



Rede de sensores wireless baseada na tecnologia LoRa para monitorização de processo na indústria

JOSÉ ANTÓNIO GAMEIRO TEIXEIRA

agosto de 2019

REDE DE SENSORES WIRELESS BASEADA NA TECNOLOGIA LoRa PARA MONITORIZAÇÃO DE PROCESSO NA INDÚSTRIA

José António Gameiro Teixeira



Mestrado em Engenharia Electrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

2019

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Disciplina de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Electrotécnica e de Computadores

Candidato: José António Gameiro Teixeira, N° 1120500, 1120500@isep.ipp.pt

Orientação científica: Ramiro Barbosa, rsb@isep.ipp.pt

Empresa: Carlos Gouveia, Carlos.gouveia@eqs-global.com



Mestrado em Engenharia Electrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

29 de julho de 2019

Agradecimentos

Com a realização deste trabalho dá-se por terminada mais uma etapa na minha vida académica. Como tal, não poderia deixar de mencionar e agradecer a todos os que fizeram parte do meu percurso e que de alguma forma contribuíram para a realização deste trabalho.

Em primeiro lugar, gostava de agradecer a toda a minha família pelo apoio, dedicação e motivação prestados. Este foi fundamental durante todo o trajeto.

Um agradecimento especial à empresa EQS, em particular aos meus orientadores, Carlos Gouveia e Gabriel Dinis, que para além de me oferecerem a oportunidade de realizar este trabalho, estiveram sempre disponíveis e acompanharam de perto a realização deste projeto.

Por fim, deixar também um agradecimento ao Engenheiro Ramiro Barbosa, pelo apoio, disponibilidade e orientação, prestados durante todo o projeto.

Resumo

Na indústria petroquímica, a monitorização de processo é da maior importância para assegurar o bom funcionamento dos equipamentos e dos processos. Com os dados fornecidos pela monitorização é possível implementar, sistemas capazes de controlar todos estes parâmetros e garantir o funcionamento de toda a estação. Todavia, é necessária a coordenação entre todos os sistemas da instalação, podendo estes estar dispersos por grandes áreas geográficas.

De forma a permitir a monitorização remota, é necessária a implementação de sistemas de monitorização *wireless*. Entre os vários tipos de comunicação de longo alcance, destaca-se o LoRa (*Long Range*), uma tecnologia de modulação de rádio frequência que oferece comunicações com longo alcance, taxas de transmissão baixas, robustez e baixos consumos energéticos.

A rede desenvolvida é constituída por nós, um *gateway* e o servidor. Nos nós é feita a interligação com o meio, sendo lidos parâmetros como a temperatura, humidade e luminosidade e enviados os dados recolhidos através do protocolo LoRaWAN. No servidor estes são primeiramente descodificados, para depois serem apresentados. A fazer a conexão entre a rede LoRaWAN e a Internet temos o *gateway*, responsável pelo encaminhamento das tramas recebidas dos nós. O sistema permite a inclusão de mais elementos na rede como sensores, nós e *gateways*.

Palavras-Chave

LoRa, LoRaWAN, WAN, LPWAN, *Wireless*, Dragino, IoT.

Abstract

In the petrochemical industry the process, monitoring is of major importance to ensure the proper functioning of all equipments and processes. With the data provided it is possible to implement systems capable of controlling all these parameters and guarantee the correct operation of the entire station. However, it is necessary to coordinate all the systems of the station, even if they are spread over a large geographic area.

In order to allow remote monitoring it is necessary to implement wireless monitoring systems. Among the various types of long-range communication, WAN and LPWAN, the LoRa was used which is a type of radio frequency modulation that offers communications with low transmission rates, ruggedness, long range and low power consumption.

The developed network consists of nodes, a gateway and a server. The nodes do the connection with the field reading parameters like temperature, humidity and luminosity. These data is sent through the LoRaWAN protocol. The server does the decoding first and then displays the data. To make the connection between the LoRaWAN network and the Internet we have the gateway, which is responsible for forwarding the received frames of the nodes. The system allows the inclusion of more elements in the network like sensors, nodes and gateways.

Keywords

LoRa, LoRaWAN, WAN, LPWAN, Wireless, Dragino, IoT.

Índice

AGRADECIMENTOS	V
RESUMO	VII
ABSTRACT	IX
ÍNDICE	XI
ÍNDICE DE FIGURAS	XIII
ÍNDICE DE TABELAS	XVI
ACRÓNIMOS	XVII
1. INTRODUÇÃO	1
1.1. EMPRESA EQS	1
1.2. CONTEXTUALIZAÇÃO	2
1.3. OBJETIVOS	3
1.4. CALENDARIZAÇÃO	4
1.5. ORGANIZAÇÃO DO RELATÓRIO	4
2. ESTADO DA ARTE	7
2.1. INDÚSTRIA PETROQUÍMICA	7
2.2. COMUNICAÇÃO ENTRE MÁQUINAS	9
2.3. ARQUITETURAS DE REDES SEM FIOS.....	12
2.4. REDES DE SENSORES <i>WIRELESS</i>	14
2.5. <i>WIRELESS INDUSTRY STANDARDS</i>	16
2.6. CONCLUSÃO	28
3. PROTÓTIPO	31
3.1. CONTEXTO INICIAL	32
3.2. REDE DE TESTE LoRA	32
3.3. REDE DE TESTE LoRAWAN.....	37
3.4. MICROCHIP LoRA MOTE	62
3.5. <i>GATEWAY</i>	77
3.6. CONCLUSÃO	80
4. ARQUITETURA	83
4.1. INTRODUÇÃO	83

4.2.	DESENVOLVIMENTO.....	85
5.	SOFTWARE.....	93
5.1.	DRAGINO.....	93
5.2.	MIRCROCHIP MOTE.....	96
6.	TESTES.....	101
6.1.	DISTÂNCIA.....	101
6.2.	QUALIDADE DE SINAL.....	105
7.	CONCLUSÕES.....	109
	REFERÊNCIAS DOCUMENTAIS.....	113
	ANEXO A. ESQUEMÁTICO DOS NÓS DRAGINO.....	119
	ANEXO B. SENSOR DE TEMPERATURA LM35.....	120

Índice de Figuras

Figura 1 Logótipo e slogan da Empresa [1].	2
Figura 2 Planta de processos típica de uma refinaria [58].	9
Figura 3 Arquitetura de comunicações entre máquinas [5].	10
Figura 4 Arquitetura de uma rede <i>Peer-to-peer</i> [40].	13
Figura 5 Arquitetura de uma rede Cliente/Servidor [41].	14
Figura 6 Exemplo de uma WSN [42].	15
Figura 7 Topologias de rede [11].	16
Figura 8 Arquitetura ISA 100.11a [17].	19
Figura 9 Exemplo de uma arquitetura WirelessHART [43].	20
Figura 10 Vantagens da tecnologia NB-IoT [22].	22
Figura 11 Exemplo de uma Arquitetura LoRa [23].	23
Figura 12 Topologia de rede com protocolo LoRaWAN [24].	24
Figura 13 Janelas de receção das diferentes classes dos nós [29].	25
Figura 14 Camadas da comunicação LoRa [30].	25
Figura 15 Arquitetura de uma rede SigFox [33].	26
Figura 16 Arquitetura RPMA [45].	27
Figura 17 Arquitetura da rede de teste.	32
Figura 18 Conceito da rede de teste.	33
Figura 19 Rotina principal do nó central.	34
Figura 20 Rotina principal do nó.	35
Figura 21 Formatos de dados.	37
Figura 22 Arquitetura LoRaWAN[48].	38
Figura 24 Arquitetura LoRaWAN [50].	40
Figura 25 <i>Gateway</i> registado no servidor TTN.	41
Figura 26 Nó registado no TTN.	42
Figura 27 Arquitetura do sistema.	42
Figura 28 Configurações da rádio frequência.	43
Figura 29 Configurações do protocolo.	44
Figura 30 Rotina principal [55].	45
Figura 31 Tráfego de dados do <i>gateway</i> .	46

Figura 32 Dados recebidos pela aplicação	47
Figura 33 Sistema desenvolvido.....	47
Figura 34 Dados recebidos pelo servidor	48
Figura 35 Dados enviados pelo nó	49
Figura 36 Circuito do sensor LM35 [60].....	50
Figura 37 Instrução de leitura do sensor.....	50
Figura 38 Comunicações do nó para o servidor	51
Figura 39 Registo de comunicações do servidor	52
Figura 40 Registo de comunicações na aplicação do servidor	53
Figura 41 Registo das comunicações do nó com sensores	54
Figura 42 Registo das comunicações do nó com GPS	54
Figura 43 Estrutura de comandos NMEA [66].....	55
Figura 44 Dados enviados pelo módulo GPS.....	56
Figura 45 Dados enviados pelo nó com GPS	57
Figura 46 Localização do nó	57
Figura 47 Percurso de teste	59
Figura 48 Gateway LG01-P Dragino [53].....	61
Figura 49 Nó LoRa da Microchip (DM164138) [54].....	62
Figura 50 Diagrama de blocos do Microchip LoRa MOTE.....	63
Figura 51 PICkit3 e LorA Mote da Microchip.....	64
Figura 52 Ambiente de trabalho do PICkit 3.....	65
Figura 53 Dispositivo criado no servidor TTN	66
Figura 54 Arquitetura da rede	66
Figura 55 Interface gráfica Lora Suite.....	67
Figura 56 Configuração da camada MAC.....	68
Figura 57 Configuração da camada física	68
Figura 58 Transmissão sem confirmação	69
Figura 59 Fluxograma do funcionamento do MOTE	70
Figura 60 Comandos GPIO [46]	71
Figura 61 Comandos para controlar o sensor e LED	72
Figura 62 Dados recebidos pelo <i>gateway</i>	73
Figura 63 Dados recebidos pela aplicação	73
Figura 64 Conexões entre PIC18LF46K22 e o <i>Transceiver</i> SX1276 [57].....	74
Figura 65 Funcionamento do novo software do Mote.....	75
Figura 66 Fluxograma da rotina principal	76

