de propagação *irregular terrain model* (ITM ou modelo Longley-Rice) e resolução de 90 metros (SRTM3). A *gateway* foi considerada o emissor, com uma antena dipolo vertical de ganho 0,6 (Lynx), potência de emissão de 14 dB e instalada a 3 metros de altura. Considerou-se que os recetores eram colocados a 1 metro de altura, com um ganho de 2,14 dBi e utilizando uma antena dipolo vertical. A Figura 5.1 mostra os resultados obtidos, sendo importante ressaltar que estas simulações não incluem a propagação no interior de edifícios ou a atenuação por estes causada. Esta simulação permitiu perceber que deveria existir sinal em todo o recinto, embora nas zonas mais afastadas a potência recebida possa ser baixa para DR5.

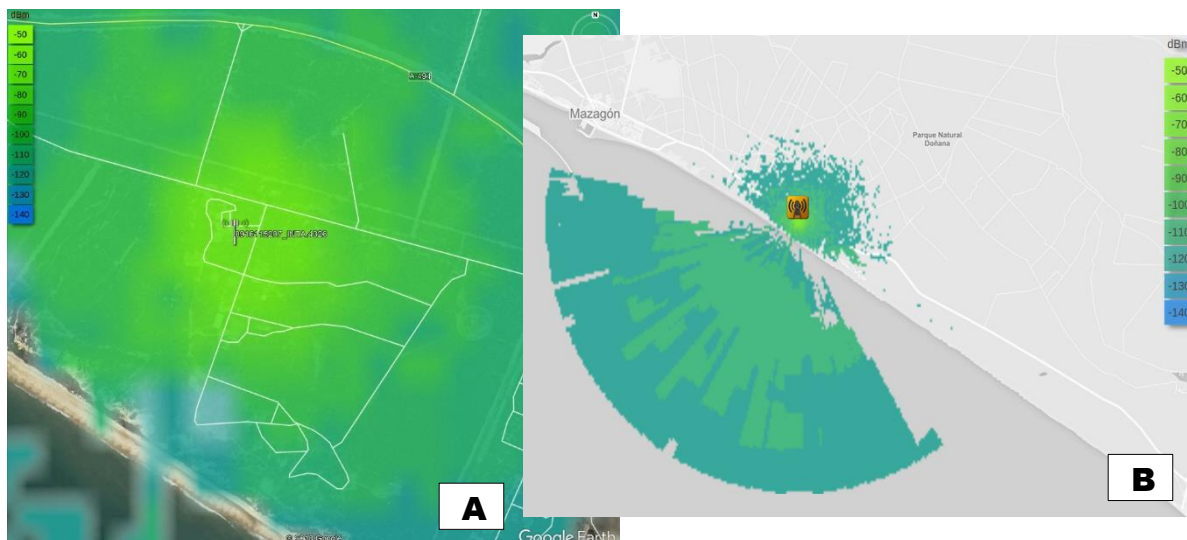


Figura 5.1 - Simulação de potência recebida, na área da INTA. Resultado da simulação em CloudRF (A) e mesma simulação em Google Earth (B)

Na visita à INTA, pretendeu-se verificar em contexto real se se mantinham as coberturas simuladas na área dos edificios principais e o alcance para os veículos. O setup LoraWAN (*gateway*, *router* Wi-Fi e servidor de aplicações Thingsboard) foi colocado no interior da sala dos armários elétricos, onde existem também os dois bancos de baterias. Para perceber o alcance e as condições de receção do sinal, um nó LoPy+Pytrack, alimentado por *powerbank*, percorreu várias zonas do complexo INTA. A antena utilizada na *gateway* foi idêntica à dos Lopy.

No interior e zonas ao redor dos edificios, a qualidade do sinal Lora foi sempre alta, sendo o RSSI mínimo obtido de -88 dB com DR5 e a LSNR positiva. Assim, no edificio e ao seu redor a cobertura será sempre garantida com o débito mais elevado. A Figura 5.2 mostra o resultado obtido.

As deslocações efetuadas no recinto da INTA permitiram perceber como o sinal irá variar com a distância. Até sensivelmente 250 metros conseguiu-se assegurar a receção de dados, com DR5. A partir desta distância, verificaram-se algumas perdas de pacotes. Este facto pode ser verificado pelos saltos no contador FCnt e pela não receção de alguns pacotes, tal como apresentado na Figura 5.3, no gráfico inferior esquerdo. Esta situação verificou-se para um RSSI inferior a -100 dBm e um LSNR abaixo de -4 dB. A origem das perdas passa provavelmente pela proximidade do valor de LSNR ao limite de -7,5 dB, a partir do qual não se conseguem decodificar as tramas.

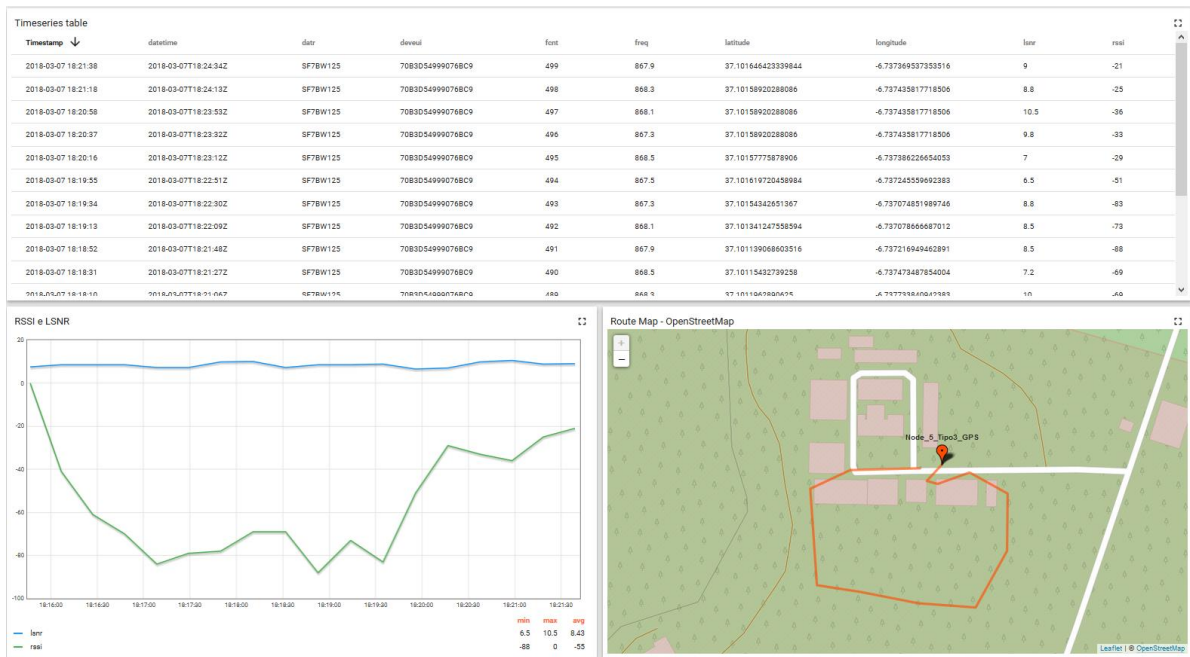


Figura 5.2 - Site survey em redor do edifício principal, imagem da dashboard Thingsboard

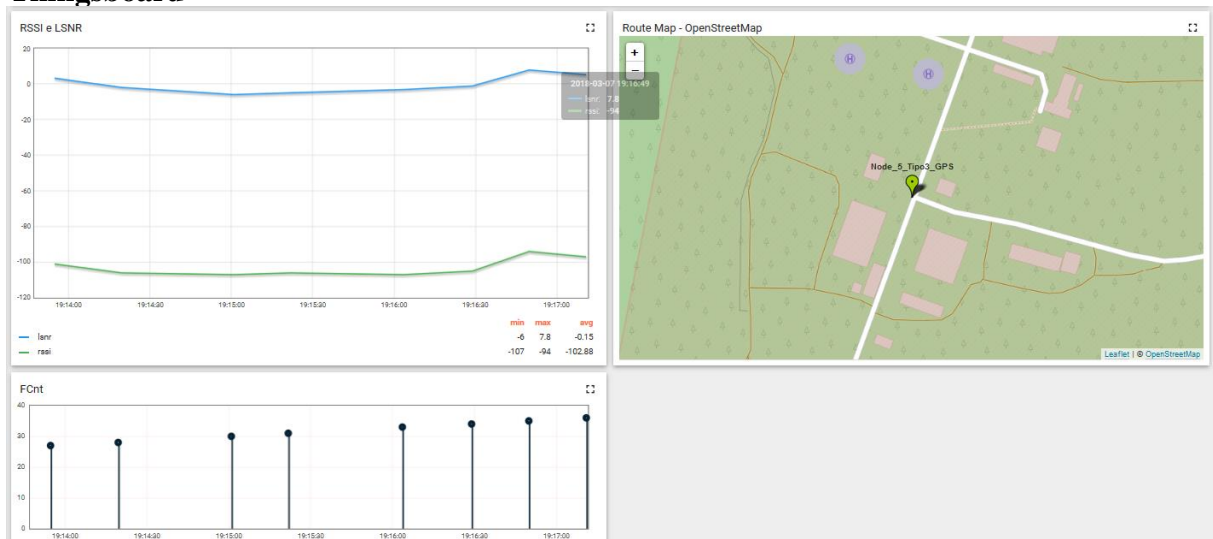


Figura 5.3 - Estado de receção à distância aproximada de 250 metros com DR5, imagem da dashboard Thingsboard

A partir desta região forçou-se a alteração do débito para DR3, tendo-se verificado que a perda ocasional de pacotes se mantinha. Nesta fase dos testes obteve-se uma média de -8,26 dB para o LSNR e -106.38 dBm para o RSSI, tendo conseguido um alcance de 450 metros. Nesta situação, a LSNR aproximou-se do limite de -12 dB, sendo que a variação da mesma em torno deste valor, pode levar a que certos pacotes tenham um LSNR inferior e o pacote não seja corretamente decodificado no recetor. Nesta distância forçou-se a alteração de débitos para DR0, tendo-se conseguido aumentar a receção de dados até uma distância de 670 metros, mas com uma elevada perda de pacotes. Mediu-se um LSNR médio de -12,16 dB e RSSI de

-109,6 dBm, que são valores distintos dos teóricos, mas reais, dadas as condições do teste. A Figura 5.4 (topo esquerdo) mostra o trajeto onde foram feitas as medições. Na mesma figura, em cima, à direita apresenta-se a potência de recepção simulada, no local mais afastado onde se obteve sinal. Na parte de baixo da figura, temos o perfil de propagação simulado, para esse mesmo local. O valor real obtido (-109,6 dBm) e o valor simulado (-98 dBm), apresentam uma diferença substancial.

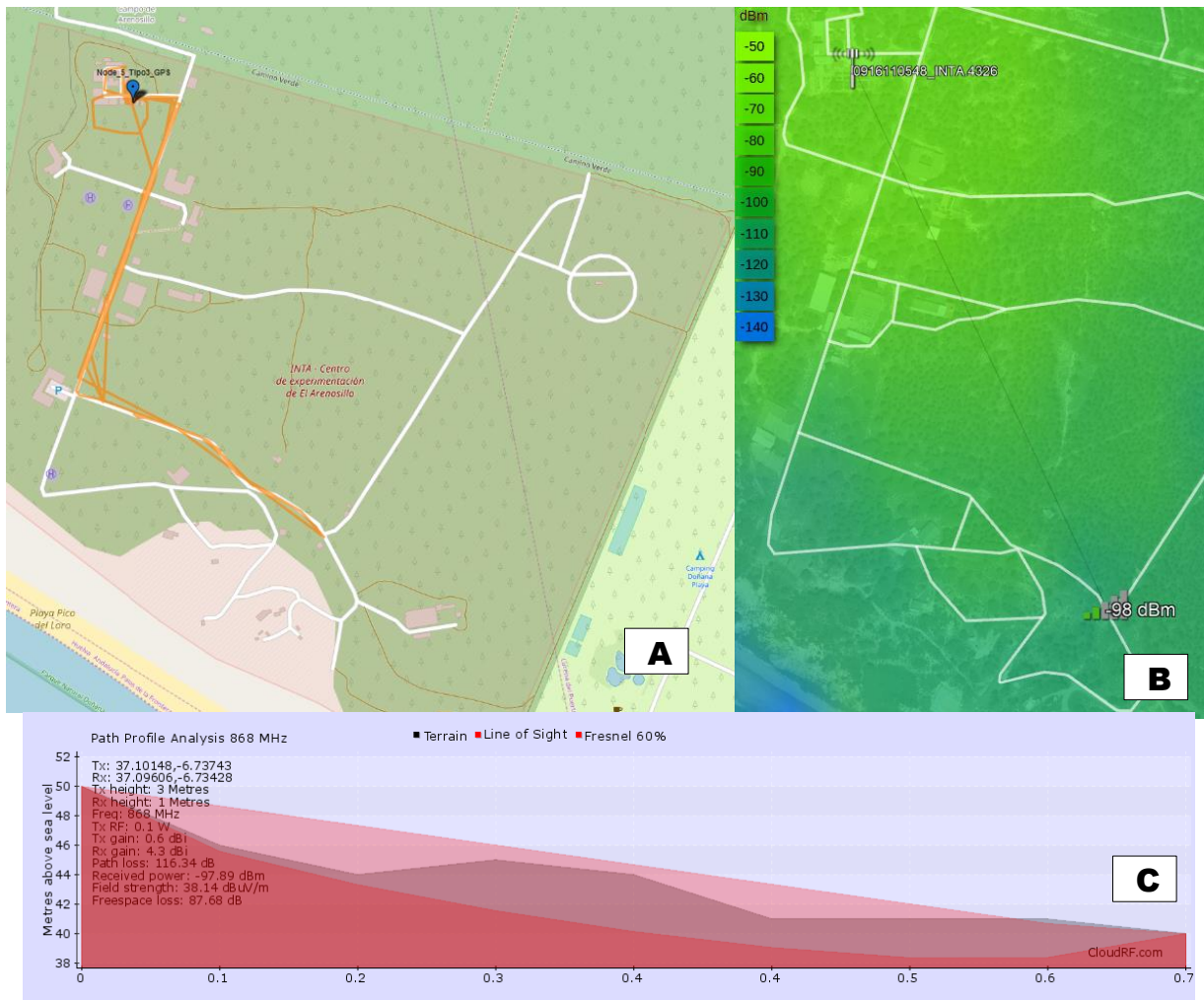


Figura 5.4 – Site survey ao recinto (A), potência recebida simulada no local extremo do site survey (B) e perfil de transmissão do sinal simulado (C)

Os teste realizados permitiram perceber que a localização e qualidade da antena da *gateway* são elementos importantes para se alcançar a cobertura de aproximadamente 1 km. Os dados obtidos na fase das simulações faziam prever um resultado mais satisfatório, embora estas não considerem as atenuações dos obstáculos nem interferências. O local onde foi colocada a antena durante o teste, está suscetível de muita interferência proveniente dos diversos equipamentos de alta potência e que criam bastante ruído eletromagnético. Ao mesmo tempo,

a antena estava dentro de um edifício com telhados e portões em chapa metálica, o que não é benéfico para a propagação do sinal.

Os resultados obtidos permitiram definir que na implementação final ir-se-á optar por colocar a antena num local exterior ao edifício e mais elevado. Utilizaremos também a antena Linx (referida no capítulo 4.3.2.4) de qualidade superior.

5.1.2 ENSAIO DE PERDA DE PACOTES LORAWAN

De forma a medir como se comportam diferentes configurações de DR LoraWAN, foram colocados 4 nós idênticos a transmitir nas mesmas condições, durante uma semana. Os quatro nós enviavam 8 *bytes* de dados, de 20 em 20 segundos, numa localização urbana a sensivelmente 400 metros da antena recetora, localizada no Laboratório 160 do Instituto Superior de Engenharia (ISE), da Universidade do Algarve. A Figura 5.5 mostra o *setup* utilizado.

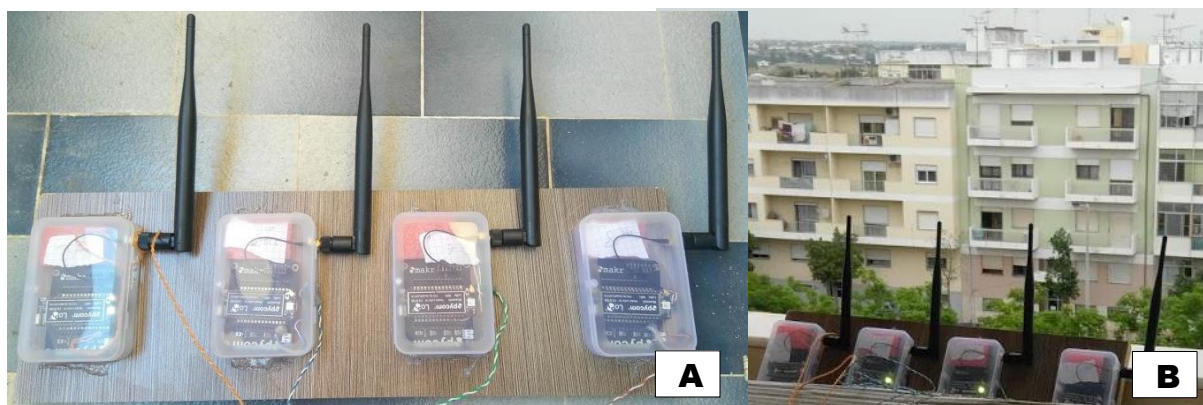


Figura 5.5 - Setup de 4 Lopy (A) e local de colocação do mesmo (B)

Dos 4 nós, um deles tinha ADR ativo, enquanto os outros tinham DR fixos de 5, 2, e 0. Utilizaram-se os mesmos 8 canais definidos no capítulo 4.3.2.5. Após esta semana, os dados foram recolhidos da plataforma ThingsBoard, em formato “csv”, de forma a permitir uma análise em Excel. Os parâmetros analisados incluíram a perda de pacotes global, as médias e variação de LSNR e RSSI durante a totalidade da semana, as percentagens de transmissões por canal, os valores de LSNR e RSSI por canal e por dia, e as perdas de pacotes diárias e horárias. No caso ADR, analisou-se também a percentagem de envios por canal e por DR. Por comparação, verificou-se a perda de pacotes que obtínhamos num nó fixo a DR5, presente no Laboratório 51, do edifício em U do ISE (aproximadamente a 40 metros da antena recetora), e que envia a temperatura e humidade a cada 60 segundos. A Tabela 5.1 resume os valores

obtidos, onde podemos constatar que obviamente, um DR mais baixo tem menor perda de pacotes. Os valores de RSSI são em média idênticos pois a distância não se altera. O ruído no canal, é difícil de quantificar pelo seu valor médio, no entanto o valor mínimo permite constatar que em DR0 se decodificaram mensagens com um valor inferior. O mesmo sucedeu com ADR, pois este atingiu DR1.

	ADR	DR5	DR2	DR0	Lab (DR5)
Total de pacotes enviados	28352	28370	28368	28195	14394
Pacotes perdidos	6943	12824	1464	668	72
Pacotes perdidos (%)	24.49	45.20	5.16	2.37	0.50
LSNR Média (dB)	-7.97	-4.50	-2.52	-4.62	8.87
LSNR Máximo (dB)	1.5	2.2	7.8	5.0	11.2
LSNR Mínimo (dB)	-19.5	-10.5	-17.50	-21.20	3.2
LSNR Mediana	-7.2	-4.8	-1.8	-4.2	9.0
LSNR Desvio Padrão	3.47	2.30	4.62	5.05	1.04
RSSI Média (dBm)	-115	-115	-113	-114	-80
RSSI Máxima (dBm)	-95	-105	-85	-35	-73
RSSI Mínima (dBm)	-123	-120	-131	-125	-85
RSSI Mediana	-115	-114	-113	-114	-80
RSSI Desvio Padrão	2.39	2.14	2.79	2.80	1.84

Tabela 5.1 - Valores totais obtidos

Sabemos que os nós escolhem os canais de forma pseudoaleatória, sendo que a distribuição de probabilidade deveria ser uniforme. Verificou-se que as mensagens recebidas em DR5 e ADR, não estão distribuídas uniformemente pelos canais. Tal pode dever-se a ruído nestas frequências, levando a perdas de pacotes nesses canais. A nível de perdas de pacote por dia da

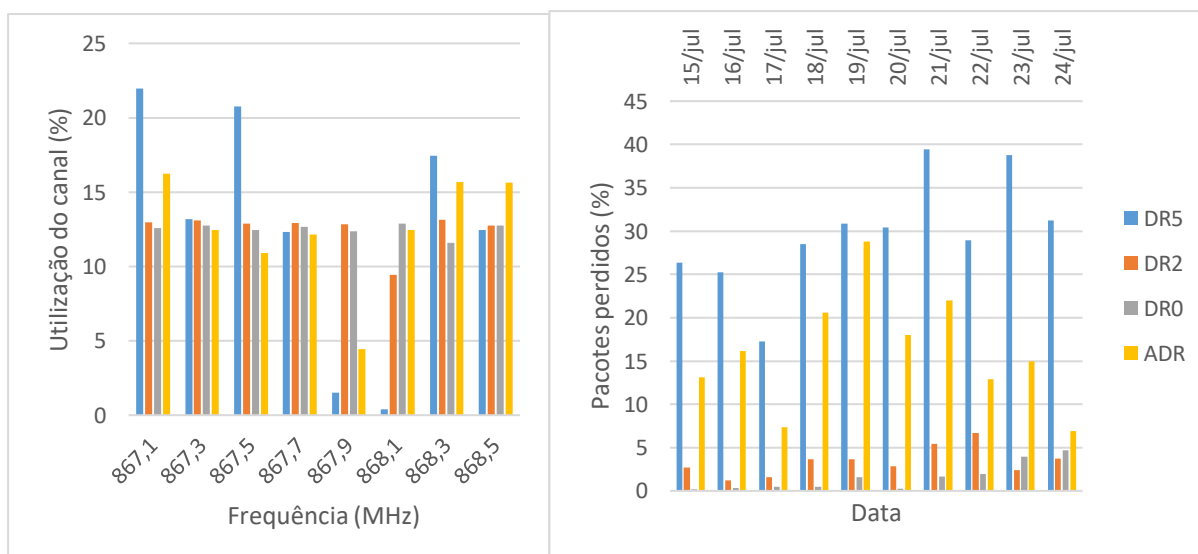


Figura 5.6 - Gráfico de mensagens recebidas por canal (esquerda) e pacotes perdidos por dia da semana (direita), em percentagem

semana, não se verificou nenhuma relação, sendo que nem ao fim de semana (dias 21 e 22) se notaram diferenças significativas na qualidade de recepção. Os gráficos da Figura 5.6 demonstram visualmente estes resultados.

A qualidade de recepção RSSI e o relação sinal ruído Lora, embora sofram variações, em termos médios não se verificam alterações entre os dias de semana e o fim de semana. A Figura 5.7 mostra os resultados obtidos.

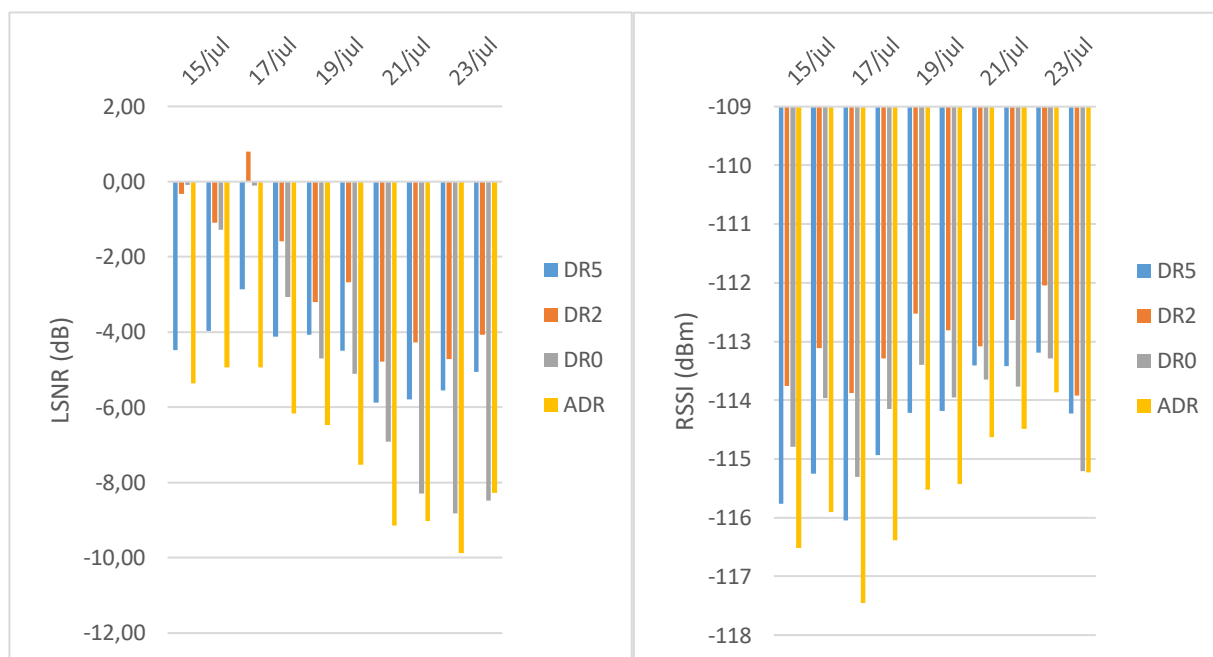


Figura 5.7 - LSNR e RSSI médias por dias

Em relação à perda de pacotes, quando analisada de forma individual e separada em horas, verifica-se que esta é maior durante o horário laboral diurno. Nos gráficos da Figura 5.8, vê-se uma maior ocorrência de perdas nesse horário, estando estes separados por dias e sobrepostos no mesmo gráfico.

Relacionando os dados de qualidade do sinal e a perda de pacotes, verifica-se que sempre que existem alterações do LSNR negativas, perdem-se mais pacotes. Curiosamente, estas alterações fazem subir o RSSI. A causa desta degradação da LSNR não é conhecida, no entanto estas ocorrem no período laboral, sendo por isso expectável que se deva a mais interferência proveniente das atividades realizadas na cidade. Nos gráficos da Figura 5.9, verificam-se estas ocorrências, onde nos traços a vermelho se podem ver as perdas de pacotes em valores absolutos (as quais devem ser lidas no eixo azul da esquerda e multiplicadas por 10) e a qualidade de recepção (RSSI e LSNR). Os dados foram calculados em períodos de uma hora, para os dias analisados, estando o eixo vertical dividido em grupos de 6 horas.

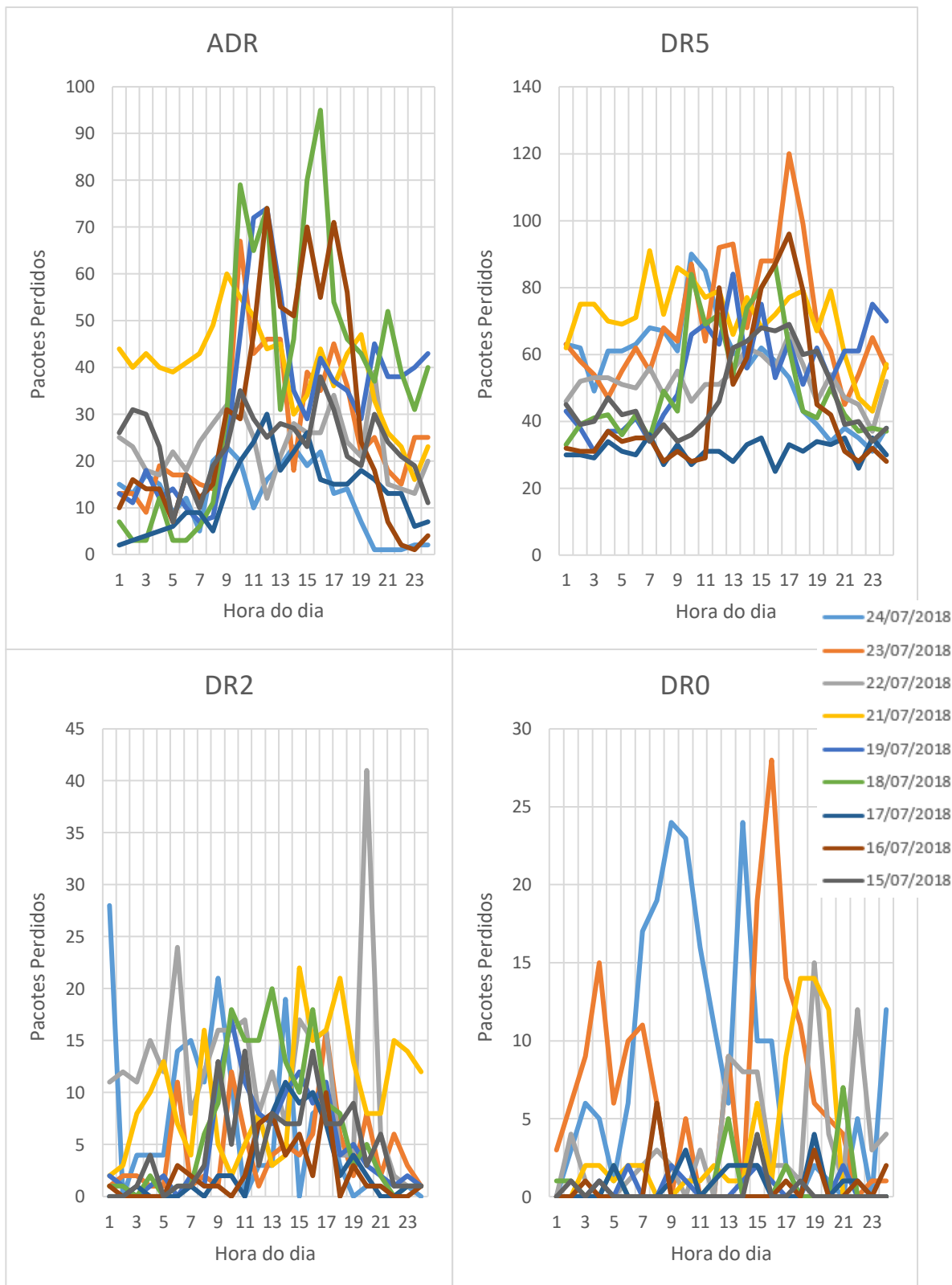


Figura 5.8 - Perda de pacotes por dia e hora, para os vários DR

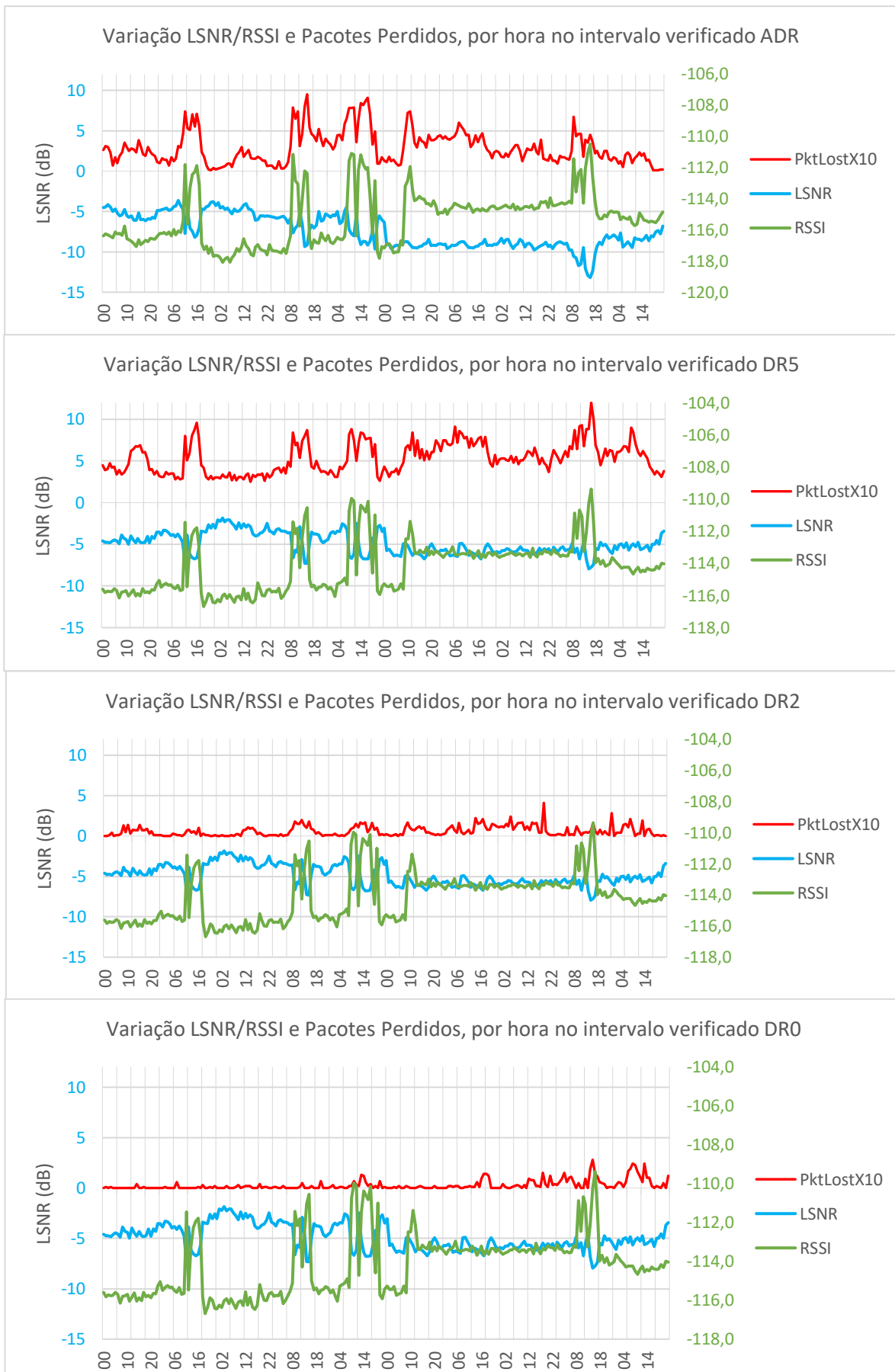


Figura 5.9 - Perda de pacotes e RSSI/LSNR, separados por configuração e por horário, a partir das 00:00 de dia 15

5.1.3 ENSAIO DOS MÓDULOS DE MEDIÇÃO COM HO 10-P

De forma a testar os módulos de leitura de corrente, foram efetuadas diversas leituras de corrente (entre 0 e 23 Amperes) numa carga controlável existente na INTA. Utilizou-se um módulo de medição de 4 correntes (DC_4i), obtendo os valores brutos do ADC em cada transdutor (tr0, tr1, tr2, tr3), e posteriormente efetuando a conversão com a fórmula (24). A obtenção dos valores em bruto permitiu-nos saber o que o ADC estava a obter, podendo verificar como se comportava a fórmula teórica e adaptando-a se necessário. Aos valores em bruto é retirado o valor lido pelo ADC à tensão de referência (9792, equivalente a 1,227 Volt, medidos na placa) e um *offset* individual, obtido de uma média de leituras com zero corrente (87, 105, 123 e 171 respetivamente). É então feita a conversão. Os dados obtidos são apresentados na tabela seguinte, onde para além dos valores em bruto, se apresenta a conversão para corrente primária (*I_p*) e o erro absoluto e relativo percentual. Não foram efetuadas várias leituras, de forma a verificar a estabilidade das leituras ao longo do tempo e corrigir eventuais *offsets*

Ho 10-P	Leituras Raw				Leituras convertidas											
	tr0	tr1	tr2	tr3	I _{p_0} (A)	Erro Absoluto	Erro Relativo (%)	I _{p_1} (A)	Erro Absoluto	Erro Relativo (%)	I _{p_2} (A)	Erro Absoluto	Erro Relativo (%)	I _{p_3} (A)	Erro Absoluto	Erro Relativo (%)
0.001	9875	9893	9905	9899	-0.01	0.01	726.72	-0.01	0.01	726.72	-0.02	0.02	1666.80	-0.10	0.10	10127.54
1.019	10476	10514	10533	10634	0.94	0.08	8.21	0.97	0.05	5.13	0.97	0.05	4.98	1.05	-0.03	3.17
2.020	11113	11159	11174	11273	1.93	0.09	4.29	1.98	0.04	2.11	1.97	0.05	2.35	2.05	-0.03	1.61
3.019	11754	11806	11827	11913	2.94	0.08	2.69	2.99	0.03	0.93	3.00	0.02	0.77	3.06	-0.04	1.20
4.018	12404	12460	12481	12564	3.96	0.06	1.54	4.02	0.00	0.06	4.02	0.00	0.06	4.08	-0.06	1.42
5.017	13055	13114	13138	13208	4.98	0.04	0.81	5.04	-0.02	0.47	5.05	-0.03	0.65	5.08	-0.07	1.34
6.017	13705	13770	13793	13854	5.99	0.02	0.37	6.07	-0.05	0.85	6.08	-0.06	0.98	6.10	-0.08	1.32
7.017	14353	14422	14450	14500	7.01	0.01	0.10	7.09	-0.07	1.04	7.11	-0.09	1.26	7.11	-0.09	1.31
8.015	14993	15070	15097	15143	8.01	0.00	0.03	8.11	-0.09	1.12	8.12	-0.10	1.30	8.12	-0.10	1.26
9.016	15630	15709	15741	15784	9.01	0.01	0.06	9.11	-0.09	1.00	9.13	-0.11	1.24	9.12	-0.10	1.16
10.016	16268	16354	16386	16432	10.01	0.01	0.06	10.12	-0.10	1.01	10.14	-0.12	1.23	10.14	-0.12	1.19
11.016	16915	17007	17042	17079	11.02	-0.01	0.07	11.14	-0.12	1.13	11.17	-0.15	1.37	11.15	-0.13	1.21
12.017	17562	17660	17694	17726	12.04	-0.02	0.17	12.16	-0.15	1.22	12.19	-0.17	1.42	12.16	-0.15	1.22
13.017	18209	18310	18346	18372	13.05	-0.03	0.26	13.18	-0.16	1.26	13.21	-0.19	1.48	13.18	-0.16	1.22
14.018	18858	18967	19002	19023	14.07	-0.05	0.36	14.21	-0.19	1.38	14.24	-0.22	1.57	14.20	-0.18	1.26
15.017	19503	19617	19657	19669	15.08	-0.06	0.41	15.23	-0.21	1.41	15.26	-0.25	1.64	15.21	-0.19	1.27
16.018	20148	20270	20310	20317	16.09	-0.07	0.45	16.25	-0.23	1.46	16.29	-0.27	1.68	16.22	-0.20	1.28
17.018	20798	20922	20964	20965	17.11	-0.09	0.53	17.27	-0.26	1.50	17.31	-0.29	1.73	17.24	-0.22	1.29
18.017	21444	21574	21616	21612	18.12	-0.10	0.57	18.30	-0.28	1.55	18.33	-0.32	1.75	18.25	-0.23	1.30
19.018	22091	22227	22270	22258	19.13	-0.12	0.61	19.32	-0.30	1.58	19.36	-0.34	1.79	19.26	-0.25	1.29
20.018	22738	22880	22922	22907	20.15	-0.13	0.65	20.34	-0.32	1.62	20.38	-0.36	1.81	20.28	-0.26	1.31
21.019	23382	23533	23577	23556	21.16	-0.14	0.65	21.36	-0.35	1.65	21.41	-0.39	1.84	21.30	-0.28	1.33
22.018	24031	24183	24230	24204	22.17	-0.16	0.71	22.38	-0.37	1.66	22.43	-0.41	1.87	22.31	-0.29	1.34
23.018	24678	24837	24888	24850	23.19	-0.17	0.73	23.41	-0.39	1.69	23.46	-0.44	1.92	23.33	-0.31	1.33

Tabela 5.2 - Valores obtidos em módulo de leitura de 4 correntes

A tabela permite concluir que cada transdutor tem um comportamento algo distinto. Embora para a corrente nominal, todos estejam dentro da tolerância de 1,35%, noutros valores podemos ter medições com erros acima deste valor. Podemos verificar que o erro absoluto varia de uma forma aproximadamente linear, mas que quanto maior a corrente, mais este se afasta do valor real. Do mesmo modo, para correntes abaixo de 3 Amperes o erro relativo é superior. Estes comportamentos individuais exigem uma calibração rigorosa, para minimizar os erros de leitura. Com mais medições, poderemos melhorar a fórmula de cálculo, aplicando uma regressão linear aos dados obtidos e melhorar assim a precisão das leituras.

5.1.4 ENSAIO DOS MÓDULOS DE MEDIÇÃO COM LV 25-P

Para ensaiar os módulos de leitura de tensão contínua, utilizou-se uma fonte de tensão regulável e efetuaram-se leituras em múltiplos de 50 Volt, partindo do valor zero até atingir a tensão seguinte à nominal (e.x.: para uma tensão nominal de 273, medimos até ao múltiplo de 50 seguinte, ou seja 300 Volt). A Figura 5.10 mostra a bancada de testes utilizada. Foram repetidas 2 medições, para tentar verificar possíveis variações. Novamente, foram recolhidos os valores em bruto do ADC, de forma a confirmar os resultados das fórmulas (33), (34) e (35). Cada módulo (DC_1V, DC_1V_1i e DC_1V_3i) foi testado individualmente e respeitando os valores nominais de funcionamento. Foram também mantidos num valor superior à tensão nominal, durante 15 minutos, para verificar o aquecimento de cada um. A temperatura não foi medida nestes testes iniciais, mas espera-se que em casos de instalação no interior de caixas, se possa efetuar esta medição, de forma a garantir que não há aquecimento em demasia. Fica em aberto a opção de ventilar a caixa, se necessário.

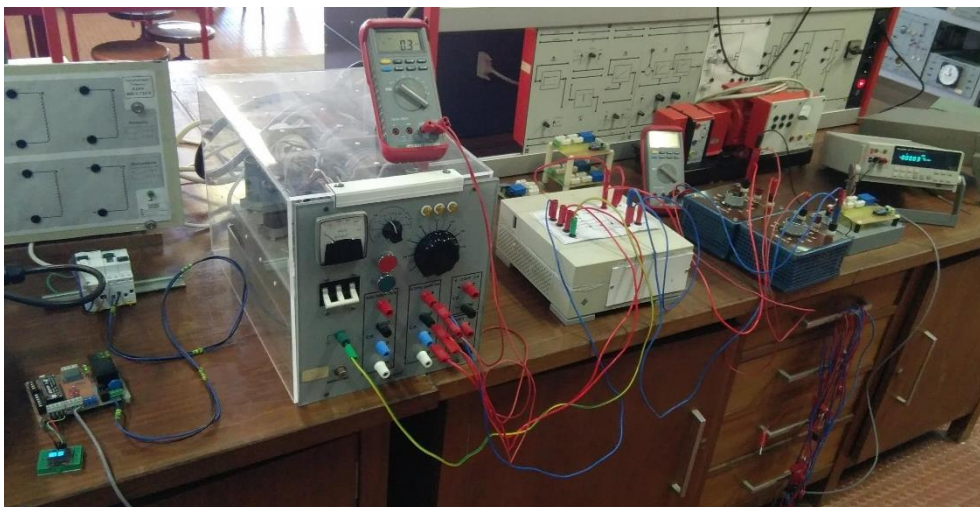


Figura 5.10 - Bancada de ensaio para os módulos de tensão

As fórmulas utilizadas baseiam-se nos valores teóricos de resistência no primário. Devido às variações de fabrico, o circuito real tem alguma variação desta resistência, conduzindo ao cálculo de valores errados. Assim, para corrigir estes valores foram medidas as resistências primárias reais no circuito, e aplicadas às fórmulas (33), (34) e (35), verificando se os erros diminuam. A resistência do secundário foi mantida nos 118 ohm.

As tabelas no Apêndice V mostram todos os valores obtidos, para as resistências teóricas e reais. Para além dos valores teóricos esperados de tensão na resistência de carga (V_{Rs}) e valor no ADC, temos a tensão aplicada à entrada (V_p), a corrente primária (I_p), o valor lido pelo ADC no secundário e respetiva conversão em tensão primária (V_{p_conv}) e os erros absoluto e relativo de cada medida individual.

A tabela 5.3 mostra o resultado dos valores médios das medidas individuais de erro absoluto e relativo. Esta média foi calculada em cada uma das duas medições efetuadas, para cada valor múltiplo de 50 V aplicado na entrada, excetuando o valor 0. Foi calculada a variância de cada um dos erros para perceber a dispersão dos mesmos. A tabela mostra também os valores da resistência primária teórica e real, para cada ponto de medida.

Ponto de medição	Rp (ohm)		Medição 1				Medição 2			
			Erro Absoluto Médio (V)	Variância Erro Absoluto	Erro Relativo Médio (%)	Variância Erro Relativo	Erro Absoluto Médio (V)	Variância Erro Absoluto	Erro Relativo Médio (%)	Variância Erro Relativo
DC BUS	Teórico	40228	-1,91	0,70	0,88	0,26	-2,09	1,06	0,88	0,15
	Real	39903	0,38	1,55	0,41	0,08	0,19	1,46	0,32	0,04
PV 1	Teórico	32228	7,03	9,65	3,09	0,07	7,17	10,19	3,15	0,10
	Real	33140	0,55	0,01	0,34	0,07	0,70	0,02	0,41	0,11
PV 2	Teórico	32228	5,81	8,06	2,46	0,01	5,76	7,95	2,44	0,00
	Real	32990	0,37	0,10	0,15	0,00	0,32	0,08	0,13	0,00
PV 3	Teórico	32228	4,23	4,84	1,73	0,02	3,32	4,61	1,26	0,14
	Real	32786	0,21	0,12	0,12	0,01	-0,71	0,07	0,55	0,39
Li-Ion	Teórico	27428	-1,52	0,36	0,88	0,04	-1,32	0,46	0,65	0,03
	Real	27157	0,37	0,20	0,19	0,02	0,56	0,05	0,35	0,03

Tabela 5.3 - Resultados dos ensaios dos módulos de medição de tensão.

Podemos verificar que a utilização do valor real da resistência no primário (R_p), conduz a um erro absoluto inferior em todos os casos, tornando o valor convertido mais preciso e próximo do valor real na entrada do circuito. Em todos os casos reais, o erro médio absoluto (na tabela manteve-se o sinal, de forma a perceber se a variação era positiva ou negativa) foi inferior à unidade, bastante abaixo dos valores obtidos a partir de R_p teórico. Ao mesmo tempo, a variância deste, também foi inferior em praticamente todos os casos, revelando uma maior exatidão das leituras. Apenas no caso do DC_BUS, a variância é superior. O erro

relativo foi também, em todos os casos reais, bastante inferior comparativamente ao caso teórico.

Com esta análise, comprova-se que o valor de R_p real deverá ser o escolhido para a conversão no microcontrolador, uma vez que os valores são bastante mais próximos dos reais.

5.2 SETUP PARA IMPLEMENTAÇÃO INICIAL NAS INSTALAÇÕES INTA

Durante uma visita à INTA para análise da instalação, verificou-se que alguns dos módulos desenhados não seriam de fácil colocação física no local. Como tal, decidiu-se efetuar algumas alterações no formato dos mesmos de forma a facilitar a montagem, na micro rede piloto da INTA. Uma das alterações mais significativas foi a separação dos transdutores de corrente do módulo de leituras. As ligações passaram a ser feitas por cabos terminados com conectores JST-XH de 4 pinos. Ao mesmo tempo, verificou-se que alguns dos pontos de medida estariam fisicamente noutra local ou inativos, levando a alterações no agrupamento de leituras. A Figura 5.11 apresenta todos os componentes criados, já agrupados e instalados em caixas plásticas, prontos para a instalação na micro rede. Na imagem representam-se os transdutores de corrente isolados e ligados por cabo, assim como os cabos entre os módulos emissores e os módulos de medição (Lopy Power Board 4, 5 e 6). Os módulos emissores Lopy Power Board 1, 2 e 3 serão ligados aos equipamentos de medição Modbus, presentes na imagem junto ao emissor 1. Na imagem todos os módulos estão ligados, e a enviar dados utilizando LoRaWAN. Este conjunto de equipamentos esteve ligado durante vários dias, para verificar o seu correto funcionamento antes da efetiva instalação na INTA. As configurações utilizadas no NS e AS encontram-se descritas no Apêndice U.

Na Figura 5.12 (lado esquerdo) podemos ver um pormenor da instalação no armário da micro rede INTA, com a colocação de alguns dos transdutores de corrente (a tracejado amarelo) e da caixa plástica que junta todos os módulos de leitura. Na mesma imagem, do lado esquerdo, está um módulo emissor ligado a um medidor via Modbus. Estas alterações não aparecem no corpo da tese, pois foram efetuadas mais recentemente, no entanto, os dimensionamentos descritos mantiveram-se iguais. Devido as estas alterações, não foram também feitos testes a alguns dos componentes aqui descritos (DC_1V_1i e DC_1V_3i), uma vez que sofreram modificações.

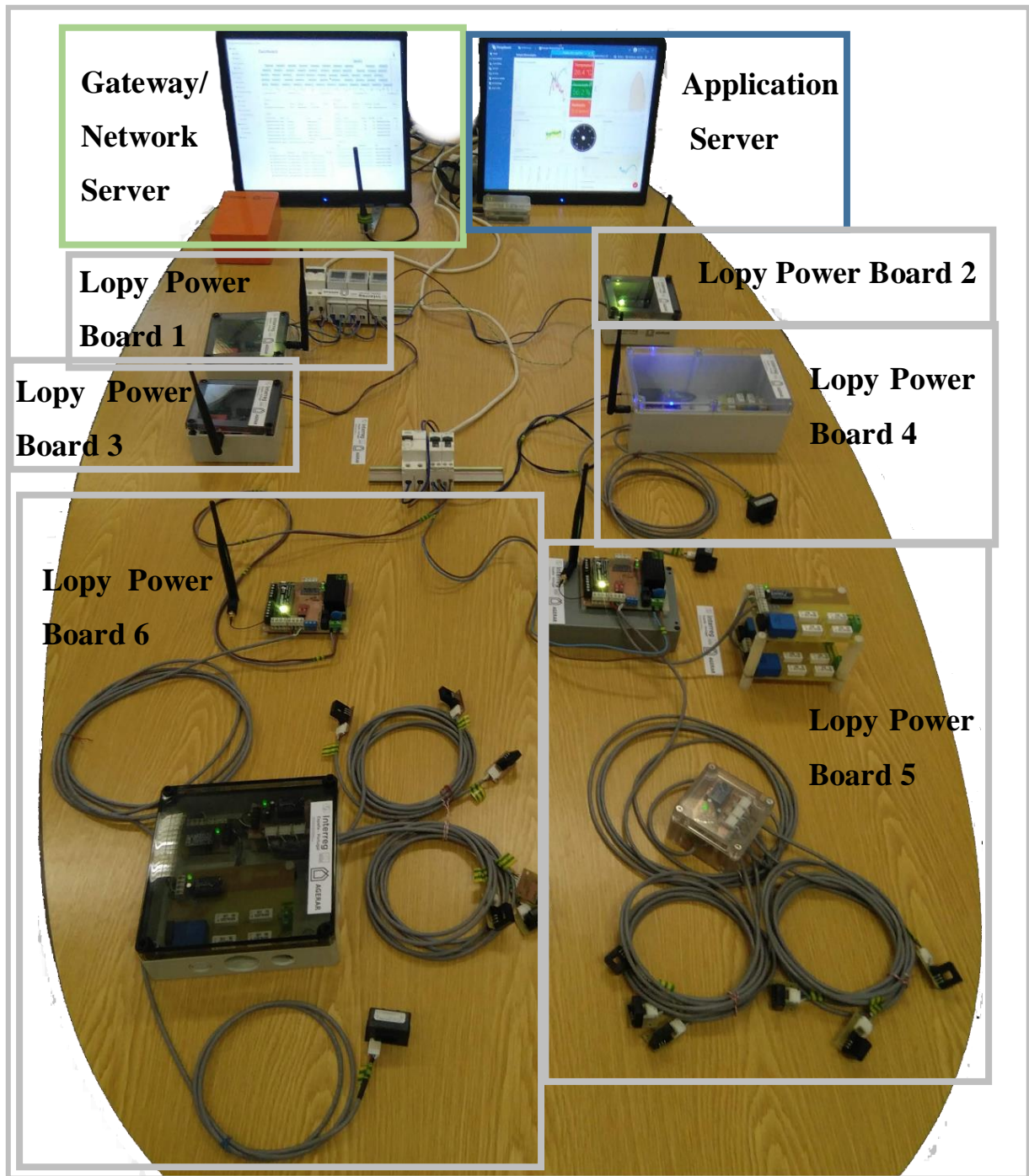


Figura 5.11 - Todos os componentes do sistema, pronto a implementar na microrrede



Figura 5.12 - Pormenor da instalação na micro rede INTA. Transdutores de corrente a tracejado amarelo e caixa com todos os conversores ADC (A). Módulo emissor Lopy Power Board, por cima do quadro elétrico, para conectar via Modbus a medidores de energia no interior do quadro elétrico (B)

6. CONCLUSÕES E TRABALHO FUTURO

6.1 CONCLUSÕES

Ao longo desta tese foi desenvolvido um conjunto de equipamentos que permite a monitorização e controlo de um sistema de energia, aplicável em *smart grids* mas com possibilidade de integrar outras monitorizações distintas, como no caso, os dados ambientais. O trabalho efetuado permitiu adquirir conhecimentos sobre a tecnologia LPWAN LoraWAN, e nas particularidades de implementar uma rede de sensores sem fios utilizando esta tecnologia.

Em relação às características inicialmente pretendidas no sistema, conseguimos cumprir com os pressupostos base:

- Todos os componentes utilizados são *open source*, o que nos permite alterar o que for necessário para os nossos objetivos.
- A rede implementada é totalmente privada, acessível apenas a utilizadores credenciados, assim como é independente de operadores externos. Apesar do inconveniente de recair sobre o utilizador toda a configuração do sistema, os custos envolvidos na manutenção e utilização da rede são inferiores. As comunicações são encriptadas, garantindo a segurança necessária.
- A nível de custos dos equipamentos criados, estes ficaram abaixo dos 95 euros. Apesar de não ter as certificações e segurança de um produto comercial, a nossa implementação tem a possibilidade de acrescentar funcionalidades aos módulos emissores, tornando-a uma solução flexível e abrangente. Estes, foram desenhados de forma a deixar os pinos do microcontrolador disponíveis, permitindo assim ligar outros dispositivos externos, não incluídos neste trabalho. É possível reduzir o preço de alguns módulos, escolhendo corretamente os componentes e fornecedores dos mesmos. Pode também abdicar-se de algumas funcionalidades de segurança como o isolamento galvânico ou a proteção contra curto circuitos nas fontes de alimentação. O material LoraWAN tem também vindo a descer de preço, assim como têm surgido novas soluções no mercado.
- Os componentes foram desenhados com o objetivo de serem modulares, podendo agrupar os necessários para efetuar as medições do local de instalação. Desta forma,

cumprimos a premissa do sistema ser modular, permitindo a sua instalação em diferentes redes elétricas.

- A escalabilidade da rede é garantida pelo LoraWAN, sendo que a rede permite acrescentar mais dispositivos, apenas configurando o NS. No nosso caso, apenas 7 deles estão ativos simultaneamente, o que não é problemático embora possam ocorrer colisões esporadicamente. No entanto, se houver necessidade de aumentar a rede, será sempre possível, bastando acrescentar mais *gateways*.
- Nesta fase inicial e com o material utilizado, o fator alcance ainda não é importante. Conseguimos cobertura em todos os pontos da instalação, uma vez que todos estes se encontram num raio de 50 metros, em redor da antena. Os testes efetuados mostram que num raio de 250 metros não se perdiam transmissões, mesmo na montagem de teste que estava bastante mal otimizada. Posteriormente, com a introdução das medições nos veículos, deveremos verificar se superamos o alcance de 670 metros, que foi obtido no ensaio efetuado na INTA. Na instalação que decorre, ir-se-á colocar a antena elevada e exterior ao edifício, esperando por isso um alcance melhorado.
- Garantimos também isolamento galvânico das redes primárias, nos sensores utilizados, uma mais valia de segurança.

Os resultados obtidos nos ensaios de perda de pacotes em ambiente urbano, mostram que o nosso *setup* apresenta uma percentagem considerável de perdas de pacotes. A instalação dos nós a 400 metros, revela uma percentagem de perda de pacotes em DR 5 muito elevada, na ordem dos 45,2 %. Já nos DR mais baixos, 2 e 0, os resultados são claramente melhores, de 5,16 % e 2,37 % respetivamente. A utilização de ADR apresenta um resultado de 24,49 %. Estes valores elevados podem revelar que o *setup* não é o mais indicado ou que existe demasiada interferência nos canais. Ao mesmo tempo, em relação aos ensaios efetuados nas instalações da Inta, verificamos uma grande diferença. Nestas instalações existem poucos edifícios, não afetando assim substancialmente a transmissão de dados, contrariamente ao que acontece no ensaio urbano. Para melhorar a receção, uma das soluções pode passar pela utilização de uma antena na *gateway* de ganho superior, de forma a tentar aumentar o LSNR. Apesar de este teste não corresponder à aplicação real, mostra o funcionamento *best effort* da tecnologia.

Os resultados mostram também que importa considerar a existência de pacotes perdidos. Mesmo no melhor cenário (entre Laboratórios do ISE), verificou-se uma perda de 0,5 % dos pacotes. Conclui-se, portanto, que dificilmente se obterá uma fiabilidade a 100 % nesta tecnologia, sendo por isso necessário considerar esta situação e garantir que a perda ocasional

de dados não interfira no funcionamento do sistema, nem seja crítica. A utilização dos mecanismos de *acknowledgement* poderá melhorar estes valores, mas prejudica a nível de *duty cycle*, devido às transmissões de confirmação adicionais que impõe. Uma possível solução será implementar mecanismos de correção de erros na aplicação, sendo desta forma possível recuperar alguma informação perdida. Em cada pacote enviado, seriam incluídas informações adicionais que permitam recuperar alguns pacotes perdidos, utilizando os últimos pacotes recebidos. Obviamente, irá existir um *overhead* adicional assim como um atraso na disponibilidade dos dados, uma vez que serão necessários guardar alguns pacotes anteriores para descodificar os dados. No entanto esta solução não irá, para já, ser explorada.

Os ensaios efetuados aos módulos de medição de corrente permitiram verificar que, para a corrente nominal, o erro relativo se mantém abaixo do valor do fabricante (1,35 %). O erro absoluto máximo atingiu 0,44 A. Verificou-se também que cada transdutor, apresenta variações distintas na linearidade dos resultados. Assim, conclui-se que existe a necessidade de uma calibração precisa, efetuada de forma individual para cada transdutor, de forma a minimizar estes erros.

Os módulos de medição de tensão apresentaram erros relativos baixos, após efetuar a calibração dos mesmos, sendo que o valor médio ficou abaixo de 0,55 %. Este valor é inferior ao definido pelo fabricante, de 0,8 %. O erro absoluto máximo foi de 3,17 V, no entanto, os valores médios deste erro mantiveram-se abaixo de 0,71 V. Podemos concluir que estes sensores, apresentam o comportamento esperado.

Em relação ao controlo, no decorrer do projeto AGERAR pretende-se que este mesmo sistema suporte o envio de mensagens que resultem de algoritmos de escalonamento informando os nós, e os equipamentos associados, da hora de entrada e de saída de funcionamento.

6.2 TRABALHO FUTURO

Uma vez que o trabalho desenvolvido nesta tese está inserido num projeto em curso, todo o sistema está em processo de finalização de instalação na micro rede experimental INTA. Após finalizar esta instalação, serão recolhidos e analisados dados de forma a verificar a fiabilidade das leituras e efetuar correções necessárias para melhorar a precisão das mesmas. Esta deverá ser feita por comparação com valores obtidos no sistema SCADA da INTA.

Iremos também avaliar o funcionamento e comportamento do sistema a médio-longo prazo, assim como integrá-lo nos componentes de outros parceiros do projeto.

Iremos também avançar com a parte de atuação remota, criando o servidor de escalonamentos e explorar as funcionalidades de alertas da plataforma Thingsboard.

Posteriormente, iremos proceder à criação de módulos de monitorização para aplicação nos veículos elétricos da INTA.

A nível de módulos, estes deverão ser refeitos com novas PCBs contruídas com métodos de fotolitografia, tornando o produto com um aspeto profissional.

6.3 PUBLICAÇÕES

No decorrer do desenvolvimento deste trabalho, efetuaram-se as seguintes publicações:

- **N. Pinto, D. Cruz, J. Monteiro, C. Cabrita, J. Semião, P. Cardoso, L. Oliveira e J. Rodrigues**, “IoE based Control and Monitoring of Electrical Grids: a Smart Grid’s perspective”, Book chapter accepted in **Harnessing the Internet of Everything (IoE) for Accelerated Innovation Opportunities**. IGI Global.
- **P. J. S. Cardoso, J. Monteiro, N. Pinto, D. Cruz, J. M. F. Rodrigues**, “Application of Machine Learning Algorithms to the IoE”, Book chapter accepted in **Harnessing the Internet of Everything (IoE) for Accelerated Innovation Opportunities**. IGI Global.
- **D. Cruz, N. Pinto, J. Monteiro, P. Cardoso, C. Cabrita, J. Semião, L. Oliveira e J. Rodrigues**, “Development of an Energy Management System for the Charge Scheduling of Plug-in Electric Vehicles”, in **Universal Access in Human-Computer Interaction. Virtual, Augmented, and Intelligent Environments**, pp. 214-225. Springer, 2018. DOI: 10.1007/978-3-319-92052-8_17

BIBLIOGRAFIA E REFERÊNCIAS

- [1] S. Borlase, *Smart Grids: Advanced Technologies and Solutions*. CRC Press, 2017.
- [2] M. La Scala, S. Bruno, C. A. Nucci, S. Lamonaca, and U. Stecchi, *From Smart Grids to Smart Cities: New Challenges in Optimizing Energy Grids*. John Wiley & Sons, 2017.
- [3] Associação empresarial Aiminho, “Guia de orientação para a utilização das energias renováveis”. [Online]. Disponível em: http://www.aiminho.pt/imgAll/file/Sustentar/Guia_de_Orientacao_para_a_utilizacao_da_s_Energias_Renovaveis.pdf [Acedido no dia 25/09/2017].
- [4] Associação de energias renováveis APREN. [Online]. Disponível em: <http://www.apren.pt/pt/dadostecnicos/index.php?id=1147&cat=266> [Acedido no dia 25/09/2017].
- [5] P. A. da Silva Costa, “Atlas do potencial eólico para Portugal continental,” *Faculdade de Ciências da Universidade de Lisboa, Lisboa*, 2004.
- [6] REN, “Produção renovável suficiente para abastecer o consumo de eletricidade em Portugal durante 63 horas”. [Online]. Disponível em: https://www.ren.pt/pt-PT/media/comunicados/detalhe/producao_renovavel_suficiente_para_abastecer_o_consumo_de_eletricidade_em_portugal_durante_63_horas_2/ [Acedido no dia 30/03/2018].
- [7] C. Wang, J. Wu, J. Ekanayake, and N. Jenkins, *Smart Electricity Distribution Networks*. CRC Press, 2017.
- [8] A. Sendin, M. A. Sanchez-Fornie, I. Berganza, J. Simon, and I. Urrutia, *Telecommunication networks for the smart grid*. Artech House, 2016.
- [9] B. L. R. Stojkoska and K. V. Trivodaliev, “A review of Internet of Things for smart home: Challenges and solutions,” *Journal of Cleaner Production*, vol. 140, pp. 1454–1464, 2017.
- [10] Y. Saleem, N. Crespi, M. H. Rehmani, and R. Copeland, “Internet of things-aided smart grid: Technologies, architectures, applications, prototypes, and future research directions,” *arXiv preprint arXiv:1704.08977*, 2017.
- [11] H. M. A. Fahmy, *Wireless Sensor Networks: Concepts, Applications, Experimentation and Analysis*. Springer, 2016.
- [12] Gilchrist, *Industry 4.0: the industrial internet of things*. Apress, 2016.
- [13] Ali, G. A. Shah, M. O. Farooq, and U. Ghani, “Technologies and challenges in developing Machine-to-Machine applications: A survey,” *Journal of Network and Computer Applications*, 2017.
- [14] Nordic Semi, J. Leonard “A brief guide to Bluetooth 5”, 2017. [Online]. Disponível em: <https://www.nordicsemi.com/eng/News/ULP-Wireless-Update/A-brief-guide-to-Bluetooth-5> [Acedido no dia 11/10/2017].
- [15] Nordic Semi, “Taking a deeper dive into Bluetooth 5”, 2017. [Online]. Disponível em: <https://devzone.nordicsemi.com/blogs/1093/> [Acedido a 10/10/2017].
- [16] TI blog, “How does Bluetooth® 5 increase the achievable range of a Bluetooth low energy connection?”, 2017. [Online]. Disponível em: http://e2e.ti.com/blogs_/b/connecting_wirelessly/archive/2017/01/30/how-does-bluetooth-5-increase-the-achievable-range-of-a-bluetooth-low-energy-connection [Acedido no dia 10/10/2017].
- [17] Bluetooth SIG, “Bluetooth core specification V4”. [Online]. Disponível em: <https://www.bluetooth.com/specifications> [Acedido no dia 10/10/2017].
- [18] M. Tailor, “What’s New with Bluetooth 5.0?”, 2017. Disponível em <http://www.summitdata.com/blog/whats-new-with-bluetooth-5-0/> [Acedido no dia 10/10/2017].

- [19] Bluetooth SIG documentation, 2017. [Online]. Disponível em: <https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works/le-mesh> [Acedido no dia 11/10/2017].
- [20] Ericsson, “Bluetooth mesh networking” whitepaper, 2017. [Online]. Disponível em: <https://www.ericsson.com/en/publications/white-papers/bluetooth-mesh-networking> [Acedido no dia 11/10/2017].
- [21] U-blox “Bluetooth technology” website, 2017. [Online]. Disponível em: <https://www.u-blox.com/en/bluetooth-technology> [Acedido no dia 11/10/2017].
- [22] S. M. Darroudi and C. Gomez, “Bluetooth Low Energy Mesh Networks: A Survey,” *Sensors*, vol. 17, no. 7, p. 1467, 2017.
- [23] Dynastream, “Ant Blaze Product Brief”, 2017. [Online]. Disponível em: https://www.dynastream.com/assets/D52/Dynastream.ANT_BLAZE.Product.Brief.pdf [Acedido no dia 12/10/2017].
- [24] Mahmood, N. Javaid, and S. Razzaq, “A review of wireless communications for smart grid,” *Renewable and sustainable energy reviews*, vol. 41, pp. 248–260, 2015.
- [25] Zigbee Alliance “Webinar presentation: Introducing ZigBee 3.0”, 2014. [Online]. Disponível em: <http://www.zigbee.org/download/introducing-zigbee-3-0-webinar-presentation> [Acedido no dia 12/10/2017].
- [26] T. Lennvall, S. Svensson, and F. Hekland, “A comparison of WirelessHART and ZigBee for industrial applications,” in *Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on, 2008*, pp. 85–88.
- [27] A. Koubâa, M. Alves, and E. Tovar, “IEEE 802.15. 4: a federating communication protocol for time-sensitive wireless sensor networks,” *Sensor Networks and Configurations: Fundamentals, Techniques, Platforms, and Experiments*, pp. 19–49, 2006.
- [28] Phoenix Contact, “Getting the Most Out of Your WirelessHART® System” whitepaper, 2015. [Online]. Disponível em: https://www.phoenixcontact.com/assets/downloads_ed/global/web_dwl_promotion/EN_Whitepaper_IE_WirelessHART_LoRes.pdf [Acedido no dia 13/10/2017].
- [29] R. Zurawski, “Industrial communication technology handbook”. CRC Press, 2014.
- [30] E. Araújo, “O protocolo WirelessHART (Parte1)”. [Online]. Disponível em: <https://www.automacaoindustrial.info/o-protocolo-wirelesshart-parte-1/> [Acedido no dia 12/10/2017].
- [31] Siemens, “WirelessHART@Siemens” whitepaper, 2010. [Online]. Disponível em: https://w5.siemens.com/web/at/de/industry/ia_dt/aktuelles/veranstaltungen/wireless-days2011/Documents/02_WirelessHART_ProductPresentation_en.pdf [Acedido no dia 12/10/2017].
- [32] EnOcean Alliance “Introducing the EnOcean Eco-System”, 2016. [Online]. Disponível em: https://www.enocean-alliance.org/wp-content/uploads/2016/11/Whitepaper_Introducing_the_EnOcean_Ecosystem.pdf [Acedido no dia 13/10/2017].
- [33] EnOcean “Radio Protocol 2”. [Online]. Disponível em: <https://www.enocean.com/erp2/> [Acedido no dia 13/10/2017].
- [34] EnOcean “Radio Protocol 1”. [Online]. Disponível em: <https://www.enocean.com/erp1/> [Acedido no dia 13/10/2017].
- [35] EnOcean “868 MHz EnOcean for Europe”. [Online]. Disponível em: https://www.enocean.com/en/enocean_modules/ [Acedido no dia 13/10/2017].
- [36] P. Jain and S. Taneeru, “Performance Evaluation of IEEE 802.11 ah Protocol in Wireless Area Network,” in *Micro-Electronics and Telecommunication Engineering (ICMETE), 2016 International Conference on, 2016*, pp. 578–583.

- [37] V. Baños-Gonzalez, M. S. Afaqui, E. Lopez-Aguilera, and E. Garcia-Villegas, “IEEE 802.11 ah: a technology to face the IoT challenge,” *Sensors*, vol. 16, no. 11, p. 1960, 2016.
- [38] M. Zemedede, “Explosion of the Internet of Things: What does it mean for wireless devices”, Keysight, 2015. [Online]. Disponível em: http://www.keysight.com/upload/cmc_upload/All/IoT_Seminar_Session1_Explosion_of_the_Internet_of_Things.pdf [Acedido no dia 18/10/2017].
- [39] Domazetovi'c, E. Kovcan, and A. Mihovska, “Performance evaluation of IEEE 802.11 ah systems,” in *Telecommunications Forum (TELFOR)*, 2016 24th, 2016, pp. 1–4.
- [40] W. Sun, M. Choi, and S. Choi, “IEEE 802.11 ah: A long range 802.11 WLAN at sub 1 GHz,” *Journal of ICT Standardization*, vol. 1, no. 1, pp. 83–108, 2013.
- [41] M. Weyn, G. Ergeerts, R. Berkvens, B. Wojciechowski, and Y. Tabakov, “DASH7 alliance protocol 1.0: Low-power, mid-range sensor and actuator communication,” in *Standards for Communications and Networking (CSCN)*, 2015 IEEE Conference on, 2015, pp. 54–59.
- [42] Dash7 Alliance “Wireless Sensor and Actuator Network Protocol Version 1.1”, 2017. [Online]. Disponível em <http://www.dash7-alliance.org/product/d7ap1-1/> [Acedido no dia 13/10/2017].
- [43] Y. Tabakov “DASH7 Alliance Protocol”, DASH7 Alliance, 2014. [Online]. Disponível em <http://dash7-alliance.org/wp-content/uploads/2014/08/005-Dash7-Alliance-Mode-technical-presentation.pdf> [Acedido no dia 13/10/2017].
- [44] M. Weyn, G. Ergeerts, R. Berkvens, B. Wojciechowski, and Y. Tabakov, “DASH7 alliance protocol 1.0: Low-power, mid-range sensor and actuator communication,” in *Standards for Communications and Networking (CSCN)*, 2015 IEEE Conference on, 2015, pp. 54–59.
- [45] Nokia, “LTE Evolution for IoT connectivity”, 2017. [Online]. Disponível em <https://resources.ext.nokia.com/asset/200178> [Acedido no dia 09/10/2017].
- [46] 3GPP “Low Power Wide Area Technologies” whitepaper, 2016. [Online]. Disponível em <https://www.gsma.com/iot/3gpp-low-power-wide-area-technologies-white-paper/> [Acedido no dia 09/10/2017].
- [47] S. Dawaliby, A. Bradai, and Y. Pousset, “In depth performance evaluation of LTE-M for M2M communications,” in *The 12th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob'16*, 2016.
- [48] S. Tabbane, “IoT Network Planning” ITU, 2016. [Online]. Disponível em <https://www.itu.int/en/ITU-D/Regional-Presence/AsiaPacific/SiteAssets/Pages/Events/2016/Dec-2016-IoT/IoTtraining/IoT%20network%20planning%20ST%2015122016.pdf> [Acedido no dia 10/10/2017].
- [49] GSMA “LTE_M Deployment Guide”, 2017. [Online]. Disponível em <https://www.gsma.com/iot/wp-content/uploads/2017/09/LTE-M-Deployment-Guide-CLP.29-v1.0.pdf> [Acedido no dia 11/10/2017].
- [50] Sequans “Narrowband LTE: Which apps need Cat M1 and which need Cat NB1?”, 2016. [Online]. Disponível em <http://www.sequans.com/narrowband-lte-which-apps-need-cat-m1-and-which-need-cat-nb1> [Acedido no dia 09/10/2017].
- [51] GSMA “NB-IoT Deployment Guide”, 2017. [Online]. Disponível em <https://www.gsma.com/iot/wp-content/uploads/2017/08/CLP.28-v1.0.pdf> [Acedido a 11/10/2017].
- [52] D.Flore, “3GPP Standards for the Internet-of-Things”, 3GPP, 2016. [Online]. Disponível em http://www.3gpp.org/images/presentations/3GPP_Standards_for_IoT.pdf [Acedido a 10/10/2017].

- [53] Rohde&Schwarze, “Narrowband Internet of Things” whitepaper, 2016. [Online]. Disponível em https://www.rohde-schwarz.com/pt/applications/narrowband-internet-of-things-white-paper_230854-314242.html [Acedido a 11/10/2017].
- [54] Keysight, “NB-Iot Technical Fundamentals” whitepaper, 2017. [Online]. Disponível em http://www.keysight.com/upload/cmc_upload/All/20170612-A4-JianHuaWu-updated.pdf [Acedido a 16/10/2017]
- [55] Sequans “Narrowband LTE: Which apps need Cat M1 and which need Cat NB1?”, 2016. [Online]. Disponível em <http://www.sequans.com/narrowband-lte-which-apps-need-cat-m1-and-which-need-cat-nb1/> [Acedido a 09/10/2017]
- [56] ABI Research, “Best fit use Cases for LPWANs” whitepaper, 2016. [Online]. Disponível em <https://www.ingenu.com/portfolio/best-fit-use-cases-for-lpwans> [Acedido a 16/10/2017].
- [57] EverythingRF “What is EC-GSM-IoT?” website, 2017. [Online]. Disponível em <https://www.everythingrf.com/community/what-is-ec-gsm-iot> [Acedido a 15/10/2017]
- [58] R. Sanchez-Iborra and M.-D. Cano, “State of the art in LP-WAN solutions for industrial IoT services,” *Sensors*, vol. 16, no. 5, p. 708, 2016.
- [59] K. Al Agha, G. Pujolle, and T. A. Yahiya, *Mobile and Wireless Networks*. John Wiley & Sons, 2016.
- [60] R.Quinnell, “Low power wide-area networking alternatives for the IoT”, 2015. [Online]. Disponível em <https://www.edn.com/design/systems-design/4440343/Low-power-wide-area-networking-alternatives-for-the-IoT> [Acedido a 16/10/2017]
- [61] Sigfox, “Sigfox technical overview”, 2017. [Online]. Disponível em <https://www.disk91.com/wp-content/uploads/2017/05/4967675830228422064.pdf> [Acedido a 16/10/2017]
- [62] Disk91.com, “One day at Sigfox”, 2015. [Online]. Disponível em <https://www.disk91.com/2015/news/technologies/one-day-at-sigfox/> [Acedido a 16/10/2017]
- [63] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, “Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios,” *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, 2016.
- [64] É. Morin, M. Maman, R. Guizzetti, and A. Duda, “Comparison of the Device Lifetime in Wireless Networks for the Internet of Things,” *IEEE Access*, vol. 5, pp. 7097–7114, 2017.
- [65] Hernandez, G. Peralta, L. Manero, R. Gomez, J. Bilbao, and C. Zubia, “Energy and coverage study of LPWAN schemes for Industry 4.0,” in *Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)*, 2017 IEEE International Workshop of, 2017, pp. 1–6.
- [66] L. Krupka, L. Vojtech, and M. Neruda, “The issue of LPWAN technology coexistence in IoT environment,” in *Mechatronics-Mechatronika (ME)*, 2016 17th International Conference on, 2016, pp. 1–8.
- [67] W. Webb, *Understanding Weightless: Technology, Equipment, and Network Deployment for M2M Communications in White Space*. Cambridge University Press, 2012.
- [68] G. Margelis, R. Piechocki, D. Kaleshi, and P. Thomas, “Low throughput networks for the IoT: Lessons learned from industrial implementations,” in *Internet of Things (WF-IoT)*, 2015 IEEE 2nd World Forum on, 2015, pp. 181–186.
- [69] Ingenu, “How RPMA works” whitepaper. [Online]. Disponível em <https://www.ingenu.com/portfolio/how-rpma-works-white-paper/> [Acedido a 17/10/2017].
- [70] Ingenu, “Ingenu webinar: How RPMA works”. [Online]. Disponível em <https://www.youtube.com/watch?v=4beoZapuBXw> [Acedido a 17/10/2017].

- [71] Ingenu, “RPMA Development kit”. [Online]. Disponível em <https://www.ingenu.com/get-started/hardware/rpma-devkit/> [Acedido a 17/10/2017]
- [72] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, “A study of LoRa: Long range & low power networks for the internet of things,” *Sensors*, vol. 16, no. 9, p. 1466, 2016.
- [73] LoRa Alliance, “A technical overview of LoRa and LoRaWAN”. [Online]. Disponível em https://docs.wixstatic.com/ugd/eccc1a_ed71ea1cd969417493c74e4a13c55685.pdf [Acedido a 26/10/2017]
- [74] Microchip “LoRaWAN 101 Class” presentation. [Online]. Disponível em <http://www.spincraft.com/hackers/wp-content/uploads/2017/01/LoRaWAN-101-Class-v2-MARCOM-1.pdf> [Acedido a 19/10/2017].
- [75] Semtech Corporation, “Lora Modem Designer’s guide – AN1200.13 application note”. [Online]. Disponível em: <https://www.semtech.com/images/datasheet/LoraDesignGuide STD.pdf> [acedido a 19/10/2017].
- [76] Semtech Corporation, “LoRa Modem Low Energy Consumption Design – AN1200.17 application note”. [Online]. Disponível em: <http://www.semtech.com/images/datasheet/LoraLowEnergyDesign STD.pdf> [acedido a 19/10/2017]
- [77] Semtech Corporation, “Lora Modulation Basics – AN1200.22 application note”. [Online]. Disponível em: <http://www.semtech.com/images/datasheet/an1200.22.pdf> [acedido a 19/10/2017]
- [78] D. Magrin, M. Centenaro, and L. Vangelista, “Performance evaluation of LoRa networks in a smart city scenario,” in *Communications (ICC), 2017 IEEE International Conference on*, 2017, pp. 1–7.
- [79] Lora Alliance “What is it?” whitepaper. [Online]. Disponível em: https://docs.wixstatic.com/ugd/eccc1a_ed71ea1cd969417493c74e4a13c55685.pdf [Acedido a 19/10/2017].
- [80] TheThingsNetwork “WiKi LoRaWAN”. [Online]. Disponível em <https://www.thethingsnetwork.org/wiki/LoRaWAN/Home> [Acedido a 19/10/2017].
- [81] M. H. Cintuglu, O. A. Mohammed, K. Akkaya, and A. S. Uluagac, “A survey on smart grid cyber-physical system testbeds,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 446–464, 2017.
- [82] N. Andreadou, M. O. Guardiola, and G. Fulli, “Telecommunication technologies for smart grid projects with focus on smart metering applications,” *Energies*, vol. 9, no. 5, p. 375, 2016.
- [83] Semtech Corporation, “Low power long range transmitter”, Patente EU EP2763321 A1, 6 Ago 2014. [Online]. Disponível em <https://www.google.com/patents/EP2763321A1>
- [84] B. Sikken, “Decoding Lora”. [Online]. Disponível em: <https://revspace.nl/DecodingLora> [Acedido em 24/10/2017]
- [85] M. Knight, “Decoding the LoRa PHY”. [Online]. Disponível em <https://github.com/BastilleResearch/gr-lora/wiki/Decoding-the-LoRa-PHY> [Acedido a 24/10/2017]
- [86] A. Springer, W. Gugler, M. Huemer, L. Reindl, C. Ruppel, and R. Weigel, “Spread spectrum communications using chirp signals,” in *EUROCOMM 2000. Information Systems for Enhanced Public Safety and Security. IEEE/AFCEA, 2000*, pp. 166–170.
- [87] J. Petäjäjärvi, K. Mikhaylov, M. Pettissalo, J. Janhunen, and J. Iinatti, “Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage,” *International Journal of Distributed Sensor Networks*, vol. 13, no. 3, p. 1550147717699412, 2017.