Figura 67 <i>Gateway</i> Dragino	77
Figura 68 <i>Gateway</i> Microchip.....	79
Figura 69 Estado do <i>gateway</i> no servidor	80
Figura 70 Arquitetura do sistema	84
Figura 71 Diagrama de funcionamento da placa rádio [68]	86
Figura 72 Diagrama de funcionamento da placa central do <i>gateway</i>	87
Figura 73 Fluxograma do funcionamento do nó Dragino	89
Figura 74 Fluxograma do nó da Microchip.....	91
Figura 75 Fluxograma de funcionamento do nó Dragino	94
Figura 76 Rotina de envio de dados	95
Figura 77 Fluxograma de funcionamento do nó Microchip.....	97
Figura 78 Fluxograma de envio dos dados.....	99
Figura 79 Fluxograma de funcionamento da rotina <code>wake_up()</code>	100
Figura 80 Mapeamento dos locais de teste.....	102
Figura 81 Mapeamento da rede	103
Figura 82 Mapeamento dos novos locais de teste	104
Figura 83 Mapeamento da rede	105
Figura 84 Parâmetros da comunicação.....	106
Figura 85 Equação de cálculo da distância com base no RSSI [69]	107

Índice de Tabelas

Tabela 1 Planeamento do projeto	4
Tabela 2 Camadas da WirelessHART segundo o modelo OSI [44].....	20
Tabela 3 Comparação entre LPWAN.....	28
Tabela 4 Dados adquiridos no teste.....	107

Acrónimos

ABP	–	<i>Activation By Personalization</i>
ADR	–	<i>Adaptative Data Rate</i>
AES	–	<i>Advanced Encryption Standard</i>
ASCII	–	<i>American Standard Code for Information Interchange</i>
BAC	–	<i>Building Automation and Control</i>
FSK	–	<i>Frequency-Shift-Keying</i>
GAN	–	<i>Global Area Network</i>
GSM	–	<i>Global System for Mobile Communications</i>
GPIO	–	<i>General Purpose Input/Output</i>
GPS	–	<i>Global Positioning System</i>
HART	–	<i>Highway Adressable Remote Transducer Protocol</i>
IP	–	<i>Internet Protocol</i>
IR	–	<i>Infrared Radiation</i>
LAN	–	<i>Local Area Network</i>
LED	–	<i>Light Emitting Diode</i>
LoRa	–	<i>Long Range</i>
LoRaWAN	–	<i>LoRa Wide Area Network</i>
LPWAN	–	<i>Low Power Wide Area Network</i>

M2M	–	<i>Machine-to-Machine</i>
MAC	–	<i>Medium Access Control</i>
MQTT	–	<i>Message Queuing Telemetry Transport</i>
NMEA	–	<i>National Marine Electronics Association</i>
OSI	–	<i>Open System Interconnection</i>
OTAA	–	<i>Over-The-Air-Activation</i>
RPMA	–	<i>Random Phase Multiple Access</i>
SCADA	–	<i>Supervisory Control and Data Acquisition</i>
SF	–	<i>Spreading Factor</i>
TCP	–	<i>Transmission Control Protocol</i>
TTFB	–	<i>Time To First Fix</i>
USB	–	<i>Universal Serial Bus</i>
USB-TTL	–	<i>Universal Serial Bus-Transistor to Transistor Logic</i>
WAN	–	<i>Wide Area Network</i>
WLAN	–	<i>Wireless Local Area Network</i>
WSAN	–	<i>Wireless Sensor to Actuator Network</i>
WSN	–	<i>Wireless Sensor Network</i>

1. INTRODUÇÃO

Neste capítulo é feita uma introdução ao tema, abordando o objetivo principal do projeto e a sua contextualização. Também faz parte desta capítulo a apresentação da EQS, empresa onde foi realizado este projeto. O capítulo termina com o planeamento do trabalho e a organização do relatório.

1.1. EMPRESA EQS

A EQS, Engenharia, Qualidade e Segurança (Figura 1), é uma empresa inserida nas mais variadas áreas da engenharia e tecnologia como são exemplo a indústria de petróleo, gás, processos, energias renováveis, infraestruturas, equipamentos industriais, entre outras.

Fornecer serviços de inspeção e garantia de qualidade, testes e monitorização de ativos, gestão de risco e consultoria, formações técnicas e digitais, consultoria e supervisão de projetos de engenharia. No que diz respeito ao *outsourcing*, a EQS, desenvolve soluções de gestão de risco, gestão de ativos e digitais para as empresas.

Por outro lado, como projetos de investigação e desenvolvimento, a EQS aposta no desenvolvimento de tecnologias no âmbito da Indústria 4.0, com recurso ao financiamento de programas como o Portugal 2020 e Horizon 2020.

Projetos de *hardware* e *software*, como criação de sistemas integrados inteligentes para monitorização e modelação preditiva do desempenho de ativos críticos industriais, plataformas de integração IoT, aplicações dedicadas à gestão de grandes quantidades de dados, simulação de cenários, análises de risco, sistemas que integra inteligência artificial e plataformas de gestão de ativos industriais, são alguns dos projetos executados neste âmbito.



Figura 1 Logótipo e slogan da Empresa [1].

1.2. CONTEXTUALIZAÇÃO

A monitorização do processo na indústria petroquímica, em particular nas refinarias, é fulcral na medida em que permite a segurança, eficiência e produtividade da instalação fornecendo dados sobre toda a planta de processo bem como dos equipamentos. Esta permite a proteção automática de toda a instalação, manutenção de parâmetros de operação, controlo de indicadores do processo bem como a sua visualização, registo e criação de relatórios. As grandezas a monitorizar são a pressão, temperatura, níveis de líquidos, fluxo, velocidade, composição, densidade e peso. Por outro lado, temos também outros parâmetros mecânicos e elétricos como a vibração, posição, corrente e tensão. Para a supervisão de todos os parâmetros de interesse é necessária a implementação de soluções de informação, que permitam a comunicação total entre todos os sistemas, sendo que estes podem estar dispersos.

Atualmente, assistimos a uma migração dos sistemas de monitorização e controlo com cablagem para os sistemas *wireless*, sendo que estes últimos permitem uma monitorização remota, redução de custos e tempo de implementação. Para a introdução destas novas tecnologias na indústria é necessário preencher os requisitos de segurança, confiança, robustez, qualidade de serviço, integração com sistemas existentes, capacidade de suporte de vários dispositivos, de expansão e equipamentos de suporte. Nesta indústria, todavia, existem alguns obstáculos adicionais em relação às demais, uma vez que, estas instalações são caracterizadas por uma forte presença de estruturas metálicas que bloqueiam as

comunicações e criam reflexões, dificultando assim a implementação de comunicações sem fio.

Na área das comunicações sem fio podemos distinguir dois tipos de comunicação, a comunicação LAN (*Local Area Network*) e WAN (*Wide Area Network*), sendo que o que as distingue é o seu alcance. No primeiro caso este parâmetro é menor, sendo que a sua aplicabilidade é dirigida a redes menos dispersas. Já, no que diz respeito ao segundo caso, esta cobre maiores áreas como por exemplo cidades.

Entre os vários tipos de comunicação WAN, destaca-se o LoRa (*Long Range*) pois para além de apresentar as vantagens associadas a este tipo de tecnologias, também oferece comunicações com taxas de transmissão baixas, robustez e baixos consumos energéticos. Sendo que na aplicação prática deste projeto estas são todas características vantajosas e de interesse nesta indústria [2].

1.3. OBJETIVOS

O objetivo principal deste projeto passa pela criação de uma rede de sensores *wireless* baseada na tecnologia de comunicação LoRa para monitorização de processo na indústria petroquímica. Para tal, fazem parte deste trabalho as seguintes etapas:

- Análise e compreensão das necessidades de monitorização do processo na indústria petroquímica;
- Apresentação do estado da arte das comunicações *wireless* aplicadas à indústria;
- Apresentação da comunicação LoRa bem como comparação desta com as demais ofertas de mercado;
- Implementação de uma rede de sensores sem fio LoRa, utilizando o *kit* de desenvolvimento Dragino, para monitorização de parâmetros;
- Criação de uma base de dados para armazenar a informação;
- Implementação da rede LoRa em laboratório para monitorização dos parâmetros do processo (temperatura, fluxo, pressão e consumos energéticos).

1.4. CALENDARIZAÇÃO

O planeamento deste projeto (Tabela 1) encontra-se dividido nas seguintes fases:

- Trabalho de pesquisa do estado da arte das comunicações sem fios utilizadas na indústria petroquímica, bem como recolha de informações sobre a tecnologia LoRa e a referida indústria;
- Testes com o *hardware*;
- Implementação e testes de uma rede experimental LoRa RAW, utilizando o *kit* de desenvolvimento Dragino e testes;
- Implementação e testes da rede LoRaWAN, numa primeira fase com o *kit* Dragino, e posteriormente com este e o *kit* de desenvolvimento da Microchip;
- Implementação da arquitetura da rede final, com ambos os nós dos *kits*, o *gateway* da Microchip, sensores e o servidor TTN;
- Documentação do trabalho.

Tabela 1 Planeamento do projeto

Mês	Março				Abril				Maio				Junho				Julho			
Semana	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Trabalho de pesquisa do estado da arte																				
Testes com <i>hardware</i>																				
Implementação da rede LoRa RAW																				
Testes com rede LoRa RAW																				
Implementação da rede LoRaWAN																				
Testes com rede LoRaWAN																				
Arquitetura final e testes																				
Documentação do trabalho																				
Entrega da documentação do trabalho																				

1.5. ORGANIZAÇÃO DO RELATÓRIO

No Capítulo 1 é feita a apresentação do trabalho, dando uma introdução ao tema e ao contexto envolvente. É também apresentada a empresa EQS, onde o estágio foi realizado, os objetivos do trabalho, a sua calendarização e a estrutura do relatório.