- [88] Gaussianwaves, “Channel Capacity & Shannon’s theorem – demystified”. [Online]. Disponível em <https://www.gaussianwaves.com/2008/04/channel-capacity/> [Acedido em 24/10/2017]
- [89] R. E. Ziemer, “Fundamentals of spread spectrum modulation,” Synthesis Lectures on communications, vol. 2, no. 1, pp. 1–79, 2007.
- [90] R. Pickholtz, D. Schilling, and L. Milstein, “Theory of spread-spectrum communications- a tutorial,” IEEE transactions on Communications, vol. 30, no. 5, pp. 855–884, 1982.
- [91] L. Vangelista, “Frequency Shift Chirp Modulation: the LoRa Modulation” IEEE Signal Processing Letters, vol. PP, no. 99, pp. 1–1, 2017.
- [92] Semtech Corporation, “Signal concentrator device”, Patente US20160020932 A1, 21 Jan 2016. [Online]. Disponível em <https://www.google.com/patents/US20160020932>
- [93] Stackexchange, “Understanding the relationship between LoRa chips, chirps, symbols and bits”. [Online]. Disponível em <https://electronics.stackexchange.com/questions/278192/understanding-the-relationship-between-lora-chips-chirps-symbols-and-bits> [acedido a 24/10/2017]
- [94] CEPT ECC, “Recomendation 70-03” Disponível em <http://www.erodocdb.dk/Docs/doc98/official/pdf/REC7003E.PDF> [Acedido a 26/10/2017]
- [95] ETSI, “EN300.220-1 v3.1.0”. [Online]. Disponível em http://www.etsi.org/deliver/etsi_en/300200_300299/30022001/03.01.00_20/en_30022001v030100a.pdf [Acedido a 26/10/2017]
- [96] Semtech Corporation, “SX1272/73 Datasheet”. [Online]. Disponível em <https://www.semtech.com/uploads/documents/sx1272.pdf> [acedido a 19/10/2017]
- [97] Semtech Corporation, “LoRa Modem Calculator”. [Online]. Disponível em <http://www.semtech.com/apps/filedown/down.php?file=SX1272LoRaCalculatorSetup1%271.zip> [acedido a 19/10/2017]
- [98] TheThingNetwork, “LoRa(WAN) airtime calculator”. [Online]. Disponível em <https://docs.google.com/spreadsheets/d/1QvcKsGeTTPpr9icj4XkKXq4r2zTc2j0gsHLrnplzM3I/edit#gid=0> [Acedido a 26/10/2017]
- [99] P. Egli, “Overview of emerging technologies for low power wide area networks in Internet of Things (IoT) and M2M scenarios” [online] Disponível em <https://pt.scribd.com/document/258682723/Overview-of-emerging-technologies-for-low-power-wide-area-networks-in-Internet-of-Things-IoT-and-M2M-scenarios> [Acedido a 26/10/2017]
- [100] D. Magrin, “Network level performances of a LoRa system,”, dissertação de mestrado, Università degli Studi di Padova, Padova, 2016
- [101] LoRa Alliance, “LoRaWAN 101 A technical introduction whitepaper”. [Online]. Disponível em <https://www.lora-alliance.org/what-is-lora> [Acedido a 26/10/2017]
- [102] J. Marais, R. Malekian, and A. Abu-Mahfouz, “LoRa and LoRaWAN Testbeds: a Review,” in Africon 2017, 2017.
- [103] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, “Understanding the limits of LoRaWAN,” IEEE Communications Magazine, vol. 55, no. 9, pp. 34–40, 2017.
- [104] LoRa Alliance, “Lorawan specificationV1R0”. [Online]. Disponível em <https://www.rs-online.com/designspark/rel-assets/ds-assets/uploads/knowledge-items/application-notes-for-the-internet-of-things/LoRaWAN%20Specification%201R0.pdf> [Acedido a 26/10/2017]
- [105] N. Varsier and J. Schwoerer, “Capacity limits of LoRaWAN technology for smart metering applications” in Communications (ICC), 2017 IEEE International Conference on, 2017, pp. 1–6.

- [106] P. Ferrari, A. Flammini, M. Rizzi, E. Sisinni, and M. Gidlund, “On the evaluation of LoRaWAN virtual channels orthogonality for dense distributed systems” in Measurement and Networking (M&N), 2017 IEEE International Workshop on, 2017, pp. 1–6.
- [107] J. de Carvalho Silva, J. J. Rodrigues, A. M. Alberti, P. Solic, and A. L. Aquino, “LoRaWAN—A low power WAN protocol for Internet of Things: A review and opportunities” in Computer and Energy Science (SpliTech), 2017 2nd International Multidisciplinary Conference on, 2017, pp. 1–6.
- [108] LoRa Alliance, “A technical overview of LoRa and LoRaWAN”. [Online]. Disponível em https://docs.wixstatic.com/ugd/eccc1a_ed71ea1cd969417493c74e4a13c55685.pdf [Acedido a 26/10/2017]
- [109] D. Bankov, E. Khorov, and A. Lyakhov, “On the Limits of LoRaWAN Channel Access” in Engineering and Telecommunication (EnT), 2016 International Conference on, 2016, pp. 10–14.
- [110] R. Sanchez-Iborra and M.-D. Cano, “State of the art in LP-WAN solutions for industrial IoT services” Sensors, vol. 16, no. 5, p. 708, 2016.
- [111] L. Vangelista, A. Zanella, and M. Zorzi, “Long-Range IoT Technologies: The Dawn of LoRaTM” in Future Access Enablers for Ubiquitous and Intelligent Infrastructures, 2015, pp. 51–58.
- [112] F. V. den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, “Scalability analysis of large-scale LoRaWAN networks in ns-3” arXiv preprint arXiv:1705.05899, 2017.
- [113] N. Vatcharatiansakul, P. Tuwanut, and C. Pornavalai, “Experimental performance evaluation of LoRaWAN: A case study in Bangkok” in Computer Science and Software Engineering (JCSSE), 2017 14th International Joint Conference on, 2017, pp. 1–4.
- [114] LoRa Alliance, “Recommended SX1272 settings for EU868 LoRaWAN network operations, AN1200.23”. [Online]. Disponível em <https://www.semtech.com/uploads/documents/an1200.23.pdf> [Acedido a 26/10/2017]
- [115] LoRa Alliance, “LoRaWAN Specification v1.1”. [Online]. Disponível em <https://loralliance.org/resource-hub/lorawantm-specification-v11> [Acedido a 20/03/2018]
- [116] Semtech Lora-net “Basic communication protocol between Lora gateway and server”. [Online]. Disponível em https://github.com/Lora-net/packet_forwarder/blob/master/PROTOCOL.TXT [Acedido a 20/03/2018]
- [117] Semtech, “Gateway to server interface ANNWS.01.2.1.w.sys”. [Online]. Disponível em <https://www.thethingsnetwork.org/forum/uploads/default/original/1X/4fbda86583605f4aa24dcedaab874ca5a1572825.pdf> [Acedido a 20/03/2018]
- [118] “Introducing JSON”. [Online]. Disponível em <https://www.json.org/> [Acedido a 20/03/2018]
- [119] LoRa Alliance, “LoRaWAN Back-End Interfaces v1.0”. [Online]. Disponível em <https://loralliance.org/resource-hub/lorawantm-back-end-interfaces-v10> [Acedido a 20/03/2018]
- [120] IMST, “iC880A LoRaWAN Concentrator Datasheet”. [Online]. Disponível em https://wireless-solutions.de/downloads/Radio-Modules/iC880A/iC880A_Datasheet_V1_0.pdf [Acedido a 18/01/2018]
- [121] “I/Q Data for dummies”. [Online]. Disponível em <http://whiteboard.ping.se/SDR/IQ> [Acedido a 16/01/2018]
- [122] Sparkfun, “SPI Tutorial”. [Online]. Disponível em <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi> [Acedido a 16/01/2018]
- [123] Semtech, “SX1301 datasheet”. [Online]. Disponível em <https://www.semtech.com/uploads/documents/sx1301.pdf> [Acedido a 16/01/2018]

- [124] Semtech, Lora-net, “Lora gateway”. [Online]. Disponível em https://github.com/Lora-net/lora_gateway [Acedido a 16/01/2018]
- [125] Linx technologies, “ANT-868-ID datasheet”. [Online]. Disponível em <https://linxtechnologies.com/wp/wp-content/uploads/ant-868-id-llll-ccc.pdf> [Acedido a 21/04/2018]
- [126] Semtech, Lora-net, “Packet forwarder”. [Online]. Disponível em https://github.com/Lora-net/packet_forwarder [acedido a 16/01/2018]
- [127] MyDevicesIOT, “Cayenne Low Power Payload”. [Online]. Disponível em <https://github.com/myDevicesIoT/cayenne-docs/blob/master/docs/LORA.md> [Acedido a 14/02/2018]
- [128] IPSO Alliance, “IPSO Smart Objects”. [Online]. Disponível em <https://www.ipso-alliance.org/> [Acedido a 14/02/2018]
- [129] The Things Networks forum, “Best practices when sending GPS location data”. [Online]. Disponível em <https://www.thethingsnetwork.org/forum/t/best-practices-when-sending-gps-location-data/1242> [Acedido a 24/02/2018]
- [130] Eastron, “Electricity meter”. [Online]. Disponível em <http://www.eastrongroup.com/products/10.html> [Acedido a 1/03/2018]
- [131] Vigortronix, “5 Watt ACDC PCB Power Modules datasheet”. [Online]. Disponível em <http://www.vigortronix.com/5WattACDCPCBPowerModule.aspx> [Acedido a 5/03/2018]
- [132] Recom, “Powerline AC-DC Converter datasheet”. [Online]. Disponível em https://www.recom-power.com/pdf/Powerline-AC-DC/RAC10-S_DC_277.pdf [Acedido a 17/03/2018]
- [133] NXP, “I2C-bus specification and user manual”. [Online]. Disponível em <https://www.nxp.com/docs/en/user-guide/UM10204.pdf> [Acedido a 5/03/2018]
- [134] TI, “I2C infographic flyer”. [Online]. Disponível em <http://www.ti.com/lit/sg/slyt658/slyt658.pdf> [Acedido a 5/03/2018]
- [135] Sparkfun, “Level translator breakout”. [Online]. Disponível em <https://www.sparkfun.com/products/11955> [Acedido a 5/03/2018]
- [136] Eastron, “SDM230 Modbus protocol V1.2”. [Online]. Disponível em <http://www.flanesi.it/blog/download/sdm230modbus/SDM230-Modbus%20protocol%20V1.2.pdf> [Acedido a 5/03/2018]
- [137] Eastron, “SDM530CT-MT Modbus protocol V1.2”. [Online]. Disponível em http://www.eastrongroup.com/data/uploads/Eastron_SDM530CT-MT_protocol_V1_2.pdf [Acedido a 5/03/2018]
- [138] Microchip, “MCP1700 Low quiescent current LDO”. [Online]. Disponível em <https://www.microchip.com/wwwproducts/en/MCP1700> [Acedido a 5/03/2018]
- [139] Recom, “RD line DC/DC Converter datasheet”. [Online]. Disponível em <https://www.recom-power.com/pdf/Econoline/RD.pdf> [Acedido a 5/03/2018]
- [140] M. R. Senouci and A. Mellouk, *Deploying Wireless Sensor Networks: Theory and Practice*. Elsevier, 2016.
- [141] O. Çetinkaya and O. Akan, "Use of Wireless Sensor Networks in Smart Homes", *Emerging Communication Technologies Based on Wireless Sensor Networks*, pp.233-258, 2016.
- [142] M. H. Rehmani and A.-S. K. Pathan, *Emerging communication technologies based on wireless sensor networks: current research and future applications*. CRC Press, 2016.
- [143] A. Förster, *Introduction to Wireless Sensor Networks*. John Wiley & Sons, 2016.
- [144] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, “Wireless sensor networks: a survey on recent developments and potential synergies”, *The Journal of supercomputing*, vol. 68, no. 1, pp. 1–48, 2014.

- [145] J. Cao and X. Liu, "Wireless Sensor Networks for Structural Health Monitoring". Springer, 2016.
- [146] T. Hirata, K. Terada, M. Toyota, Y. Takada, K. Matsumoto, and M. S. Tanaka, "Proposal of a power saving network for rice fields using LoRa" in Consumer Electronics (GCCE), 2017 IEEE 6th Global Conference on, 2017, pp. 1–4.
- [147] R. Jedermann, U. Praeger, and W. Lang, "Challenges and opportunities in remote monitoring of perishable products", Food Packaging and Shelf Life, vol. 14, pp. 18–25, 2017.
- [148] Smart city Santander. [Online]. Disponível em: <http://www.smartsantander.eu/> [Acedido no dia 30/03/2018].
- [149] Smart city Lagoa. [Online]. Disponível em: <https://www.cm-lagoa.pt/index.php/pt/smartcity-lagoa> [Acedido no dia 30/03/2018].
- [150] ST Microelectronics, "STM32L0 Discovery kit LoRa, Sigfox, low-power wireless". [Online]. Disponível em: <http://www.st.com/en/evaluation-tools/b-l072z-lrwan1.html> [acedido a 22/01/2018]
- [151] Adafruit, "Adafruit feather 32u4 with Lora radio". [Online]. Disponível em <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-feather-32u4-radio-with-lora-radio-module.pdf> [Acedido a 22/01/2018]
- [152] Pycom, "Lopy development board". [Online]. Disponível em <https://pycom.io/product/lopy/> [Acedido a 22/01/2018]
- [153] Libelium, "Wasmote open source sensor node for IoT". [Online]. Disponível em <http://www.libelium.com/products/wasmote/> [Acedido a 22/01/2018]
- [154] Cooking Hacks, "Wasmote loRaWAN SMA 4.5 dBi - Europe". [Online]. Disponível em <https://www.cooking-hacks.com/wasmote-lorawan-sma-4-5-dbi-868-mhz> [Acedido a 22/01/2018]
- [155] Microchip, "LoRa(R) Technology Evaluation Kit - 800". [Online]. Disponível em <https://www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=dv164140-1> [Acedido a 22/01/2018]
- [156] Multitech, "MultiConnect® mDot™ Long Range LoRa® Module". [Online]. Disponível em <https://www.multitech.com/documents/publications/data-sheets/86002171.pdf> [Acedido a 22/01/2018]
- [157] Seeedstudio, "Seeeduino LoRaWAN". [Online]. Disponível em http://wiki.seeedstudio.com/Seeeduino_LoRAWAN/ [Acedido a 22/01/2018]
- [158] Sandbox Eletronics, "LoRaGo PORT – SX1301/SX1308 Based LoRaWAN Concentrator Module". [Online]. Disponível em <https://sandboxelectronics.com/?p=2669> [Acedido a 22/01/2018]
- [159] Dragino, "Lora/GPS HAT". [Online]. Disponível em http://wiki.dragino.com/index.php?title=Lora/GPS_HAT [Acedido a 22/01/2018]
- [160] Wireless Solutions, "iC880A - LoRaWAN® Concentrator 868MHz". [Online]. Disponível em <https://wireless-solutions.de/products/radiomodules/ic880a.html> [Acedido a 22/01/2018]
- [161] Laird Tech, "Sentrus™ RG1xx LoRa®-Enabled Gateway + Wi-Fi / Bluetooth / Ethernet". [Online]. Disponível em <https://www.lairdtech.com/products/rg1xx-lora-gateway> [Acedido a 23/01/2018]
- [162] Multitech, "MultiConnect® Conduit™ Programmable Gateway for the IfT(MTCDDT Series)" . [Online]. Disponível em <https://www.multitech.com/brands/multiconnect-conduit> [Acedido a 23/01/2018]
- [163] Kerlink, "Wirnet Station". [Online]. Disponível em <https://www.kerlink.com/product/wirnet-station/> [Acedido a 23/01/2018]

- [164] LoraServer, "LoRa Server, open-source LoRaWAN network-server". [Online]. Disponível em <https://www.loraserver.io/> [Acedido a 23/01/2018]
- [165] Lora Alliance, "LoRaWAN Regional Parameters v1.0.2revB". [Online]. Disponível em <https://lora-alliance.org/resource-hub/lorawantm-regional-parameters-v102rb> [Acedido a 03/05/2018]
- [166] Lorawan-server, "Lorawan-server, Compact server for private LoRa networks". [Online]. Disponível em <https://gotthardp.github.io/lorawan-server/> [Acedido a 23/01/2018]
- [167] Freeboard, "Dashboards for the IoT". [Online]. Disponível em <https://freeboard.io/> [Acedido a 01/02/2018]
- [168] Thingsboard, "Open-source IoT Platform". [Online]. Disponível em <https://thingsboard.io/> [Acedido a 01/02/2018]
- [169] Lorient, "LoRaWAN Services and Software". [Online]. Disponível em <https://www.loriot.io/> [Acedido a 01/02/2018]
- [170] ResIOT, "LoRaWAN Network Server and IoT Platform". [Online]. Disponível em <https://www.resiot.io/en/> [Acedido a 01/02/2018]
- [171] The Things Network, "Building a global internet of things network together". [Online]. Disponível em <https://www.thethingsnetwork.org/> [Acedido a 01/02/2018]
- [172] MQTT.ORG, "MQTT". [Online]. Disponível em <http://mqtt.org/> [Acedido a 21/03/2018]
- [173] Honeywell, "Hall Effect Sensing and Application". [Online]. Disponível em <https://sensing.honeywell.com/hallbook.pdf> [Acedido a 03/04/2018]
- [174] LEM, "Isolated current and voltage transducers. Characteristics - Applications - Calculations". [Online]. Disponível em <https://www.lem.com/zh/file/3139/download> [Acedido a 03/04/2018]
- [175] Tamura Corporation, "Open Loop Hall Effect Current Sensors". [Online]. Disponível em www.tamuracorp.com/uploads/currentsensor/OpenloopPDF.pdf [Acedido a 03/04/2018]
- [176] Tamura Corporation, "Closed loop Hall Effect Sensors". [Online]. Disponível em www.tamuracorp.com/uploads/currentsensor/ClosedloopPDF.pdf [Acedido a 03/04/2018]
- [177] S. A. Dyer, Wiley Survey of Instrumentation and Measurement. John Wiley & Sons, 2004.
- [178] Adafruit, "Adafruit 4-Channel ADC Breakouts". [Online]. Disponível em <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-4-channel-adc-breakouts.pdf> [Acedido a 09/04/2018]
- [179] TI, "Ads1115 datasheet". [Online]. Disponível em www.ti.com/lit/ds/symlink/ads1115.pdf [Acedido a 09/04/2018]
- [180] LEM, "HO 10-P datasheet". [Online]. Disponível em https://www.lem.com/sites/default/files/products_datasheets/ho-p_series.pdf [Acedido a 09/04/2018]
- [181] Tamura Corporation, "Hall Effect Current Sensors L31S***S05FS Series". [Online]. Disponível em <http://tamuracorp.com/clientuploads/pdfs/engineeringdocs/L31SXXXS05FS.pdf> [Acedido a 09/04/2018]
- [182] TI, "LM358 Dual operational amplifiers datasheet". [Online]. Disponível em www.ti.com/lit/gpn/lm158 [Acedido a 10/04/2018]
- [183] B. Carter and R. Mancini, Op Amps for Everyone. Newnes, 2017.
- [184] TI, "Designing Gain and Offset in Thirty Seconds". [Online]. Disponível em www.ti.com/lit/an/sloa097/sloa097.pdf [Acedido a 09/04/2018]
- [185] LEM, "LV 25-P datasheet". [Online]. Disponível em https://www.lem.com/sites/default/files/products_datasheets/lv_25-p.pdf [Acedido a 16/04/2018]

APÊNDICES

A. CARACTERÍSTICAS DE REDES WSN

Para compreensão de alguns dos conceitos abordados ao longo do capítulo 2, descrevem-se aqui as principais considerações quando se fala de redes de sensores sem fios.

A.1 ARQUITETURAS DE REDES WSN

Na arquitetura de rede, as WSNs seguem geralmente uma organização que compreende não só os nós sensores, mas também um dispositivo que comunica com esses nós, disponibilizando aos utilizadores a informação obtida, ou encaminhando-a para algum outro tipo de rede, normalmente a Internet. Este último é geralmente designado de *gateway*, embora cada tecnologia possa adotar outros termos (*base station* por exemplo). Uma das organizações mais comuns é a de salto único, onde cada nó comunica diretamente com o *gateway*. No entanto, dentro das WSN podem existir outras organizações consoante as ligações sejam ponto a ponto (*ad hoc*) ou em malha (*mesh*). No caso *ad hoc* os dispositivos comunicam diretamente entre eles, não necessitando de nenhuma infraestrutura adicional. Podem, no entanto, encaminhar também pacotes de dados a outros nós, um pouco como nas redes *mesh*. Redes em malha

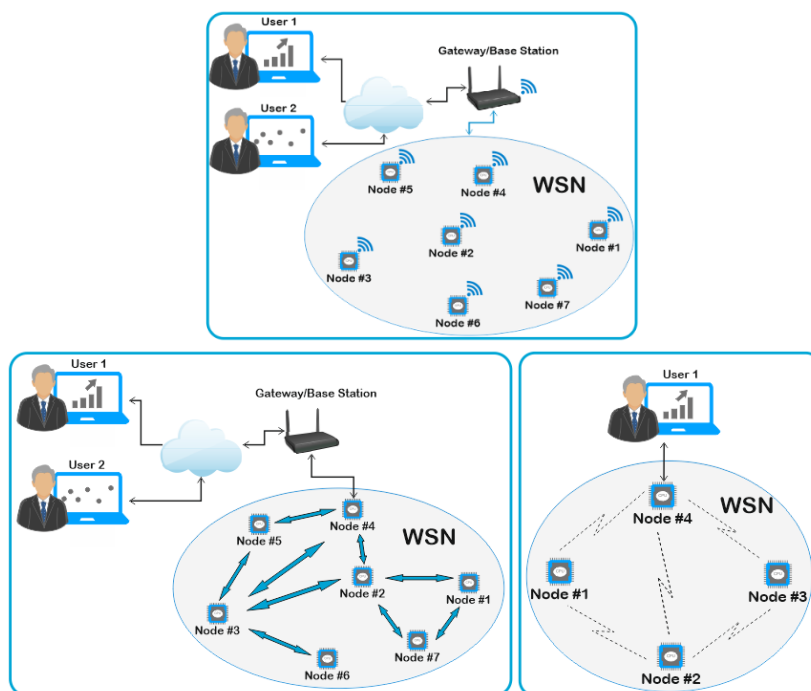


Figura A.1 - Arquitetura de rede em single hop (em cima), em malha (em baixo, esquerda) e ad hoc (em baixo, direita)

aproveitam os diversos nós para reencaminhar mensagens até ao destino ou a um nó que atue como *gateway* [11]. A Figura A.1 mostra um esquema dos 3 tipos de redes.

A.2 CONSIDERAÇÕES WSN

No sentido de implementar uma solução em redes de sensores sem fios vários critérios devem ser tidos em consideração. Os principais pontos a considerar são [12][60]:

- Alcance máximo da tecnologia. Esta é uma das mais importantes métricas de qualquer sistema de telecomunicações sem fios. Irá dar-nos a distância de comunicação permitida para cada tecnologia, designada de cobertura da rede, embora existam fatores externos que alteram este valor. Implementações em ambientes com obstrução do sinal, subterrâneos ou aquáticos diferem bastante de implementações em campo aberto, diminuindo a capacidade de conexões de rede. Obviamente a cobertura não será uniforme em todo o raio de ação e poderá variar ao longo do dia, devido a interferências aleatórias.
- Autonomia do dispositivo (duração de baterias ou eficiência energética do dispositivo). Devido ao constrangimento da energia limitada, os nós devem ser o mais eficientes possível de forma a aumentar o tempo de vida útil da rede. Juntamente com a eficiência, um bom planeamento de localização, de métodos e frequências de transmissão ou de esquemas de conservação de energia torna-se essencial para maximizar a duração do nó.
- Tempo de vida da rede. Este parâmetro que nos dá o tempo expectável de funcionamento da mesma. Uma vez que utilizamos recursos energéticos limitados, a substituição destes mesmos pode ser difícil ou impossível, levando a que a rede tenha um tempo de vida baixo. Por exemplo numa rede de monitorização de animais selvagens, pode não ser possível recuperar o nó quando acaba a bateria, sendo este inutilizado a partir desse momento.
- Quantidade de dados transferidos por unidade de tempo, ou débito (*data rate*).
- Tolerância a falhas e robustez a interferências. A natureza destes sensores leva a situações em que os mesmos possam não funcionar corretamente, necessitando de mecanismos para lidar com essas situações. A redundância pode ser uma das soluções.
- Escalabilidade da rede, ou seja, quantidade de nós ligados e respetiva possibilidade de escalabilidade.

- Confiabilidade dos dados. Numa rede com bastantes sensores muitas vezes heterogéneos, é importante que todos os dados sejam credíveis. Devem ser considerados esquemas de correção e deteção de erros de forma a minimizar a probabilidade de problemas associados a estas falhas.
- Segurança da rede. Engloba requisitos de proteção, por forma a que os dados sensíveis estejam protegidos e não sejam alterados e decodificados por terceiros;
- Arquitetura da rede e gama de frequências de trabalho, permitindo esta escolha garantir interoperacionalidade a nível mundial (ex, 2,4 GHz) ou não (868 MHz na Europa e 900 MHz na América).
- Comunicação uni ou bidirecional.
- Disponibilização dos dados. Refere-se à forma como os dados serão disponibilizados às aplicações dos utilizadores.
- Custo de implementação da solução. Cada tecnologia tem necessidades diferente em todo o material necessário. Se há tecnologias que exigem apenas o custo de alguns nós (ex: Zigbee), outras exigem uma infraestrutura complexa e bastante mais cara (ex: *Random Phase Multiple Access* RPMA). Do mesmo modo existem tecnologias que assentam em operadores de serviço os quais levam a um pagamento regular para utilização da rede (ex: redes baseadas em LTE).
- Roaming dos dispositivos. Uma vez que nem todos os nós estarão fixos na sua posição é relevante considerar métodos que suportem a mudança entre estações base.

A.3 CATEGORIZAÇÃO DE REDES SEM FIOS

As redes de comunicação podem ser categorizadas em função da sua área geográfica de cobertura. Esta última define diversos tipos de redes, consoante o seu alcance, nomeadamente:

- BAN, *body area network*. Os chamados *wearables*, presentes em muita da eletrónica de consumo atual e que incluem sensores de batimento cardíaco, de passada ou de pressão arterial são alguns dos exemplos de utilização. Geralmente os alcances são curtos, na ordem dos centímetros ou poucos metros. Muitas das tecnologias utilizadas aqui são comuns às utilizadas em nas *personal area network*, sendo por vezes considerada uma divisão desta última.
- PAN, *personal area network*. Bastante desenvolvida, este tipo de rede assenta em comunicações na ordem dos poucos metros, sendo o Bluetooth clássico o melhor exemplo da mesma.

- HAN, *home area network*, normalmente uma rede numa habitação residencial, com alcance de algumas dezenas de metros. Pode ser também considerada uma pequena rede de área local LAN, uma vez que utiliza as mesmas tecnologias. As redes de domótica referem regularmente este termo.
- LAN, *local area network*, rede de pequena dimensão, geralmente numa casa ou edifício. As redes Wi-Fi e *Ethernet* são as mais comuns. Os alcances podem ser de poucas centenas de metros.
- MAN, *metropolitan area network*, normalmente cobrindo uma cidade ou pequena área geográfica de poucos quilómetros. Geralmente, as redes *backhaul* de operadores podem ser consideradas deste tipo.
- WAN, *wide area networks*, onde as áreas de cobertura são por norma grandes, na casa dos quilómetros, podendo ser um país por exemplo.

Para além da distância de cobertura, característica sempre importante em qualquer tipo de rede, WSN podem ser caracterizadas por outros fatores como [140]:

- Número de sensores, podendo ser consideradas redes pequenas, médias, grandes ou muito grandes, partindo de poucos sensores até milhares.
- As capacidades de cada sensor, podendo a rede ser homogénea se todos os sensores são idênticos ou heterogénea se existem sensores com mais capacidades que outros.
- Rede estacionária, móvel ou híbrida consoante os nós possam estar sempre no mesmo local ou estejam em constante movimento.
- Rede plana ou hierárquica se todos os dispositivos desempenham a mesma função na rede ou se existem alguns com funções distintas como *gateways*;
- Número de saltos para transmitir os dados. Os nós podem ser de *single hop* se a comunicação for direta com a *gateway* ou *multi hop* se existirem vários saltos para os dados chegarem á *gateway*. Esta última acontece nas redes em malha, enquanto a outra é comum nas ligações Bluetooth.

A Figura A.2 mostra um exemplo de duas redes distintas, uma baseada em nós com um simples salto e a outra uma organização em malha.

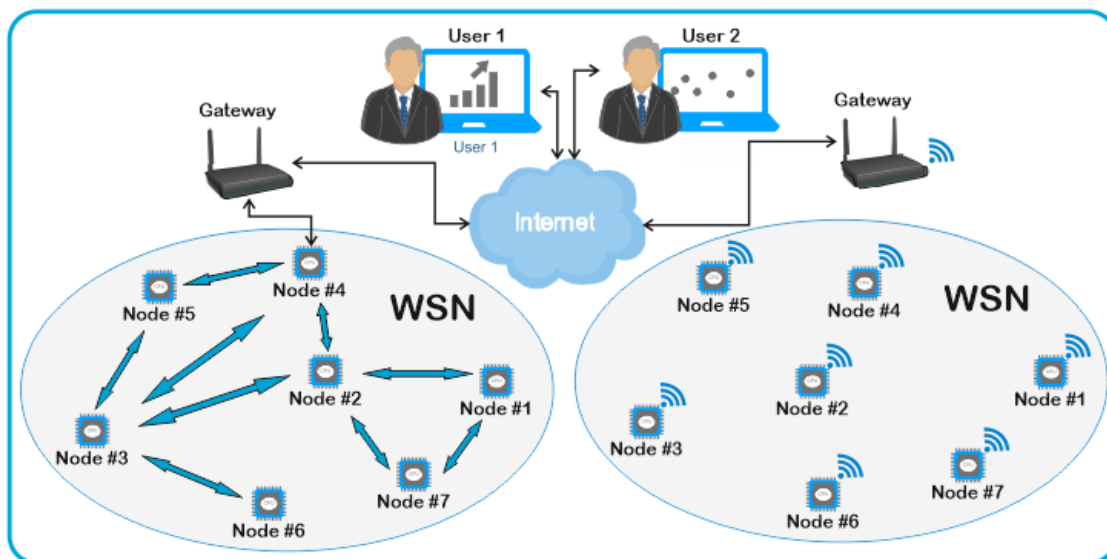


Figura A.2 Redes homogênea com 1 salto (À direita) e heterogênea com vários saltos (esquerda)

A.4 CONSTITUIÇÃO DE UM NÓ

Os nós sensores (também designados de *motés*) são dispositivos que devem ser capazes de ler algum fenómeno ambiental através dos respetivos sensores, processar esses dados e ter um sistema de comunicações sem fios capaz de transmitir esses dados para uma localização remota. Normalmente instalam-se em locais de difícil acesso, podendo estar fixos ou móveis. Podem ou não correr um sistema operativo, consoante o tipo de hardware do nó. São normalmente constituídos pelos componentes descritos de seguida, resumidos na Figura A.3.

- Uma unidade de processamento geralmente um microprocessador, FPGA ou ASIC. Os microcontroladores são os mais comuns, dispositivos que incluem um pequeno processador, memória e portas de entrada e saída, onde se podem ligar outros dispositivos externos como sensores, transdutores ou conversores ADC. O seu baixo custo, flexibilidade de utilização e baixo consumo energético tornam-no a escolha ideal. A sua capacidade de processamento e memória disponível é geralmente baixa. No entanto devem ser capazes de processar os dados, gerir temporizações e periféricos ou tomar decisões. A memória disponível é bastante pequena, normalmente na casa das dezenas a centenas de *kilobytes*.

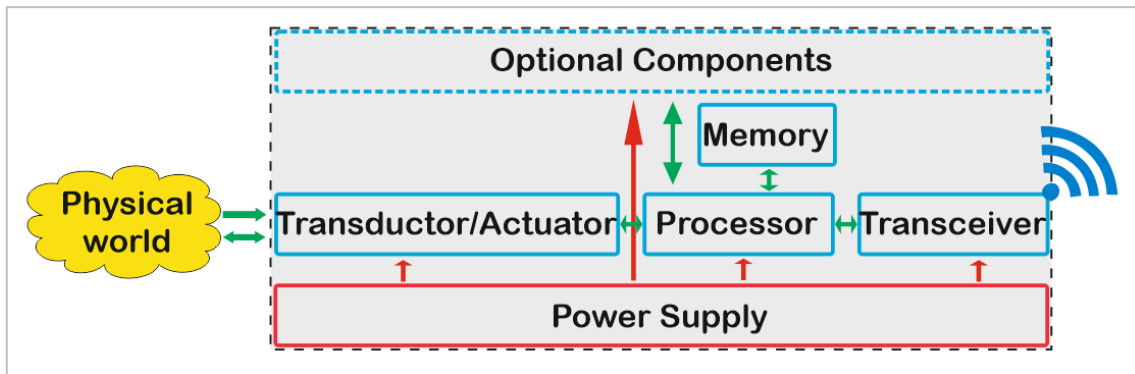


Figura A.3 Constituição de nó

Um nó pode correr um sistema operativo, o qual gere os recursos de *software* e *hardware* de forma eficiente [141].

- Os *transceiver* rádio são os componentes responsáveis pelas comunicações rádio e fazem a conversão dos sinais elétricos em eletromagnéticos e vice-versa. Existem para todas as tecnologias rádio, como Bluetooth ou Zigbee. Podem ter vários estados de funcionamento como acordados, a dormir, a transmitir ou a escutar. Estes 2 últimos são normalmente onde se gasta a maior parte da energia num nó, sendo por isso fundamental gerir estes estados corretamente. Normalmente é o microcontrolador a tomar essas decisões [142].
- Os sensores ou transdutores nos quais se traduzem os fenómenos físicos que se pretendem saber em informação elétrica que o microcontrolador pode interpretar. Estes podem ser passivos, ligados às entradas analógicas/digitais do microcontrolador, ou ativos e necessitando de alimentação para o seu funcionamento. Podem ler fenómenos de origem elétrica, mecânica, térmica, química, biológica, ótica ou magnética, permitindo desta forma ler um grande número de variáveis ambientais. Muitos destes sensores já existiam como dispositivos volumosos e de elevado consumo energético, não sendo viável a sua utilização autónoma em dispositivos operados a baterias. As novas técnicas em micro e nano eletrónica (por exemplo sistemas micro eletromecânicos MEMS) permitiram a criação de sensores pequenos e de alta eficiência, ideais para dispositivos pequenos, portáteis e operados a baterias.
- Uma fonte de alimentação que irá disponibilizar energia a todos os componentes do nó, normalmente uma bateria ou pilhas. No entanto existem soluções autónomas, que captam energia do meio, o chamado *energy harvesting*. A mais comum é a utilização de painéis solares, no entanto existem soluções que recolhem energia mecânica, química ou eletromagnética. Podem também existir soluções híbridas que incluem *energy harvesting* e baterias [143]. Obviamente a utilização de baterias leva a que a energia disponível seja

limitada pelo que os consumos de todos os componentes têm de ser dimensionados para maximizar a duração do nó e respetivo tempo de vida da rede. A utilização de bateria exige também ter em conta fatores como a estabilidade da voltagem ao longo do tempo, a auto descarga, a capacidade de fornecer corrente, o tempo de vida útil ou a capacidade de recarga. Estes fatores são importantes pois muitos dos nós atuais planeiam-se para durações até 10 anos [141].

Adicionalmente a estes e consoante a aplicação, podem-se incluir outros periféricos como câmaras de vídeo, cartões de memória, recetores GPS, ecrãs ou dispositivos de interação homem máquina.

A.5 APLICAÇÕES WSN

Com todas as soluções técnicas existentes no mercado, existe uma infinidade de aplicações para redes sem fios, em distintas áreas de atuação. Listam-se de seguida apenas alguns exemplos de utilização:

- Aplicações domésticas, normalmente associadas a domótica e automação onde se monitoriza o estado de diversas variáveis como temperaturas, estado de portas ou janelas, consumos de eletricidade, água ou gás entre outras [142].
- Aplicações para telemedicina, onde se monitorizam os sinais vitais de doentes de forma a acompanhar em tempo real o estado destes. (ex: AlarmNet) [140].
- Monitorização ambiental ou de vida selvagem. Várias aplicações têm sido desenvolvidas para seguir animais (ex: ZebraNet) [144], efetuar monitorização de vulcões (ex: Volcan Reventator) ou glaciares (ex: GlacsWeb) [144], e até mesmo de deteção de incêndios (ex: Firesense) [140].
- Aplicações industriais como monitorização de processos e matérias primas ou verificação de integridade estrutural em infraestruturas (pontes, edifícios) [145]
- Aplicações militares como vigilância remota e deteção de posição de alvos ou tropas (ex: PinPtr) [144].
- Utilização em agricultura, permitindo a monitorização de campos agrícolas (condições ambientais, dos solos, níveis de água, entre outros) [144] e a chamada agricultura de precisão [146]. Em monitorização da saúde animal em quintas, também é utilizado [144].
- Operações industriais ou logísticas, nomeadamente na monitorização do estado de processos industriais [140] ou bens perecíveis [147] e no seguimento de bens móveis [144].

- *Smart cities* onde se monitorizam diversos parâmetros da cidade como luminosidade, níveis de ruído, qualidades da água e ar, lugares de estacionamento vagos, entre outros, e se disponibilizam esses dados em tempo real aos cidadãos. Destes existem vários exemplos de aplicação como o projeto SmartSantander [148] ou na nossa região, a *smart city* de Lagoa [149].

A.6 APLICAÇÕES WSN EM SMART GRIDS

A aplicação de WSN em *smart grids* pode ocorrer em diferentes partes da rede, com diversas aplicações distintas para monitorização, otimização ou controlo. Em [9] definem-se algumas potenciais utilizações de sistemas IoT em *smart grids* como os descritos na Tabela A.1 seguinte.

Produção e Geração	Transmissão	Distribuição	Consumo
Monitorização em tempo real da produção	Controlo das linhas de transmissão e equipamentos	Controlo de transformadores em estações	Monitorização de consumos automática
Controlo das estações de produção	Monitorização de potência	Monitorização de cabos e condutas subterrâneas	Gestão energética de edifício
Controlo de fontes energéticas alternativas			Gestão de painéis solares
Monitorização de fontes distribuídas DER			Previsão de produção solar ou eólica consoante as condições climáticas do local

Tabela A.1 - Exemplos de possíveis utilizações IoT em smart grids

Nas redes elétricas atuais, desde a produção até à distribuição, os sistemas de monitorização assentam maioritariamente em sistemas cablados (*Power Line Communication* PLC, ADSL ou fibra), pois os sistemas de comunicação sem fios, ainda não oferecem a fiabilidade necessária para adoção em sistemas considerados críticos [9]. Na área do consumidor e em *Smart Home/Grid/Building*, existem várias opções de WSN que foram adotadas, nomeadamente Zigbee, ou a rede móvel (GSM/LTE) [2][7]. Estas, presentes no mercado há bastantes anos, têm maturidade suficiente para os fabricantes e operadores confiarem no seu funcionamento. As novas tecnologias LPWAN (ex: Sigfox, LoraWAN ou NB-IoT) são recentes no mercado, mas têm muito boas características para aplicações em sistemas de

monitorização de redes elétricas. Apesar de estas virem permitir novas aplicações na *grid* ainda não são utilizadas pelos operadores. Em *micro* redes podem, no entanto, ser aplicadas para toda a monitorização e gestão da rede, sendo uma solução económica e funcional.

B. TABELA COMPARATIVA TECNOLOGIAS WSN

	Curto alcance						Médio Alcance		Longo Alcance								
	Bluetooth 5	Bluetooth 4	ANT	Zigbee	WirelessHART	EnOcean	WiFi Halow	DASH7	NB-IoT	LTE M	EC-GSM-iot	Weightless N	Weightless P	Weightless W	Sigfox	RPMA	LoRaWAN
Banda de frequências	2,4 GHz	2,4 GHz	2,4 GHz	2,4 GHz ou 868 MHz	2.4 GHz	868 MHz	900MHz (868 MHz)	868 MHz ISM	LTE	LTE	GSM	Sub GHz ISM	Sub GHz ISM	400-800 MHz (espectro de TV)	868MHz ISM	2,4 GHz	868MHz ISM
Largura do canal	1 ou 2 MHz		1 MHz	2 MHz	2MHz		1, 2, 4, 8 ou 16 MHz	25kHz ou 200 kHz	200kHz	1.4 MHz	200 kHz	200Hz UNB	12.5 kHz	5MHz	100Hz UNB	1 MHz	125, 250 e 500 kHz
Alcance Urbano	40 m	10 m		30 m		30 m									10 Km	10 Km	10 Km
Alcance Rural ou LOS	200 m	50 m	30 m	150 m	250 m	300 m	1 Km	5 Km	22 Km	22 Km		3 Km	2 Km	5 Km	50 Km	200 Km	30 Km
Potência do emissor (máxima)	20 dBm	7dBm		20 dBm	10 dBm	(dependem do regulador)	10 mW	27 dBm	20/23 dBm	20/23 dBm	23 dBm	17 dBm	17 dBm	17 dBm	20 dBm	27 dBm	20dBm
DataRate no Uplink	2 Mbps	1 Mbps	60 Kbps	250 kbps ou 20 kbps(868 MHz)	250 Kbps	125 kbps	Entre 150 kbps e 346.6 Mbps	9.6 kbps, 55.5 kbps e 166.6 kbps	250 kbps	1 Mbps	70 a 240 kbps	100 bps	200bps a 100kbps	1kbps a 10Mbps	100 bps até ao máximo de 140 mensagens por dia	624 kbps	300 bps a 50 kbps
DataRate no Downlink	2 Mbps	1 Mbps	60 Kbps	250 kbps ou 20 kbps(868 MHz)	250 Kbps	125 kbps	Entre 150 kbps e 346.6 Mbps	9.6 kbps, 55.5 kbps e 166.6 kbps	170 kbps	1 Mbps	70 a 240 kbps	Sem Downlink	200bps a 100kbps	1kbps a 10Mbps	100 bps até ao máximo de 4 mensagens por dia	156 kbps	300 bps a 50 kbps
Topologia	Nó-a-nó ou malha	Nó-a-nó ou malha	Nó-a-nó, estrela, malha	Malha	Malha, estrela ou estrela-malha	Malha, estrela ou ponto a ponto	Estrela ou árvore	Estrela, árvore ou nó-a-nó	Estrela	Estrela	Estrela	Estrela	Estrela	Estrela	Estrela	Estrela ou arvore	Estrela de estrelas
Número de dispositivos	Ilimitado	Ilimitado	500	64000	1500		8191 (por ponto de acesso)	Ilimitado	200000	>20000	50000	Sem limite	Sem limite	Sem limite	Sem limite	384000	1000000 no uplink, <100k no downlink
Tamanho do pacote Uplink (dados uteis)	31 a 255 bytes	17-20 bytes		73 a 100 bytes	127 bytes	14 bytes	Até 7991 bytes (com possibilidade de chegar a 65535 bytes)	256 bytes	1000 bits (por TBS)	100 a 1000 bytes		20 bytes	A partir de 10 bytes	A partir de 10 bytes	12 bytes	de 6 a 10k bytes	de 8 a 250 bytes
Tamanho do pacote Downlink (dados uteis)	31 a 255 bytes	17-20 bytes		73 a 100 bytes	127 bytes	14 bytes	Até 7991 bytes (com possibilidade de chegar a 65535 bytes)	256 bytes	680 bits (por TBS)	100 a 1000 bytes		20 bytes	A partir de 10 bytes	A partir de 10 bytes	8 bytes	de 6 a 10k bytes	de 8 a 250 bytes
Link Budget	111dBm	92 dBm		108 dBm			140 dBm	140 dBm	164 dBm	156 dBm	164 dBm	160dBm	160dBm	160dBm	159 dBm	172 dB	157 dB
Duração máxima de baterias (estimada)				10 anos					10 anos	10 anos	10 anos	10 anos	8 anos	5 anos	20 anos	10 anos	10 anos
Necessidade de operador	Não	Não	Não	Não	Não	Não	Não	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Não
Duty Cycle (caso exista)						1%									1%		1%
Encriptação de dados			AES-128	AES-128	AES-128	AES-128		AES - 128	Rede	Rede	Rede		AES-128/256		Rede	Rede	AES-128

Tabela B.1 - Tabela Comparativa de tecnologias WSN

C. MATERIAL LORAWAN

Neste Apêndice iremos descrever o material (*hardware* e *software*) LoraWAN analisado no final de 2017. Consultamos várias lojas europeias para saber que produtos se encontravam disponíveis, as características dos mesmos e os respetivos preços. Verificamos também, que soluções de *software* existiam.

C.1 MÓDULOS LORAWAN

Existem no mercado módulos completos com microcontrolador e *transceiver* Lora, ou apenas o *transceiver*. Apesar destes últimos serem mais baratos e possíveis de integrar em microcontroladores estilo Arduino, procuramos soluções já integradas para simplificar e aumentar a fiabilidade. Fica a referência da existência de módulos de comunicação baseados nos *transceivers* HopeRF 95W, Microchip RN2483, RisingHF76, Multitech mDot ou Semtech SX1276. Existe bastante mais material no mercado, mas focámo-nos na disponibilidade imediata e em venda na Europa. Analisamos os módulos em função dos seus microcontroladores, memória disponível, entradas e saídas analógicas ou digitais, interfaces de comunicação, consumos de corrente, características Lora, linguagens de programação, certificações LoraWAN e custo de aquisição. De seguida resume-se o material analisado no final de 2017. **STM32L072Z-LRWAN1**

Plataforma da STmicroelectronics, composta por um microcontrolador STM32L072CZ ARM M0+ 32 MHz de 192 kB de FLASH, 20 kB de RAM e 20 kB de EEPROM. Tem 12 entradas analógicas e 2 saídas analógicas. Tem entrada USB para programação e alimentação, funcionando a 3,3V. Pode comunicar com outros dispositivos utilizando os protocolos UART, I2C e SPI. O *transceiver* Lora é um Murata CMWX1ZZABZ-09 com *chip* Semtech SX1276, com sensibilidade até -137 dBm. A corrente em receção é de 10 mA, não sendo indicadas corrente em *sleep* e transmissão. Inclui conectores uFL e SMA e também antena SMA.

A programação pode ser feita utilizando a plataforma Mbed ou softwares IAR Ewarm ou Keil MDK-ARM. É um produto certificado LoraWAN classe A e C. Adicionalmente tem conexão Sigfox. O custo de aquisição ronda os 40 euros [150].

C.1.2 ADAFRUIT FEATHER M0

Esta plataforma utiliza um microcontrolador Atmel ATSAMD21G18, de arquitetura ARM M0 com 48 MHz de relógio, 256 kB de memória FLASH e 32 kB de RAM. A *board* inclui programador bastando por isso ligar o cabo USB. Funciona a 3,3V com regulador interno capaz de fornecer 500 mA, e inclui conector para bateria LiPo externa e respetivo circuito de carga. Tem 20 pinos GPIO, com 10 entradas analógicas e 1 saída analógica. Estes pinos permitem PWM. Inclui pinos UART, I2C e SPI para ligação a outros periféricos. A nível Lora incluem um módulo RFM96 baseado no *chip* Semtech SX1776. Não inclui conector para antenas, sendo necessário utilizar uma antena de fio com 8,2 cm ou soldar um conector uFL no espaço apropriado. Os consumos rondam os 120mA em transmissão, 40 mA em escuta e 300 uA em modo *sleep*.

É programado no Arduino IDE sendo por isso fácil de programar e de encontrar exemplos de referência. O custo ronda os 29 euros mas não é certificado LoraWAN [151].

C.1.3 PYCOM LOPY

Este módulo é baseado no microcontrolador ESP32 160 MHz de 32 bits, com 512 kB de RAM e 4 MB de FLASH. Inclui 8 entradas e 2 saídas analógicas (ADC/DAC), permite até 24 GPIO e tem conexão I2C, SPI e UART. Com consumos em modo *sleep* de 30 uA e 100 mA em funcionamento, conta com uma entrada de tensão entre 3.3 e 5.5V. O *transceiver* Lora é um Semtech SX1272, certificado para classes A e C. Juntamente com isto esta placa inclui Bluetooth (normal e LE) e WiFi 802.11b/g/n.

A programação é em MicroPython e pode ser programado em VisualStudio, Atom ou os *scripts* enviados via *file transfer protocol* (FTP). A programação pode ser feita diretamente com um adaptador série ou utilizando uma das suas placas de expansão. O custo ronda os 30 euros, excluindo antena. Atualmente (Junho 2018), este módulo deixou de ser comercializado, sendo substituído pelo Lopy4 que inclui conectividade Sigfox, 4 MB de RAM, 8 MB de Flash e mantendo o mesmo preço [152].

C.1.4 LIBELIUM WASPMOTE LORAWAN

Este módulo baseia-se na plataforma modular Waspote da Libellium e conta com uma placa principal com um microcontrolador Atmega 1281 a 14.74 MHz, 128 kB de memória FLASH, 8 kB de RAM e EEPROM com 4 kB. O consumo desta varia entre os 57 mA e os 7uA. Permite ligar cartões SD e conta com 7 entradas analógicas e 8 pinos I/O digitais assim

como comunicações SPI e I2C. É alimentado a 3.3 ou 5 V via USB. O módulo LoRaWAN é baseado no *transceiver* Microchip RN2483, permitindo uma sensibilidade de -136 dBm e um consumo de 38,9 mA.

É programado em linguagem C num IDE próprio e certificado LoraWAN classe A. O custo base anda nos 228 euros. A empresa disponibiliza também um serviço na *cloud* onde se podem aceder aos dados [153][154].

C.1.5 MICROCHIP DV164140-1

Este conjunto de desenvolvimento é constituído por uma placa principal que funciona como *gateway* e pequenos nós Lora. A placa principal tem um microcontrolador PIC24EP512GU810, de 16 bits com 512KB de RAM, 32.768 MHz, interface *ethernet* e um ecrã TFT 3.2 polegadas. Os nós (DM164138) têm um microcontrolador PIC18LF45K50 48 MHz de 8 bits, FLASH de 32 kB, RAM com 2048 *bytes* e 256 *bytes* de EEPROM. e um *transceiver* Microchip RN2483. Incluem um pequeno ecrã, alimentação por 2 pilhas AAA, sensor de luz e temperatura. Não se indicam os consumos de corrente.

O conjunto custa 426 euros enquanto os nós custam 60 euros. São certificados LoraWAN classe A [155].

C.1.6 MULTITECH MULTICONNECT MDOT

Módulos de aplicação industrial, utilizando um microcontrolador STM32F411RET a 100 MHz, incluem 512 kB de memória FLASH e 128 kB de RAM. O consumo divulgado é de 40 uA em modo *sleep* e 42 mA em envio. Possui comunicação SPI, I2C, UART, 21 pinos GPIO e 11 entradas analógicas, podendo ser alimentado a 3.3 ou 5 V.

A programação é feita pela plataforma Mbed em linguagem C/C++ e cada módulo tem um custo de 45 euros [156].

C.1.7 SEEDSTUDIO SEEDUINO LORAWAN

Esta placa baseada num microcontrolador ATSAMD21G18 a 48MHz, 256 kB de Flash, 32 kB de RAM tem um *transceiver* RHF76 com uma sensibilidade de -140 dBm. Conta com 20 pinos de GPIO, 6 entradas e uma saída analógicas e um consumo mínimo de 80 uA. Existe numa versão com GPS integrado e trás chip de gestão de carregamento de bateria de lítio.

A programação é feita no Arduino IDE e é compatível com LoraWAN A e C. O custo ronda os 40 euros [157].

C.1.8 COMPARAÇÃO DE CARACTERÍSTICAS TÉCNICAS DOS MÓDULOS

		STM32L072Z-LRWAN1	ST P-NUCLEO-LRWAN1	Adafruit Feather M0	Pycom Lopy	Libelium Waspote	Microchip DV164140-1	Multitech MultiConnect mDot	Seedstudio LoraWAN
Processador	MCU	STM32L072CZ	STM32L073RZ	Atmel ATSAM21G18	Espressif ESP32	Atmel Atmega 1281	PIC18LF45K50	STM32F411RET	Atmel ATSAM21G18
	MCU CLK	32 MHz	32 MHz	48 MHz	160 MHz	14.74 MHz	48 MHz	100 MHz	48 MHz
Memória	FLASH	192 kB	192 kB	256 kB	4 MB	128 kB	32 kB	512 kB	256 kB
	RAM	20 kB	20 kB	32 kB	512 kB	8 kB	2048 bytes	128 kB	32 kB
	EEPROM	20 kB	6 kB				256 bytes		
I/O analógicas	A IN (ADC)	12		10	8	7		11	6
	A OUT (DAC)	2		1	2				1
I/O Digitais	GPIO			20	24	8	13	21	20
Tensão de alimentação	VDD	3.3	1.65 - 3.6	3.3	3.3 - 5.5		3.3	3.3 - 5	3.3
Interfaces	UART	X		X	X			X	
	SPI	X		X	X	X		X	
	I2C	X		X	X	X		X	
Consumo	Corrente (rx, tx, sleep)	10mA,-,-	10mA,-,-	40mA, 120mA, 300uA	-, 100 mA, 30 uA	-, 57 mA, 7 uA		-, 42 mA, 40 uA	
Transceiver Lora		SX1276 (Murata CMWX1ZZABZ-09)	SX1272	SX1276 (RFM96)	SX1272	Microchip RN2483	Microchip RN2483		Rising HF RHF76-052
Linguagem de programação		C/C++ (mbed, IAR, Keil)	C/C++ (mbed, IAR, Keil)	C/C++ (Arduino IDE)	MicroPython (Atom, VisualStudio)	C		C/C++ (mbed)	C/C++ (Arduino IDE)
Certificação LoraWAN		A, C	A		A, C	A	A	A	A/C
Custo aquisição (euros)		40	60	29	30	228	426 (60)	45	41
Outras características		Sigfox			Bluetooth, WiFi		Inclui gateway		Inclui GPS

Tabela C.1 - Tabela comparativa de módulos emissores LoRaWAN

C.2 GATEWAY

Nas *gateways* temos a opção de comprar uma completamente funcional, ou construir uma com vários componentes. Com um microcomputador tipo Raspberry Pi ou Beaglebone e uma placa concentradora, podemos construir uma *gateway* utilizando o *software hardware abstraction layer (HAL) packet forwarder* de referência da Semtech [126]. Esta opção é mais configurável. De seguida descrevem-se algumas das soluções analisadas.

C.2.1 LORAGO PORT

Esta é uma placa concentradora baseada nos *transceivers* Semtech SX1257/SX1301, permitindo a receção de 8 canais simultâneos. A interface pode ser feito via SPI ou via USB para um computador. O custo ronda os 150 euros incluindo antena, GPS e HAT para Raspberry PI. Pode ser alimentado com 4 a 12 V, com um consumo máximo de 420 mA [158].

C.2.2 SEEDSTUDIO LORA HAT

Esta placa HAT para Raspberry Pi assenta no *transceiver* SX1278 da Semtech, sendo que apenas apresenta 1 canal de receção e uma sensibilidade de -148 dBm. Inclui módulo GPS e antena, com conector SMA e um custo de aproximadamente 26 euros. A alimentação é feita através do *header* e não é divulgado o consumo [159].

C.2.3 IMST IC880A

Esta é uma placa concentradora baseada nos *transceiver* Semtech SX1257/1301 e permitindo 8 canais de receção com uma sensibilidade de -138 dBm. A interface é feita pelos pinos de SPI. Necessita de alimentação externa de 5 V, consumindo um máximo de 433mA. Permite a ligação de um recetor GPS soldado na placa (não incluído) e para a ligação da antena inclui um conector SMA. O custo ronda os 119 euros, mas não inclui nenhum HAT nem a antena [160].

C.2.4 LAIRD SENTRIUS RG186

Esta *gateway* profissional, corre um sistema operativo Linux num processador Atmel A5 a 536 MHz. Disponibiliza *interfaces* wi-fi, Bluetooth e *ethernet*. Conta com caixas estanques opcionais IP65 para montagem exterior. A nível Lora conta com *transceiver* Semtech SX1257/1301 que garante 8 canais. O *software* permite ligação direta aos serviços TTN, IOT-X e Lorient.io, e configurável a outros serviços. O custo ronda os 216 euros [161].

C.2.5 MULTITECH CONDUIT GATEWAY IP67 BASE STATION

As gamas de material da Multitech são desenhadas para a indústria sendo por isso de bastante qualidade e completas. Correm um sistema operativo mLinux, num processador ARM a 400 MHz e disponibilizam *backhauls* LTE, Wi-Fi e *ethernet*. Podem ser programadas em diversas linguagens (C/C++, Node-RED, java, python, ...) e já incluem pacotes com

serviços adicionais (servidores web, bases de dados,...). Existem variantes de *indoor* e *outdoor*. Os preços começam nos 480 euros [162].

C.2.6 KERLINK WIRNET STATION

Dentro desta gama de gateways, a Kerlink disponibiliza várias opções profissionais com *hardware* e *software* proprietário, próprias para montagem exterior e interior. Este modelo assenta num processador ARM, com 128 MB de memória, a correr um sistema operativo Linux. Permite 9 canais e tem preços a partir dos 450 euros, sendo uma boa solução para aplicações industriais. Incluem suporte a funcionalidades extra como *firewall*, *gps*, alimentação via *power over ethernet* (PoE), ligações *backhaul* a redes celulares e *ethernet* ou geolocalização [163].

C.2.7 COMPARAÇÃO DE CARACTERÍSTICAS TÉCNICAS DAS GATEWAYS

	Placa concentradora		
	LoraGo PORT	Seedstudio Lora Hat	IMST iC880A
Transceiver Lora	Semtech SX1257/SX1301	Semtech SX1278	Semtech SX1257/SX1301
Sensibilidade	-	-148	-138
Canais	8	1	8
SPI	X	-	X
USB	X	N	N
Alimentação	4-12	5	5
Consumo	420	-	433
Custo (€)	150	26	119
Outras características	Inclui antena, GPS e HAT para RPI	Inclui GPS e antena	-

Tabela C.2 - Tabela comparativa de placas concentradoras

		Gateway completa		
		Laird Sentrius RG186	Multitech Conduit GW IP67	Kerlink Wirnet
Processador		Atmel A5 536 MHz	ARM 400 MHz	ARM
Sistema Operativo		Linux	mLinux	Linux
Transceiver Lora		Semtech SX1257/SX1301	-	-
Canais		8	-	9
Interfaces	Wi-Fi	X	X	
	Ethernet	X	X	X
	3G/LTE		X	X
	Bluetooth	X	-	-
Custo (€)		216	480	450
Outras características		Caixa IP65 Opcional	-	GPS, PoE, firewall

Tabela C.3 - Tabela comparativa de gateways comerciais completas

C.3 SERVIDORES LORAWAN NS

Muitos dos servidores de aplicação existentes *online* funcionam também como servidores de rede. No entanto procuramos que soluções existiam que nos permitissem correr um servidor privado e que fossem gratuitas e *open source*. Encontramos 2 soluções descritas de seguida.

C.3.1 LORASERVER.IO

A implementação “Loraserver.io” [164] assenta num esquema modular, onde cada componente é independente. A versatilidade desta solução permite ter em máquinas distintas os diversos componentes, como por exemplo o servidor de rede e de aplicações em diferentes locais. No entanto também permite incluir tudo na mesma máquina, de forma a que num único local de rede estejam todos os componentes necessários ao correto funcionamento LoRaWAN. Funciona em sistemas operativos Linux, Windows ou OSX. A comunicação entre módulos assenta no protocolo MQTT, assim como a comunicação para o exterior utiliza este mesmo protocolo ou *http*. Os dados internos como configurações de nós são guardados numa base de dados PostgreSQL assim como os dados temporários numa base de dados REDIS. Estas são instaladas e configuradas à parte. Suporta dispositivos de classe A, B e C, ADR, ativações ABP e OTA e reconfiguração automática dos canais em todas as bandas ISM especificadas na norma [165]. Permite gestão e controlo remoto dos diferentes componentes utilizando as APIs

gRPC ou REST. Suporta uma divisão por entidades (*tenants*) podendo o mesmo servidor tratar nós de diferentes *tenants*. Está dividida em 4 componentes organizados na arquitetura da Figura C.1, sendo eles o *LoRa Gateway Bridge*, *LoRa Channel Manager*, *LoRa Server* e *Lora App Server*. Na imagem é possível ver que todos os componentes dentro da nuvem cinza podem estar dentro da mesma máquina. As aplicações e a *gateway bridge* podem ou não estar, consoante a rede e *hardware* a utilizar.

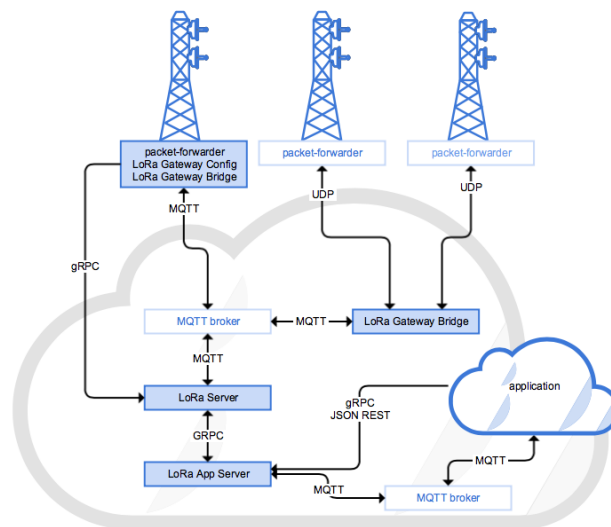


Figura C.1 - Arquitetura Loraserver.io

C.3.1.1 LoRa Gateway Bridge

Este módulo utilizado em *gateways* ou no servidor, efetua uma conversão entre o protocolo UDP (vem da implementação de referência) em MQTT, de forma a que possa enviar os dados a qualquer cliente MQTT à escuta. Uma *gateway* típica está pensada para enviar os seus dados em JSON sobre UDP, num esquema de encaminhador de pacote, e poderá não implementar nenhum mecanismo de segurança. A utilização deste módulo visa garantir que as comunicações são fiáveis e seguras, devido às funcionalidades de qualidade de serviço do MQTT (ver capítulo do MQTT). Assim o módulo deve correr direto na *gateway*, convertendo os dados e enviando ao *broker* no NS. Em casos onde não seja possível instalar o módulo na *gateway* a conversão deverá ser feito utilizando o módulo no NS, que irá receber os dados via UDP das *gateways* e efetuar a conversão localmente. Em *gateways* que utilizem um sistema operativo baseado em Linux (ex. *gateways* em RaspberryPI) é possível instalar o módulo, efetuando este a conversão e envio ao *broker* respetivo. A utilização deste componente permite uma melhor gestão de encaminhamentos entre servidor e *gateway* assim como uma maior

segurança e fiabilidade utilizando o MQTT com TLS. Ao mesmo tempo torna mais fácil o acesso à informação, pois basta-nos inscrever o tópico respetivo e temos acesso aos dados.

C.3.1.2 LoRa Channel Manager

Este é um módulo simples que permite modificar os canais utilizados por uma *gateway* baseada em *packet-forwarder*, de forma simples e rápida. Periodicamente é verificado se houve alterações destes canais no Network Server, utilizando gRPC. Se existiram alterações estas são copiadas para a *gateway* e esta reiniciada. As informações de canais são posteriormente disponibilizadas aos nós.

C.3.1.3 LoRa Server

Este componente implementa o servidor de rede responsável por efetuar toda a gestão de nós e *gateways*, cumprindo os requisitos LoRaWAN. As suas responsabilidades são:

- Garantir que nós se podem ligar á rede (ativação ABP e OTA) e com que configurações, mediante informações que este pede ao LoRa App Server;
- Verificar e eliminar pacotes duplicados, de forma a enviar os dados unicamente uma vez às aplicações;
- Encripta e descripta os dados das aplicações;
- Gere e agenda os envios de dados *downlink* aos nós, caso existam;
- Envia comandos MAC aos nós de forma a adaptar estes às necessidades do canal físico (ADR, mudança de canais), saber o estado destes ou reconfigurar a rede;
- Gerir as diferentes *gateways* presentes na rede.

Neste momento suporta dispositivos classe A, B e C, autenticações ABP e OTAA, mensagens confirmadas e não confirmadas, ADR e múltiplas bandas ISM.

A nível interno, os dados são enviados e recebidos via *broker* MQTT, nomeadamente utilizando o Mosquitto. As informações de configurações persistentes, respetivas a *gateways* ou aplicações, são mantidas em bases de dados PostgreSQL. Informações que não são permanentes, nomeadamente dados em transição, são mantidas numa base de dados Redis. Todos estes 3 componentes são configurados na instalação e essenciais para o funcionamento correto do sistema. Parâmetros como a banda a utilizar, IP dos outros *brokers* MQTT ou certificados de segurança devem também ser configurados no arranque.

C.3.1.4 LoRa App Server

Apesar do nome, este não corre nenhuma aplicação final para o utilizador. É o componente responsável por criar uma interface de utilizador para gerir o tratamento dos *payloads*, nós e utilizadores da rede. Cria e gere toda a organização da rede utilizando a *interface web* ou remotamente a partir sua API (utilizando gRPC ou RESTful JSON). Envia os *payloads* de dados recebidos às aplicações respetivas (externas ao módulo) assim como mantém uma fila de dados a enviar aos nós. Utiliza o MQTT para estas trocas de dados. Suporta dispositivos classe A, B e C assim como mensagens confirmadas e não confirmadas. Similarmente ao LoRa Server, utiliza uma base de dados PostgreSQL para guardar dados persistentes, Redis para dados não persistentes e MQTT para comunicações de dados.

O módulo permite a criação de utilizadores, organizações de rede, perfis de serviço ou dispositivo, aplicações, dispositivos, servidores de rede e *gateways*.

C.3.2 LORAWAN-SERVER

O servidor Lorawan-Server [166] apresenta uma estrutura mais compacta, funcionando apenas num único módulo como servidor de rede e aplicações. É um servidor leve que funciona em sistemas operativos Windows, Linux, OSX e sistemas embebidos como RPI, sendo escrito em linguagem Erlang de forma a torná-lo mais fiável e tolerante a falhas. Implementa a especificação 1.0.2, suportando nós de classe A e C, ADR, mensagens confirmadas e não confirmadas, *multicast* e todas as regiões definidas na especificação. Permite enviar dados externamente via MQTT, *http*, AMQP e *websockets* assim como guardar as mensagens diretamente numa base de dados MongoDB. Inclui integração direta em diversas plataformas IoT na *cloud* como Amazon AWS, Thingspeak, IBM Watson e Microsoft Azure. Podemos também filtrar os *payloads* permitindo desta forma uma separação dos dados, antes do envio às aplicações.

As *gateways* existentes na rede comunicam as mensagens utilizando o protocolo UDP de referência para o porto de escuta do servidor. Este verifica a integridade e duplicação das mensagens e se tudo estiver correto prepara-as para o envio ao servidor de aplicações (designados neste de *backends*). Este processo é transparente para o utilizador. As mensagens recebidas, após as verificações iniciais, são disponibilizadas a um *handler* que pode formatar o *payload* (separar bytes por exemplo) e adicionar à mensagem campos informativos da *gateway* como a qualidade do sinal LSNR ou data e hora de receção dos mesmos na *gateway*. Com os dados preparados, estes são enviados para um ou vários *connectors* que como o nome

indica são as conexões para o exterior. Aqui, todos os campos de dados provenientes dos *handlers* são estruturados num formato adaptado ao envio, como JSON ou *webform*. Os métodos (*http*, *websockets*, ...) e endereços para envio ou os padrões de tópicos são definidos aqui. Com estas configurações, as mensagens podem então ser enviadas às aplicações, pelo *backend*.

Adicionalmente, este NS pode gerir várias redes, assim como gerir e configurar nós comissionados e diferentes perfis de dispositivos, útil para quando na rede se utilizam nós com o mesmo *hardware*. Pode ser controlado remotamente utilizando a REST API disponibilizada. A escolha acabou por cair neste NS, devido aos diversos *backends* que disponibiliza, assim como à leveza do servidor, que iria correr num RPI.

C.4 SERVIDORES DE APLICAÇÃO AS

A necessidade de existir uma aplicação para processar os dados disponibilizados levou-nos na busca de soluções interessantes para visualização destes. A criação de uma aplicação bem elaborada obrigaria a um esforço adicional e ocuparia bastante tempo. Assim procuramos as melhores soluções existentes que nos permitissem visualizar os dados obtidos na rede e que fossem versáteis na sua utilização. Existem bastantes servidores de aplicação preparados para LoraWAN *online* mas para uma utilização *offline* a escolha é bastante mais restrita. De seguida resumimos algumas das soluções analisadas.

C.4.1 AS OFFLINE

Estes servidores podem ser instalados em qualquer máquina privada, sendo por isso interessantes para redes deste estilo. Apesar disso podem manter comunicação para fora da rede, para acessos aos dados de forma remota a partir de qualquer lado com acesso à internet. São utilizados para soluções IoT em geral, não sendo específicos para LoraWAN.

C.4.1.1 Freeboard

A plataforma gratuita Freeboard assenta na utilização de *dashboards*, painéis com mostradores onde os dados dos utilizadores são representados em tempo real, de forma visual em gráficos, mapas ou tabelas por exemplo. As diferentes *dashboards* são configuradas e customizadas pelo utilizador com *widgets*, que contêm visualizações pré-definidas. No entanto não inclui muitas funcionalidades, como multiutilizadores ou separação por zonas [167].

C.4.1.2 Thingboard

A plataforma *open source* gratuita Thingsboard, é desenhada para IoT e permite a recolha, processamento, armazenamento e visualização de dados de diversos dispositivos. Permite criar *dashboards* com gráficos, mapas e tabelas onde mostra os dados recolhidos. Permite também manipulação destes dados e atuação baseada em eventos, como alarmes ou *emails* de alerta. A nível de conectividade, permite receber dados via MQTT, CoAP ou HTTP, sendo assim bastante versátil na integração com nós sensores. Permite múltiplos utilizadores, gestão de dispositivos e conexões seguras. Permite ainda a utilização de um componente que liga diretamente Sigfox, servidores OPC-UA ou Modbus [168].

C.4.2 AS ONLINE

A utilização de servidores *online* permite que os dados estejam disponíveis a partir de qualquer local com ligação à internet. Ao mesmo tempo podem integrar o servidor de rede no mesmo serviço, unificando desta forma a criação da rede. Destes existem redes comerciais que permitem integrar alguns nós ou *gateways* gratuitos e existem redes totalmente gratuitas. Apesar de não podermos utilizar nenhum destes serviços, pela imposição da rede privada, estes são ótimos para testar pequenas redes e o funcionamento das mesmas. Fica a referência que foi testado o funcionamento dos serviços Loriot [169], ResIoT [170] e TheThingsNetwork (TTN) [171]. Este último merece destaque por ser uma plataforma gratuita composta por centenas de *gateways* dos utilizadores, que as disponibilizam como pontos de acesso à rede TTN e que permitem a utilização das mesmas por qualquer outro utilizador, criando uma rede global gratuita.

D. INSTALAÇÃO E CONFIGURAÇÃO INICIAL DO RASPBERRYPI3

```
1 #Os exemplos seguintes utilizam um utilizador "iseualg" de pass "iseualg", apenas para demonstração.
2 #Utilizar outros nomes
3 #
4 #####
5 #
6 #   Raspberry Pi configuração inicial
7 #
8 #####
9
10 #Após instalação completa do Raspbian, no menu de configuração
11 sudo raspi-config
12 #Ativar spi e ssh (e VNC para acesso gui remoto - Neste adicionar utilizador iseualg)
13 #Hostname: LoraGatewayIseUalg
14
15 #Criar utilizador iseualg@iseualg e dar-lhe todos os privilegios
16 sudo adduser iseualg
17 sudo visudo
18 #na linha #user privilege spec adicionar
19 agerar ALL=(ALL:ALL) ALL
20 #ctrl+O , Y, ctrl+X para guardar e sair
21
```

```

22 #Mudar password root para "iseualg"
23 sudo passwd root
24
25 #desativar login automatico nas configurações e reiniciar, entrando com user: iseualg
26 #remover utilizador pi
27 sudo deluser pi
28
29 #Adicionar NTP UALG
30 sudo leafpad /etc/systemd/timesyncd.conf
31 #descomentar #NTP= para
32 NTP=ntp.ualg.pt
33 #ctrl+s para Salvar
34
35 #Arrancar serviço
36 timedatectl set-ntp true
37 #Verificar com:
38 timedatectl status
39 reboot
40
41 #se for para usar wifi abrir wpa_supplicant.conf(ou criar antes se não existir)
42 sudo leafpad /etc/wpa_supplicant/wpa_supplicant.conf
43 #adicionar as seguintes linhas (com exemplo da rede da sala 160)
44 ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
45 network={
46     ssid="sensornet160"
47     psk="sensor123"
48 }
49
50 #Configurar um ip fixo na interface respetiva, no ficheiro de config dhcp
51 #Neste caso será 192.168.1.20
52 sudo leafpad /etc/dhcpd.conf
53 #Adicionar as seguintes linhas com os ip's respectivos
54 #Configurar tambem router para não utilizar dhcp em toda a gama de endereços, de forma a deixar
55 #livres os que queremos fixos >>ex: DHCP 192.168.1.100 a 192.168.1.254, desta forma todos os
56 #endereços entre o 1 e o 99 não são atribuidos por dhcp
57 interface wlan0
58 static ip_address=192.168.1.20/24
59 static routers=192.168.1.1
60 static domain_name_servers=192.168.1.1
61
62 #Fazer um cleanup de programas desnecessários
63 #ver espaço com
64 df -h
65 #ver o que tá instalado e desinstalar o que não é necessário
66 dpkg-query -l> list_packages.txt
67 #Remover programas não utilizados, e atualizar sistema
68 sudo apt-get remove --purge wolfram-engine bluej greenfoot nuscratch scratch sonic-pi claws-mail
69 sudo apt-get clean
70 sudo apt-get autoremove
71 sudo reboot
72 sudo apt-get update && sudo apt-get upgrade -y
73

```

E. RESET DO CONCENTRADOR IC880

```

1 #####
2 #
3 #   Reset do Concentrador
4 #
5 #####
6
7 #Confirmar funcionamento dos pinos GPIO/livraria wiring
8 #Verificar instalação pedindo a versão (vem de origem no stretch)
9 gpio -v
10 #Se devolver versão está instalado se não der verificar instalação em
11 http://wiringpi.com/download-and-install/
12 #Fazer leitura do estado dos pinos
13 gpio readall
14
15 #Criar script para reiniciar iC880-SPI_reset.c O pino 0 corresponde ao pino fisico 11 do rpi - ver
16 wiringpi pinout
17 #Tudo o que seja relacionado com Lora fica na pasta /usr/Lora, criar pasta e ficheiro *.c com código
18 de reset
19 sudo mkdir /usr/Lora
20 sudo touch /usr/Lora/iC880-SPI_reset.c
21 sudo chmod 777 /usr/Lora/iC880-SPI_reset.c
22 sudo leafpad /usr/Lora/iC880-SPI_reset.c
23
24 #No ficheiro adicionar o seguinte código.
25 #O funcionamento é simples, define-se o pino wiring de reset, 0 neste caso
26 # O pino wiring 0 corresponde ao pino fisico rpi 11 >> Ver pag web wiring
27 #colocá-lo a nivel logico alto durante 5 segundos baixando de seguida

```

```

26 '
27 #include <wiringPi.h>
28 #include <unistd.h>
29 #define GPIO_RESET_PIN 0
30
31 int main(){
32     wiringPiSetup();
33     pinMode(GPIO_RESET_PIN, OUTPUT);
34     digitalWrite(GPIO_RESET_PIN, HIGH);
35     sleep(5);
36     digitalWrite(GPIO_RESET_PIN, LOW);
37     return 0;
38 }
39 '
40
41 #Compilar o código e torná-lo executavel, dando as respetivas permissões de execução
42 cd /usr/Lora/
43 #Compilar com gcc já vem no stretch
44 gcc -Wall -o iC880-SPI_reset iC880-SPI_reset.c -lwiringPi
45
46 #Para correr basta executar
47 sudo ./iC880-SPI_reset
48 #Para confirmar a execução verificar se o pino 11 está como output e a nível 0
49 gpio readall
50
51 #Colocar no ficheiro de arranque da máquina (ver linha 259 deste texto)
52 sudo leafpad /etc/rc.local
53 #adicionar caminho do executável
54 sudo /usr/Lora/iC880-SPI_reset
55 #reiniciar e verificar se o pino 11 está como output e a nível 0
56 reboot
57 gpio readall

```

F. CONFIGURAÇÃO DA GATEWAY

```

1 #Os exemplos seguintes utilizam um utilizador "iseualg" de pass "iseualg", apenas para demonstração.
2 #Utilizar outros nomes
3 #
4 #####
5 #
6 # Lora gateway lib e packet forwarder
7 #
8 #####
9
10 #instalar git se não estiver instalado
11
12 sudo apt-get install git
13 #criar pastas para livraria e pkt fwd e dar permissões
14 mkdir -p /usr/Lora/lora_gateway
15 mkdir -p /usr/Lora/packet_forwarder/
16 cd /usr/Lora/
17 sudo chmod -R 777
18 #Sacar lora_gateway e pcktforwarder
19 git clone http://github.com/Lora-net/lora_gateway.git
20 git clone http://github.com/Lora-net/packet_forwarder.git
21
22 #Para debug editar /lora_gateway/libloragw/library.cfg colocar a l os que queremos ver na saída ex:
23 DEBUG_AUX= 0
24 DEBUG_SPI= 1
25 DEBUG_REG= 1
26 DEBUG_HAL= 1
27 DEBUG_LBT= 0
28 DEBUG_GPS= 0
29 #Salvar ficheiro
30
31 #editar a velocidade SPI no ficheiro /lora_gateway/libloragw/src/loragw_spi.native.c
32 '
33 #define SPI_SPEED          1000000
34 '
35
36 #No usr/Lora/packet_forwarder/lora_pkt_fwd/src/lora_pkt_fwd.c substituir caminho existente para
37 caminho completo:
38 #Resolve problema no arranque
39 @ Linha 982
40 /* configuration file related */
41 char *global_cfg_path= "/usr/Lora/packet_forwarder/lora_pkt_fwd/global_conf.json"; /* contain
42 global (typ. network-wide) configuration */
43 char *local_cfg_path = "/usr/Lora/packet_forwarder/lora_pkt_fwd/local_conf.json"; /* contain node
44 specific configuration, overwrite global parameters for parameters that are defined in both */
45 char *debug_cfg_path = "/usr/Lora/packet_forwarder/lora_pkt_fwd/debug_conf.json"; /* if present,
46 all other configuration files are ignored */
47
48 #Este passo pode ser efectuado posteriormente na configuração do NS de forma similar

```

```

45 #Não necessitam da compilação, portanto alteram-se em qualquer altura!
46 #Configurar a gateway para os valores necessários nos ficheiros seguintes.
47 #definir gateway ID, endereço do servidor e porto (estes 2 últimos, podem manter os valores por
48 sudo leafpad /usr/Lora/packet_forwarder/lora_pkt_fwd/global_conf.json
49 #definir gateway ID
50 sudo leafpad /usr/Lora/packet_forwarder/lora_pkt_fwd/local_conf.json
51
52 #Compilar e testar >sempre primeiro a lora gateway e depois packet forwarder >> dependencias
53 cd /usr/Lora/lora_gateway/
54 make all
55 cd /usr/Lora/packet_forwarder/
56 make all
57
58 #Testar funcionamento do envio com
59 cd /usr/Lora/lora_gateway/util_tx_test/
60 ./util_tx_test -r 1257 -f 866.5
61 #deverá devolver algo como:
62 sudo ./util_tx_test -r 1257 -f 866.5
63 Sending -1 LoRa packets on 866500000 Hz (BW 125 kHz, SF 10, CR 1, 16 bytes payload, 8 symbols
preamble) at 14 dBm, with 1000 ms between each
64 lgw_board_setconf:427: Note: board configuration; lorawan_public:1, clksrc:1
65 lgw_rxf_setconf:488: Note: rf_chain 0 configuration; en:1 freq:866500000 rssi_offset:0.000000
radio_type:2 tx_enable:1 tx_notch_freq:129000
66 lgw_rxf_setconf:488: Note: rf_chain 1 configuration; en:1 freq:866500000 rssi_offset:0.000000
radio_type:2 tx_enable:0 tx_notch_freq:129000
67 lgw_connect:532: INFO: no FPGA detected or version not supported (v0)
68 Note: success connecting the concentrator
69 lgw_setup_sx125x:407: Note: SX125x #0 version register returned 0x21
70 lgw_setup_sx125x:415: Note: SX125x #0 clock output disabled
71 lgw_setup_sx125x:469: Note: SX125x #0 PLL start (attempt 1)
72 lgw_setup_sx125x:407: Note: SX125x #1 version register returned 0x21
73 lgw_setup_sx125x:412: Note: SX125x #1 clock output enabled
74 lgw_setup_sx125x:469: Note: SX125x #1 PLL start (attempt 1)
75 lgw_start:823: Note: calibration started (time: 2300 ms)
76 lgw_start:844: Note: calibration finished (status = 191)
77 Info: Initialising AGC firmware...
78 Info: putting back original RADIO_SELECT value
79 INFO: concentrator started, packet can be sent
80 Sending packet number 1 ...INFO: tx_start_delay=1495 (1495.500000) - (1497, bw_delay=1.500000,
notch_delay=0.000000)
81 INFO: Enabling TX notch filter
82 58.a0.0.0.6a.5.1.2.0.9a.10.0.0.8.0.0.54.45.53.54.0.1.61.62.63.64.65.66.67.68.69.6a.end
83
84 #se existir erros aqui, não avançar sem os corrigir primeiro.
85 #se não existirem erros podemos desligar os debugs, mantendo apenas o HAL
86 #editar /lora_gateway/libloragw/library.cfg :
87 DEBUG_AUX= 0
88 DEBUG_SPI= 0
89 DEBUG_REG= 0
90 DEBUG_HAL= 1
91 DEBUG_LBT= 0
92 DEBUG_GPS= 0
93 #Salvar ficheiro e recompilar
94 cd /usr/Lora/lora_gateway/
95 make all
96 cd /usr/Lora/packet_forwarder/
97 make all
98
99 #Criar pasta de logs
100 cd /var/log/
101 sudo mkdir Lora
102 cd Lora
103 sudo touch packet_forwarder.log
104 sudo chmod 777 packet_forwarder.log
105
106 #adicinar no arranque
107 sudo leafpad /etc/rc.local
108 #abre o ficheiro, adicionar o caminho do encaminhador de pacotes e redirecionar stdout para o
ficheiro de log (linha 242)
109 '
110 #!/bin/sh -e
111 #
112 # rc.local
113 #
114 # This script is executed at the end of each multiuser runlevel.
115 # Make sure that the script will "exit 0" on success or any other
116 # value on error.
117 #
118 # In order to enable or disable this script just change the execution
119 # bits.
120 #
121 # By default this script does nothing.
122
123 # Print the IP address
124 _IP=$(hostname -I) || true
125 if [ "$_IP" ]; then

```

```

126     printf "\n\n\n          ***** ISE_UALG *****          \n          My IP
address is %s\n\n\n" "$_IP"
127 fi
128
129 sudo /usr/Lora/ic880-SPI_reset
130 printf "\n          ***   ic880-SPI reset   *** \n"
131 sleep 5
132
133 sudo /usr/Lora/packet_forwarder/lora_pkt_fwd/lora_pkt_fwd >> /var/log/Lora/packet_forwarder.log 2>&1
&
134 sleep 5
135
136 exit 0
137 '
138 #Reiniciar e verificar arranque no ficheiro de log.
139
140 #Se não existirem problemas configurar logrotate
141 #Indicar caminha da pasta dos logs, a periodicidade de execução(daily), o tamanho maximo do ficheiro
(10M) e os dias que ele mantem
142 #os logs (7)
143 sudo touch /etc/logrotate.d/packet_forwarder.conf
144 sudo leafpad /etc/logrotate.d/packet_forwarder.conf
145 #colar
146 /var/log/Lora/*{
147     daily
148     rotate 7
149     size 10M
150     compress
151     delaycompress
152 }
153
154 #testar com
155 sudo logrotate -d /etc/logrotate.d/packet_forwarder.conf
156
157 #Não deve ser preciso mais nada, pois ele corre logrotate de origem, mas confirmar
158 #caso não corra verificar permissões
159 /bin/chown root.root /etc/logrotate.conf
160 /bin/chmod 600 /etc/logrotate.conf /usr/sbin/logrotate -v /etc/logrotate.conf
161
162 #Caso se pretenda replicar várias gateways efectuar cópia de segurança de todo o cartão aqui.