No segundo Capítulo é feita a apresentação do estado da Arte. Este aborda as comunicações *wireless* entre máquinas, no geral e na indústria, os *standards* deste tipo de tecnologia e faz um ponto da situação na indústria petroquímica.

No terceiro Capítulo são apresentados os vários protótipos desenvolvidos até chegar à arquitetura final da rede, tendo-se utilizado diversas soluções quer com diferentes equipamentos quer com diferentes protocolos de comunicações.

O quarto Capítulo é composto pela arquitetura criada que soluciona o problema em questão, apresentando o conceito central do trabalho.

O quinto Capítulo faz referência ao *software* desenvolvido para a solução encontrada.

No sexto Capítulo são apresentados os testes realizados com a rede final. Foram executados testes de distância e de medição da força de sinal.

O sétimo e último Capítulo faz o balanço do projeto, apresentando as suas conclusões e futuras melhorias ao mesmo.

2. ESTADO DA ARTE

Esta secção consiste no trabalho de pesquisa científica sobre as tecnologias envolvidas neste projeto bem como a apresentação de soluções, existentes no mercado, na área dos sistemas de monitorização de processo sem fio na indústria. Com este capítulo foi possível compreender as tecnologias em questão bem como oferecer uma visão alargada sobre as soluções já desenvolvidas, ambiente em que estas se inserem e também a escolha dos equipamentos que mais se adequam ao projeto em questão. Assim, este foi o ponto de partida deste projeto.

2.1. INDÚSTRIA PETROQUÍMICA

A indústria petroquímica traduz-se na atividade de produção industrial de petróleo e seus derivados. O principal foco desta é a produção de uma vasta gama de produtos, derivados do petróleo, com o menor custo e melhor qualidade possíveis. Deste processo de produção fazem parte a extração da matéria prima, o petróleo, o refino e a sua transformação, no produto final.

No que diz respeito à instrumentação, nesta indústria, fazem parte os sistemas de monitorização e controlo da planta de processo. A instrumentação é composta por sensores, transmissores de sinal, controladores, indicadores, alarmes, válvulas de atuação,

circuitos lógicos e interfaces de operador. Todos estes elementos permitem a monitorização de toda a estrutura do processo.

Esta monitorização é feita sobre os indicadores que permitem controlar o funcionamento do processo. Assim sendo, temos como parâmetros do processo a pressão, temperatura, níveis de líquidos, fluxo, velocidade, composição, densidade e peso. Por outro lado, temos também outros parâmetros que são os elétricos e mecânicos, como a vibração, posição, corrente e tensão.

No que diz respeito à temperatura, esta está diretamente ligada com o processo de aquecimento ou arrefecimento quer de gases ou líquidos, no processo produtivo. Assim são necessários elementos de medição desta grandeza, como termopares e termo resistências. Por norma, estes processos são levados a cabo por elementos permutadores de calor, como a água, *glycol* ou óleo quente, originando a necessidade destes elementos de medição. Por outro lado, temos também a pressão.

Nesta indústria, muitos dos processos que envolvem gases, líquidos ou elementos petroquímicos, requerem um certo nível de pressão, sendo estas grandezas medidas por sensores de pressão.

Outra grandeza a controlar é o fluxo, este controla, por exemplo, a taxa de transferência do processo petroquímico ou o fluxo nos compressores. Os equipamentos de medição são sensores de pressão, que transmitem as informações para controladores de pressão.

Por outro lado, são também necessários sensores de nível, uma vez que o processo envolve quantidades de líquidos e gases. Para além destes equipamentos, são também utilizados flutuadores magnéticos, medidores de pressão diferencial, entre outros.

Por último, temos também as vibrações. Este é um parâmetro que pode indicar o estado do funcionamento do processo. Na origem destas vibrações irregulares podem estar falhas em equilíbrios magnéticos, mecânicos e aerodinâmicos. Exemplos destas falhas são aberturas de ar desiguais, fontes assimétricas, falhas no rotor, estator e desequilíbrios nas cargas do motor. A Figura 2 apresenta os vários processos típicos de uma refinaria.

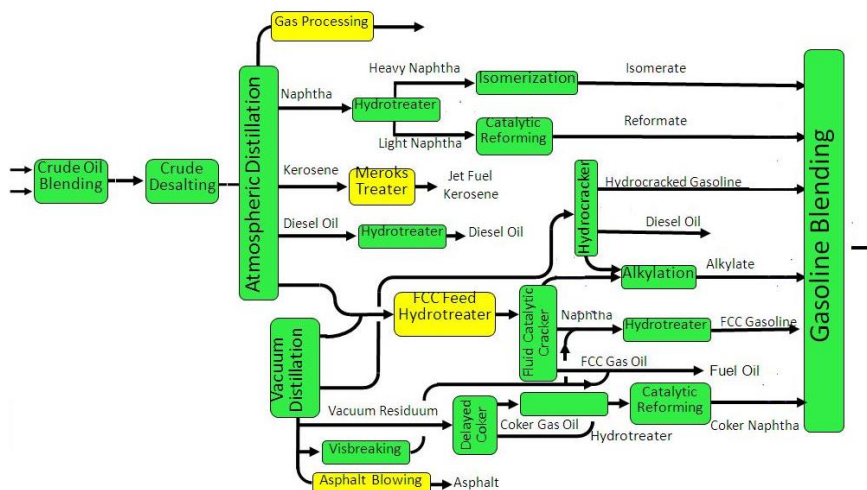


Figura 2 Planta de processos típica de uma refinaria [58]

Através das ferramentas de instrumentação e automação é possível a monitorização dos parâmetros e posterior controlo. Estes tanto podem ser armazenados localmente ou nas salas de controlo. A parte da automação é aplicada quando existe uma predefinição dos valores destes parâmetros, ou seja, existe uma programação do sistema, que pode permitir o acionamento de atuadores ou alarmes nos casos em que estes valores excedem os limites definidos.

Como principais atuadores destes sistemas temos as válvulas, amortecedores, bombas e compressores. Estes elementos fazem parte do processo produtivo, sendo que o seu controlo mantém o correto funcionamento das operações [3][4].

2.2. COMUNICAÇÃO ENTRE MÁQUINAS

A tecnologia de comunicação entre máquinas faz referência à transferência de informação, entre máquinas, de forma autónoma, quer com a utilização de cablagem ou não. As primeiras aplicações práticas desta tecnologia foram na área da indústria de produção, com a finalidade de monitorizar, adquirir e controlar os processos industriais. Assim nasceram os sistemas SCADA (*Supervisory Control and Data Acquisition*).

Apesar das diferenças, atualmente, esta tecnologia é a base para o conceito IoT (*Internet of Things*) porém também está presente nas mais diversas áreas, quer a nível empresarial quer a nível doméstico.

Através da transferência de informação entre máquinas é possível realizar o principal objetivo desta tecnologia que é a recolha de informações de sensores e posterior envio para a rede. Estas normalmente utilizam redes como Ethernet ou GSM (*Global System for Mobile Communications*), uma vez que são de fácil acesso, permitem uma flexibilização nos custos. Este tem sido um ponto chave para o crescimento destas tecnologias, uma vez que assistimos à expansão e globalização destas redes de comunicação.

Como principais funcionalidades desta tecnologia temos a aquisição de dados e o seu envio. Todavia, muitos destes equipamentos não são capazes de fazer estas tarefas, sendo necessário adicionar outros dispositivos, capazes de interagir com o meio, transformando grandezas em medições, e receber, enviar e tratar estes dados.

Deste modo, como componentes principais temos os sensores e atuadores, para interagirem com o meio, os protocolos de comunicação, como Wi-Fi ou rede móvel, para o envio de informação, e por último, os equipamentos que interpretam os dados, os elementos computacionais.

O Instituto Europeu de Normas de Telecomunicação (ETSI) definiu o *standard* da arquitetura da rede, com cinco elementos, e 3 domínios. O domínio do equipamento, da rede e da aplicação. Desta forma foi possível distinguir os elementos que recebem e transmitem dados, o canal que interliga os domínios de equipamento e aplicação, permitindo a transmissão de dados, e a aplicação, que para além de comunicar com o cliente, inclui o servidor. A Figura 3 apresenta a arquitetura.

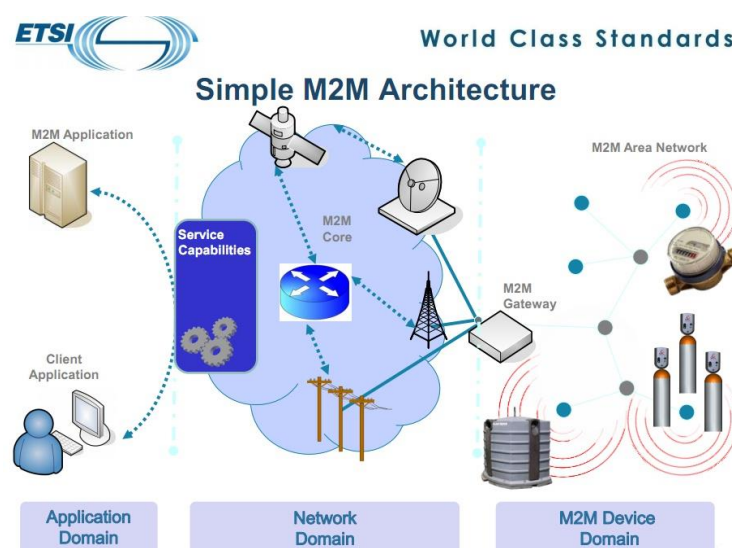


Figura 3 Arquitetura de comunicações entre máquinas [5]

As principais vantagens são a redução de tempo necessário para a transmissão de informação, a redução de custos com a manutenção dos equipamentos, um melhor serviço prestado ao cliente, uma vez que as ações podem ser tomadas de forma proactiva, ou seja antes da falha dos equipamentos, devido à sua monitorização, uma alta fiabilidade, sendo que podem ser aplicados a sistemas de emergência, baixo consumo energético. Sendo que certas aplicações até podem implementar mecanismos de controlo de tráfego, suportando atrasos na transmissão.