```

G. CONFIGURAÇÃO DO SERVIDOR DE REDE NS

```

1  #Os exemplos seguintes utilizam um utilizador "iseualg" de pass "iseualg", apenas para demonstração.
2  #Utilizar outros nomes
3  #
4  #####
5  #
6  #   Servidor de rede >> No mesmo RPI da gateway
7  #
8  #####9
9  #sacar o instalador <VERSION>.deb mais recente de
10 https://github.com/gotthardp/lorawan-server/releases
11 #na pasta onde o ficheiro se encontre, instalar com:
12 dpkg -i lorawan-server-<VERSION>.deb
13
14
15 #Configurar o necessário no ficheiro JSON de configuração em
16 sudo leafpad /usr/lib/lorawan-server/releases/<VERSION>/sys.config
17 #aqui configura-se a porta do servidor, certificados de segurança e credenciais de acesso.
18
19 #Os valores da porta devem corresponder aos utilizados em
20 sudo leafpad /usr/Lora/packet_forwarder/lora_pkt_fwd/global_conf.json
21 # sendo estes os importantes: server_address, serv_port_up, serv_port_down
22
23 #Exemplo de parte do ficheiro
24 "[
25 % server configuration
26 {lorawan_server, [
27     % UDP port listening for packets from the packet_forwarder Gateway
28     {packet_forwarder_listen, [{port, 1680}]},
29     % HTTP port for web-administration and REST API
30     {http_admin_listen, [{port, 8080}]},
31     % HTTP/SSL port
32     % {http_admin_listen_ssl, [
33     %     {port, 8443},
34     %     {certfile, "cert.pem"},
35     %     {keyfile, "key.pem"}
36     % ]},
37     % default username and password for the admin interface
38     {http_admin_credentials, {<<"iseualg">>, <<"iseualg">>}},
39     % amount of rxframes retained for each device/node
40     {retained_rxframes, 50},
41     % websocket expiration if client sends no data

```

```

42     {websocket_timeout, 3600000} % ms
43   ]},
44   {os_mon, [ .....]
45 }
46 #Arrancar manualmente com lorawan server
47 sudo systemctl start lorawan-server
48
49 #listas de serviços >> verificar loraserver
50 systemctl list-unit-files
51
52 #Iniciar no arranque de forma automatica
53 sudo systemctl enable lorawan-server.service
54 sudo reboot

```

H. SERVIDOR DE APLICAÇÕES AS

```

1  #Os exemplos seguintes utilizam um utilizador "iseualg" de pass "iseualg", apenas para demonstração.
2  #Utilizar outros nomes
3  #
4  #####
5  #
6  #   Servidor de aplicações ThingsBoard
7  #
8  #####
9
10 #Instalar o RPI da mesma maneira que anteriormente explicado.
11 #Adicionar livreria para corrigir o funcionamento do touchscreen
12 sudo apt-get install xserver-xorg-input-evdev
13 #definir que o tipo de dispositivo de entrada (pointer) usa o evdev e não o libinput
14 #editar o ficheiro em
15 sudo leafpad /usr/share/X11/xorg.conf.d/40-libinput.conf
16 #alterar driver de libinput para evdev e salvar
17 '
18 Section "InputClass"
19     Identifier "libinput pointer catchall"
20     ...
21     MatchDevicePath "/dev/input/event*"
22     Driver "evdev"
23 EndSection
24 '
25
26 #Instalar servidor de base de dados postgresql
27 sudo apt-get update
28 sudo apt-get install postgresql postgresql-contrib
29 sudo service postgresql start
30
31 #adicionar user, db e pass
32 sudo -u postgres createuser --interactive
33 >>Enter name of role to add: iseualg
34 >>Shall the new role be a superuser? (y/n) y
35 sudo -u iseualg createdb thingsboard                                #criar thingsboard db
36 sudo -u iseualg psql
37 iseualg=# \password                                                #mudar password de (user)iseualg para iseualg
38
39 #Sacar thingsboard service (no exemplo está a versão 1.3.1, já desatualizada!)
40 wget https://github.com/thingsboard/thingsboard/releases/download/v1.3.1/thingsboard-1.3.1.deb
41 sudo dpkg -i thingsboard-1.3.1.deb
42 export JAVA_OPTS="$JAVA_OPTS -Dplatform=rpi -Xms256M -Xmx256M"
43
44 #descomentar postgres e comentar hsqldb em
45 sudo leafpad /etc/thingsboard/conf/thingsboard.yml
46 #adicionalmente utilizar user e pass da db postgresql
47     username: "${SPRING_DATASOURCE_USERNAME:iseualg}"
48     password: "${SPRING_DATASOURCE_PASSWORD:iseualg}"
49
50 #instalar
51 sudo /usr/share/thingsboard/bin/install/install.sh
52 #Arrancar serviço
53 sudo service thingsboard start
54
55 #Entrar em http://localhost:8080 com
56     login - sysadmin@thingsboard.org
57     password - sysadmin
58
59 #Criar tenant
60 Name: Lab160
61 email: nfpinto@ualg.pt
62
63 #Criar user no tenant
64 email: nfpinto@ualg.pt
65 #iR AO Link de ativação e definir password
66 pass: iseualg
67

```



```

68
69 #Configurar o arranque do browser chromium
70 cp /etc/xdg/lxsession/LXDE-pi/autostart /home/pi/.config/lxsession/LXDE-pi/autostart
71 sudo leafpad /home/pi/.config/lxsession/LXDE-pi/autostart
72
73 #colocar isto no ficheiro de autostart
74 @lxpanel --profile LXDE-pi
75 @pcmanfm --desktop --profile LXDE-pi
76 #@point-rpi
77 @xscreensaver -no-splash # comment this line out to disable screensaver
78 @xset s off
79 @xset -dpms
80 @xset s noblank
81 @chromium-browser --start-maximized --kiosk http://localhost:8080

```

I. SCRIPT DE LEITURA DA ESTAÇÃO METEREOLÓGICA

O *script* de leitura utiliza os módulos de Python3 *setuptools*, *wheel*, *lxml*, e *requests* para o processamento da página da estação. Este envia os dados para o AS e adicionalmente para o servidor *emoncms* local. O módulo *requests* faz a leitura da página sendo o *lxml* responsável por extrair os dados da página. Com a extração dos dados feita basta criar um *payload* com os formatos necessários para as plataformas os receberem e enviá-los. O Thingsboard recebe os dados num campo de dados dum pedido *POST* enquanto no *emoncms* os dados são enviados no *url* através de um pedido *GET*. Posteriormente gravam-se na base de dados e verificam-se as ligações de rede e estado da bateria e leituras, enviando alertas se necessário. As verificações de ligação de rede são feitas por simples *ping* ao endereço do *logger* e a um endereço externo, verificando desta forma se a falha é local ou não. Ocorrem a cada trinta minutos. Os emails são enviados utilizando o servidor de *smtp* da UALG, e consistem num simples texto indicando o problema. Os emails apenas são enviados uma vez por dia, mesmo quando o problema persiste. Estas duas periodicidades são configuradas no código. A base de dados foi criada externamente no *software* Sqlitestudio3 e consiste em tabelas simples com as variáveis lidas e a data e hora de leitura, conforme se pode ver na Figura I.1.

	ID	data	tmed	hmed	radmed	velvent	dirvent	pluvio	vbatt	patmos
191	191	2017/12/15 16:30:01.400	16.836824	87.263792	29.20524	3.776667	262.42462	0	12.28621	1.916233
192	192	2017/12/15 16:31:01.458	16.835558	87.273808	28.72825	4.003334	263.15366	0	12.28621	1.916365
193	193	2017/12/15 16:32:01.514	16.836802	87.34308	27.99932	2.006666	267.69812	0	12.28621	1.91752
194	194	2017/12/15 16:33:00.570	16.842996	87.3754	26.592294	2.743334	262.5217	0	12.28621	1.918544
195	195	2017/12/15 16:34:00.627	16.845488	87.391408	24.561934	2.77	240.1515	0	12.28621	1.9179
196	196	2017/12/15 16:35:00.685	16.846044	87.386712	21.909884	2.76	259.73624	0	12.28621	1.916993
197	197	2017/12/15 16:36:00.756	16.841448	87.324376	18.66246	3.268334	272.31784	0	12.28621	1.914816
198	198	2017/12/15 16:37:00.799	16.825026	87.2456	15.57277	2.95	260.70148	0	12.28621	1.914684
199	199	2017/12/15 16:38:00.857	16.81901	87.301272	13.30231	2.935	254.16918	0	12.28621	1.912743
200	200	2017/12/15 16:39:00.915	16.80649	87.256632	11.9991	1.958333	254.0781	0	12.28621	1.911721
201	201	2017/12/15 16:40:00.971	16.798698	87.19604	11.462624	3.48	250.9518	0	12.28621	1.911211
202	202	2017/12/15 16:41:01.030	16.791616	87.211712	11.510972	2.46	253.19486	0	12.28621	1.911576
203	203	2017/12/15 16:42:01.088	16.795334	87.244056	12.25377	1.776667	256.49804	0	12.28621	1.912494

Figura I.1 - Excerto de conteúdo da base de dados SQLite

Adicionalmente a este são guardados dois ficheiros que contêm a hora POSIX de envio do último email e da última verificação do estado da estação. Estes são abertos na execução do *script* e garantem a periodicidade dos verificações e emails, evitando um *spam* de emails, de minuto a minuto no caso de alguma falha. É também criado um ficheiro de *log* na mesma pasta, para onde os resultados de cada execução são enviados, muito útil para *debug* de erros após o agendamento estar a correr. O agendamento é feito no **Crontab** de Linux com:

Crontab -e

Acrescentar a seguinte linha no ficheiro **Crontab**, com os caminhos corretos para os *scripts* e *log*:

```
*/* * * * * /usr/bin/python3 /home/user/Documentos/dataTakerLog/
MeteoEmonDBv121.py >> /home/user/Documentos/dataTakerLog/Error.log 2>&1
```

Com todos os ficheiros (base de dados, variáveis e *logs*) na pasta e o agendamento criado, teremos um sistema de monitorização e armazenamento de variáveis ambientais, funcional e muito útil para funções de previsão das condições meteorológicas, as quais permitem otimizar os escalonamentos de cargas, dependentes de fontes renováveis solares ou eólicas.

O código fonte, presente no ficheiro **MeteoEmonDBv121.py** é o seguinte:

```
1 ## Nelson Pinto nfpinto@ualg.pt
2 ## Python 3.6
3 ## V1.1 Fazer loop de 1m a 1m via cron;
4 ## Envio de dados ao emon e base de dados local. Links e base de dados configuraveis nos campos de
Variaveis
5 ## TODO- Converter strings em floats, antes do envio para a base de dados, se houver necessidade -
Comportamento SQLITE3 >>https://sqlite.org/datatype3.html
6 ## V1.2 02/01/2018
7 ## Acrescentada escrita na base de dados;
8 ## apikey errada no emonurl local > estava com a conta agerar@agerar d90a6f9a6bfc4d44931e69021ca6bbe;
9 ## Substituido para conta ualg@_____ fbede4619d3abcd3adee909c1501fc6b ; Conta externa OK
10 ## V1.2.1 03/01/2018
11 ## Criação dinamica do URL, para permitir várias contas, nós e hosts; Definida função para obter dados
de um url e colocar dados num dict temporario
12 ## Escrita nas bases de dados, tabelas de geracao e consumo;
13 ## V1.2.2 30/01/2018
14 ## Added timeout in requests (3 s)
15 ## TODO Add a warning if there's something wrong with the readings
16 ## V2.0 14/02/2018
17 ## Problema dos timeouts não resolvido, ocorreu este fds mesmo após timeouts
18 ## Acrescentado stream=false Confirmar se funciona
19 ## Implementado tudo numa função, com aviso para timeout/erros no request.get
20 ## V2.1 19/02/2018
21 ## Adicionada publicação da estacao meteo ao thingsboard
22 ## V2.1.1 20/02/2018
23 ## Valores convertidos em floant antes do envio para tb - Não faz medias dos valores pois vao em
string. Adicionados headers corretos
24 ## Alterada formatacao dos prints de time para facilitar debugs - Todos em formato : 2018-02-20
14:23:04.294457
25 ## Adicionados timeouts em todos os requests - Apenas o da estacao meteo dá chatisses mas ficam logo em
todos
26 ## if page.code != 200 alterado para dentro do try
27 ## 2.1.2 19/03/2018
28 ## Corrigido time no send mail timeout mas mantem-se erro da conection refused apos uns dias
29 ## Adicionada envio e antispam de mail de timeout, agora funcional
30 ## 2.1.3 26/03/2018
31 ## Adicionado contador de leituras para verificar ao fim de quantas ele dá o erro de timeout
32
33
34 from lxml import html
35 from time import mktime
36 from email.message import EmailMessage
37 import requests
38 import json
39 import datetime
40 import sqlite3
```

```

41 import sys
42 import smtplib
43 import os
44 import time
45 from datetime import datetime
46
47
48 ###Variaveis. agerar@agerar (emonurl) e ualg@qwertyui (emonurl2)
49 datatakerlink = 'http://10.10.140.141/html/channels.shm'
50
51 #ADD v1.2.1
52 emon_local_url = 'http://10.11.143.130/emoncms/input/post.json?'
53 emon_online_url = 'http://emoncms.org/input/post.json?'
54 emon_online_get_data_url = 'https://emoncms.org/input/get/'
55 emon_local_apikey_ualg = '&apikey=fbede4619d3abcd3adee909c1501fc6b'
56 emon_local_apikey_agerar = '&apikey=d90a6f9a6bfc4d44931e69021ca6bbee'
57 emon_online_apikey_ualg = '&apikey=6bbf33f500278de5a9921f536c073e95'
58 node_name_geracao = 'Geracao'
59 node_name_consumo51 = 'Consumo51'
60 node_name_datataker = 'node=dataTaker'
61
62 #ADD V2.1
63 thingsboard_url_datataker = "http://10.11.143.13:8080/api/v1/agerar_datataker/telemetry"
64
65 path_to_db = '/home/janio/Documents/dataTakerLog/datatakerdb.db'
66 #path_to_db = 'C:\sqlite\datatakerdb.db' #@db teste no windows
67 table_meteo = 'datataker'
68 table_geracao = 'geracao'
69 table_consumo51 = 'consumo51'
70
71 #variaveis para alertas
72 recipients1 = 'nfpinto@ualg.pt'
73 recipients = 'nfpinto@ualg.pt' + ', ' + 'jmmonte@ualg.pt'
74 #recipients = 'nfpinto@ualg.pt' + ', ' + 'jmmonte@ualg.pt' + ', ' + 'cmsantos@ualg.pt'
75 loggerIp = '10.10.140.141'
76 testHost = 'google.pt'
77 loggerWebPage = 'http://10.10.140.141/html/channels.shm'
78 smtpHost = 'smtp.ualg.pt'
79 mail_periodicity = 86400
80 #mailPeriodicity = 120 #for debug only
81 pathToTemp_mail = '/home/janio/Documents/dataTakerLog/warningemail'
82 pathToTemp_check = '/home/janio/Documents/dataTakerLog/warningcheck'
83 pathToRead_Count = '/home/janio/Documents/dataTakerLog/readcount'
84
85 timeout_value = 3
86 check_time = 1800 #30 minutos
87
88
89
90
91 #####
92 #
93 #   Funções
94 #
95 #####
96
97 #Escreve na tabela table da base de dados sqlite3 em path. Em caso de erros, envia estes para o log
98 def database_write(data_type):
99     #Se der algum erro descomentar as 2 linhas seguintes
100     #global geracao
101     #global consumo51
102
103     try:
104         #Inicia ligação à base de dados
105         conn = sqlite3.connect(path_to_db)
106         c = conn.cursor()
107         try:
108             #Tenta inserir os valores na tabela table da bd, efetuando o commit e terminando a ligação logo de
109             #seguida
110             if data_type == 'datataker':
111                 c.execute("INSERT INTO %s VALUES (NULL, ?, ?, ?, ?, ?, ?, ?, ?)" % table_meteo , (data[0],
112                 tmed[0], hmed[0], radmed[0], velvent[0], dirvent[0], pluvio[0], vbatt[0], patmos[0] ) )
113             if data_type == 'geracao':
114                 #Versão com temperatura do painel
115                 c.execute("INSERT INTO %s VALUES (NULL, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)" %
116                 table_geracao
117                 , (geracao["timestamp"], geracao["I_ACL1"], geracao["I_ACL2"], geracao["I_ACL3"], geracao["I_S1_inv"], geracao["
118                 I_S2_inv"],
119                 geracao["P_ACL1"], geracao["P_ACL2"], geracao["P_ACL3"], geracao["P_S1_inv"], geracao["P_S2_inv"], geracao["V_A
120                 CL1"],
121                 geracao["V_ACL2"], geracao["V_ACL3"], geracao["V_S1_inv"], geracao["V_S2_inv"], geracao["total_energy_produced
122                 _inv"], geracao["panel_temperature"] ) )
123             if data_type == 'consumo51':
124                 c.execute("INSERT INTO %s VALUES (NULL, ?, ?, ?, ?, ?, ?, ?, ? , ?, ?, ?, ?, ?, ?)" %
125                 table_consumo51

```

```

, (consumo51["timestamp"], consumo51["I1"], consumo51["I2"], consumo51["I3"], consumo51["V12"], consumo51["V23"]
, consumo51["V31"],
118
consumo51["V1N"], consumo51["V2N"], consumo51["V3N"], consumo51["P"], consumo51["Q"], consumo51["S"],
119
consumo51["PF"], consumo51["F"], consumo51["EPi"], consumo51["EPe"], consumo51["EQi"], consumo51["EQe"] ) )
##FINALIZAR
120     conn.commit()
121     conn.close()
122     #print("Escrito na base de dados!")
123     except sqlite3.Error as e:
124         print("Não foi possível escrever na base de dados! Erro: ", e)
125 except sqlite3.Error as e :
126     print("Não foi possível encontrar a base de dados em %s . Erro : %s "% (path, e) )
127 return
128
129
130def emon_send(url):
131     #Só faz prints se o código for diferente de 200, minimiza os logs do crontab
132     #envia ao emon
133     try:
134         r = requests.get(url, timeout=timeout_value)
135         if r.status_code != 200 :
136             print(url)
137             print(r.status_code)
138             print(r.content)
139     except:
140         print("Sem conexão à página !" + url)
141
142     return
143
144def thingsboard_send(url, payload):
145     try:
146         r = requests.post(url, data=json.dumps(payload), headers={'Content-Type': "application/json"},
147         timeout=10)
148         if r.status_code != 200 :
149             print(url)
150             print(r.status_code)
151             print(r.content)
152     except:
153         print("Sem conexão à página !" + url)
154
155     return
156
157def emon_get(url):
158     tmp_dict = {}
159     try:
160         page = requests.get(url, timeout=timeout_value)
161         if page.status_code != 200 :
162             print("Não foi possível aceder ao site!")
163             print(url)
164             print(page.status_code)
165             print(page.content)
166     except:
167         print("Sem conexão à página: " + url)
168
169     get_dict = json.loads(page.content.decode('utf-8'))
170     for key in get_dict:
171         tmp_dict[key] = get_dict[key]['value']
172
173     return tmp_dict
174
175
176
177def emon_local_get(url, node):
178     tmp_dict = {}
179     try:
180         page = requests.get(url, timeout=timeout_value)
181         if page.status_code != 200 :
182             print("Não foi possível aceder ao site!")
183             print(url)
184             print(page.status_code)
185             print(page.content)
186     if page.status_code == 200:
187         get_dict = json.loads(page.content.decode('utf-8'))
188         for key in get_dict:
189             if key["nodeid"] == node: #Enviar geracao ou Consumo51, verificar se é algum deles e tirar o
190                 valor
191                 #Geracao
192                 if key["name"] == "I_ACL1":
193                     tmp_dict["I_ACL1"] = key["value"]
194                     tmp_dict["timestamp"] = key["time"]
195                     if key["name"] == "I_ACL1":tmp_dict["I_ACL1"] = key["value"]
196                 if key["name"] == "I_ACL2":tmp_dict["I_ACL2"] = key["value"]
197                 if key["name"] == "I_ACL3":tmp_dict["I_ACL3"] = key["value"]
198                 if key["name"] == "I_S1_inv":tmp_dict["I_S1_inv"] = key["value"]
199                 if key["name"] == "I_S2_inv":tmp_dict["I_S2_inv"] = key["value"]

```

```

199     if key["name"] == "P_ACL1":tmp_dict["P_ACL1"] = key["value"]
200     if key["name"] == "P_ACL2":tmp_dict["P_ACL2"] = key["value"]
201     if key["name"] == "P_ACL3":tmp_dict["P_ACL3"] = key["value"]
202     if key["name"] == "P_S1_inv":tmp_dict["P_S1_inv"] = key["value"]
203     if key["name"] == "P_S2_inv":tmp_dict["P_S2_inv"] = key["value"]
204     if key["name"] == "V_ACL1":tmp_dict["V_ACL1"] = key["value"]
205     if key["name"] == "V_ACL2":tmp_dict["V_ACL2"] = key["value"]
206     if key["name"] == "V_ACL3":tmp_dict["V_ACL3"] = key["value"]
207     if key["name"] == "V_S1_inv":tmp_dict["V_S1_inv"] = key["value"]
208     if key["name"] == "V_S2_inv":tmp_dict["V_S2_inv"] = key["value"]
209     if key["name"] == "V_S1_inv":tmp_dict["V_S1_inv"] = key["value"]
210     if key["name"] == "total_energy_produced_inv":tmp_dict["total_energy_produced_inv"] =
key["value"]
211     if key["name"] == "panel_temperature":tmp_dict["panel_temperature"] = key["value"]
212
213     #Consumo51
214     if key["name"] == "I1":
215         tmp_dict["I1"] = key["value"]
216         tmp_dict["timestamp"] = key["time"]
217     if key["name"] == "I2":tmp_dict["I2"] = key["value"]
218     if key["name"] == "I3":tmp_dict["I3"] = key["value"]
219     if key["name"] == "V12":tmp_dict["V12"] = key["value"]
220     if key["name"] == "V23":tmp_dict["V23"] = key["value"]
221     if key["name"] == "V12":tmp_dict["V12"] = key["value"]
222     if key["name"] == "V1N":tmp_dict["V1N"] = key["value"]
223     if key["name"] == "V2N":tmp_dict["V2N"] = key["value"]
224     if key["name"] == "V3N":tmp_dict["V3N"] = key["value"]
225     if key["name"] == "P":tmp_dict["P"] = key["value"]
226     if key["name"] == "Q":tmp_dict["Q"] = key["value"]
227     if key["name"] == "S":tmp_dict["S"] = key["value"]
228     if key["name"] == "PF":tmp_dict["PF"] = key["value"]
229     if key["name"] == "F":tmp_dict["F"] = key["value"]
230     if key["name"] == "EPi":tmp_dict["EPi"] = key["value"]
231     if key["name"] == "EPe":tmp_dict["EPe"] = key["value"]
232     if key["name"] == "EQi":tmp_dict["EQi"] = key["value"]
233     if key["name"] == "EQe":tmp_dict["EQe"] = key["value"]
234 except:
235     print("Sem conexão à página: " + url)
236
237 return tmp_dict
238
239
240
241def send_mail(msg1):
242     try:
243         #Envia mensagem pelo smtp da ualg
244         s = smtplib.SMTP(smtpHost)
245         s.send_message(msg1)
246     except smtplib.SMTPException as e:
247         print("Erro no envio do mail smtp: ", e)
248         s.quit()
249
250     return
251
252#####Envia email de aviso para valor 0 ou ---
253def send_mail_inv(vbatt):
254
255     #cria mensagem com texto plano e html
256     msg = EmailMessage()
257     msg['Subject'] = '[ERRO DATALOGGER ISE] Valor de tensão de bateria do datalogger ISE inválido => ' +
str(vbat)
258     msg['From'] = 'nfpinto@ualg.pt'
259     msg['To'] = recipients
260     msg.preamble = 'Um erro ocorreu!'
261     msg.set_content("""
262     Olá!
263
264     Ocorreu um erro na leitura de tensão de bateria do datalogger do ISE.
265     O valor lido foi "" + str(vbatt) + ""
266     http://10.10.140.141/html/channels.shm
267
268
269     Verifique o problema!
270
271     """)
272
273     msg.add_alternative("""\
274     <html>
275     <head></head>
276     <body>
277     <p>Olá!</p>
278     <p>Ocorreu um erro na
279     <a href="http://10.10.140.141/html/channels.shm">
280     leitura de tensão de bateria
281     </a> do datalogger do ISE.
282     </p>
283     <p>O valor lido foi "" + str(vbatt) + "" </p>
284     <p>Verifique o problema!</p>

```

```

285 </body>
286 </html>
287 """ , subtype='html')
288
289 send_mail(msg)
290
291 return
292
293#####Envia email de aviso para valor de tensão inferior a 8 V
294def send_mail_min(vbatt):
295
296 #cria mensagem com texto plano e html
297 msg = EmailMessage()
298 msg['Subject'] = '[ERRO DATALOGGER ISE] Tensão de bateria do datalogger ISE abaixo de 8 Volt => ' +
str(vbat[0]) + ' Volt'
299 msg['From'] = 'nfpinto@ualg.pt'
300 msg['To'] = recipients
301 msg.preamble = 'Um erro ocorreu!'
302 msg.set_content("""
303 Olá!
304
305 A leitura de tensão de bateria, no datalogger do ISE, encontra-se abaixo dos 8 Volt.
306 O valor lido foi de "" + str(vbatt[0]) + "" Volt.
307 http://10.10.140.141/html/channels.shm
308
309
310 Verifique o problema e substitua bateria se necessário!
311
312 """)
313 msg.add_alternative("""\
314 <html>
315 <head></head>
316 <body>
317 <p>Olá!</p>
318 <p>A leitura de
319 <a href="http://10.10.140.141/html/channels.shm">
320 tensão de bateria, no datalogger do ISE,
321 </a> encontra-se abaixo dos 8 Volt.
322 </p>
323 <p>O valor lido foi de "" + str(vbatt[0]) + "" Volt.</p>
324 <p>Verifique o problema e substitua bateria se necessário!</p>
325 </body>
326 </html>
327 """, subtype='html')
328
329 send_mail(msg)
330
331 return
332
333def send_mail_not_connected():
334
335 #cria mensagem com texto plano e html
336 msg = EmailMessage()
337 msg['Subject'] = '[ERRO DATALOGGER ISE] Sem conexão ao datalogger ISE'
338 msg['From'] = 'nfpinto@ualg.pt'
339 msg['To'] = recipients
340 msg.preamble = 'Um erro ocorreu!'
341 msg.set_content("""
342 Olá!
343
344 Não foi possível ligar ao datalogger. Um ping foi feito ao endereço "" + loggerIp + "" , em "" +
str(datetime.datetime.now()) + "" sem sucesso.
345
346 Verifique o estado da ligação.
347
348 """)
349
350 msg.add_alternative("""\
351 <html>
352 <head></head>
353 <body>
354 <p>Olá!</p>
355 <p>Não foi possível ligar ao datalogger. Um ping foi feito ao endereço "" + loggerIp + "" , em
"" + str(datetime.datetime.now()) + "" sem sucesso. </p>
356 <p>Verifique o estado da ligação.</p>
357 </body>
358 </html>
359 """, subtype='html')
360
361 send_mail(msg)
362
363 return
364
365
366def send_mail_timeout(excpt):
367
368 #cria mensagem com texto plano e html
369 msg = EmailMessage()

```

```

370 msg['Subject'] = '[ERRO DATALOGGER ISE] Timeout'
371 msg['From'] = 'nfpinto@ualg.pt'
372 msg['To'] = recipients1
373 msg.set_content("""
374 Olá!
375
376 Não foi possível ligar ao datalogger. TIMEOUT "" + loggerIp + "" , em "" + str(time.time()) +
377 "" + str(excpt) + ""
378
379 Verifique o estado da ligação.
380
381 """)
382
383 msg.add_alternative("""\
384 <html>
385 <head></head>
386 <body>
387 <p>Olá!</p>
388 <p>Não foi possível ligar ao datalogger. TIMEOUT "" + loggerIp + "" , em "" + str(time.time()) +
389 "" . </p>
389 <p>"" + str(excpt) + ""</p>
390 <p>Verifique o estado da ligação.</p>
391 </body>
392 </html>
393 """, subtype='html')
394
395 send_mail(msg)
396
397 return
398
399
400#####
401#
402#   Main
403#
404#####
405
406#hora de inicio do script
407timenow = time.time()
408
409#Ficheiro com timestamp da ultima verificação
410chk = open(pathToTemp_check, 'r+')
411last_check = float(chk.readline())
412read_count_tmp = open(pathToRead_Count, 'r+')
413read_count = float(read_count_tmp.readline())
414
415#Tenta aceder ao site, caso não consiga ligar por falta de ligação termina execução
416#V2 add stream=False
417try:
418     page = requests.get(datatakerlink, timeout=timeout_value, stream=False)
419     valid_readings = True
420     #Alterado para aqui na v2.1.1: Tenta evitar aceder a page se existir excepção
421     #Caso ligue mas a resposta do acesso à página seja diferente de 200, faz print para debug, termina
422     #execução para não enviar valores errados
423     if page.status_code != 200:
424         print("Erro a aceder ao site!")
425         print(str(datetime.now()))
426         print(url)
427         print(page.status_code)
428         print(page.content)
429         valid_readings = False
430     #ADD 2.1.3
431     read_count += 1
432     read_count_tmp.seek(0)
433     read_count_tmp.write(str(read_count) )
434 except Exception as e:
435     last_mail = open(pathToTemp_mail, 'r+')
436     lastmail = float(last_mail.readline())
437     if (timenow - lastmail) > mail_periodicity:
438         try:
439             send_mail_timeout(e)
440             print("DEBUG mail enviado por timeout. Tempo entre o ultimo email: " + str(timenow -
441 lastmail) )
442             last_mail.seek(0)
443             last_mail.write(str(timenow) )
444         except Exception as e:
445             print("DEBUG Não enviou email de timeout: ", e)
446     last_mail.close()
447     print("##### Sem conexão à página !
448 #####")
449     print(str(datetime.now()))
450     print("Não foi possível aceder ao site devido a ")
451     print(e)
452     valid_readings = False
453     #tentar fechar a pagina antes de sair #ADD 7 02 2018

```

```

452 #Verificar comportamento na v2.1.1 >> Se No futuro der erro de page apagar o page.close >> em caso
de timeout ele não tem page e dá erro
453 #page.close()
454
455#V2 se valores são válidos atualiza variaveis e envia dados
456if valid_readings:
457 #Parsing da página web e recolha das variaveis da tabela
458 #Acrescentar aqui mais valores se necessário
459 tree = html.fromstring(page.content)
460 data = tree.xpath('/html/body/table[4]/tbody/tr[33]/td[5]/text()')
461 tmed = tree.xpath('/html/body/table[4]/tbody/tr[35]/td[3]/text()')
462 hmed = tree.xpath('/html/body/table[4]/tbody/tr[38]/td[3]/text()')
463 radmed = tree.xpath('/html/body/table[4]/tbody/tr[40]/td[3]/text()')
464 velvent = tree.xpath('/html/body/table[4]/tbody/tr[42]/td[3]/text()')
465 dirvent = tree.xpath('/html/body/table[4]/tbody/tr[44]/td[3]/text()')
466 pluvio = tree.xpath('/html/body/table[4]/tbody/tr[45]/td[3]/text()')
467 patmos = tree.xpath('/html/body/table[4]/tbody/tr[13]/td[3]/text()')
468 vbatt = tree.xpath('/html/body/table[4]/tbody/tr[55]/td[3]/text()')
469 #ADD 7 02 2018
470 page.close()
471
472 #Processar data para emon >> só recebe em posix
473 #d = datetime.
474 d = datetime.strptime(data[0], '%Y/%m/%d %H:%M:%S.%f')
475 dataemon = mktime(d.timetuple())
476
477 #dict com todos os valores que queremos para emon
478 payload = {"Tmed": tmed[0], "Hmed": hmed[0], "Radmed": radmed[0], "Velvent": velvent[0], "Dirvent":
dirvent[0], "Pluvio": pluvio[0], "Patmos": patmos[0], "Vbatt": vbatt[0], "dataemon": dataemon }
479 #ADD V2.1.1 convert to float
480 payload_thingsboard = {"Tmed":float(tmed[0]), "Hmed":float(hmed[0]), "Radmed":float(radmed[0]),
"Velvent":float(velvent[0]), "Dirvent":float(dirvent[0]), "Pluvio":float(pluvio[0]),
"Patmos":float(patmos[0]), "Vbatt":float(vbatt[0]) }
481
482 #ADD V2.1
483 thingsboard_send(thingsboard_url_datataker, payload_thingsboard)
484
485 #Escreve na tabela dos dados da estacao meteorológica e envia ao emon
486 database_write("datataker")
487
488 #ADD V1.2.1 Criar os url e enviar para a função
489 emon_local = emon_local_url + node_name_datataker + emon_local_apikey_ualg + "&json="
490 emon_local += json.dumps(payload, sort_keys=True)
491 emon_send(emon_local)
492
493 #2.1.4 Deixa de enviar, serviços pagos
494 #emon_online = emon_online_url + node_name_datataker + emon_online_apikey_ualg + "&json="
495 #emon_online += json.dumps(payload, sort_keys=True)
496 #emon_send(emon_online)
497
498 ##ADD 1.2.1 cria função para obter o conteudo da pagina
499 #Obtem dados da geracao, guarda num dic geracao e envia para bd
500 #emon_get_online = emon_online_get_data_url + node_name_geracao + emon_online_apikey_ualg
501 emon_get_local = emon_local_url + emon_local_apikey_ualg
502 geracao = emon_local_get(emon_get_local, "Geracao")
503 database_write("geracao")
504
505 #Obtem dados do consumo
506 #emon_get_online = emon_online_get_data_url + node_name_consumo51 + emon_online_apikey_ualg
507
508 consumo51 = emon_local_get(emon_get_local, "Consumo51")
509 database_write("consumo51")
510
511
512
513#Se já passaram mais que check_time segundos
514if (timenow - last_check) > check_time:
515 #print(str(timenow))
516 #Testa conexão ping ao datalogger, se falhar tenta a outro host. se conseguir ligar envia mail,
caso contrário termina execução
517 #Se houver conexão ao datalogger, lê bateria
518 #Ficheiro com timestamp do ultimo email
519 last_mail = open(pathToTemp_mail, 'r+')
520 lastmail = float(last_mail.readline())
521
522 #ping ao datalogger, se não pingar tenta testhost(está o google) de forma a perceber
523 #que a falha é na ligação do datalogger. Se pingar ao google envia mail. Se não pingar vai para o
log
524 #Atenção que o log fica com os resultados dos ping, como tal implementar logrotate se a
periodicidade for curta
525 test_conn_local = os.system("ping -q -c 1 " + loggerIp ) #Linux
526 if test_conn_local != 0:
527 #print("Sem conectividade ao datalogger. Ping feito a " + loggerIp)
528 test_conn_global = os.system("ping -q -c 1 " + testHost ) #Linux
529 if test_conn_global != 0:
530 #Sem ligação à rede >> envia para log
531 #print("Sem conectividade à rede global. Ping feito a " + testHost + ". Sem possibilidade
de enviar email devido a falha de rede.")

```



```

532     else:
533         print("Com conectividade à rede global. Ping feito a " + testHost)
534         if (timenow - lastmail) > mail_periodicity:
535             try:
536                 send_mail_not_connected()
537                 print("DEBUG mail enviado por falta de conectividade. Tempo entre o ultimo email: " +
+ str(timenow - lastmail) )
538                 last_mail.seek(0)
539                 last_mail.write(str(timenow) )
540             except Exception as e:
541                 print("DEBUG Não enviou email: ", e)
542
543     if valid_readings:
544         if (vbatt == ['---']) or ( vbatt == ['0'] ):
545             print("Erro! Valor inválido: ", vbat)
546             if (timenow - lastmail) > mail_periodicity:
547                 try:
548                     send_mail_inv(vbatt)
549                     print("DEBUG mail enviado por valor invalido. Tempo entre o ultimo email: " +
str(timenow - lastmail) )
550                     last_mail.seek(0)
551                     last_mail.write(str(timenow) )
552                 except Exception as e:
553                     print("DEBUG Não enviou email de valor inválido: ", e)
554
555             elif (float(vbatt[0]) < 8):
556                 print("Valor de tensão crítico: ", vbat)
557                 if (timenow - lastmail) > mail_periodicity:
558                     try:
559                         send_mail_min(vbatt)
560                         print("DEBUG mail enviado por tensão abaixo minimo. Tempo entre o ultimo email: " +
str(timenow - lastmail) )
561                         last_mail.seek(0)
562                         last_mail.write(str(timenow) )
563                     except Exception as e:
564                         print("DEBUG Não enviou email de tensão abaixo minimo: ", e)
565
566     last_mail.close()
567     chk.seek(0)
568     chk.write(str(timenow) )
569
570#fecha ficheiro com tempo de check
571chk.close()
572read_count_tmp.close()
573#linha no ficheiro de log
574print("---")
575if valid_readings:
576     print(str(datetime.now())+ " OK ; Cnt = " + str(read_count) )
577sys.exit()
578

```

J. MQTT

O *message queieuing telemetry transport* (MQTT) [172] é um protocolo de comunicação *open source* normalizado (ISO/IEC 20922), desenvolvido pela IBM de forma a ser eficiente em largura de banda e consumo, sendo ideal para dispositivos de baixo consumo. Baseia-se numa arquitetura de publicação/subscrição de mensagens, as quais são por norma curtas e com um cabeçalho de apenas 2 *bytes*. O software que corre o protocolo também é pequeno, assíncrono e baseado em TCP, sendo assim facilmente utilizável em dispositivos com constrangimentos de memória, tais como microcontroladores. Ao utilizar TCP, cumpre-se o requisito LoraWAN de comunicação NS-AS. Com estas características tornou-se popular em redes de sensores WSN e aplicações IoT/M2M, como método de enviar e receber dados da camada de aplicação. Ocupa as camadas de sessão, apresentação e aplicação do modelo OSI.

O funcionamento do protocolo assenta na simplicidade. É composto por clientes MQTT e *broker* MQTT, conforme representado na Figura J.1. Este *broker*, equiparado a um servidor,

está encarregue de encaminhar mensagens entre os diferentes clientes. Um dos mais conhecidos é o Mosquitto, disponível para variados sistemas operativos. Os clientes MQTT são obviamente dispositivos que enviam e recebem dados. As mensagens identificam-se pelos chamados tópicos, que são um tipo de canal de informação, e que permite ao *broker* efetuar o encaminhamento correto das mensagens.

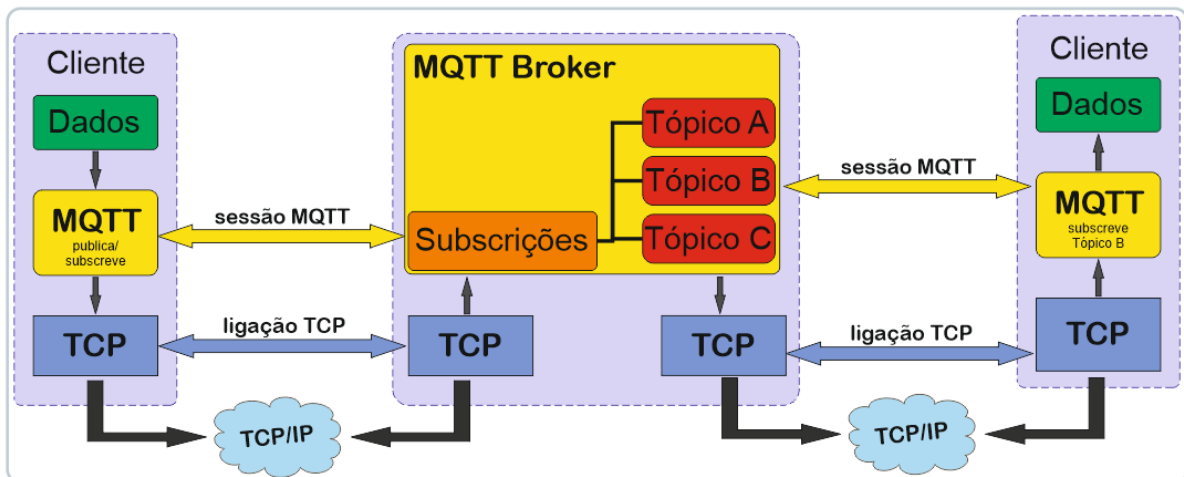


Figura J.1 - Arquitetura típica MQTT

Os clientes que queiram receber dados devem subscrever no *broker* o respetivo tópicos, de forma a receber todos os dados que surjam nesse mesmo tópicos. No caso de dispositivos que pretendam enviar dados, devem publicá-los para o *broker* num determinado tópicos. Esta publicação é feita quando for necessário, de forma assíncrona, não necessitando desta forma de nenhum mecanismo de sincronismo entre os dispositivos. Por exemplo, um sensor publica os seus dados num tópicos enviado ao *broker*. Quem desejar obter esses dados, deve subscrever esse mesmo tópicos no *broker*, ficando o *broker* encarregue de, quando recebe mensagens, encaminhá-las aos diferentes subscritores. Neste método os clientes apenas conhecem os tópicos, não sabendo mais nada de outros clientes, incluindo de quem criou a mensagem. A Figura J.2 demonstra este processo.

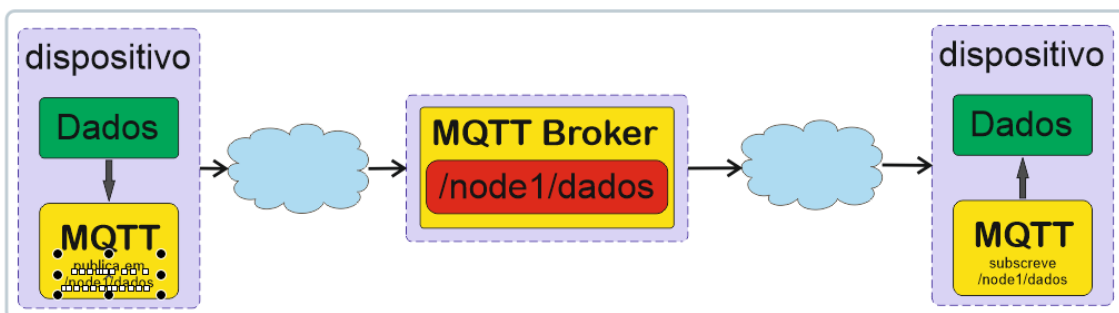


Figura J.2 - Publicação e subscrição de dados em “/node1/dados”

Um tópicos é identificado por apenas uma *string*, podendo ter várias hierarquias separadas por uma barra, como por exemplo: *Casa/Sensor1/Temperatura*. Neste tópicos estaríamos a referir-nos à temperatura do sensor 1 de casa (Figura J.3). No caso desse sensor publicar mais tópicos, poderíamos subscrever todos utilizando o caracter # (o chamado *multilevel wildcard*), utilizando o tópicos *Casa/Sensor1/#*. Ou poderíamos subscrever um tópicos para obter apenas as temperaturas de todos os sensores de casa, utilizando o caracter + (*single level wildcard*) como *Casa/+/Temperatura*. A Tabela J.1 resume estes exemplos.

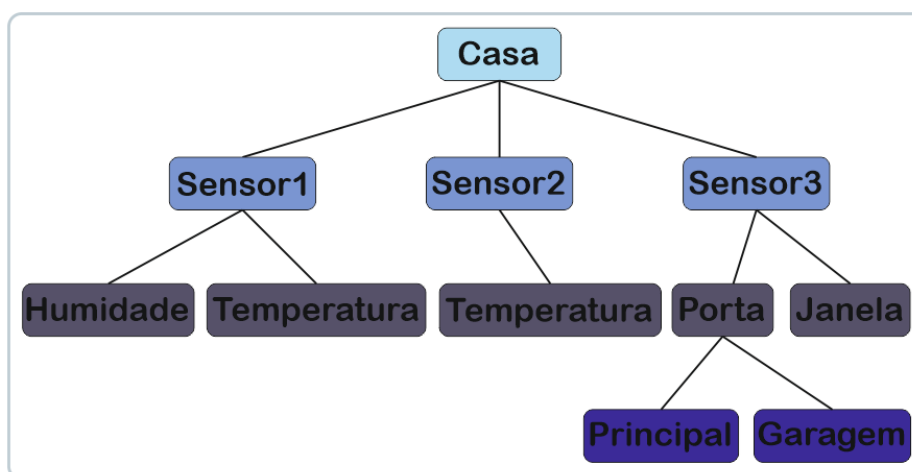


Figura J.3 - Exemplo de estrutura de tópicos, com várias hierarquias

	Caracter	Exemplo de subscrição	Tópicos obtidos
Nível único	+	<i>Casa/+/Temperatura</i>	<i>Casa/Sensor1/Temperatura</i> <i>Casa/Sensor2/Temperatura</i>
Multi nível	#	<i>Casa/Sensor3/#</i>	<i>Casa/Sensor3/Janela</i> <i>Casa/Sensor3/Porta/Garagem</i> <i>Casa/Sensor3/Porta/Principal</i>

Tabela J.1 - Exemplo de wildcards e tópicos MQTT

O protocolo inclui três níveis de qualidade de serviço (QoS), que podem ser implementados consoante a necessidade. O nível 0 envia as mensagens, mas não garante entrega, numa tipologia *best effort*. É o mais leve a nível de *overheads* adicionais, sendo o que consome menos largura de banda. O nível 1 garante que a mensagem é entregue pelo menos uma vez. No entanto podem existir mensagens duplicadas. O nível 2 garante que a mensagem é entregue apenas uma única vez. Este é o mais pesado em termos de largura de banda, uma vez que os cabeçalhos necessários são maiores. Ao mesmo tempo exige reter a mensagem nos

dispositivos algum tempo. De forma a garantir a segurança dos dados é possível utilizar ligações seguras SSL/TLS e encriptação nos dados enviados. Uma vez que o protocolo não lida com os dados e apenas os encaminha, estes podem ter qualquer representação (inteiros, vírgula flutuante, *bytes*, ...), sendo transparente para o MQTT.

As conexões iniciam-se quando um cliente envia uma mensagem de **CONNECT** ao *broker* o qual deve responder com um *acknowledge* **CONNACK**, contendo informações sobre a aceitação ou não do pedido. Este passo estabelece uma sessão que será mantida até o cliente enviar um pedido **DISCONNECT**. Com a sessão estabelecida o cliente pode subscrever tópicos utilizando **SUBSCRIBE** ou publicar dados com **PUBLISH**. Se não quiser receber mais dados de determinado tópico basta enviar um comando **UNSUBSCRIBE** indicando o tópico. Os comandos de subscrição recebem sempre confirmação por parte do *broker* **SUBACK** e **UNSUBACK**. As confirmações de publicações **PUBACK** só acontecem quando se utiliza os níveis de QoS 1 ou 2.

K. ENVIO DE MENSAGENS PARA NÓS LORAWAN (DOWNLINK)

Para enviar dados aos nós (mensagens *downlink*), devemos enviar uma mensagem via MQTT do SS para o NS, com determinadas informações. De forma a que o NS encaminhe as mensagens aos nós LoraWAN corretos, estes devem ser identificados pelo seu DevEUI, presente no corpo da mensagem ou no tópico de publicação. Por exemplo, para enviar os dados “01”, os quais podem significar ativar uma saída do nó “70B3D54901234567”, basta apenas contruir uma mensagem JSON com os campos “*data*” e “*deveui*”, e publicá-la no tópico respetivo:

```
{"deveui":"70B3D54901234567","data":"01"}
```

Os dados de utilizador são enviados dentro do campo “*data*”, sendo que este deverá conter informação que o nó interprete, não existindo nenhuma restrição. Pode inclusive ter variadas funções para um mesmo nó, sendo estas funcionalidades definidas no *software*.

Deve-se ter também em consideração a classe dos nós, uma vez que o NS nada sabe da mesma. Nós de classe C exigem que no corpo da mensagem exista o campo “*time*” e “*port*”. O “*port*” identifica apenas um porto de receção enquanto o “*time*” diz ao NS quando deve enviar a mensagem, se imediatamente ou numa data definida. Nós de classe A não necessitam destes campos, uma vez que a mensagem fica retida no NS até que este receba um *uplink* do nó, enviando então a mensagem na primeira janela de receção livre. Por exemplo, para enviar

a mensagem não confirmada “123” imediatamente, ao nó “70B3D54901234567”, o SS deve publicar no tópico escolhido a seguinte mensagem:

```
{"deveui":"70B3D54901234567", "data":"123", "port":2, "time":"immediately"}
```

Se pretendermos agendar a mensagem para as 20:01:10 horas do dia 1/02/2018 devemos enviar a mensagem:

```
{"deveui":"70B3D54901234567", "data":"123", "port":2, "time":"2018-02-01T20:01:10.0000"}
```

O tópico que o NS deve subscrever é configurável (Nos *Connectors* do NS, no campo *Subscribe*), e neste caso foi escolhido o *downlink_nodes/{deveui}*, identificando o nó no tópico. Existe também a possibilidade do NS publicar eventos (em *Connectors*, *Publish Events*), como nós que se juntaram à rede ou enviar confirmações de mensagens, num outro tópico. Escolhemos o *publish_node_events/{deveui}*.

O mecanismo de confirmações de mensagens, parte do SS o qual deve incluir no corpo da mensagem a confirmar, um identificador aleatório “*receipt*” e o campo de “*confirmed*” a *true*. O valor de “*receipt*” fica guardado no NS, até que a confirmação da receção daquela mensagem chegue do nó. Quando a confirmação é recebida no NS, este publica no tópico dos eventos, uma mensagem com o valor de “*receipt*”. O SS tem, obviamente, de ter subscrito o tópico dos eventos, de forma a receber as confirmações.

Assim, se quisermos enviar imediatamente a mensagem hexadecimal “001122AAFF” para o nó classe C com **DevEUI** “70B3D54997F53293” na porta 2, devemos publicar em:

```
downlink_nodes/70B3D54997F53293
```

A mensagem com o seguinte conteúdo:

```
{"data":"001122AAFF", "port":2, "time":"immediately"}
```

No caso de querermos enviar a mensagem confirmada “FFAA221100”, para o nó classe A com **DevEUI** “70B3D54997F53299”, o qual deve publicar a confirmação “1123”, devemos publicar em:

```
downlink_nodes/70B3D54997F53299
```

A mensagem com o seguinte conteúdo:

```
{"data":"04", "time":"immediately", "confirmed":true, "receipt":"1123"}
```

Quando o NS recebe a confirmação da entrega, irá publicar a mensagem “1123” em *publish_node_events/{deveui}*, informando desta forma ao SS, a entrega da mensagem respeitante aquele valor de *receipt*.

L. PONTOS DE LEITURA NA REDE INTA

Medição	Amplitude Nominal	Amplitude Máxima	Quantidade	Nome	Identificação INTA	Secção	Localização	Notas
Corrente DC (A)	-	10	3	INV DC/AC	i13.1, i13.2, i13.3	6	Armário Micro rede	
	6.7	16	4	PV1	i1.1, i1.2, i1.3, i1.4	10		
	9	16	2	PV2, PV3	i3, i5	6		
	10	16	1	PV4	i7	6	Parque Veículos	
	10	30	2	PV2-DC/DC, PV3-DC/DC	i4, i6	16?	Armário Micro rede	
	-	32?	1	Eletrolyzer	i12	4		
	-	30	1	PV1-DC/DC	i2	16?		
	-	30	1	DC Load	i15	16		
	-	30	1	DC Source	i23	6		
-	30	1	AERO AC-DC	i22	10			
Corrente Bidirecional DC (A)	-	32	1	Pb-Ac	i9	16		
	-	32	1	Li-Ion DC/DC	i10	10		
	-	120	1	Li-Ion	i11	16		
Tensão DC (V)	>250	350	1	PV1	u1	-	Armário Micro rede	250-330 (nominal)
	>150	350	2	PV2, PV3	u2, u3	-		150-320
	360	600	1	PV4	u4	-		260-500 (nominal MPPP)
	>350	420	1	Pb-Ac	u6	-		Nominal 408
	>202	273	1	Li-Ion	u7	-		(sistema usa 210-265)

Tabela L.1 - Pontos de leitura DC

Medição	Amplitude Máxima	Quantidade	Nome	Identificação INTA	Localização
Sensores Corrente AC (A)	63	3	PV4-AC, SKT1, INV EV	i8, i16, i24	Parque Veículos
		1	INV DC/AC	i14	Armário Micro rede
Sensores Tensão AC (V)	230	3	PV4-AC, SKT1, INV EV	u5, u9, u15	Parque Veículos
		1	INV DC/AC	u8	Armário Micro rede
Sensores Corrente AC Trifásicos (A)		2	SKT2, SKT3	i17, i18	Parque Veículos
		2	IN, IN-SAI	i19, i20	Armário Elétrico
		1	AEROG	i21	Armário Micro rede
Sensores Tensão AC Trifásico (V)	380	2	SKT2, SKT3	u10, u11	Parque Veículos
		2	IN, IN-SAI	u12, u13	Armário Elétrico
		1	AEROG	u14	Armário Micro rede

Tabela L.2 - Pontos de leitura AC

M. COMPONENTES DOS MÓDULOS DC

M.1 SENSORES DE CORRENTE

Para medição das correntes DC seriam necessárias várias soluções de transdutores, uma vez que as escalas de medição serão distintas entre os diferentes pontos, não sendo assim possível criar uma solução para todos os cenários. Mais uma vez pretendeu-se garantir o máximo de isolamento da rede, pelo que a escolha recaiu sobre sensores de efeito Hall. Estes garantem o isolamento galvânico, não existindo ligação física entre o circuito a medir (primário) e o circuito de medição (secundário) e funcionam em correntes contínuas e alternadas. O funcionamento baseia-se no efeito de Hall, onde um material semiconductor produz uma tensão aos seus terminais proporcional ao campo magnético que o atravessa. Esta tensão deverá ser lida por um conversor analógico digital. A Figura M.1 seguinte representa o esquema de funcionamento básico destes sensores. Num sensor de formato como o da imagem, quando alimentado por uma qualquer tensão contínua nos terminais superior e inferior e sem nenhum campo magnético aplicado, existe uma passagem de elétrons direta entre estes terminais, sendo que nenhuns se desviam para as faces laterais. A presença de um campo magnético provoca a deflexão de elétrons para as faces laterais surgindo assim uma diferença de potencial proporcional ao fluxo magnético.

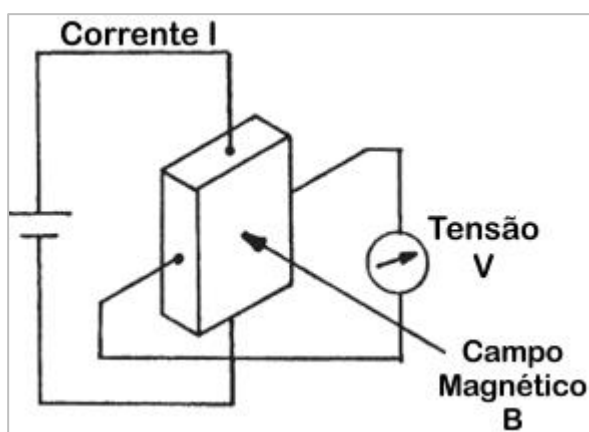


Figura M.1 - Esquema de sensor de efeito Hall

Existem várias características elétricas importantes nestes transdutores, a ter em consideração no momento de escolha dos mesmos. Para medições de correntes alternadas devem considerar-se as características relativas aos tempos de resposta (subida e descida), assim como a largura de banda do transdutor. Para todos os tipos de correntes temos também que considerar a escala de leituras pretendida (corrente máxima) ou corrente nominal, a

linearidade da resposta, as temperaturas de funcionamento (as quais alteram ganhos, *offset* e linearidades), a precisão e a tensão máxima de isolamento. É importante saber também se a saída é em corrente ou tensão, assim como quais os valores nominais destes nas condições de corrente nominal e de pico no primário. Correntes superiores à máxima podem levar à magnetização do núcleo, o que criará um fluxo magnético parasita constante e um *offset* na saída. É possível fazer desmagnetização com procedimentos específicos, no entanto é sempre melhor dimensionar a utilização baseada neste valor máximo para evitar problemas. Juntamente com isto temos algumas características mecânicas, como dimensões do transdutor e da abertura de passagem do primário, os métodos de conexão aos circuitos e de fixação (em pcb ou calha DIN por exemplo) [173][174][175][176][177].

M.2 TIPOS DE TRANSDUTORES DE CORRENTE

Para medições de corrente existem tradicionalmente 2 tipos de transdutores, os de *loop* aberto e de *loop* fechado. Nos de *loop* aberto, o condutor que queremos medir (primário) passa no interior de um núcleo metálico, onde surge um fluxo magnético proporcional à corrente do primário. Este núcleo é interrompido num local onde se insere um sensor de *hall*. Desta forma, o fluxo magnético induzido pelo primário no núcleo atravessa o dispositivo, e provoca o aparecimento de uma queda de tensão aos terminais deste, a qual pode ser lida externamente por um circuito amplificador. A Figura M.2, do lado esquerdo, demonstra este método. Estes sensores são pequenos, têm um custo reduzido, são bastante simples eletronicamente e consomem pouca energia. O tempo de resposta, largura de banda moderada e variações no ganho com mudanças de temperatura são as suas maiores limitações. A saída normal deste tipo de transdutor é em tensão, podendo ter ou não *offset* ajustável [174].

Os sensores de *loop* fechado têm adicionalmente um enrolamento à volta do núcleo (secundário), que permite fazer um balanceamento da corrente induzida pelo primário, aumentando a performance do dispositivo. Esta compensação permite corrigir os desvios de ganho causados pela temperatura. É uma técnica mais complexa a nível de eletrónica adicional, mas têm uma precisão superior, maior linearidade, maior largura de banda e tempos de resposta mais rápidos. No entanto gastam mais energia e são têm um custo maior [174]. A Figura M.2 do lado direito, exemplifica este esquema.

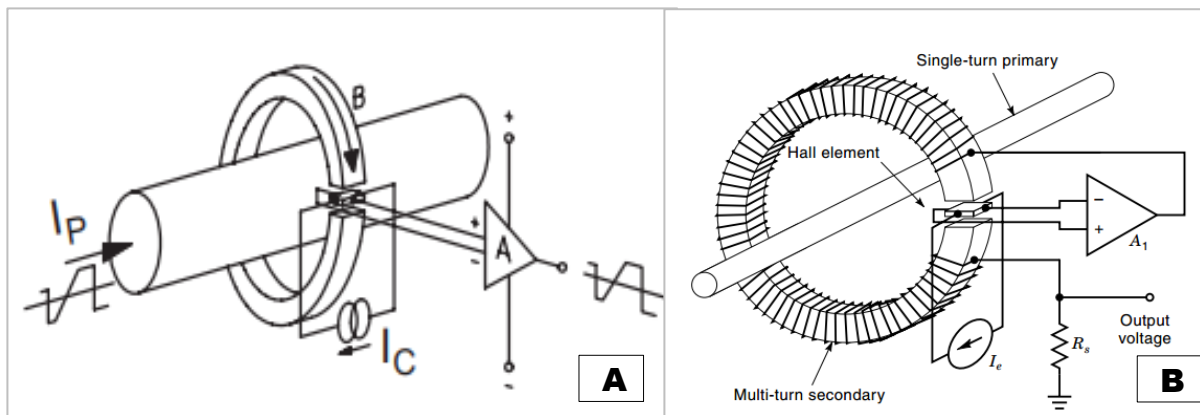


Figura M.2 – Esquema de transdutor de efeito de hall em loop aberto (A) [174] e em loop fechado (B) [177]

Independente do tipo de transdutor, as saídas podem ser em tensão ou corrente. Se a saída for em corrente necessita de uma resistência de carga para leituras de tensão. As duas tecnologias podem incluir um circuito de condicionamento do sinal, podendo alterar o tipo de saída de tensão para corrente e vice-versa, assim como ajustar o *offset* se existirem entradas para tensões de referência. Muitos dos transdutores podem também ler correntes bidirecionais, sendo que para correntes de 0 Amperes, a tensão de saída será normalmente metade da tensão de alimentação, como mostra a Figura M.3 num exemplo de alimentação a 5 V.

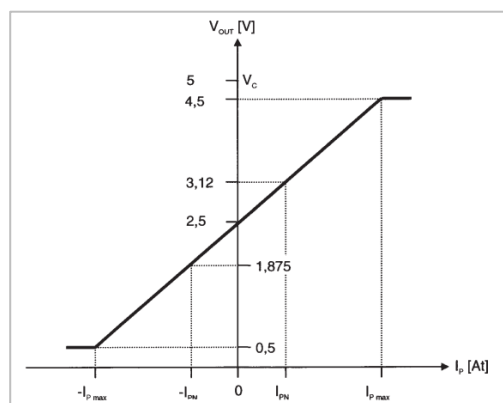


Figura M.3 - Gráfico de tensão de saída em função de corrente primária[174]

M.3 CONVERSOR ANALÓGICO DIGITAL ADS1115

Para efetuar as leituras das saídas em tensão dos transdutores, considerou-se um conversor analógico digital aplicado no módulo. Devido à alta interferência dentro do armário da instalação, apenas se pretende instalar os transdutores de corrente e um conversor ADC no mesmo módulo, enviando a conversão já em formato digital e por cabo blindado para o

exterior do armário. Este cabo permitirá a alimentação dos transdutores a 5 V e a transmissão via I2C das leituras para o exterior. O conversor ADC escolhido foi o TI ADS1115, no módulo Adafruit ADS1115 *Breakout Board* [178], com 16 *bits* de resolução, 860 amostras por segundo e 4 entradas simples ou diferenciais. Permite 4 endereços distintos (48h, 49h, 4Ah e 4Bh) selecionados num pino ligado à massa, Vcc, DAS ou SCL. Este tem um amplificador integrado e permite várias escalas de leituras (*full scale range*, FSR, de valores entre ± 256 mV até $\pm 6,144$ V), configuráveis por *software*. Dentro das opções, e considerando a alimentação a 5 V, optámos pela FSR de $\pm 4,096$ V. As conversões serão mostradas num formato de complemento para 2, significando que os valores entre 0 e 4,096 V (+FSR) correspondem a uma saída entre 0000h e 7FFFh. Os valores entre -125 μ V e -4,096 V (-FSR) correspondem a valores entre FFFFh e 8000h (ver Tabela M.1). Na prática, se apenas utilizarmos a escala positiva, corresponderá a uma resolução de 15 *bits*, com resolução de 125 μ V no *least significant byte* (LSB). O ADC poderá assim ler numa escala de 0 a 4,096 V, sendo este o valor máximo que pode ser lido [179]. Para o caso dos transdutores de corrente LEM HO, as tensões nunca são negativas, sendo por isso possível de utilizar este esquema. Já no modelo da Tamura, iremos condicionar o sinal para aproveitar a parte negativa do ADC.

Vin(FSR=4,096V)	Vin (mV)	Output Code (hex)	Output Code (binary)
+FSR	4096	7FFFh	01111111 11111111b
+FSR/2 ¹⁵	0,125	0001h	00000000 00000001b
0	0	0000h	00000000 00000000b
-FSR/2 ¹⁵	-0,125	FFFFh	11111111 11111111b
-FSR	-4096	8000h	10000000 00000000b

Tabela M.1 - Valores exemplo de conversão do ADS1115

M.4 DIMENSIONAMENTO DO TRANSDUTOR DE CORRENTE LEM HO 10-P

O transdutor LEM HO 10-P (Figura M.4) foi o escolhido para leituras de corrente até 32 A. A abertura é de 8x8 mm, permite uma passagem dos cabos existentes na instalação. Estes dão-nos a saída em tensão, bastando assim efetuar a leitura pelo ADC. Embora possível, nestes casos não seria necessário adicionar um circuito de condicionamento do sinal. As saídas dos HO variam entre a tensão de referência aplicada na entrada de referência e um máximo

correspondente à tensão de alimentação, embora para qualquer valor de corrente abaixo dos 32 A, nunca se atinja esse máximo.

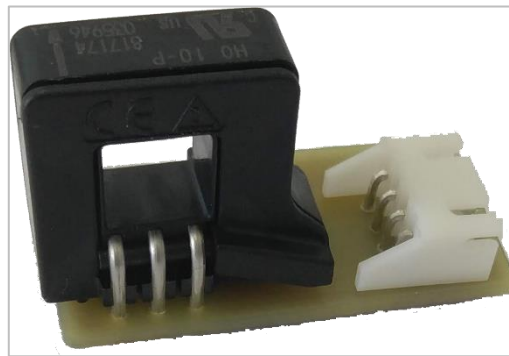


Figura M.4 - Transdutor LEM HO 10-P montado em pcb

Foram calculadas as tensões de saída (V_s) em função da corrente de entrada (I_p), sensibilidade do modelo (G) e tensão de referência aplicada (V_{ref}), baseadas na fórmula [180]:

$$V_s = V_{ref} + G * I_p [V] \quad (36)$$

No HO 10-P, a tensão de saída varia entre 1,2 e 3,76 V para 0 e 32 A, respetivamente. Todos estes valores se encontram dentro da escala do ADC. Esta fórmula não considera os desvios causados por temperatura, linearidade ou o *offset* de cada transdutor, sendo que é apenas uma aproximação teórica. A fórmula mais real será dada por:

$$V_s = V_{ref} + G * I_p + \sum errors [V] \quad (37)$$

Nesta fórmula, o somatório deve incluir os erros associados à variação de temperatura (que atuam no *offset* e na sensibilidade), erros de linearidade e erros de sensibilidade. No entanto, estes são difíceis de calcular, sendo por isso mais fácil utilizar a precisão do mesmo à temperatura ambiente. Esta precisão dá-nos o máximo valor que o transdutor apresentará na saída, em relação ao valor linear, conforme demonstra a Figura M.5. No caso dos modelos

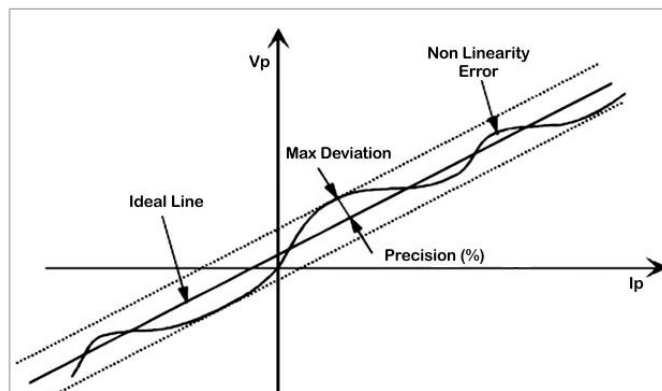


Figura M.5 - Gráfico de não linearidade

HO-P, esta precisão é de $\pm 1.35\%$ da corrente nominal, correspondendo a variações de $\pm 0,135$ A no HO-10P. Adicionalmente é dado um valor máximo para o *offset* elétrico de 10 mV.

Para gerar a tensão de referência a partir da alimentação de 5 V, bastaria um simples divisor de tensão. No entanto, pelo baixo custo e fiabilidade, optou-se por um regulador de tensão linear de baixo *dropout* (*low dropout regulator* LDO). Analisaram-se as características técnicas dos modelos MCP1825S, MCP1826S e MCP1700. Foram calculadas as escalas de medida do HO 10-P, consoante as tensões de referência aplicada [180]. Verificou-se que se podem medir correntes até 32 A, utilizando tensões de referência até sensivelmente 1,9 V, conforme os valores da Tabela M.2. No entanto, esta gera na saída uma tensão de 4,46 V, superior à escala do ADC. Assim, pelo ADC estávamos limitados a 1,5 V de tensão de referência. Como a medida mais comum de LDO é 1,2 ou 0,8 V, optou-se pela variante de 1,2 V, escolhendo o *chip* MCP1700, com um invólucro TO-92 de 3 pinos, com capacidade de colocar 250 mA na saída (com alimentação a 5 V) e 3% de precisão na regulação de carga. Adicionalmente inclui ainda proteção para correntes e temperaturas elevadas, assim como uma corrente quiescente baixa, de 1.6 μ A. Estas características chegam para gerar a tensão de referência necessárias aos transdutores [138].

Limites	HO 10-P			
	Vref (V)	I Min (A)	I Max (A)	Vout (V) @ 32 A
0,5	0	50	3,06	4,5
0,6	-1,25	48,75	3,16	4,5
0,7	-2,5	47,5	3,26	4,5
0,8	-3,75	46,25	3,36	4,5
0,9	-5	45	3,46	4,5
1	-6,25	43,75	3,56	4,5
1,1	-7,5	42,5	3,66	4,5
1,2	-8,75	41,25	3,76	4,5
1,3	-10	40	3,86	4,5
1,4	-11,25	38,75	3,96	4,5
1,5	-12,5	37,5	4,06	4,5
2	-18,75	31,25	4,56	4,5

Tabela M.2 - Correntes máximas e mínimas, em função de tensão de referência utilizada [180]

M.5 DIMENSIONAMENTO DO TRANSDUTOR DE CORRENTE TAMURA L31S050S05FS

Para o dimensionamento do circuito onde se ligaria o transdutor Tamura L31S050S05FS (Figura M.6) [181], aplicável nos locais onde existam correntes bidirecionais (i_9 , i_{10} , i_{11}), foi necessário ter em consideração as correntes máximas dos locais de aplicação. Decidiu-se aplicar um circuito de condicionamento de sinal, transformando as tensões de saída do transdutor ($0V > 2,5V > 5V$), nas tensões FSR do ADS1115 ($-4,096V > 0V > 4,096V$). Por exemplo, uma corrente de 0 A no primário do transdutor, gera uma saída de 2,5V a qual é aplicada na entrada do condicionador. Este circuito transformaria os 2,5 V em 0 V, para aplicar numa entrada do ADC, que o converte no valor 0000h. Uma saída de 0 V no Tamura, deveria resultar numa tensão de -4,096V à saída do condicionador, a qual aplicada no ADC resultaria no valor 8000h. Esta última, significaria que a corrente era inversa ao sentido indicado no transdutor. Este exemplo foi apenas demonstrativo, pois na realidade os nossos extremos não seriam os 0 e 5 V, mas sim os valores expectáveis para as correntes máximas a ler, descritos de seguida.



Figura M.6 - Transdutor Tamura L31S050S05FS

De forma a gerar nas saídas as tensões simétricas, optou-se por utilizar um *ampop* LM358 [182], alimentado a $\pm 12V$. Para achar os valores teóricos de saída, que nos permitem dimensionar o circuito de condicionamento do ADC, utilizou-se a seguinte expressão:

$$V_s = V_{ref} \pm \frac{(G * I_p * N)}{I_n} [V] \quad (38)$$

Como a corrente nominal (I_n) de 50 A, nos dá uma variação de 625 mV (G) na saída, a aplicação em locais onde a corrente máxima seja de apenas de 32 A, resultaria numa variação de apenas 400 mV. A sensibilidade do mesmo é de 12,5mV/A. De início, considerou-se a

possibilidade de dar várias voltas do cabo primário no transdutor, mas após uma visita à INTA, verificou-se que as dimensões dos cabos e dificuldade de acesso nos locais de aplicação, não permitiriam esta solução. Assim, todo o dimensionamento foi efetuado para uma única passagem de cabo ($N = 1$). Assim, calculou-se a segunda parte da equação $\frac{(G \cdot I_p \cdot N)}{I_n}$, variando I_p , de forma a perceber qual seria a variação desta componente. Para os 2 locais com correntes bidirecionais de 32 A (i9, i10), a saída do transdutor apresentava tensões entre 2,1 V (para $I_p = -32$ A) e 2,9 V (para $I_p = 32$ A).

$$V_{out} = 2,5 \pm \frac{(0,625 * 32 * 1)}{50} = 2,5 \pm 0,4 [V] \quad (39)$$

No ponto de leitura das baterias de íões de lítio (i11), a corrente máxima de 120 A de descarga, traduz-se em saídas que variam entre 1 V (para $I_p = -120$ A) e 4 V (para $I_p = 120$ A)

$$V_{out} = 2,5 \pm \frac{(0,625 * 120 * 1)}{50} = 2,5 \pm 1,5 [V] \quad (40)$$

Para efetuar este condicionamento optou-se por uma montagem diferencial, similar à representada na Figura M.7. Esta rege-se pela equação:

$$V_{out} = \left(1 + \frac{R4}{R3}\right) * \left(\frac{R2}{R1 + R2}\right) * V_1 + \left(-\frac{R4}{R3}\right) * V_2 [V] \quad (41)$$

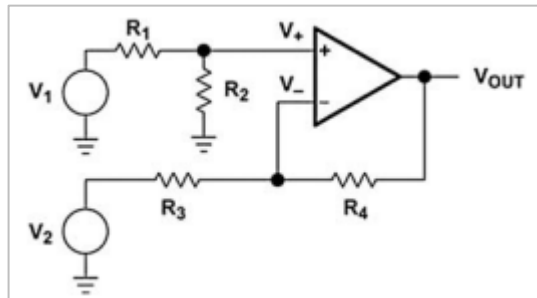


Figura M.7 - Ampop em montagem diferencial

Esta equação pode ser aproximada pela equação de reta $y = mx + b$, sendo $b = \left(-\frac{R4}{R3}\right) * V_2$, $m = \left(1 + \frac{R4}{R3}\right) * \left(\frac{R2}{R1 + R2}\right)$, $x = V_1$ e $y = V_{out}$. Adicionalmente pode-se simplificar esta expressão se se igualarem as resistências $R1 = R3$ e $R2 = R4$, obtendo:

$$V_{out} = (V_1 - V_2) * \left(\frac{R4}{R3}\right) [V] \quad (42)$$

Utilizando a tensão de referência de 2,5 V, pode-se subtrair esta à tensão de entrada não inversora (V_1), aplicando-a na entrada inversora do amplificador (V_2). Seguindo o método de dimensionamento de [183] e [184], foram calculadas as equações das retas, de forma a retirar os coeficientes m e b , baseados nas tensões máximas e mínimas de entrada e saída. Para o caso de 32 A:

- $V_{out_{fullscale}} = 4,096 \text{ V}$
- $V_{out_{zeroscale}} = -4,096 \text{ V}$
- $V_{in_{fullscale}} = 2,9 \text{ V}$
- $V_{in_{zeroscale}} = 2,1 \text{ V}$
- $V_{ref} = 2,5 \text{ V}$

$$\begin{cases} -4,096 = m * 2,1 + b \\ 4,096 = m * 2,9 + b \end{cases} \Leftrightarrow m = \frac{V_{out_{fullscale}} - V_{out_{zeroscale}}}{V_{in_{fullscale}} - V_{in_{zeroscale}}} = \frac{4,096 - (-4,096)}{2,9 - 2,1} = 10,24 \quad (43)$$

$$b = V_{out_{zeroscale}} - m * V_{in_{zeroscale}} = -4,096 - 10,24 * 2,1 = -25,6 \quad (44)$$

Para a escolha de resistências deve-se obter o rácio entre as mesmas, a partir do valor de b :

$$b = \left(-\frac{R4}{R3}\right) * V_{ref} \Leftrightarrow -25,6 = \left(-\frac{R4}{R3}\right) * 2,5 \Leftrightarrow R4 = 10,24 * R3 \quad (45)$$

Escolhendo $R3 = 10 \text{ k}\Omega$, $R4$ deverá ser $102,4 \text{ k}\Omega$. O valor mais próximo de resistências existente é o de $102 \text{ k}\Omega$. Calculando os extremos de tensão de saída do transdutor ($V_{out_{zero_scale}}$ com $V_1 = 2,1 \text{ V}$ ou $V_{out_{full_scale}}$ com $V_1 = 2,9 \text{ V}$), baseados nestas resistências e com $V_2 = V_{Ref}$, teremos à saída do *ampop* V_{out} os seguintes valores:

$$V_{out} = (V_1 - V_{Ref}) * \left(\frac{102}{10}\right) [V] \quad (46)$$

$$V_{out_{zero_scale}} = (2,1 - 2,5) * \left(\frac{102}{10}\right) = -4,08 \text{ V} \quad (47)$$

$$V_{out_{full_scale}} = (2,9 - 2,5) * \left(\frac{102}{10}\right) = 4,08 \text{ V} \quad (48)$$

Neste caso podemos simplificar a expressão em:

$$V_{out} = V_1 * 10,2 - 25,5 [V] \quad (49)$$

Estes valores confirmam que a escala resulta alterada pelo circuito e que fica agora dentro da escala de entrada $-4,096V/4,096V$ do ADC. Apesar da tensão de saída máxima não corresponder exatamente à tensão de saída desejável, é uma boa aproximação, ficando abaixo desta e não saturando as leituras do ADC nos casos extremos. Verificamos também, que a sensibilidade à saída do condicionador passou a ser de $127,5\text{ mV/A}$. De forma a confirmar os resultados, o circuito foi simulado em no *software* Multisim, variando a tensão de entrada entre os $2,1$ e $2,9\text{ V}$, e confirmando a tensão na saída. A Figura M.8 mostra esses resultados.

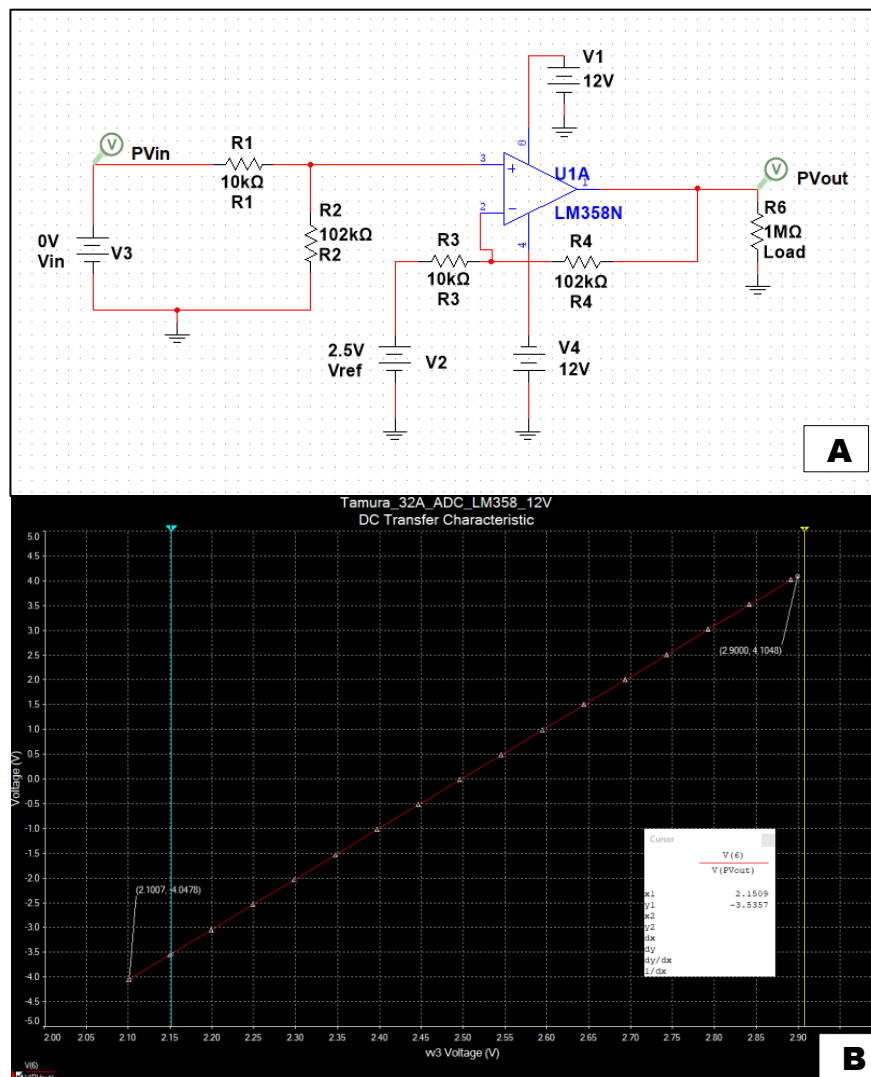


Figura M.8 - Simulação de circuito de condicionamento, 32 A. Circuito utilizado (A) e gráfico de tensão de saída em função da tensão de entrada (B)

Repetimos o mesmo procedimento para o caso de i_{11} , 120 A .

- $V_{out\ fullscale} = 4,096\text{ V}$
- $V_{out\ zeroscale} = -4,096\text{ V}$

- $V_{in_{fullscale}} = 4 V$
- $V_{in_{zeroscale}} = 1 V$
- $V_{ref} = 2,5 V$

$$\begin{cases} -4,096 = m * 1 + b \\ 4,096 = m * 4 + b \end{cases} \Leftrightarrow m = \frac{V_{out_{fullscale}} - V_{out_{zeroscale}}}{V_{in_{fullscale}} - V_{in_{zeroscale}}} = \frac{4,096 - (-4,096)}{4 - 1} = 2,73 \quad (50)$$

$$b = V_{out_{zeroscale}} - m * V_{in_{zeroscale}} = -4,096 - 2,73 * 1 = -6,826 \quad (51)$$

$$b = \left(-\frac{R4}{R3}\right) * V_{ref} \Leftrightarrow -6,826 = \left(-\frac{R4}{R3}\right) * 2,5 \Leftrightarrow R4 = 2,7304 * R3 \quad (52)$$

Escolhendo $R3 = 10 \text{ k}\Omega$, $R4$ seria $27,3 \text{ k}\Omega$. A resistência mais próxima seria a de $27,4 \text{ k}\Omega$, que resultaria numa tensão à saída superior aos $4,096V$, optando assim por uma inferior de $27 \text{ k}\Omega$.

$$V_{out} = (V_1 - V_{Ref}) * \left(\frac{27}{10}\right) [V] \quad (53)$$

$$V_{out_{zero_scale}} = (1 - 2,5) * \left(\frac{27}{10}\right) = -4,05 V \quad (54)$$

$$V_{out_{full_scale}} = (4 - 2,5) * \left(\frac{27}{10}\right) = 4,05 V \quad (55)$$

$$V_{out} = V_1 * 2,7 - 6,75 [V] \quad (56)$$

A sensibilidade neste caso é de $33,75 \text{ mV/A}$. Mais uma vez, simulou-se o circuito em Multisim, conforme a Figura M.9.