Tendo em conta as vantagens associadas a esta tecnologia vemos a sua aplicação tanto em áreas mais práticas como as *smart homes*, onde é possível minimizar o consumo energético, e na medicina, onde, por exemplo, é possível a monitorização, em tempo real, dos pacientes. Também nas áreas industriais, como são exemplo a logística, nos sistemas de gestão de *stocks*, os sistemas SCADA, redes inteligentes de energia, automação de edifícios, robótica, controlo de tráfego e segurança vemos estas tecnologias.

No ambiente industrial, conforme mencionado anteriormente, os primeiros sistemas deste tipo foram os sistemas SCADA. Estes monitorizam e controlam processos e sistemas, com base em dados fornecidos por sensores e instrumentos de medição. Ou seja, estes sensores obtêm medições de parâmetros do processo e, posteriormente, enviam estas informações para equipamentos de controlo. Estes, por sua vez, vão proceder à análise e interpretação destes dados e atuar sobre os processos através de dispositivos de controlo, como são exemplo os PLC (*Programmable Logic Controller*) e PID (*Proportional Integrative Derivative Controler*). O sistema SCADA permite fazer a interligação entre todos estes equipamentos e o seu controlo.

Vemos também aplicações na automação e controlo de edifícios, como são exemplo os sistemas BAC (*Building Automation and Control System*). Estes são sistemas centralizados que monitorizam, controlam e registam os serviços de edifícios. Vemos a sua aplicação em sistemas de regulação de temperatura, gestão de energia, desperdícios de água, sistemas de segurança, ventilação, acessos, luz, entre outros. Todas estas características melhoram o conforto dos ocupantes bem como a eficiência das construções.

Com os avanços tecnológicos, estes sistemas, que antigamente eram sistemas fechados, hoje, permitem um controlo bastante abrangente de equipamentos e posterior envio de informações para bases de dados. Em 1987 foi feita a standardização destas redes, dando

origem ao protocolo BACnet que veio possibilitar a comunicação entre diferentes equipamentos.

Em suma, no contexto industrial, a aplicação desta tecnologia resume-se a sistemas de monitorização e manutenção de equipamentos, supervisionamento, controlo energético, controlo de planta de processo e logística, como é exemplo a gestão de inventários [5][6][7].

2.3. ARQUITETURAS DE REDES SEM FIOS

Com o avanço das tecnologias de comunicação sem fios surgiram inúmeras aplicações quer na indústria quer para o uso pessoal, sendo que se sentiu a necessidade de criar *standards* de forma a criar padrões para estes sistemas

Assim, as arquiteturas de rede definem a estrutura, organização e o modo de funcionamento dos vários equipamentos implementados na rede. Estes podem incluir especificações ao nível do *software*, *hardware*, protocolos e meios de transmissão. Dentre as várias topologias podemos distinguir dois grandes grupos de arquiteturas, as P2P (*Peer-to-peer*) e Cliente/Servidor ou em camadas.

2.3.1. PEER-TO-PEER

Este tipo de arquitetura (Figura 4) é caracterizado por dispositivos sem níveis hierárquicos, distintos e com os mesmos recursos, dentro da rede. Sendo que a gestão da rede é feita por cada um dos dispositivos.

Como principais vantagens estas redes não apresentam a necessidade de um servidor dedicado, o que implica menos custos. Por outro lado, aumenta a sua robustez, uma vez que o não funcionamento de um dispositivo não impede o funcionamento da rede, uma vez que esta continua a funcionar com os restantes. Por último, temos também os aspetos de implementação e configuração, que são de fácil acesso, uma vez que já existem ferramentas de suporte embutidas nos sistemas operativos.

Já, no que diz respeito às desvantagens, estas redes apresentam grandes desafios no que toca à sua escalabilidade, uma vez que com o aumento dos dispositivos na rede, aumenta a complexidade e dificuldade em atingir determinados valores para os parâmetros de

funcionamento, segurança e acesso. De forma adicional, podemos afirmar que o facto de ser necessário a cada equipamento fazer o *backup* dos dados, leva um risco maior na perda destas informações uma vez que só são guardadas por um único dispositivo.



Figura 4 Arquitetura de uma rede *Peer-to-peer* [40]

2.3.2. CLIENTE/SERVIDOR

Numa arquitetura do tipo Cliente/Servidor (Figura 5) temos no centro um servidor, alocado a um equipamento com alta capacidade de processamento, ao qual os clientes se conectam. Estes comunicam com o servidor, sendo que cabe a este a gestão da rede e manipulação dos recursos dos clientes.

Nesta topologia cabe ao servidor o controlo dos recursos e a segurança dos dados, sendo que existe a facilidade de acesso a estas informações através de inúmeras plataformas. Por outro lado, a escalabilidade destes sistemas é, a par das características mencionadas anteriormente, uma das vantagens desta arquitetura.

Os servidores são equipamentos bastante dispendiosos, dado as suas especificações, apresentam necessidades de manutenção com maior regularidade que outros equipamentos, para manter o seu correto funcionamento, e são uma peça fundamental neste tipo de arquitetura. Sendo que este é um tipo de arquitetura centralizada todas as desvantagens estão relacionadas com problemas afetos ao servidor [8].

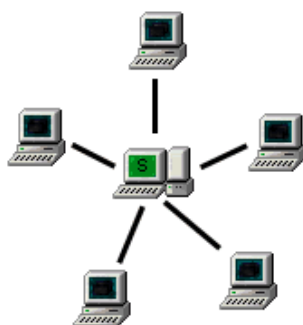


Figura 5 Arquitetura de uma rede Cliente/Servidor [41]

2.4. REDES DE SENSORES WIRELESS

Na indústria, já desde os anos 70 que as redes de sensores eram utilizadas todavia, neste caso, estas recorriam à utilização de cabelagem, como meio de comunicação entre os vários equipamentos da rede.

A par do avanço nas tecnologias de redes sem fios, dos microcontroladores e outras tecnologias associadas, foi possível desenvolver as redes de sensores sem fios. Estas permitem a interligação do meio industrial, físico, à Internet, facilitando assim o acesso, monitorização e controlo de parâmetros e processos.

Atualmente as WSN (*Wireless Sensor Network*) têm uma forte presença em aplicações comerciais e industriais (Figura 6) sendo que entre as várias aplicações temos como exemplos mais comuns as redes de monitorização de condições atmosféricas como a temperatura, humidade, luminosidade, e também redes de segurança, que alertam para situações de incêndio, através de sensores de deteção de fogo ou fumo, inundações, acessos. Por outro lado, vemos aplicações também nas áreas da deteção inteligente, processamento e de dados, rastreio de parâmetros, localização GPS (*Global Positioning Service*), entre outras.

Devido às inúmeras aplicações nas áreas da medicina, ambiente, transportes, militar, entretenimento, defesa de território e aplicações inteligentes, espera-se que as WSN, dentro de poucos anos, estejam presentes na maioria das tarefas do dia-a-dia. Para esta tecnologia, este é um avanço muito significativo, uma vez que vemos a evolução de sistemas direcionados para a indústria, a serem aplicados aos mais diversos cenários.

Como principais objetivos das WSN temos a redução dos consumos energéticos, a qualidade das comunicações, uma vez que podem operar em ambientes com bastantes interferências, baixos recursos computacionais, o mínimo processamento de dados e a flexibilidade para o aumento dos nós da rede.

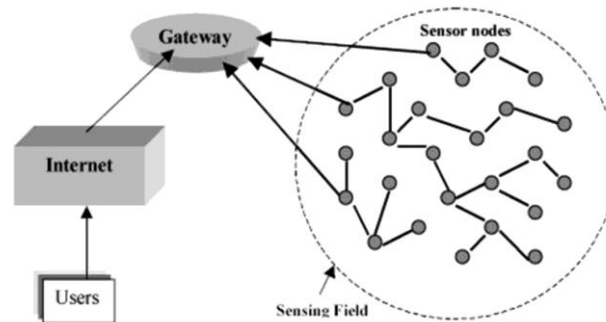


Figura 6 Exemplo de uma WSN [42]

A nível técnico, estas redes são compostas por elementos de processamento de informação, módulos de comunicação sem fios, *transceivers*, e dispositivos computacionais, com baixos consumos energéticos, todos implementados num nó. Cabe a estes elementos a criação da rede, sendo que são multifuncionais e eficientes na utilização da energia.

Só desta forma é possível obter uma rede de sensores que cumpra os requisitos de cobertura de grandes áreas geográficas, utilização de dispositivos de pequenas dimensões alimentados a bateria, uma vez que para além da necessidade de processamento e comunicação, associada à captura, manipulação e transmissão dos dados, é também necessária uma longevidade de operação do sistema.

Por outro lado, podemos afirmar que uma das grandes vantagens deste tipo de tecnologia é que a quantidade de nós pode variar desde as centenas até aos milhares [9][10].

2.4.1. ARQUITETURA DA REDE DE SENSORES

Dentro das várias redes podemos distinguir dois grandes grupos. As redes *ad hoc* e as redes estruturadas. No que diz respeito às redes *ad hoc*, estas permitem a comunicação entre nós, sem a necessidade de utilização de uma estação base, seguindo uma arquitetura do tipo *peer-to-peer*. Por outro lado, temos também as redes estruturadas, que ao contrário das *ad hoc*, a comunicação é sempre feita entre os nós e as estruturas físicas, sendo estas últimas um ponto de passagem obrigatório.

Uma WSN é constituída pelos nós sensores, para a interação com o meio físico em questão, nós terminais, responsáveis pelo envio dos dados para o *router*, routers, que encaminham a informação para a rede, e por último, os *gateways*, que recebem os dados e fazem o posterior processamento.