Verificou-se nas primeiras montagens que cada transdutor apresentava um *offset* diferente. Assim para corrigir o mesmo, pode ser necessário adicionar um circuito de compensação do mesmo. Este permite ajustar a tensão zero, à saída do mesmo. Variações nas tolerâncias dos componentes fazem com que este valor não exatamente igual ao dimensionado, sendo por isso necessário calibrar e compensar essas variações. Calibrações adicionais que venham a ser necessárias após a montagem do circuito, deverão ser efetuadas em *software*, compensando as variações das resistências e dos transdutores, à temperatura ambiente. Para tal devem ser feitas leituras com vários valores distintos e conhecidos de corrente, de forma a calibrar as leituras.

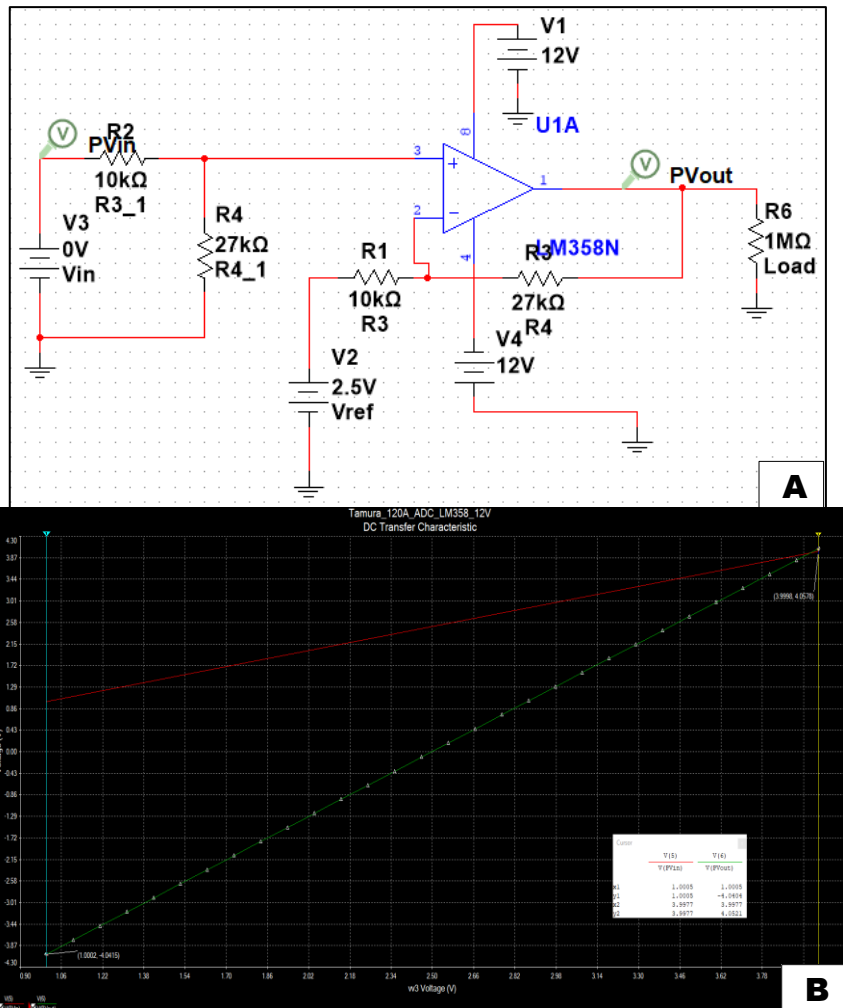


Figura M.9 – Simulação de circuito de condicionamento, 120 A. Esquema (A) e gráfico de tensão de saída em função de entrada (B)

M.6 DIMENSIONAMENTO DO TRANSDUTOR DE TENSÃO LEM LV-25P

O *datasheet* do transdutor LEM LV-25P (Figura M.10) [185] diz-nos que este tem uma relação de conversão de 2500:1000, resultando na seguinte relação entre corrente no primário (I_p) e corrente no secundário (I_s):

$$I_p * 2500 = I_s * 1000 \Leftrightarrow I_s = \frac{I_p * 2500}{1000} \Leftrightarrow I_s = I_p * 2.5 \quad [A] \quad (57)$$

O circuito interno do transdutor é o da Figura M.11. Nesta figura, verificamos também o circuito externo aconselhado pelo fabricante, sendo que no circuito primário se utiliza uma resistência R_1 , para limitar a corrente que flui pelo transdutor. Pelo *datasheet* sabemos que a



Figura M.10 - Transdutor LEM LV 25-P

corrente nominal primária deve ser 10 mA, o que corresponde a dizer que ao aplicar na entrada a tensão nominal de funcionamento do circuito, a resistência R_1 deve ser dimensionada para uma corrente de 10 mA. O *datasheet* indica também que a corrente máxima no primário deverá ser 14 mA, sendo 40 % superior à corrente nominal. A partir deste valor, o transdutor não conseguirá debitar uma corrente no secundário, correspondente à tensão no primário, sendo por isso importante efetuar o dimensionamento considerando este máximo. Poderemos assim medir sobretensões até 40%, algo que pode ocorrer em cargas indutivas, como no arranque de motores. Apesar de muitos destes sensores servirem para medir as tensões de painéis solares, serão também aplicados num *bus* DC, onde podem estar ligadas qualquer tipo de cargas.

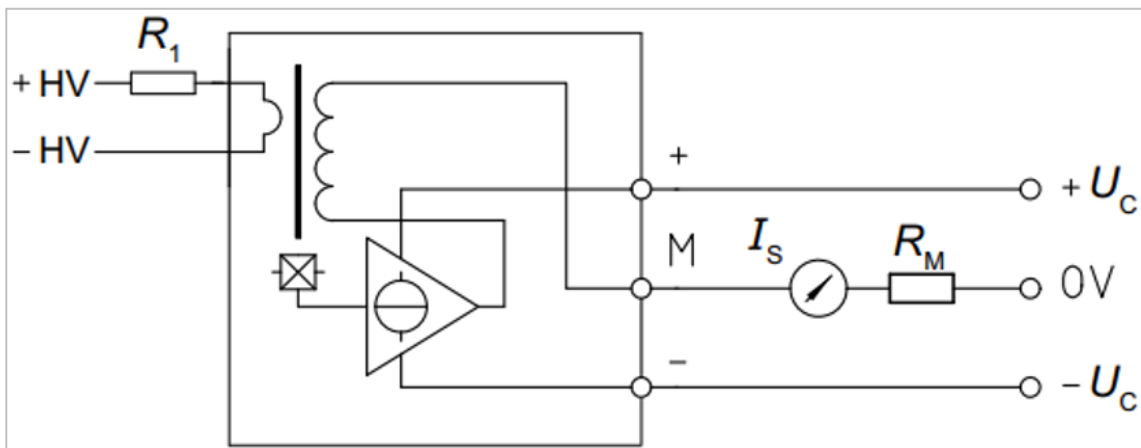


Figura M.11 - Circuito interno e montagem aconselhada do LV 25-P

Para dimensionar a resistência do primário R_1 com 10mA devemos utilizar a tensão nominal do circuito que queremos medir e aplicar a lei de Ohm. Adicionalmente devemos considerar a resistência do enrolamento interno R_p , não representada na figura. Esta varia

consoante a temperatura, sendo que a 85 °C o *datasheet* refere 300 ohm e à temperatura ambiente (aproximadamente 20 °C) medimos aproximadamente 220 ohm (não é referida no *datasheet*).

$$R_1 + R_p = \frac{V}{10mA} \Leftrightarrow R_1 = \frac{V}{10mA} - R_p \text{ [ohm]} \quad (58)$$

Com este valor teórico de R_1 , podemos achar a potência nominal dissipada na mesma. No entanto a nível de dimensionamento devemos também considerar a tensão máxima de leitura. Neste caso assumiu-se o máximo de 40% que o *transceiver* permite ler, mas a tensão primária poderá ter picos e ultrapassar esse valor, resultando apenas na incapacidade do transdutor em lê-los. Assim, calcularam-se as potências nominais (P_n) e máxima (P_{max}) para R_1 assumindo vários valores de tensão nominal (V_{pn}) e máxima (V_{pmax}), de forma a escolher resistências que permitam dissipar o valor máximo, nunca ultrapassando as características da resistência. Os valores obtidos encontram-se na Tabela M.3. A fórmula utilizada foi a da potência:

$$P_n = \frac{V_{R1} n^2}{R_1} \text{ [W]} \quad (59)$$

INTA	Vpn (V)	Vpmax (V)	R1 (ohm)	R1-Rp (25 °C)	R1 real (ohm)	l _{pn} (R1 real) (mA)	P _n (W)	P _{max} (W)
	250	350	25000	24772	24000	10,42	2,604	5,104
Li-Ion	273	382,2	27300	27072	27000	10,11	2,760	5,410
	300	420	30000	29772	30000	10,00	3,000	5,880
PV1,2,3,4	320	448	32000	31772	33000	9,70	3,103	6,082
	350	490	35000	34772	36000	9,72	3,403	6,669
	400	560	40000	39772	39000	10,26	4,103	8,041
DCBUS	408	571,2	40800	40572	39000	10,46	4,268	8,366
	450	630	45000	44772	43000	10,47	4,709	9,230

Tabela M.3 - Resistências teóricas para diversos valores de tensão de entrada

Obviamente a utilização de uma única resistência cria problemas na dissipação de potência, uma vez que toda a tensão cai sobre a mesma. A tabela anterior mostra os valores calculados para uma única resistência, onde se podem ver potências nominais de perto de 5 watt. Assim, deve-se optar por várias resistências em série, de forma a reduzir a queda de tensão em cada uma delas e respetiva potência dissipada. Como tal, considerou-se este cenário, calculando qual a queda de tensão (V_{R1_1}), potências nominais (P_{nR1_1}) e máximas (P_{maxR1_1}), em cada uma de quatro resistências idênticas, consoante a tensão nominal de entrada e respetivos 10mA de corrente. Tentou-se sempre manter a potência dissipada em cada

resistência, abaixo dos 2 watt, dimensionando os valores de forma a utilizarmos resistências de 5 watt. Desta forma, fica-se com bastante margem de segurança e diminui-se o aquecimento excessivo das resistências, aumentando a fiabilidade das mesmas [174]. Foram consultadas as tabelas de resistências da série E24 (com 5% de tolerância) de forma a dimensionar o circuito com os valores reais mais próximos dos teóricos ($R1$). Após a escolha do valor de resistência ($R1_{total} = 4 * R1_1 + Rp$), na qual 4 resistências desse valor em série ficavam mais próximo do valor teórico ($R1$), calculou-se novamente a corrente (In_{R1}), baseada nesses valores e na tensão nominal, de forma a verificar se ficávamos perto dos 10 mA pretendidos. A Tabela M.4 resume este dimensionamento, onde podemos ver os valores escolhidos para as resistências de potência $R1_1$ (apenas mostra os valores para uma delas, mas são utilizadas quatro idênticas), em função da tensão primária nominal e máxima. Destacam-se os valores dos equipamentos INTA a amarelo.

INTA	V _{pn} (V)	V _{pmax} (V)	R1 (ohm)	R1_1 (ohm)	V_R1_1 (V)	P _{n_R1_1} (W)	P _{max_R1_1} (W)	R1_total (ohm)	Delta R1 (ohm)	In_R1 (mA)
	250	350	25000	6200	62	0,62	1,22	25028	-28	9,9888
Lilon	273	382,2	27300	6800	68	0,68	1,33	27428	-128	9,9533
	300	420	30000	7500	75	0,75	1,47	30228	-228	9,9246
PV1,2,3,4	320	448	32000	8000	80	0,8	1,57	32228	-228	9,9293
	350	490	35000	9100	91	0,91	1,78	36628	-1628	9,5555
	400	560	40000	10000	100	1	1,96	40228	-228	9,9433
DCBUS	408	571,2	40800	10000	100	1	1,96	40228	572	10,1422
	450	630	45000	11000	110	1,1	2,16	44228	772	10,1746

Tabela M.4 - Dimensionamento de resistências do primário, em função de tensão de entrada

Para dimensionar a resistência no secundário R_m , devemos ter em atenção as escalas de valores que pretendemos medir. Os conversores ADC irão fazer leituras entre 0 e 4,096 V, pelo que 4,096 V deverá corresponder à tensão máxima de entrada, ou seja 40% superior à tensão nominal. Como sabemos que os 10mA na entrada (à tensão nominal) geram 25 mA na saída (da relação de conversão, na fórmula 57), sabemos também que o máximo de 14 mA na entrada (V_{pmax}) gerará 35 mA na saída. Assim, para garantir a tensão de 4,096 V na saída, com uma corrente de 35 mA, R_m deverá ser 117,03 ohms.

$$R_m = \frac{4,096}{35m} = 117,03 [ohm] \quad (60)$$

O valor de resistência real mais próximo é de 118 ohms, sendo este o valor escolhido (na série E96). Com este valor construímos uma tabela relacionando a corrente nominal no primário (I_{pn}), a respectiva corrente nominal no secundário (I_{sn}) e o valor de tensão (V_{sn}) obtido em R_m . Desta forma sabemos qual a tensão na resistência de carga, em função da tensão nominal primária, o qual deverá ser um valor a rondar os 3 V, conforme mostra a Tabela M.5.

INTA	Primário		Secundário	
	Vpn (V)	Ipn_R1 (mA)	Isn_Rm (mA)	Vsn_Rm (V)
	250	9,989	24,972	2,947
Lilon	273	9,953	24,883	2,936
	300	9,925	24,811	2,928
PV1,2,3,4	320	9,929	24,823	2,929
	350	9,556	23,889	2,819
	400	9,943	24,858	2,933
DCBUS	408	10,142	25,355	2,992
	450	10,175	25,436	3,001

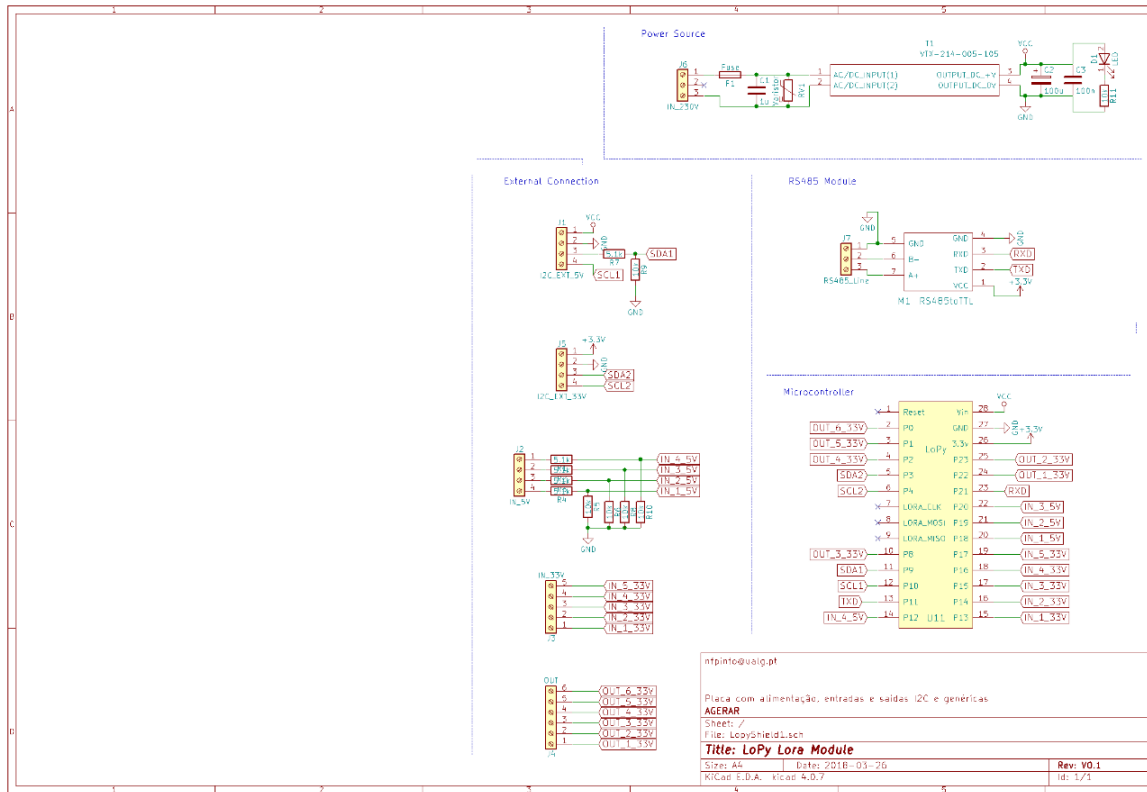
Tabela M.5 - Corrente e tensão no secundário, em função da tensão primária

N. TABELA COMPARATIVA DE TRANSDUTORES DE CORRENTE

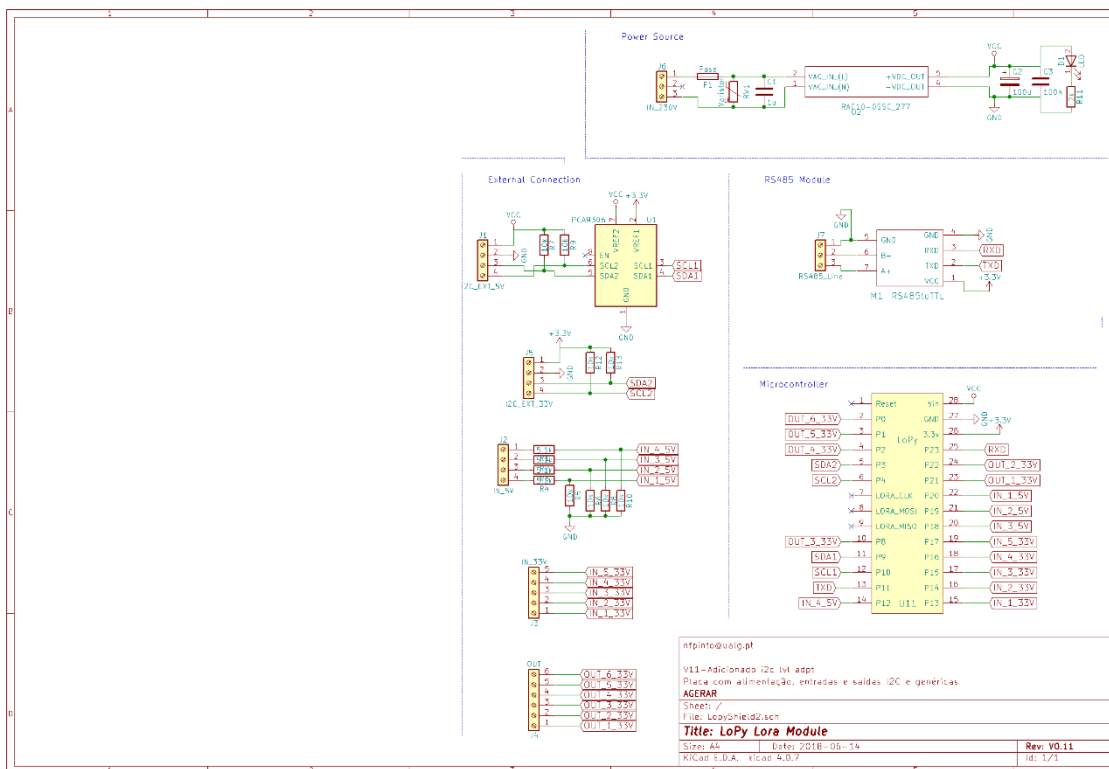
Fabricante		LEM										CY						Tamura							
Serie		HO								HTB		LA 25		BS5	ES5A	B200			L31S	L32P	L03S	L08P	S25P		
Modelo		HO 6-P/SP3 3	HO 10-P/SP3 3	HO 25-P/SP3 3	HO 6-P	HO 10-P	HO 25-P	HO 50-S/SP3 0	HO 50-S/SP3 3	HO 60-P	HTB 50-P/SP5	HTB 50-P	LA 25-P	LA 55-P	CYHCS -BS5 50A	CYHCS -ES5A-10A	CYHCS -B200 -10 A	20	25 A	L31S050S0 5FS	L32P050S05F S	L03S050D1 5	L08P050D15M 1	S25P050D15 X	
Tecnologia		OL								OL		CL		OL	CL	CL			OL	OL	OL	OL	CL		
Tensão alimentação	Vcc	3.3			5			5	3.3	5	12 .. 15 (+unipolar)	12 .. 15(- +bipolar)	12 ... 15	12 ... 15	5	5	5			5	5	15(bipolar)	15(bipolar)	12 .. 15(bipol)	
Corrente nominal	Ipn (A)	6	10	25	6	10	25	50	50	60	50		25	50	50	10	10	20	25	50	50	50	50	50	
Corrente máxima	Ipm mea. range (A)	40	50	62.5	40	50	62.5	125	125	150	150		55	70	100	30	20	40	50	150	150	150	150	55	
Precisão	(@Ipn)	1.35%			1.35%			1.25%			1%		1.25 %	0.65%	1%	0.50%	0,7%			0.50%	0.50%	1%	1%	0,5%	
	Vref min (V)	0.5								-										-	2.5	2.5	-	-	-
Tensão sem corrente	Vout @ Ipn = 0 (V)	Vref + Voffset								Vcc/2	0				2.5	2.5	2.5			Vref + Voffset	Vref + Voffset	40mv	40mv	-	
Saída em tensão	Vout @ Ipn (V)	460 mV			600m V	800 mV	800 mV	460 mV	800 mV		1,667 + Vcc/2	4				3.125	3.5			625 mV (+- 15mV)	625 mV (+- 15mV)	4	4		
Saída em corrente	Ipn (mA)	-			-			-			-		25	50	-					-	-	-	-	-	50
Variação possível	Voffset +- (mV)	7	5	6	10	10	10	5	5		30	30								25	25	40	30	-	
Sensibilidade	Gth (mV/A)	76.67	46	18.4	100	80	32	16	9.2	13,333															
Resistência de carga aconselhada	RL (Ohm)	10K			10K			10 K			10K	10K	50-400 T70 15V				10K			10K	10k	10K		15V(135-155@Ipn)	
Consumo de corrente	Ic (mA)	25			25			25			25	25	15		10mA + Is	25	20	45			15	15	20	12	16mA+Is
Dimensão abertura	(mm)	Rect 8 x 8			Rect 8 x 8			Rect 15x 8	Rect 18.8 x 8.2			Circular 16		Rect 12.7 x 7	Rect 20.5 x 10.5	Circ 8	Circ 5		Circ 5.5	Rect 20.5 x 10.5	Rect 15 x 8	Rect 20.5 x 10.5	circ 16	Rect 13x8.5	
Notas								Painel, com ficha JST SPH-00xT	Painel, com ficha JST SPH-00xT	3 condensadores 4,7 nF entre as saídas 1,3,4.			Existe em 100, 125 A	Existe SP1 com Ipm de 100 A e Ic = 25mA						Montagem em painel		Montagem em painel			
Outras características		OC pin 2.63 x Ipn			OC pin 2.63 x Ipn			OC pin 2.93 X Ipn			Sem VinRef				Sem VinRef	Com VinRef	Sem VinRef			Com VinRef	Com VinRef	Sem VinRef	Sem VinRef	Sem VinRef	
Preço	EUROS	8.76	8.76	8.76	8.76	8.76	8.76		14.92	13,84		17.04	31.22	21.16	11.66	13.42	14,45	14,45	14.45	13,43	13.48	14.49	12.27	15,97	

Tabela N.1 - Tabela comparativa de transdutores de corrente

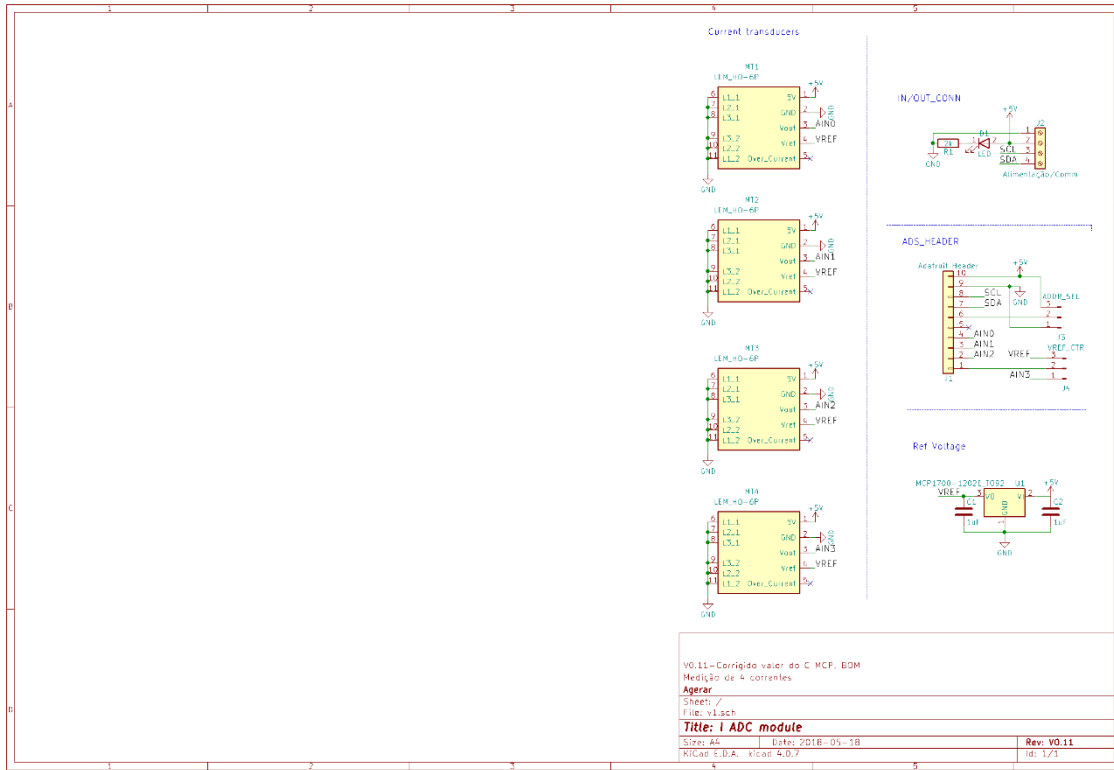
O. ESQUEMÁTICO DA PLACA LOPY POWER BOARD V1



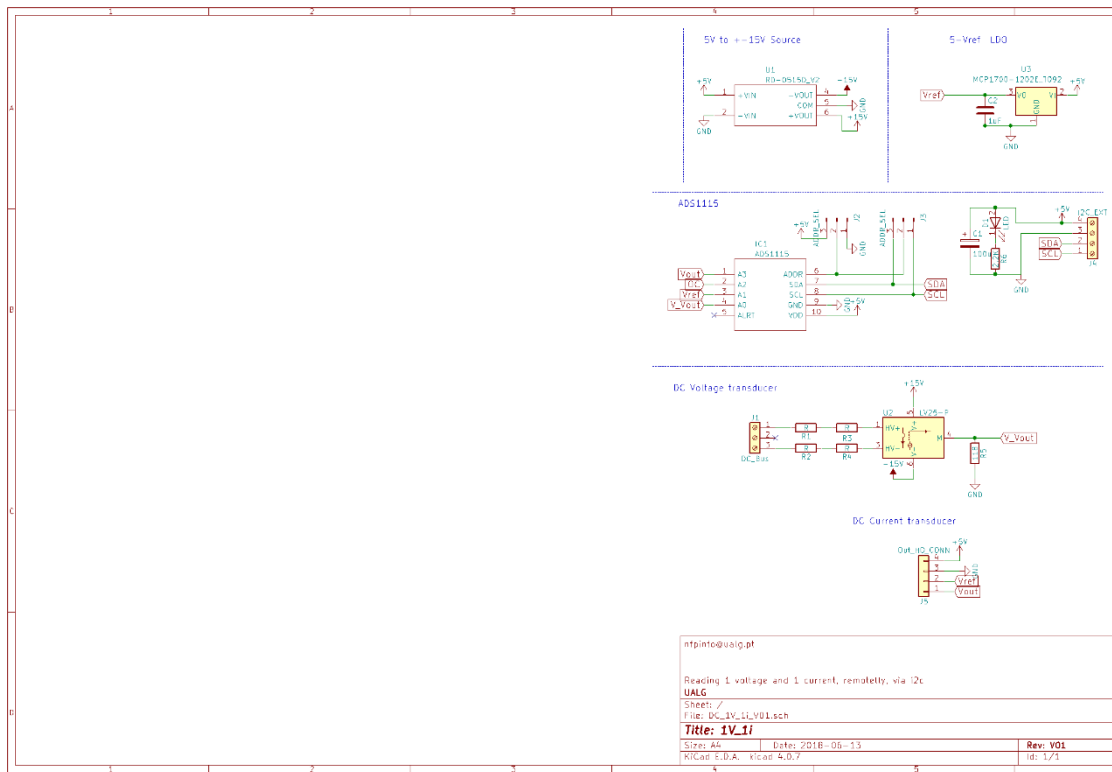
P. ESQUEMÁTICO DA PLACA LOPY POWER BOARD V2



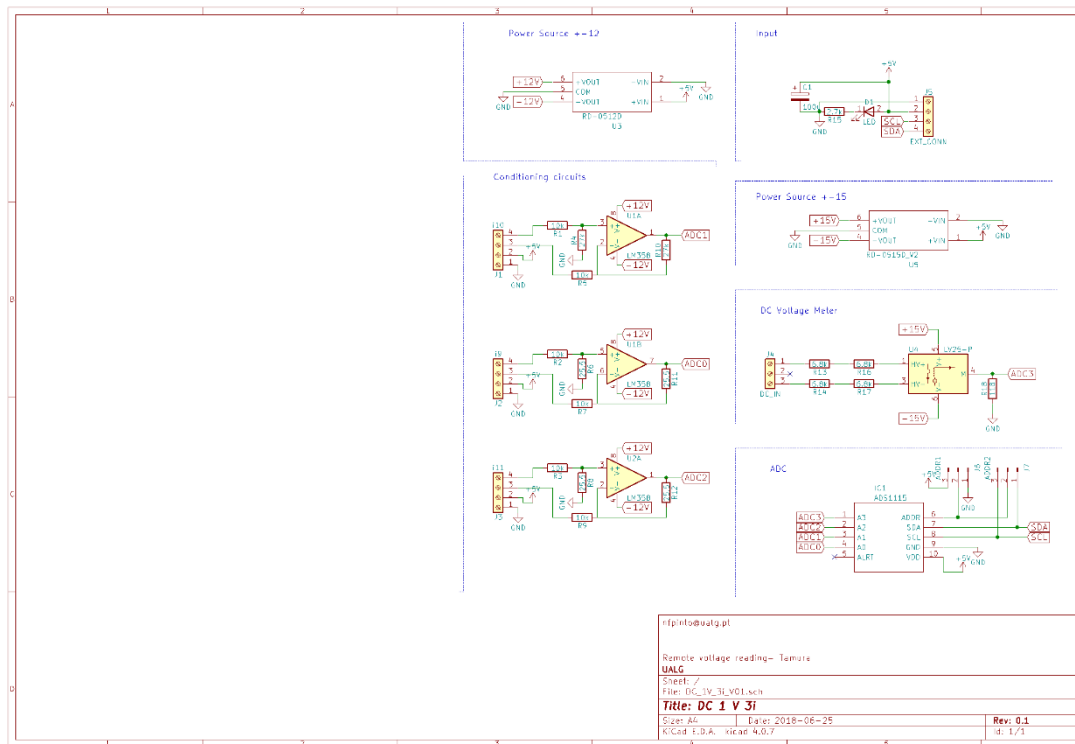
Q. ESQUEMÁTICO DA PLACA DC_4I_V01



S. ESQUEMÁTICO DA PLACA DC_1V_1I_V01



T. ESQUEMÁTICO DA PLACA DC_1V_3I_V01



U. CONFIGURAÇÕES UTILIZADAS NO NS E AS

As seguintes configurações foram utilizadas no NS e AS, na implementação descrita ao longo da tese.

U.1 SETUP DE REDE E CONFIGURAÇÕES UTILIZADAS

De forma a criar uma rede LoraWAN devem-se definir as configurações da mesma no Lorawan Server. Deve-se começar por decidir que configurações utilizar, tais como canais, chaves de segurança, configurações dos nós, filtros de *payload*, e aplicações de destino. A Figura U.1 mostra a página inicial do servidor de rede, onde se podem visualizar os nós comissionados, as mensagens recebidas ou vários eventos. Para aceder a esta página basta utilizar a URL <http://IP:8080>, onde IP representa o endereço do NS, e posteriormente inserir as credenciais de acesso (ver capítulo 4.3.3).

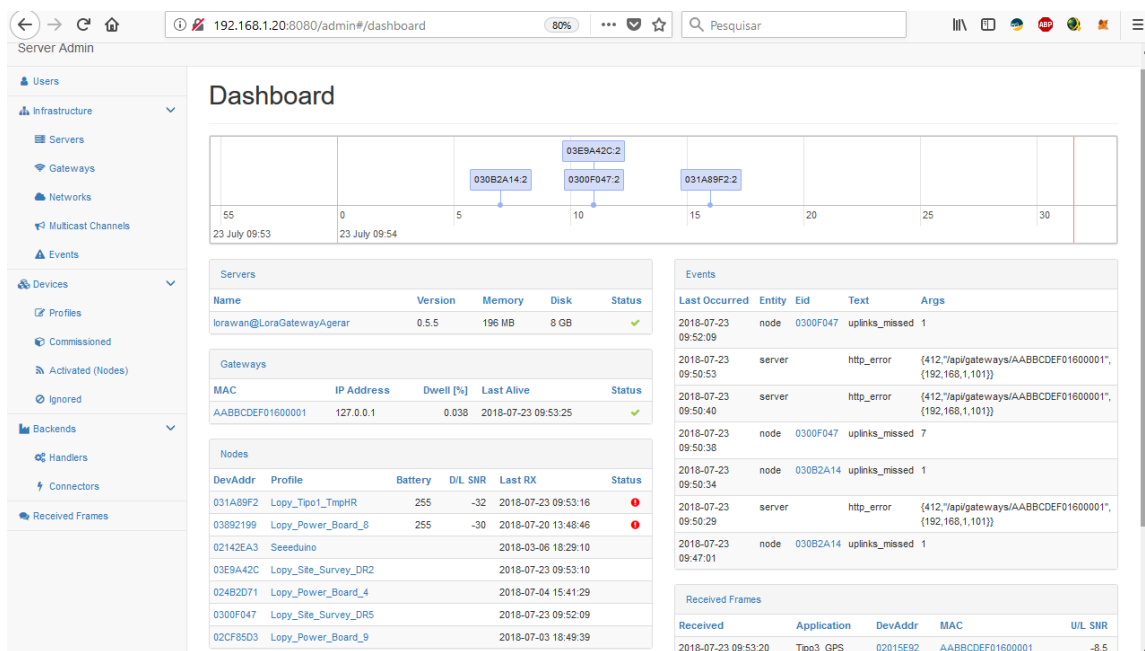


Figura U.1 - Página inicial do servidor de rede

O primeiro passo é criar a rede, sendo que este servidor permite incorporar várias redes distintas, configuradas na aba “*infrastructures* >> *networks*”. Parâmetros como a região, atrasos das janelas de receção (*RX1/2 Join Delay* e *Delay*), potência de emissão (*Gateway Power*), *coding rate*, frequências dos canais e respetivos DR (separador *Channels*, campos *Frequency* e *Min/Max DR*), e definições ADR (potências máximas e mínimas, DR máxima do canal, DRs iniciais das janelas de receção e frequência da segunda janela de receção) são aqui definidas. Posteriormente adiciona-se a *gateway* existente, na aba “*infrastructures* >>

networks”, onde se deve apenas configurar o endereço MAC atribuído à mesma (capítulo 4.3.2.5). Com este passo a infraestrutura básica da rede fica criada. A Figura U.2 mostra a *interface* destas configurações.

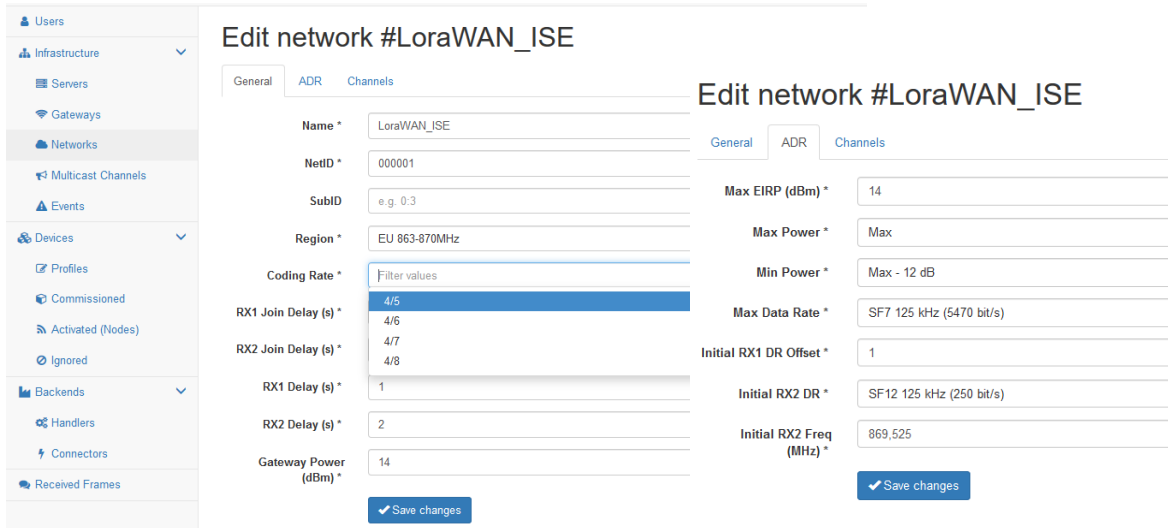


Figura U.2 - Exemplo de configurações de rede no NS

De seguida configuram-se os *backends* de rede, separados em dois componentes. Os *connectors* lidam com o envio de dados para as aplicações enquanto os *handlers* tratam do processamento das mensagens recebidas. Primeiro devem-se configurar os *handlers*, responsáveis por efetuar o tratamento dos *payloads* das mensagens LoRaWAN recebidas. Se se pretender filtrar o *payload*, deve-se configurar uma função de filtragem em linguagem Erlang, no campo *Parse Uplink*. Neste local podemos também adicionar, à mensagem a enviar às aplicações, informações de qualidade de receção (LSNR, RSSI ou FCnt, por exemplo), através do campo *Uplink Fields*. Como resultado, os dados são enviados em formato JSON aos *connectors*, já separados. Por exemplo, no Lopy Power Board 2, lemos 2 tensões e 2 correntes. No *payload* do nó, enviamos 8 bytes de dados, na sequência “u8, i14, u14, i21” em formato *little endian* e inteiro sem sinal. Adicionalmente, enviamos 1 *byte* de estado do nó. O NS filtra o *payload* recebido, separando quatro grupos de 16 *bits* e um de 8 *bits*. Depois, atribui a cada um destes grupos, um nome (a identificação “INTA”) e efetua alguma operação matemática necessária. Os primeiros 16 *bits* correspondem a u8, e devem ser neste local divididos por 100, para compensar a operação realizada na emissão, obtendo o valor correto. Ao ser enviado para o *connector* já segue no formato JSON como:

“..., “u8”:VALOR, ...”

Adicionalmente, os campos de qualidade também seguem em formato JSON na mesma mensagem, por exemplo:

“..., “*rssi*”:-99, “*lsnr*”:3.2, “*freq*”:868.1, ...“

De seguida, configuram-se os **connectors**, responsáveis pelo envio dos dados processados pelos **handlers**, para as aplicações escolhidas. No nosso caso, os dados seriam enviados para o Thingsboard via MQTT, sendo aqui configurados os tópicos de publicação (*Publish Uplinks*), o endereço do AS (em *URI*), o formato dos dados (JSON, no nosso caso) e o **handler** associado. Uma vez que precisamos do **handler** nesta configuração, o mesmo deve começar por ser criado. No caso da rede que estamos a projetar, cada Lopy Power Board tem um formato de *payload* distinto, sendo por isso necessário criar um conjunto **handler e connector** para cada placa. No caso em que existam *payloads* idênticos, não é necessário criar duplicações destes, reutilizando-os nos perfis de nó.

Os nós (na aba **Devices**) são a última coisa que deve ser configurada, sendo divididos em perfis (**Profiles**) e comissionados (**Commissioned**). Os perfis referem-se a um determinado perfil de *hardware* LoraWAN, sendo úteis para reutilizar configurações de nós idênticos, em termos de emissor. Para cada perfil configura-se a rede a utilizar (*Network*), o **handler** a utilizar (*Application*) e as configurações ADR (na aba *ADR*). Posteriormente, configuram-se os nós individuais associados a um perfil, na aba **Commissioned**. Aqui colocam-se as chaves necessárias (*DevEUI*, *AppKey* e *AppEUI*), que são igualmente configuradas nos nós. Adicionalmente, é neste ponto que se configura o *token* de acesso Thingsboard, no campo *App Arguments*. No nosso caso, os emissores são todos idênticos (Lopy), no entanto os *payloads* distintos não permitem reutilizar o perfil.

Para configuração do servidor de aplicações, basta aceder ao seu endereço IP, e inserir as credenciais escolhidas (cap. 4.3.4). No mesmo apenas precisamos de criar o dispositivo na aba “**Devices**”, atribuindo-lhe um nome e posteriormente ir à aba “**Devices >> Nome_do_Dispositivo >> Manage Credentials**” e definir no campo *Access Token*, o mesmo valor configurado em *App Arguments* do NS. A Figura U.3 mostra a *interface* da plataforma, onde se podem visualizar alguns dispositivos criados, assim como os detalhes existentes no dispositivo Lopy Power Board 1.

Os subcapítulos seguintes, resumem todas as configurações utilizadas no NS e AS, para a rede de teste da INTA.