No que diz respeito à forma como os dados são enviados para o *gateway* podemos apresentar as seguintes topologias:

- A topologia em estrela, caracterizado por comunicações somente entre os nós e o *gateway*.
- A topologia em árvore, onde temos os diversos nós de fim de percurso, para comunicarem com o *gateway*, podendo enviar a informação por nós intermédios que vão reencaminhar estes dados.
- A topologia em malha, sendo que esta última, distingue-se da anterior, pelo facto de que todos os nós da rede têm funções quer de *router* quer de nós terminais. Isto leva a que exista um aumento nas opções de encaminhamento dos pacotes até ao *gateway* [10].

A Figura 7 apresenta estas topologias.

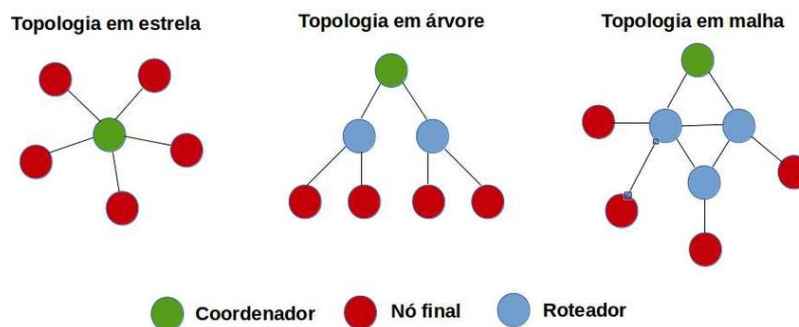


Figura 7 Topologias de rede [11]

2.5. WIRELESS INDUSTRY STANDARDS

Com a popularização das tecnologias de comunicação entre máquinas, surgiu o crescimento na procura deste tipo de sistemas. Este facto originou um alargamento nas áreas de aplicação e conduziu ao desenvolvimento de diferentes comunicações com características diversificadas, uma vez que foram aplicadas aos mais diversos cenários.

Assim surgiu a necessidade de criar uma classificação, que distinguísse as demais tecnologias.

Por exemplo temos as redes do tipo WPAN (*Wireless Personal Area Network*). Estas são de âmbito pessoal, pelo que conectam equipamentos num curto espaço. As tecnologias Bluetooth e IR (*Infrared Radiation*) estão inseridas neste tipo de redes.

Temos também as redes de telemóvel, que são redes que utilizam frequências rádio para comunicar sobre grandes áreas, denominadas células, e as redes GAN (*Global Area Network*) que dão suporte a estas, através redes *wireless* do tipo LAN e das áreas de cobertura dos satélites.

Por último temos também uma distinção baseada no alcance da rede, sendo que temos as redes de baixo alcance (LAN) e as de grande alcance (WAN). Atualmente estas redes são cada vez mais uma componente essencial nas infraestruturas de comunicação sendo usadas como o recurso principal de comunicação, não só para o envio de informação, mas também para outras funcionalidades.

Uma vez que o objetivo deste trabalho está associado à implementação, na indústria petroquímica, de uma rede de sensores, de seguida vão ser apresentadas os tipos de rede e tecnologias que são adequadas para este propósito [12].

2.5.1. LAN

O principal foco deste tipo de *standards* é a criação de redes de comunicação densas em espaços reduzidos. A grande vantagem associada a estes sistemas é a taxa de transmissão de dados um pouco mais elevada que as WLAN e, apesar de terem um alcance pequeno, através da interligação de diversos equipamentos, é possível aumentar este parâmetro. Um dos exemplos da necessidade de uso deste tipo de tecnologia são as aplicações industriais WSN, que incluem o ISA 100.11a, WirelessHart e Zigbee.

As redes WSN (*Wireless Sensor to Actuator Network*) são compostas por sensores e atuadores que comunicam entre si. Na parte da deteção e na parte da ação [13][14][15].

- ISA 100.11a

O principal objetivo da sociedade internacional de automação, na criação deste *standard*, foi oferecer ao mercado uma variada oferta de soluções para as redes sem fios industriais, capazes de suprir as necessidades da planta de processo, como são exemplo o controlo do processo, o rastreio e identificação de pessoas e bens, e as aplicações de longo alcance. Assim, este tipo de rede é descrito como uma rede *mesh* que oferece ao processo de controlo uma comunicação segura e sem fios.

A rede é composta por elementos de campo, *routers* de *backbone*, gestores de sistema, de segurança e *gateway*. No que diz respeito aos elementos de campos, estes tanto podem fazer a tarefa de *routing* como de equipamentos I/O ou até as duas.

A transmissão de informação é feita através de vários saltos, sendo que cabe aos dispositivos de *routing* a tarefa de encaminhamento dos pacotes, desde a sua origem até ao destino. São também responsáveis pela escolha de caminhos alternativos de forma a atingir uma melhoria na fiabilidade.

Nesta arquitetura, cabe ao *router* de *backbone*, o encaminhamento dos pacotes, desde a rede de origem, passando pela rede de *backbone*, até ao seu destino, sendo que este último, tanto pode ser um *gateway* como uma sub-rede. Quanto ao *gateway*, este tem como função a interligação, entre a rede de campo e a rede de controlo, onde se encontram as interfaces de controlo.

Por último, o gestor de sistema, é responsável pela configuração de toda a comunicação do equipamento de gestão, do mecanismo de controlo do *run-time* da rede e administração da rede. A Figura 8 apresenta a arquitetura deste *standard* [16].

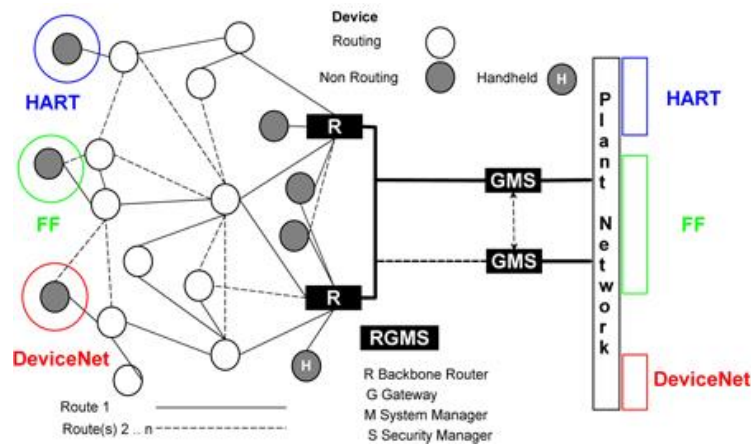


Figura 8 Arquitetura ISA 100.11a [17]

Associado a esta arquitetura temos também um protocolo, no qual podemos distinguir como principais vantagens a escalabilidade e a compatibilidade com protocolos já existentes na indústria.

- WirelessHart

Criado pela HART (*Highway Addressable Remote Transducer Protocol*), este foi o primeiro *standard* de comunicação *wireless* concebido para o controlo do processo e de aplicações na indústria. Este é dividido na camada física, MAC (*Medium Access Control*) e de rede. Sendo que as principais vantagens são a utilização de um contador de 10 ms rigoroso, alargado tempo de sincronização da rede, mudança de canais e lista negra destes, e por último, inclusão de *standards* industriais, como o AES-128. Por outro lado, outra característica que distingue este, de outros *standards* públicos, é a inclusão de um gestor de rede central.

Este dispositivo faz a manutenção e atualização das rotas bem como a calendarização das comunicações, de forma a garantir o desempenho e funcionamento da rede. A Figura 9 apresenta a um exemplo de rede WirelessHART, na qual podemos distinguir os elementos de campo, os dispositivos portáteis, o *gateway* e o gestor da rede. Os elementos de campo fazem a interligação da rede com a planta de processo. Por outro lado, os dispositivos portáteis servem para configurar, monitorizar e calibrar os equipamentos.

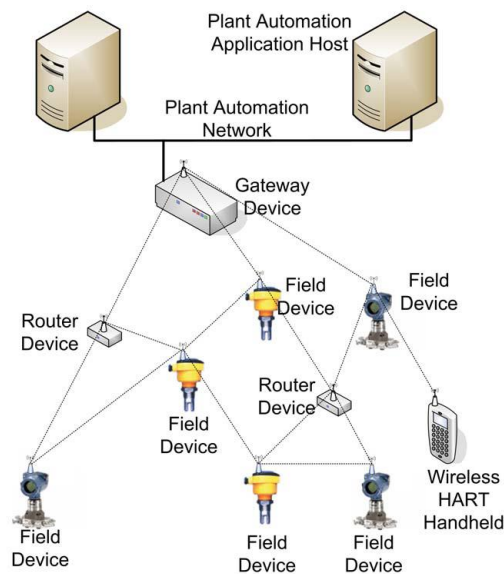


Figura 9 Exemplo de uma arquitetura WirelessHART [43]

Esta tecnologia é essencialmente descrita como uma rede sem fios centralizada, na qual cabe ao gestor central de rede a tarefa de roteamento e planeamento das comunicações. Opera na banda de rádio ISM 2,4 GHz e a topologia pode ser em estrela ou em grupo.

No que diz respeito à arquitetura do protocolo WirelessHART fazem parte as seguintes camadas, a camada física, baseada nas normas IEEE 802.15.4, camada de ligação de dados, camada de rede, camada de transporte e camada de aplicação. A Tabela 2 apresenta as camadas da rede WirelessHART.

Tabela 2 Camadas da WirelessHART segundo o modelo OSI [44]

OSI Layer	WirelessHART
Application	Command Oriented. Application Procedures
Presentation	
Session	
Transport	Auto-segmented transfer of large data sets. Reliable stream transport. Negotiated Segment sizes
Network	Power-Optimized. Redundant Path. Mesh Network
Data Link	Secure. Time Synched TDMA/CSMA, Frequency Agile with ARQ
Physical	IEEE 802.15.4 based radios. Up to 10dBm Tx Power. 2.4GHz Wireless

Todas as definições das características das ondas rádio são definidas pela camada física, como são exemplo o método de sinalização, a intensidade do sinal e a sensibilidade do equipamento.

Na camada de interligação de dados é feita a sincronização dos dados, através da característica do *time-slot*. Por outro lado, através do uso da ferramenta TDMA, são fornecidos os mecanismos de detecção de colisões e comunicações determinísticas.

De forma adicional, as camadas de rede e de transporte proporcionam uma comunicação segura e de confiança entre os dispositivos de fim de linha. Todavia, é na camada de aplicação, que é a camada de topo, que são definidos as varias ordens de comandos, respostas, tipos de trama de dados e relatórios de estado. Tendo em conta que a comunicação, nesta tecnologia, é baseada em respostas e comandos, cabe a esta camada a identificação das mensagens, a extração e execução dos comandos, e ainda, a criação das respostas [18].

2.5.2. WAN

As WAN identificam as redes de longa área. Estas tecnologias, caracterizam-se por terem taxas de transmissão relativamente baixas, aliadas a um grande alcance de transmissão. Estas características apresentam-se bastante vantajosas para sistemas em que é necessário efetuar a comunicação entre máquinas dispersas e que não necessitem de enviar grandes tramas de informação e tempo real.

No mercado, em muitas aplicações, surgiu a necessidade de criação de redes de longa área, porém, aliando esta característica a um baixo consumo energético. Assim, surgiram as LPWAN (*Low Power Wide Area Network*). Nesta temos como principais concorrentes a tecnologia LoRa e SigFox.

Dentro das redes WAN podemos distinguir os seguintes *standards*:

- NB-IoT

Este é o nome dado à tecnologia usada para criar uma rede de comunicação *wireless* de banda estreita entre sensores e dispositivos, que requerem pouca largura de banda. É baseada no modelo de comunicação e infraestruturas dos telemóveis uma vez que utiliza as

mesmas tecnologias, LTE, UMTS e GSM. Esta particularidade fez com que fosse considerada uma tecnologia CIoT (*Cellular IoT*).

As redes NB-IoT são direcionadas para aplicações portáteis com necessidades de alta durabilidade de baterias, grande alcance e zonas em que por exemplo as tecnologias 2G, 3G e 4G não conseguem aceder.

Em termos físicos, uma vez que este sistema é similar ao funcionamento da tecnologia de comunicação dos telemóveis, os dispositivos utilizam um cartão SIM, com rede para operarem. Por outro lado, os equipamentos a integrar nestas redes necessitam de operar com largura de banda pequena e enviar pequenas quantidades de dados.

Como principais vantagens esta tecnologia, apresenta um reduzido custo de implementação, a possibilidade de interligação, em massa, de equipamentos, baixos consumos energéticos, o que permite uma maior duração das baterias, elevados alcances e um maior alcance de penetração, sendo que pode operar em zonas em que outras tecnologias não chegam. De forma adicional, podemos também afirmar que uma outra grande vantagem desta rede é a sua rápida implementação e escalabilidade uma vez que os seus pilares estão assentes na rede mundial atual [19][20][21]. A Figura 10 apresenta as vantagens desta tecnologia.

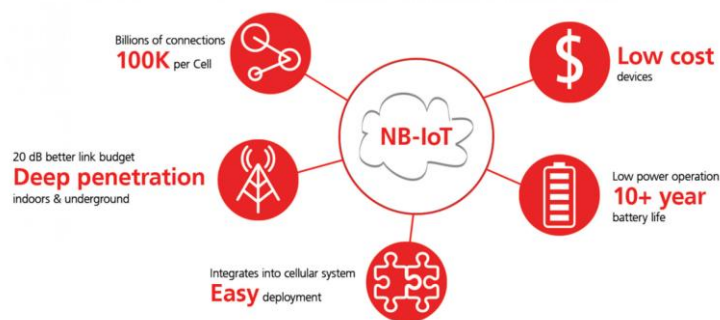


Figura 10 Vantagens da tecnologia NB-IoT [22]

- LoRa

Esta é a designação para a tecnologia de comunicação de dados, sem fio, de longo alcance, que utiliza a radiofrequência para a transmissão de informação entre equipamentos. O LoRa utiliza a modulação CSS (*Chirp Spread Spectrum*) que aliado a diferentes fatores de

propagação (*SF-Spreading Factors*) e com a largura de banda curta, permite ir de encontro às especificações de alcance necessárias para a aplicações distribuídas geograficamente.

Uma vez que este processo faz com que a comunicação perca velocidade, em troca do alcance, podemos afirmar que é mais adequada para sistemas que permitam um *delay* na transmissão dos dados. Assim, esta tecnologia foca-se em aplicações de longo alcance, com baixos custos energéticos e monetários e sem necessidade de envio de dados em tempo real.

Para além das características anteriormente referidas, que são uma das principais vantagens desta tecnologia. Um outro parâmetro o LoRa dos demais concorrentes é o seu *link budget*, uma vez que este parâmetro é superior, chegando a 156 dB.

Do ponto de vista técnico, o LoRa suporta as frequências 433, 868 e 915 MHz, que são bandas de frequência ISM (*Industrial Scientific and Medical*), ou seja, reservadas internacionalmente para o uso industrial, científico e médico.

Em suma, o LoRa é uma tecnologia que se adequa a situações em que é necessária uma comunicação sem fios, de longo alcance, com baixo consumo energético, baixo custo, segura, com baixa largura de banda e sem necessidade de transmissões de dados em tempo real. A Figura 11 apresenta o exemplo da arquitetura do LoRa.

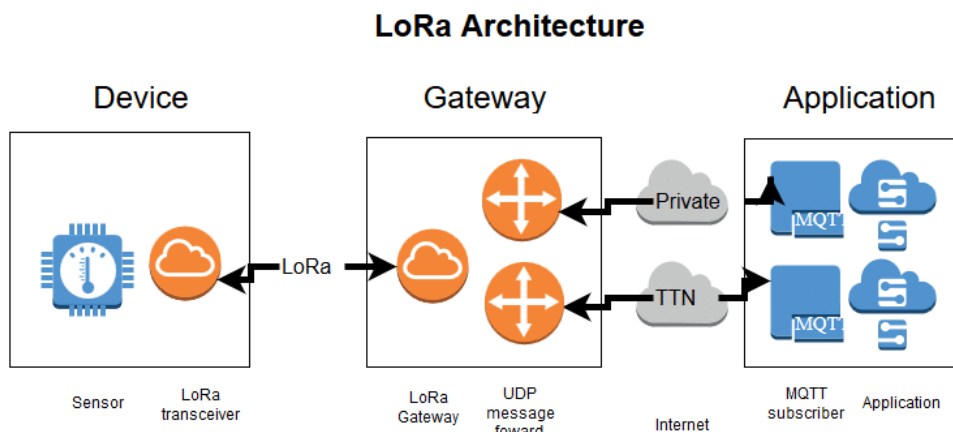


Figura 11 Exemplo de uma Arquitetura LoRa [23]

O LoRa, segundo o modelo OSI faz parte da camada física, uma vez que converte sinais binários, em sinais adequados ao meio de transmissão. Posto isto, surgiu a necessidade de criar um protocolo que controlasse as camadas superiores, nomeadamente a camada de rede.

O LoRaWAN, é um protocolo de controlo de acesso intermédio, MAC (*Media Access Control*) apoiado na camada física do LoRa. Criado pela LoRa Alliance, otimizada para a segurança, os baixos consumos, robustez, latência e baixo custo. A sua origem veio completar a solução LoRa uma vez que o protocolo e a arquitetura da rede influenciam os parâmetros citados anteriormente, como o consumo energético, a capacidade de rede, a qualidade de serviço e segurança.

A arquitetura da rede segue uma topologia em estrela na qual um nó pode-se conectar a vários *gateways*, que por sua vez fazem a transmissão das mensagens entre os equipamentos e o servidor. Estes, por sua vez, estão interligados à rede através de conexões com protocolo IP, sendo que a principal função é a conversão dos dados do formato RF para o formato IP e vice-versa. A Figura 12 apresenta a arquitetura da rede LoRa usando a topologia em estrela implementada pelo protocolo LoRaWAN.

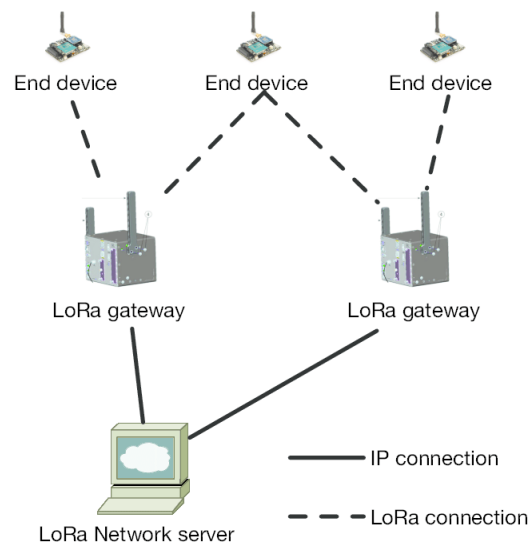


Figura 12 Topologia de rede com protocolo LoRaWAN [24]

O nó LoRa comunica, através do ar, com o *gateway*. Este permite a descodificação de até 10 transmissões concorrentes, ou seja, em simultâneo. Uma vez que podem existir vários *gateways*, no caso de uma transmissão ser detetada por vários *gateways*, o servidor decide qual o *gateway* a utilizar e envia o *acknowledgement*. Cabe ao servidor a autenticação das mensagens e o agrupamento de vários equipamentos, permitindo assim, englobar diferentes tipos de equipamentos em possíveis diferentes localizações que estejam alocados a um mesmo cliente.