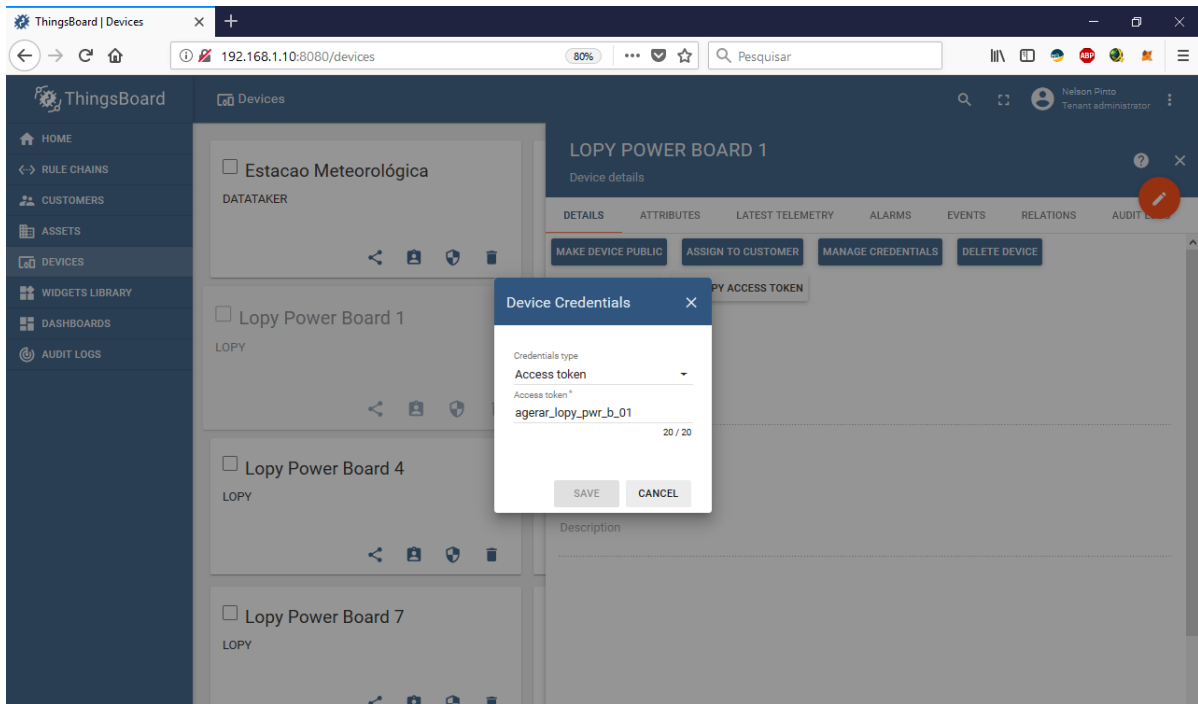


Figura U.3 - Vista de configuração do token, na plataforma Thingsboard

U.2 CONFIGURAÇÕES DE REDE

```
#####
#
#   Configurações Lorawan server NETWORK
#
#####

@general
Name:           LoraWAN_ISE
NetID:          000001
Region:         EU 863-870
Coding rate:    4/5
RX1 Join delay: 5
Rx2 Join delay: 6
RX1 Delay:      1
RX2 Delay:      2
Gateway Power:  14

@ADR
MAX EIRP:       14
Max Power:      MAX
Min Power:      MAX-12dB
Max DR:         SF7125K
Ini RX1 DR off: 1
Init RX2 DR:    SF12125K
Init RX2 Freq:  869,525

@Channels
Init chann:     0-7
Channels Freq:  867,1  867,3  867,5  867,7  867,9
Min DR:         0      0      0      0      0
Max DR:         5      5      5      5      5
```

U.3 CONFIGURAÇÕES DE GATEWAY

```
#####
#
```

```
# Configurações Lorawan server Gateway
#
#####
MAC                AABBCDEF01600001
TX Chain           0
```

U.4 CONFIGURAÇÕES DE HANDLERS

```
#####
#
# Configurações Lorawan server HANDLERS
#
#####
```

```
@Handler
Name                Lopy_Power_Board_1
Uplink Fields       deveui, datetime, fcnt, lsnr, rssi, freq, datr
Parse Uplink:
fun(Fields, <<Analiser_1_v_avg:16/little-unsigned-integer, Analiser_1_i_avg:16/little-
unsigned-integer, Analiser_2_v_avg:16/little-unsigned-integer, Analiser_2_i_avg:16/little-
unsigned-integer, Analiser_3_v_avg:16/little-unsigned-integer, Analiser_3_i_avg:16/little-
unsigned-integer, Status:8/little-unsigned-integer>>) ->
Fields#{ u5 => Analiser_1_v_avg/100, i9 => Analiser_1_i_avg/1000, u9 => Analiser_2_v_avg/100,
i16 => Analiser_2_i_avg/1000, u15 => Analiser_3_v_avg/100, i24 => Analiser_3_i_avg/1000,
status => Status }
end.

D/L Expires:       Never
```

```
@Handler
Name                Lopy_Power_Board_2
Uplink Fields       deveui, datetime, fcnt, lsnr, rssi, freq, datr
Parse Uplink:
fun(Fields, << Analiser_1_v_avg:16/little-unsigned-integer, Analiser_1_i_avg:16/little-
unsigned-integer, Analiser_2_v_avg:16/little-unsigned-integer, Analiser_2_i_avg:16/little-
unsigned-integer, Status:8/little-unsigned-integer >>) ->
Fields#{ u8 => Analiser_1_v_avg/100, i14 => Analiser_1_i_avg/1000, u14 =>
Analiser_2_v_avg/100, i21 => Analiser_2_i_avg/1000, status => Status }
end.

D/L Expires:       Never
```

```
@Handler
Name                Lopy_Power_Board_3
Uplink Fields       deveui, datetime, fcnt, lsnr, rssi, freq, datr
Parse Uplink:
fun(Fields, << Analiser_1_v_avg:16/little-unsigned-integer, Analiser_1_i_avg:16/little-
unsigned-integer, Analiser_2_v_avg:16/little-unsigned-integer, Analiser_2_i_avg:16/little-
unsigned-integer, Status:8/little-unsigned-integer >>) ->
Fields#{ u12 => Analiser_1_v_avg/100, i19 => Analiser_1_i_avg/1000, u13 =>
Analiser_2_v_avg/100, i20 => Analiser_2_i_avg/1000, status => Status }
end.

D/L Expires:       Never
```

```
@Handler
Name                Lopy_Power_Board_4
Uplink Fields       deveui, datetime, fcnt, lsnr, rssi, freq, datr
Parse Uplink:
fun(Fields, << Analiser_1_v_avg:16/little-unsigned-integer, Analiser_1_i_avg:16/little-
unsigned-integer, Analiser_2_v_avg:16/little-unsigned-integer, Analiser_2_i_avg:16/little-
unsigned-integer, Analiser_3_v_avg:16/little-unsigned-integer, Analiser_3_i_avg:16/little-
unsigned-integer, Status:8/little-unsigned-integer >>) ->
Fields#{ u4 => Analiser_1_v_avg/100, i7 => Analiser_1_i_avg/1000, u10 =>
Analiser_2_v_avg/100, i17 => Analiser_2_i_avg/1000, u11 => Analiser_3_v_avg/100, i18 =>
Analiser_3_i_avg/1000, status => Status }
end.

D/L Expires:       Never
```

```

@Handler
#NOTE: i9,i10,i11 /100
Name           Lopy_Power_Board_5
Uplink Fields  deveui, datetime, fcnt, lsnr, rssi, freq, datr
Parse Uplink:
fun(Fields, << Analiser_1_v_avg:16/little-unsigned-integer, Analiser_2_v_avg:16/little-
unsigned-integer, Analiser_2_i_avg:16/little-unsigned-integer, Analiser_3_v_avg:16/little-
unsigned-integer, Analiser_3_i_avg:16/little-unsigned-integer, Analiser_4_v_avg:16/little-
unsigned-integer, Analiser_4_i_1_avg:16/little-unsigned-integer,
Analiser_4_i_2_avg:16/little-unsigned-integer, Analiser_4_i_3_avg:16/little-unsigned-integer,
Status:8/little-unsigned-integer >>) ->
Fields#{ u1 => Analiser_1_v_avg/100, u2 => Analiser_2_v_avg/100, i3 => Analiser_2_i_avg/1000,
u3 => Analiser_3_v_avg/100, i5 => Analiser_3_i_1_avg/1000, u7 => Analiser_4_v_avg/100, i9 =>
Analiser_4_i_1_avg/100, i10 => Analiser_3_i_2_avg/100, i11 => Analiser_3_i_3_avg/100, status
=> Status }
end.

D/L Expires:   Never

```

```

@Handler
Name           Lopy_Power_Board_6
Uplink Fields  deveui, datetime, fcnt, lsnr, rssi, freq, datr
Parse Uplink:
fun(Fields, << Analiser_1_i_1_avg:16/little-unsigned-integer, Analiser_1_i_2_avg:16/little-
unsigned-integer, Analiser_1_i_3_avg:16/little-unsigned-integer,
Analiser_1_i_4_avg:16/little-unsigned-integer, Analiser_2_i_1_avg:16/little-unsigned-integer,
Analiser_2_i_2_avg:16/little-unsigned-integer, Analiser_2_i_3_avg:16/little-unsigned-integer,
Status:8/little-unsigned-integer >>) ->
Fields#{ i1_1 => Analiser_1_i_1_avg/1000, i1_2 => Analiser_1_i_2_avg/1000, i1_3 =>
Analiser_1_i_3_avg/1000, i1_4 => Analiser_1_i_4_avg/1000, i13_1 => Analiser_2_i_1_avg/1000,
i13_2 => Analiser_2_i_2_avg/1000, i13_3 => Analiser_2_i_3_avg/1000, status => Status }
end.

D/L Expires:   Never

```

```

@Handler
Name           Lopy_Power_Board_7
Uplink Fields  deveui, datetime, fcnt, lsnr, rssi, freq, datr
Parse Uplink:
fun(Fields, << Analiser_1_v_avg:16/little-unsigned-integer, Analiser_2_i_1_avg:16/little-
unsigned-integer, Analiser_2_i_2_avg:16/little-unsigned-integer,
Analiser_2_i_3_avg:16/little-unsigned-integer, Analiser_3_i_1_avg:16/little-unsigned-integer,
Analiser_3_i_2_avg:16/little-unsigned-integer, Analiser_3_i_3_avg:16/little-unsigned-integer,
Analiser_3_i_4_avg:16/little-unsigned-integer, Status:8/little-unsigned-integer >>) ->
Fields#{ u6 => Analiser_1_v_avg/100, i2 => Analiser_1_i_1_avg/1000, i4 =>
Analiser_1_i_2_avg/1000, i6 => Analiser_1_i_3_avg/1000, i12 => Analiser_1_i_4_avg/1000, i15
=> Analiser_2_i_1_avg/1000, i22 => Analiser_2_i_2_avg/1000, i23 => Analiser_2_i_3_avg/1000,
status => Status }
end.

D/L Expires:   Never

```

U.5 CONFIGURAÇÕES DE CONNECTORS

```

#####
#
#   Configurações Lorawan server Connectors
#
#####

#@Lopy_Power_Board_1_CONN
Name:           Lopy_Power_Board_1_CONN
Application:    Lopy_Power_Board_1
Format:         JSON
URI:            mqtt://192.168.1.10:1883
Pub Uplinks:   v1/devices/me/telemetry
Pub Events:    v1/devices/me/attributes

Auth:           User+Pas

```

```

Name:                {appargs}

#@Lopy_Power_Board_2_CONN
Name:                Lopy_Power_Board_2_CONN
Application:         Lopy_Power_Board_2
Format:              JSON
URI:                 mqtt://192.168.1.10:1883
Pub Uplinks:         v1/devices/me/telemetry
Pub Events:          v1/devices/me/attributes

Auth:                User+Pas
Name:                {appargs}

#@Lopy_Power_Board_3_CONN
Name:                Lopy_Power_Board_3_CONN
Application:         Lopy_Power_Board_3
Format:              JSON
URI:                 mqtt://192.168.1.10:1883
Pub Uplinks:         v1/devices/me/telemetry
Pub Events:          v1/devices/me/attributes

Auth:                User+Pas
Name:                {appargs}

#@Lopy_Power_Board_4_CONN
Name:                Lopy_Power_Board_4_CONN
Application:         Lopy_Power_Board_4
Format:              JSON
URI:                 mqtt://192.168.1.10:1883
Pub Uplinks:         v1/devices/me/telemetry
Pub Events:          v1/devices/me/attributes

Auth:                User+Pas
Name:                {appargs}

#@Lopy_Power_Board_5_CONN
Name:                Lopy_Power_Board_5_CONN
Application:         Lopy_Power_Board_5
Format:              JSON
URI:                 mqtt://192.168.1.10:1883
Pub Uplinks:         v1/devices/me/telemetry
Pub Events:          v1/devices/me/attributes

Auth:                User+Pas
Name:                {appargs}

#@Lopy_Power_Board_6_CONN
Name:                Lopy_Power_Board_6_CONN
Application:         Lopy_Power_Board_6
Format:              JSON
URI:                 mqtt://192.168.1.10:1883
Pub Uplinks:         v1/devices/me/telemetry
Pub Events:          v1/devices/me/attributes

Auth:                User+Pas
Name:                {appargs}

#@Lopy_Power_Board_7_CONN
Name:                Lopy_Power_Board_7_CONN
Application:         Lopy_Power_Board_7
Format:              JSON
URI:                 mqtt://192.168.1.10:1883
Pub Uplinks:         v1/devices/me/telemetry
Pub Events:          v1/devices/me/attributes

Auth:                User+Pas
Name:                {appargs}

```

U.6 CONFIGURAÇÕES DE PERFIS DE DISPOSITIVOS

```
#####  
#  
#   Configurações Lorawan server PROFILES  
#  
#####  
  
#@Lopy_Power_Board_1  
@general  
Name:           Lopy_Power_Board_1  
Network:        LoraWAN_ISE  
Application:    Lopy_Power_Board_1  
Can Join?      true  
FCNT Check:    Strict 32-bit_length  
Tx Windows     Auto  
@ADR  
ADR             Auto-Adjust  
Set Power      14dBm  
Set DR         SF7125K      (LoraWAN DR5)  
Max DR         SF7125K  
Set Channels   0-7  
Set Rx1DROFFSET 1  
Set RX2DR     SF12125K      (LoraWAN DR0)  
Set RX2 Freq   869,525  
Req Status     false  
  
#@Lopy_Power_Board_2  
@general  
Name:           Lopy_Power_Board_2  
Network:        LoraWAN_ISE  
Application:    Lopy_Power_Board_2  
Can Join?      true  
FCNT Check:    Strict 32-bit_length  
Tx Windows     Auto  
@ADR  
ADR             Auto-Adjust  
Set Power      14dBm  
Set DR         SF7125K      (LoraWAN DR5)  
Max DR         SF7125K  
Set Channels   0-7  
Set Rx1DROFFSET 1  
Set RX2DR     SF12125K      (LoraWAN DR0)  
Set RX2 Freq   869,525  
Req Status     false  
  
#@Lopy_Power_Board_3  
@general  
Name:           Lopy_Power_Board_3  
Network:        LoraWAN_ISE  
Application:    Lopy_Power_Board_3  
Can Join?      true  
FCNT Check:    Strict 32-bit_length  
Tx Windows     Auto  
@ADR  
ADR             Auto-Adjust  
Set Power      14dBm  
Set DR         SF7125K      (LoraWAN DR5)  
Max DR         SF7125K  
Set Channels   0-7  
Set Rx1DROFFSET 1  
Set RX2DR     SF12125K      (LoraWAN DR0)  
Set RX2 Freq   869,525  
Req Status     false  
  
#@Lopy_Power_Board_4  
@general  
Name:           Lopy_Power_Board_4  
Network:        LoraWAN_ISE  
Application:    Lopy_Power_Board_4  
Can Join?      true  
FCNT Check:    Strict 32-bit_length
```

```
Tx Windows      Auto
@ADR
ADR             Auto-Adjust
Set Power       14dBm
Set DR          SF7125K      (LoraWAN DR5)
Max DR          SF7125K
Set Channels    0-7
Set Rx1DROFFSET 1
Set RX2DR       SF12125K      (LoraWAN DR0)
Set RX2 Freq    869,525
Req Status      false
```

```
##@Lopy_Power_Board_5
```

```
@general
```

```
Name:          Lopy_Power_Board_5
Network:       LoraWAN_ISE
Application:   Lopy_Power_Board_5
Can Join?     true
FCNT Check:   Strict 32-bit_length
Tx Windows    Auto
```

```
@ADR
```

```
ADR             Auto-Adjust
Set Power       14dBm
Set DR          SF7125K      (LoraWAN DR5)
Max DR          SF7125K
Set Channels    0-7
Set Rx1DROFFSET 1
Set RX2DR       SF12125K      (LoraWAN DR0)
Set RX2 Freq    869,525
Req Status      false
```

```
##@Lopy_Power_Board_6
```

```
@general
```

```
Name:          Lopy_Power_Board_6
Network:       LoraWAN_ISE
Application:   Lopy_Power_Board_6
Can Join?     true
FCNT Check:   Strict 32-bit_length
Tx Windows    Auto
```

```
@ADR
```

```
ADR             Auto-Adjust
Set Power       14dBm
Set DR          SF7125K      (LoraWAN DR5)
Max DR          SF7125K
Set Channels    0-7
Set Rx1DROFFSET 1
Set RX2DR       SF12125K      (LoraWAN DR0)
Set RX2 Freq    869,525
Req Status      false
```

```
##@Lopy_Power_Board_7
```

```
@general
```

```
Name:          Lopy_Power_Board_7
Network:       LoraWAN_ISE
Application:   Lopy_Power_Board_7
Can Join?     true
FCNT Check:   Strict 32-bit_length
Tx Windows    Auto
```

```
@ADR
```

```
ADR             Auto-Adjust
Set Power       14dBm
Set DR          SF7125K      (LoraWAN DR5)
Max DR          SF7125K
Set Channels    0-7
Set Rx1DROFFSET 1
Set RX2DR       SF12125K      (LoraWAN DR0)
Set RX2 Freq    869,525
Req Status      false
```

U.7 CONFIGURAÇÕES DOS NÓS NO NS E AS

```
#####
#
#   Configurações dos Nós no thingsboard + Lorawan server
#
#####

#Lopy_Power_Board_1      14 - SN-1651002065      DevEUI 70b3d5499f5676dd
@Lora_Server Config
@Commissioned
@Create_Device
DevEUI:      70b3d5499f5676dd
Profile:     Lopy_power_Board_1
App arguments:  agerar_lopy_pwr_b_01
AppEUI:     AABBCDEF01600001
Appkey:     AAAABBBBCDEF01600001001234567890
Description: Parque Veiculos. 3x SDM230 +@ PV4-AC, SCK1, INV_EV

@ThingsBoard Config
Nome nó:     Lopy Power Board 1
Acess Token: agerar_lopy_pwr_b_01
Description: Parque Veiculos. 3x SDM230 +@ PV4-AC, SCK1, INV_EV

#Lopy_Power_Board_2      13 - SN-1651002485      DevEUI 70b3d54990f95be4
@Lora_Server Config
@Commissioned
@Create_Device
DevEUI:      70b3d54990f95be4
Profile:     Lopy_power_Board_2
App arguments:  agerar_lopy_pwr_b_02
AppEUI:     AABBCDEF01600001
Appkey:     AAAABBBBCDEF01600001001234567890
Description:  Armário Micro Rede. 1x SDM230 + 1x SDM530 @ INV DC/AC, AEROGERADOR

@ThingsBoard Config
Nome nó:     Lopy Power Board 2
Acess Token: agerar_lopy_pwr_b_02
Description:  Armário Micro Rede. 1x SDM230 + 1x SDM530 @ INV DC/AC, AEROGERADOR

#Lopy_Power_Board_3      12 - SN-1651002094      DevEUI 70b3d54995e2e7e4
@Lora_Server Config
@Commissioned
@Create_Device
DevEUI:      70b3d54995e2e7e4
Profile:     Lopy_power_Board_3
App arguments:  agerar_lopy_pwr_b_03
AppEUI:     AABBCDEF01600001
Appkey:     AAAABBBBCDEF01600001001234567890
Description:  Quadro Elétrico. 2x SDM530 @ IN, IN_SAI

@ThingsBoard Config
Nome nó:     Lopy Power Board 3
Acess Token: agerar_lopy_pwr_b_03
Description:  Quadro Elétrico. 2x SDM530 @ IN, IN_SAI

#Lopy_Power_Board_4      11 - SN-1651002093      DevEUI 70b3d5499178b9f9
@Lora_Server Config
@Commissioned
@Create_Device
DevEUI:      70b3d5499178b9f9
Profile:     Lopy_power_Board_4
App arguments:  agerar_lopy_pwr_b_04
AppEUI:     AABBCDEF01600001
Appkey:     AAAABBBBCDEF01600001001234567890
Description:  Parque Veiculos. 2x SDM530 + DC_1V_1i @ SKT2, SKT3, PV4

@ThingsBoard Config
Nome nó:     Lopy Power Board 4
Acess Token: agerar_lopy_pwr_b_04
Description:  Parque Veiculos. 2x SDM530 + DC_1V_1i @ SKT2, SKT3, PV4
```

```

#Lopy_Power_Board_5      10 - SN-1651002070      DevEUI 70b3d54995374f4b
@Commissioned
@Create_Device
DevEUI:      70b3d54995374f4b
Profile:     Lopy_power_Board_5
App arguments:  agerar_lopy_pwr_b_05
AppEUI:     AABBCDEF01600001
Appkey:     AAAABBBBCDEF01600001001234567890
Description: Armario Micro Rede. DC_1V + 2x DC_1V_1i + DC_1V_3i @ PV1, PV2, PV3, Pb-Ac,
Li-Ion

@ThingsBoard Config
Nome nó:     Lopy Power Board 5
Access Token: agerar_lopy_pwr_b_05
Description: Armario Micro Rede. DC_1V + 2x DC_1V_1i + DC_1V_3i @ PV1, PV2, PV3, Pb-Ac,
Li-Ion

#Lopy_Power_Board_6      9 - SN-1651002077      DevEUI 70b3d54996e87f31
@Lora_Server Config
@Commissioned
@Create_Device
DevEUI:      70b3d54996e87f31
Profile:     Lopy_power_Board_6
App arguments:  agerar_lopy_pwr_b_06
AppEUI:     AABBCDEF01600001
Appkey:     AAAABBBBCDEF01600001001234567890
Description:  Armario Micro Rede. 2x DC_4i @ PV1, DC/AC OUT

@ThingsBoard Config
Nome nó:     Lopy Power Board 6
Access Token: agerar_lopy_pwr_b_06
Description:  Armario Micro Rede. 2x DC_4i @ PV1, DC/AC OUT

#Lopy_Power_Board_7      8 - SN-1651002078      DevEUI 70b3d5499bc5cfc8
@Lora_Server Config
@Commissioned
@Create_Device
DevEUI:      70b3d5499bc5cfc8
Profile:     Lopy_power_Board_7
App arguments:  agerar_lopy_pwr_b_07
AppEUI:     AABBCDEF01600001
Appkey:     AAAABBBBCDEF01600001001234567890
Description:  Armario Micro Rede. 2x DC_4i + DC_1V @ PV1 DC/DC, PV2 DC/DC, PV3 DC/DC,
Eletrolyzer, DC Load, DC Source, AERO_AC/DC, Pb-Ac

@ThingsBoard Config
Nome nó:     Lopy Power Board 7
Access Token: agerar_lopy_pwr_b_07
Description:  Armario Micro Rede. 2x DC_4i + DC_1V @ PV1 DC/DC, PV2 DC/DC, PV3 DC/DC,
Eletrolyzer, DC Load, DC Source, AERO_AC/DC, Pb-Ac

```

V. VALORES OBTIDOS NAS LEITURAS DOS MÓDULOS DE TENSÃO

As tabelas seguintes apresentam os valores obtidos nos ensaios dos módulos de leitura de tensão.

V.1 LI-ION DC_1V_3I

Li-Ion Teórico														
Teórico			Lido Medição 1						Lido Medição 2					
Vp (V)	V_Rs(V)	ADC	Vp (V)	ADC	Ip (mA)	Vp_calc (V)	Err ABS (V)	Err Relat(%)	Vp (V)	ADC	Ip (mA)	Vp_calc (V)	Err ABS (V)	Err Relat(%)
0	0.000	0	0	9	0	0.10	-0.10	0.00	0	-26	0	-0.30	0.30	0.00
50	0.542	4338	50.3	4379	1.85	51.03	-0.73	1.46	50.4	4349	1.85	50.68	-0.28	0.56
100	1.085	8676	100.3	8721	3.69	101.63	-1.33	1.33	100.6	8707	3.7	101.47	-0.87	0.87
150	1.627	13015	150.8	13098	5.55	152.64	-1.84	1.22	150.3	13033	5.53	151.89	-1.59	1.06
200	2.169	17353	201.2	17463	7.4	203.51	-2.31	1.15	200.9	17421	7.38	203.03	-2.13	1.06
250	2.711	21691	250.7	21674	9.21	252.59	-1.89	0.75	250.1	21671	9.19	252.55	-2.45	0.98
273	2.961	23687	273.7	23740	10.05	276.67	-2.97	1.08	273.8	23721	10.06	276.45	-2.65	0.97
300	3.254	26029	300.6	26072	11.04	303.84	-3.24	1.08	301.6	26137	11.08	304.60	-3.00	1.00

Tabela V.1 - Valores obtidos no módulo DC_1V_3i, antes da calibração

Li-Ion Real														
Teórico			Lido Medição 1						Lido Medição 2					
Vp (V)	V_Rs(V)	ADC	Vp (V)	ADC	Ip (mA)	Vp_calc (V)	Err ABS (V)	Err Relat(%)	Vp (V)	ADC	Ip (mA)	Vp_calc (V)	Err ABS (V)	Err Relat(%)
0	0.000	0	0	9	0	0.10	-0.10	0.00	0	-26	0	-0.30	0.30	0.00
50	0.542	4338	50.3	4379	1.85	50.39	-0.09	0.18	50.4	4349	1.85	50.04	0.36	0.71
100	1.085	8676	100.3	8721	3.69	100.35	-0.05	0.05	100.6	8707	3.7	100.19	0.41	0.41
150	1.627	13015	150.8	13098	5.55	150.72	0.08	0.05	150.3	13033	5.53	149.97	0.33	0.22
200	2.169	17353	201.2	17463	7.4	200.95	0.25	0.13	200.9	17421	7.38	200.47	0.43	0.22
250	2.711	21691	250.7	21674	9.21	249.40	1.30	0.52	250.1	21671	9.19	249.37	0.73	0.29
273	2.961	23687	273.7	23740	10.05	273.18	0.52	0.19	273.8	23721	10.06	272.96	0.84	0.31
300	3.254	26029	300.6	26072	11.04	300.01	0.59	0.20	301.6	26137	11.08	300.76	0.84	0.28

Tabela V.2 - Valores obtidos no módulo DC_1V_3i, após calibração

V.2 DC BUS DC_1V

DC BUS Teórico														
Teórico			Medição 1						Medição 2					
Vp (V)	V_Rs(V)	ADC	Vp (V)	ADC	Ip (mA)	Vp_calc (V)	Err ABS (V)	Err Relat(%)	Vp (V)	ADC	Ip (mA)	Vp_calc (V)	Err ABS (V)	Err Relat(%)
0	0.000	0	0	19	0	0.32	-0.32	0.00	0	8	0	0.14	-0.14	0.00
50	0.369	2950	50.3	3001	1.26	51.30	-1.00	1.98	50.1	2971	1.26	50.78	-0.68	1.36
100	0.738	5900	100.2	5968	2.52	102.01	-1.81	1.81	102.3	6080	2.58	103.92	-1.62	1.59
150	1.106	8850	150.6	8898	3.85	152.09	-1.49	0.99	150.3	8920	3.78	152.47	-2.17	1.44
200	1.475	11800	201.6	11985	5.07	204.86	-3.26	1.62	200.6	11907	5.04	203.52	-2.92	1.46
250	1.844	14750	251	14901	6.31	254.70	-3.70	1.47	251.5	14905	6.32	254.77	-3.27	1.30
300	2.213	17700	301.7	17859	7.57	305.26	-3.56	1.18	301.3	17880	7.57	305.62	-4.32	1.43
350	2.581	20650	354	20944	8.88	357.99	-3.99	1.13	353.4	20914	8.87	357.48	-4.08	1.15
400	2.950	23600	400	23601	10	403.41	-3.41	0.85	400	23667	10.03	404.53	-4.53	1.13
408	3.009	24072	408	24023	10.19	410.62	-2.62	0.64	408	24074	10.21	411.49	-3.49	0.86
420	3.098	24780	420	24674	10.49	421.75	-1.75	0.42	420	24653	10.47	421.39	-1.39	0.33
450	3.319	26550	450	26502	11.24	452.99	-2.99	0.66	450	26511	11.25	453.14	-3.14	0.70

Tabela V.3 - Valores obtidos no módulo DC_1V, antes da calibração

DC BUS Real														
Teórico			Medição 1						Medição 2					
Vp (V)	V_Rs(V)	ADC	Vp (V)	ADC	Ip (mA)	Vp_calc (V)	Err ABS (V)	Err Relat(%)	Vp (V)	ADC	Ip (mA)	Vp_calc (V)	Err ABS (V)	Err Relat(%)
0	0.000	0	0	19	0	0.32	-0.32	0.00	0	8	0	0.14	-0.14	0.00
50	0.369	2950	50.3	3001	1.26	50.74	-0.44	0.88	50.1	2971	1.26	50.23	-0.13	0.27
100	0.738	5900	100.2	5968	2.52	100.91	-0.71	0.70	102.3	6080	2.58	102.80	-0.50	0.49
150	1.106	8850	150.6	8898	3.85	150.45	0.15	0.10	150.3	8920	3.78	150.82	-0.52	0.34
200	1.475	11800	201.6	11985	5.07	202.64	-1.04	0.52	200.6	11907	5.04	201.32	-0.72	0.36
250	1.844	14750	251	14901	6.31	251.94	-0.94	0.38	251.5	14905	6.32	252.01	-0.51	0.20
300	2.213	17700	301.7	17859	7.57	301.96	-0.26	0.09	301.3	17880	7.57	302.31	-1.01	0.34
350	2.581	20650	354	20944	8.88	354.12	-0.12	0.03	353.4	20914	8.87	353.61	-0.21	0.06
400	2.950	23600	400	23601	10	399.04	0.96	0.24	400	23667	10.03	400.16	-0.16	0.04
408	3.009	24072	408	24023	10.19	406.18	1.82	0.45	408	24074	10.21	407.04	0.96	0.24
420	3.098	24780	420	24674	10.49	417.19	2.81	0.67	420	24653	10.47	416.83	3.17	0.75
450	3.319	26550	450	26502	11.24	448.09	1.91	0.42	450	26511	11.25	448.25	1.75	0.39

Tabela V.4 - Valores obtidos no módulo DC_1V, após calibração

V.3 DC_1V_1i

PV1 Teórico														
Teórico			Lido Medição 1						Lido Medição 2					
Vp (V)	V_Rs(V)	ADC	Vp (V)	ADC	Ip (mA)	Vp_calc (V)	Err ABS (V)	Err Relat(%)	Vp (V)	ADC	Ip (mA)	Vp_calc (V)	Err ABS (V)	Err Relat(%)
0	0.000	0	0	-36	0	-0.49	0.49	0.00	0	-40	0	-0.55	0.55	0.00
50	0.461	3688	50.2	3538	1.51	48.45	1.75	3.49	49.9	3508	1.5	48.04	1.86	3.73
100	0.922	7375	100.2	7096	3.03	97.17	3.03	3.02	100.2	7095	3.02	97.16	3.04	3.04
150	1.383	11063	150.3	10667	4.54	146.07	4.23	2.82	150.2	10661	4.54	145.99	4.21	2.81
200	1.844	14750	200.5	14240	6.06	195.00	5.50	2.75	201.4	14298	6.08	195.79	5.61	2.79
250	2.305	18438	250.5	17800	7.57	243.75	6.75	2.70	251	17821	7.58	244.03	6.97	2.78
300	2.766	22125	300.7	21366	9.08	292.58	8.12	2.70	300.7	21375	9.09	292.70	8.00	2.66
320	2.950	23600	320.1	22753	9.66	311.57	8.53	2.67	320.7	22782	9.69	311.97	8.73	2.72
350	3.227	25813	350	24897	10.57	340.93	9.07	2.59	350.4	24893	10.58	340.87	9.53	2.72
400	3.688	29500	401	28510	12.12	390.40	10.60	2.64	399	28343	12.04	388.12	10.88	2.73

Tabela V.5 - Valores obtidos no módulo DC_1V_1i PV1, antes da calibração

PV1 Real														
Teórico			Lido Medição 1						Lido Medição 2					
Vp (V)	V_Rs(V)	ADC	Vp (V)	ADC	Ip (mA)	Vp_calc (V)	Err ABS (V)	Err Relat(%)	Vp (V)	ADC	Ip (mA)	Vp_calc (V)	Err ABS (V)	Err Relat(%)
0	0.000	0	0	-36	0	-0.51	0.51	0.00	0	-40	0	-0.56	0.56	0.00
50	0.461	3688	50.2	3538	1.51	49.68	0.52	1.03	49.9	3508	1.5	49.26	0.64	1.28
100	0.922	7375	100.2	7096	3.03	99.64	0.56	0.56	100.2	7095	3.02	99.63	0.57	0.57
150	1.383	11063	150.3	10667	4.54	149.79	0.51	0.34	150.2	10661	4.54	149.70	0.50	0.33
200	1.844	14750	200.5	14240	6.06	199.96	0.54	0.27	201.4	14298	6.08	200.78	0.62	0.31
250	2.305	18438	250.5	17800	7.57	249.95	0.55	0.22	251	17821	7.58	250.25	0.75	0.30
300	2.766	22125	300.7	21366	9.08	300.03	0.67	0.22	300.7	21375	9.09	300.15	0.55	0.18
320	2.950	23600	320.1	22753	9.66	319.50	0.60	0.19	320.7	22782	9.69	319.91	0.79	0.25
350	3.227	25813	350	24897	10.57	349.61	0.39	0.11	350.4	24893	10.58	349.55	0.85	0.24
400	3.688	29500	401	28510	12.12	400.34	0.66	0.16	399	28343	12.04	398.00	1.00	0.25

Tabela V.6 - Valores obtidos no módulo DC_1V_1i PV1, após calibração

V.4 PV2 DC_1V_1i

PV2 Teórico														
Teórico			Lido Medição 1						Lido Medição 2					
Vp (V)	V_Rs(V)	ADC	Vp (V)	ADC	Ip (mA)	Vp_calc (V)	Err ABS (V)	Err Relat(%)	Vp (V)	ADC	Ip (mA)	Vp_calc (V)	Err ABS (V)	Err Relat(%)
0	0.000	0	0	-8	0	-0.11	0.11	0.00	0	7	0	0.10	-0.10	0.00
50	0.461	3688	50.8	3626	1.54	49.65	1.15	2.26	50.2	3583	1.51	49.06	1.14	2.26
100	0.922	7375	100.4	7169	3.04	98.17	2.23	2.22	100.4	7174	3.03	98.24	2.16	2.15
150	1.383	11063	150.6	10760	4.55	147.34	3.26	2.16	150.3	10746	4.55	147.15	3.15	2.10
200	1.844	14750	200.6	14353	6.07	196.54	4.06	2.02	200.2	14309	6.05	195.94	4.26	2.13
250	2.305	18438	251.1	17942	7.6	245.69	5.41	2.15	250.1	17887	7.57	244.94	5.16	2.06
300	2.766	22125	300.1	21450	9.09	293.73	6.37	2.12	300.5	21476	9.08	294.08	6.42	2.14
320	2.950	23600	320.9	22914	9.69	313.77	7.13	2.22	320.6	22897	9.69	313.54	7.06	2.20
350	3.227	25813	350.7	25041	10.61	342.90	7.80	2.22	350.6	25040	10.6	342.89	7.71	2.20
400	3.688	29500	400	28541	12.07	390.83	9.17	2.29	400	28548	12.08	390.92	9.08	2.27

Tabela V.7 - Valores obtidos no módulo DC_1V_1i PV2, antes da calibração

PV2 Real														
Teórico			Lido Medição 1						Lido Medição 2					
Vp (V)	V_Rs(V)	ADC	Vp (V)	ADC	Ip (mA)	Vp_calc (V)	Err ABS (V)	Err Relat(%)	Vp (V)	ADC	Ip (mA)	Vp_calc (V)	Err ABS (V)	Err Relat(%)
0	0.000	0	0	-8	0	-0.11	0.11	0.00	0	7	0	0.10	-0.10	0.00
50	0.461	3688	50.8	3626	1.54	50.69	0.11	0.22	50.2	3583	1.51	50.09	0.11	0.23
100	0.922	7375	100.4	7169	3.04	100.21	0.19	0.19	100.4	7174	3.03	100.28	0.12	0.12
150	1.383	11063	150.6	10760	4.55	150.41	0.19	0.13	150.3	10746	4.55	150.21	0.09	0.06
200	1.844	14750	200.6	14353	6.07	200.64	-0.04	0.02	200.2	14309	6.05	200.02	0.18	0.09
250	2.305	18438	251.1	17942	7.6	250.81	0.29	0.12	250.1	17887	7.57	250.04	0.06	0.03
300	2.766	22125	300.1	21450	9.09	299.84	0.26	0.09	300.5	21476	9.08	300.21	0.29	0.10
320	2.950	23600	320.9	22914	9.69	320.31	0.59	0.18	320.6	22897	9.69	320.07	0.53	0.17
350	3.227	25813	350.7	25041	10.61	350.04	0.66	0.19	350.6	25040	10.6	350.03	0.57	0.16
400	3.688	29500	400	28541	12.07	398.97	1.03	0.26	400	28548	12.08	399.06	0.94	0.23

Tabela V.8 - Valores obtidos no módulo DC_1V_1i PV2, após calibração

V.5 PV3 DC_1V_1i

PV3 Teórico														
Teórico			Lido Medição 1						Lido Medição 2					
Vp (V)	V_Rs(V)	ADC	Vp (V)	ADC	Ip (mA)	Vp_cal (V)	Err ABS (V)	Err Relat(%)	Vp (V)	ADC	Ip (mA)	Vp_cal (V)	Err ABS (V)	Err Relat(%)
0	0.000	0	0	23	0	0.31	-0.31	0.00	0	85	0	1.16	-1.16	0.00
50	0.461	3688	50.7	3658	1.54	50.09	0.61	1.20	50.1	3685	1.52	50.46	-0.36	0.72
100	0.922	7375	100.8	7267	3.07	99.51	1.29	1.28	100.8	7322	3.07	100.26	0.54	0.53
150	1.383	11063	150.9	10861	4.59	148.73	2.17	1.44	150.6	10901	4.58	149.27	1.33	0.88
200	1.844	14750	200.9	14468	6.11	198.12	2.78	1.38	200.7	14512	6.11	198.72	1.98	0.99
250	2.305	18438	250.1	17982	7.61	246.24	3.86	1.54	250.1	18053	7.61	247.21	2.89	1.16
300	2.766	22125	300.4	21606	9.15	295.86	4.54	1.51	300.9	21703	9.16	297.19	3.71	1.23
320	2.950	23600	320.5	23035	9.75	315.43	5.07	1.58	320.1	23089	9.73	316.17	3.93	1.23
350	3.227	25813	350.5	25212	10.66	345.24	5.26	1.50	350.4	25264	10.67	345.95	4.45	1.27
400	3.688	29500	400	28722	12.15	393.31	6.69	1.67	400	28800	12.14	394.37	5.63	1.41

Tabela V.9 - Valores obtidos no módulo DC_1V_1i PV3, antes da calibração

PV3 Real														
Teórico			Lido Medição 1						Lido Medição 2					
Vp (V)	V_Rs(V)	ADC	Vp (V)	ADC	Ip (mA)	Vp_cal (V)	Err ABS (V)	Err Relat(%)	Vp (V)	ADC	Ip (mA)	Vp_cal (V)	Err ABS (V)	Err Relat(%)
0	0.000	0	0	23	0	0.32	-0.32	0.00	0	85	0	1.18	-1.18	0.00
50	0.461	3688	50.7	3658	1.54	50.82	-0.12	0.23	50.1	3685	1.52	51.19	-1.09	2.18
100	0.922	7375	100.8	7267	3.07	100.95	-0.15	0.15	100.8	7322	3.07	101.72	-0.92	0.91
150	1.383	11063	150.9	10861	4.59	150.88	0.02	0.01	150.6	10901	4.58	151.44	-0.84	0.56
200	1.844	14750	200.9	14468	6.11	200.99	-0.09	0.05	200.7	14512	6.11	201.60	-0.90	0.45
250	2.305	18438	250.1	17982	7.61	249.81	0.29	0.12	250.1	18053	7.61	250.80	-0.70	0.28
300	2.766	22125	300.4	21606	9.15	300.16	0.24	0.08	300.9	21703	9.16	301.50	-0.60	0.20
320	2.950	23600	320.5	23035	9.75	320.01	0.49	0.15	320.1	23089	9.73	320.76	-0.66	0.21
350	3.227	25813	350.5	25212	10.66	350.25	0.25	0.07	350.4	25264	10.67	350.97	-0.57	0.16
400	3.688	29500	400	28722	12.15	399.01	0.99	0.25	400	28800	12.14	400.10	-0.10	0.02

Tabela V.10 - Valores obtidos no módulo DC_1V_1i PV3, após calibração