Podemos também distinguir classes dentro do protocolo, classe A, B e C. Em todos estes casos temos à disposição comunicações bidirecionais, todavia, no caso da classe B para além da janela variável de receção apresentada na classe A, esta apresenta janelas de receção extra em tempos calendarizados. A Figura 13 apresenta a temporização da janela de receção das diferentes classes.

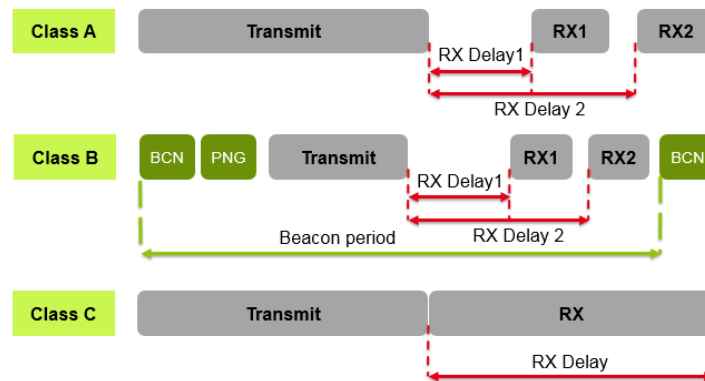


Figura 13 Janelas de receção das diferentes classes dos nós [29]

Para o equipamento poder aceder a esta janela o *gateway* fornece-lhe um período de sincronização. Já, no que toca à classe C, estes só não abrem a janela de receção quando estão a transmitir dados. Este facto, implica um aumento do consumo energético relativamente às outras duas classes, porém, este é o modo de funcionamento que apresenta menor latência entre o servidor e o equipamento final. A Figura 14 apresenta as camadas do LoRaWAN [25][26][27][28].

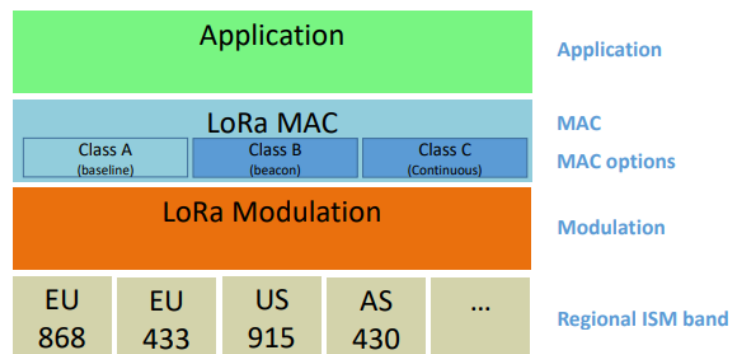


Figura 14 Camadas da comunicação LoRa [30]

- SigFox

O outro grande concorrente da tecnologia LoRa é o SigFox. Este, tal como o seu competidor, é uma tecnologia de largura de banda muito curta, porém, utiliza um método

diferente de transmissão, a modulação por deslocamento de fase (BPSK-*Binary Phase-Shift Keying*). Onde na descida, utiliza uma largura de banda de 600 Hz, para fornecer uma taxa de transmissão de 500 bit/s. Já, no que diz respeito à parte ascendente a banda é de 100 Hz, com velocidade de transmissão de 100 bit/s.

Como principais vantagens temos o elevado alcance, uma vez que este parâmetro pode exceder os 30 km em meios rurais, onde não existem grandes interferências, e a imunidade. Este último deve-se ao facto do sinal ter uma largura de banda pequena, o que implica que os recetores tenham filtro com uma largura de banda idêntica, filtrando assim, grande parte do sinal e do ruído indesejado [31].

Uma rede SigFox é composta por duas camadas, a cada *Network Equipment* e a camada *SigFox Support System*. A primeira camada inclui os dispositivos e os equipamentos que permitem a transmissão das mensagens destes até à camada seguinte. A camada *SigFox Support System*, recebe estas informações, processa-as, e faz o reencaminhamento para o cliente, sendo que é composta por equipamentos de suporte que monitorizam e garantem o bom funcionamento da rede [32]. A Figura 15 apresenta a arquitetura da rede SigFox.

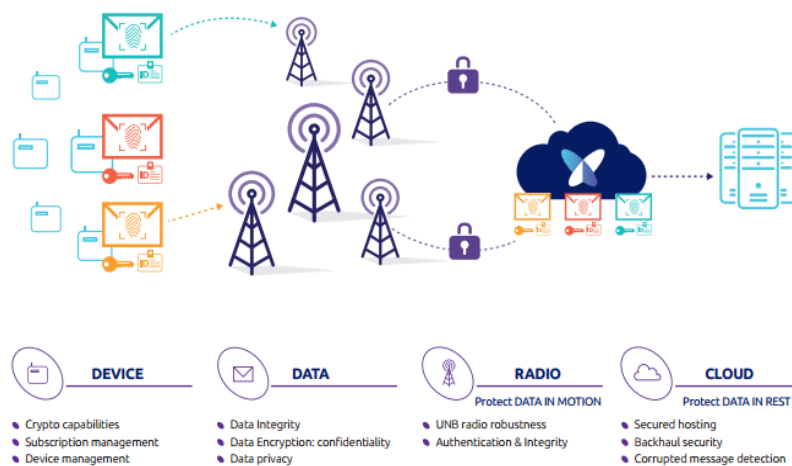


Figura 15 Arquitetura de uma rede SigFox [33]

Para codificar as informações são necessários pequenos segmentos do espectro e alterações de fase, da onda portadora. Através deste processo o SigFox, implementa um mecanismo de mitigação do ruído, uma vez que o recetor da mensagem apenas consegue receber a mensagem em pequenas partes do espectro. A solução oferecida é uma comunicação na qual toda a parte de processamento é realizada fora dos equipamentos, o que faz com que

os consumos energéticos e os custos dos dispositivos sejam diminuídos. Por outro lado, o facto de ter uma taxa de transmissão baixa, permite a utilização de componentes de baixo custo [34][35][36].

- RPMA

Desenvolvida pela companhia, Ingenu, a tecnologia RPMA (*Random Phase Multiple Access*) utiliza a banda ISM 2.4 GHz para oferecer uma solução para aplicações do tipo IoT e M2M (*Machine 2 Machine*). Através desta característica a Ingenu adota um rumo diferente das demais tecnologias, uma vez que estas operam em bandas com melhores propriedades de programação. Por outro lado, esta é uma banda que apresenta regulamentações mais flexíveis, sendo esta uma vantagem. A Figura 16 apresenta a arquitetura de uma rede deste tipo.

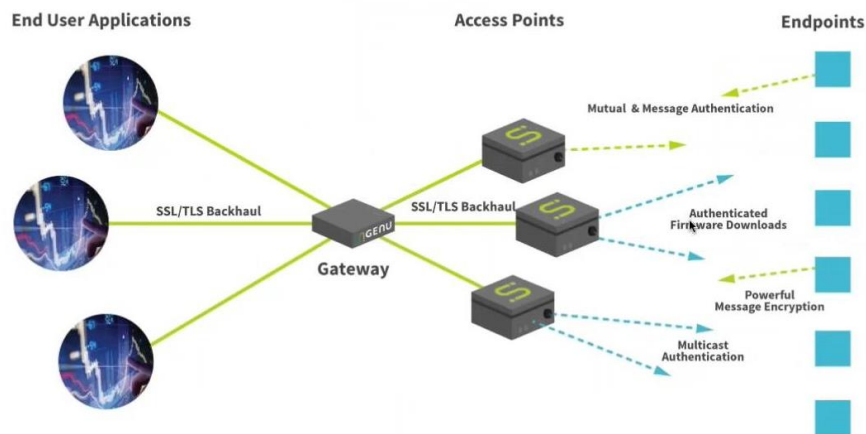


Figura 16 Arquitetura RPMA [45]

O protocolo utiliza um mecanismo patenteado RPMA *direct sequence spread-spectrum*, apenas utilizado nas conceções *uplink*. Assim é possível permitir a vários transmissores a partilha de *time slots*. Isto é possível pois é feita uma adição de um *delay offset* a cada transmissor.

Esta tecnologia tem como vantagens o envio de mensagens *broadcast*, que facilita as atualizações de *firmware*, um elevado alcance, encriptação AES (*Advanced Encryption Standard*) de 128 bits e, como faz parte das LPWAN, comporta baixos consumos energéticos.

Através da maximização da potência é possível atingir até 16 km de alcance. Por outro lado, ao contrário de outras tecnologias desta área, na Europa e nos EUA, esta não

apresenta restrições, em termos de *duty cycle*, pelo que este é um fator importante pois permite a obtenção de um melhor rendimento e maior capacidade em relação à concorrência [37][38][39].

2.6. CONCLUSÃO

Depois do esclarecimento do que são as tecnologias de comunicação entre máquinas, a apresentação dos *standards wireless* e a apresentação da indústria petroquímica, faz toda a relevância a inclusão desta secção, com a finalidade de interligar a indústria de aplicação deste projeto com as várias soluções de mercado apresentadas.

Assim, feita a apresentação da indústria a monitorizar, dos parâmetros de processo com interesse de monitorização e a sua importância é possível tirar conclusões quanto ao interesse e aplicabilidade do sistema a criar, no âmbito do tema do projeto. Tendo como foco as tecnologias LPWAN, a Tabela 3 apresenta a comparação entre as principais concorrentes nesta área, SigFox, LoRa e NB-IoT.

Tabela 3 Comparação entre LPWAN

LPWAN	LoRa	SigFox	NB-IoT
Rede privada	Sim	Não	Não
Comunicação	Half-duplex / Full-duplex	Limited Half-duplex	Half-duplex
Standard	LoRaWAN (Lora-Alliance)	Não	3GPP
Alcance	5 km (zonas urbanas) 15 km (zonas rurais)	10 km (zonas urbanas) 40 km (zonas rurais)	1 km (zonas urbanas) 10 km (zonas rurais)
Segurança	Sim (Encriptação AES-128 bits)	Não	Sim (Encriptação LTE)
Imunidade	Alta	Alta	Baixa
Modulação	CSS (<i>Chirp Spread Spectrum</i>)	BPSK (<i>Binary Phase Shift Keying</i>)	QPSK (<i>Quadrature Phase Shift Keying</i>)
Banda de Frequência	ISM (não licenciada)	ISM (não licenciada)	LTE (Licenciada)
Trama de envio máxima	243 bytes	12 bytes-8 bytes	1600 bytes
Taxa de envio máxima	50 kbps	100 bps	200 kbps

Todas estas tecnologias foram concebidas com o intuito de permitirem uma elevada escalabilidade, cobertura geográfica, alcance da rede e baixa largura de banda, a escolha deve ser feita mediante outras características de interesse ao projeto.

A solução procurada deve ter um elevado alcance, imunidade a ruídos eletromagnéticos e robustez, aliados a baixos consumos energéticos. Para além disto é fulcral a utilização de uma tecnologia segura e de confiança que permite a autenticação e encriptação dos dados enviados. Por outro lado, tal como noutras aplicações industriais, o objetivo é a implementação de uma rede privada.

Assim sendo, a escolha recai sobre uma rede do tipo LPWAN, devido aos requisitos de alcance e gastos energéticos. Por último, apesar do alcance ser uma vantagem do Sigfox em relação ao LoRa, o facto de este não implementar mecanismos de fiabilidade e segurança e não permitir desenvolver uma rede privada, faz com que a escolha recaia sobre a tecnologia LoRa [67].

3. PROTÓTIPO

Uma vez adquiridos todos os conhecimentos necessários para a compreensão e avaliação do estado da arte das tecnologias LPWAN, segue-se a escolha do *hardware* e *software* necessários para levar a cabo a concretização do objetivo deste trabalho. Utilizando assim, os conhecimentos adquiridos das tecnologias de comunicação e implementando estes numa solução que satisfaça os requisitos desejados.

No início deste projeto, foi proposta a realização de uma rede de sensores *wireless*, para monitorização de parâmetros na indústria petroquímica, com recurso ao *kit* de desenvolvimento Dragino. Tendo sido mais tarde adquirido um outro equipamento da Microchip, semelhante a este, por parte da empresa.

Assim sendo, surgiu a necessidade de testar, analisar e comparar estas duas ferramentas, de forma a decidir qual a que mais se adequava ao propósito deste trabalho. De forma adicional, também foi realizado um trabalho de pesquisa para encontrar outras soluções no mercado que significassem uma mais valia para este projeto.

Como tal, este capítulo visa a apresentação dos testes com os equipamentos adquiridos, a apresentação de outras soluções e as conclusões a que foi possível chegar.

3.1. CONTEXTO INICIAL

Primeiramente, um dos objetivos traçados para este trabalho foi a criação de uma rede de teste LoRa. Isto porque, através da criação de um sistema deste tipo era possível validar o conceito pretendido, testar a aplicabilidade da solução estudada e entender a tecnologia em si.

Inicialmente, tendo em conta a proposta apresentada, surgiu a necessidade de comparar a tecnologia LoRa com as outras tecnologias concorrentes, de forma a apresentar uma solução ótima no que diz respeito a este tipo de tecnologias, conhecer o LoRa e procurar entender o funcionamento do *hardware* disponibilizado pela empresa.

3.2. REDE DE TESTE LORA

Assim, a primeira fase passou pela criação de um sistema, que simulasse o funcionamento básico de uma rede LoRa (Figura 17). Ou seja, um sistema composto por pelo menos dois nós, clientes, e um nó central, servidor. Para esta fase do trabalho foi disponibilizado o *kit* de desenvolvimento Dragino, que inclui dois nós, conjunto *Arduino* e *Shiel YUN*, um *gateway* e vários sensores e atuadores.

Presente na placa do Arduino temos o microcontrolador ATMEGA 328p, responsável por todo o processamento dos dados. Por outro lado, nas placas temos o *transceiver* da Semtech, SX1276, responsável pela transformação dos sinais de radiofrequência para sinais perceptíveis pelo microcontrolador.

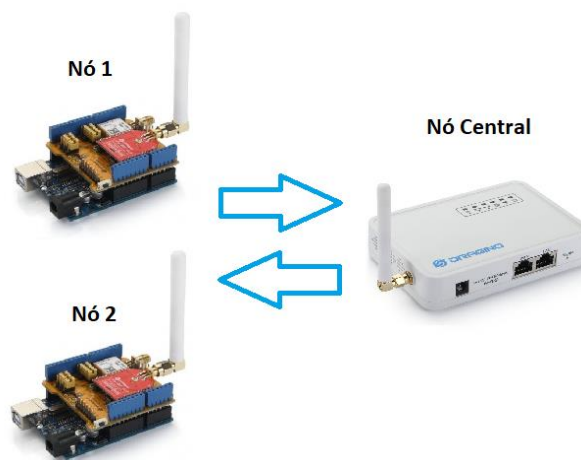


Figura 17 Arquitetura da rede de teste

Nesta fase apenas foi utilizada uma parte da tecnologia LoRa, segundo o modelo OSI (*Open Systems Interconnection*), camada física. Ou seja, nesta rede apenas foi utilizada a tecnologia de modulação, implementada pelo LoRa. Esta consiste na transformação de sinais em frequência rádio, para sinais em tensão. Assim sendo, a pilha protocolar associada à rede implementada tinha como objetivo apenas identificação dos clientes, o correto envio e receção de mensagens.

A arquitetura implementada é do tipo cliente/servidor, uma vez que cada nó da rede apenas comunica com o servidor e vice-versa. Na prática, o servidor opera como um nó central no qual eram guardadas as informações enviadas pelos nós, como resposta à solicitação deste, e feita toda a parte de gestão do funcionamento da rede. A Figura 18 apresenta o conceito inerente ao funcionamento da rede.

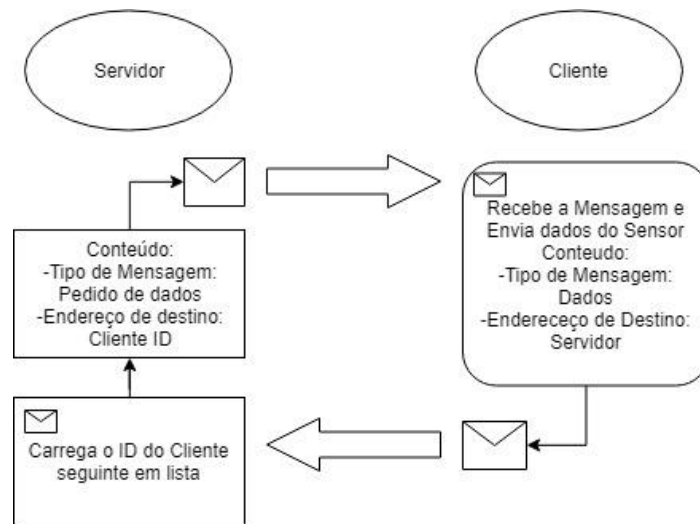


Figura 18 Conceito da rede de teste

Quanto ao funcionamento da rede, presente no nó central está todo o processamento necessário para a operação da rede, bem como o armazenamento da informação e da identificação dos clientes. Como tal, este, assim que é ligado, faz a verificação da existência de equipamentos, através do envio de mensagens *broadcast*, dentro do seu alcance e guarda a identificação destes em memória. Assim, que termina o processo, e é encontrado pelo menos um dispositivo, e o contador já passou 1 minuto, é feito um pedido de informação do estado dos sensores, presentes nestes nós, conforme apresenta o fluxograma da Figura 19.

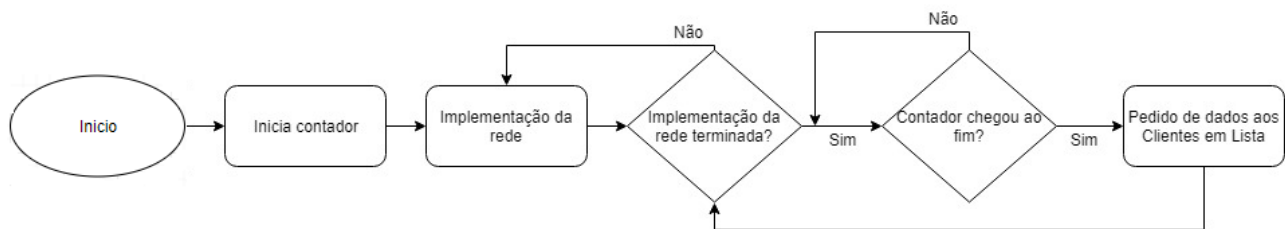


Figura 19 Rotina principal do nó central

No caso de um dos nós, ao qual esta associada a identificação, não responder ao pedido de informação, ao fim de 5 tentativas, este é eliminado da lista e o processo continua até ter sido executado o pedido a todos os nós em lista. De seguida, é feita uma nova verificação e no final deste processo no caso do nó central ter ainda pelo menos um cliente em lista, este entra em ciclo, pedindo informação aos clientes em lista. Na situação inversa, significa que este não tem em rede nenhum dispositivo, pelo que entra no modo de implementação da rede outra vez.

Desta forma é possível para o servidor eliminar equipamentos que já não estejam ao seu alcance ou que estejam em manutenção, e posteriormente voltar a adicioná-los à rede no caso de estes voltarem a estar operacionais e dentro do alcance ou adicionar novos elementos.

Por outro lado, do lado dos nós estes, assim que ficam operacionais, ficam em espera até receberem uma mensagem do nó central, para se juntarem à rede. Assim que a recebem, respondem com o seu ID e aguardam a chegada da mensagem de confirmação da adesão.

Uma vez confirmada a sua inclusão na rede, estes entram em ciclo, no qual aguardam até que o servidor lhes envie uma mensagem com um pedido de dados sobre os sensores, conforme mostra o fluxograma da Figura 20.

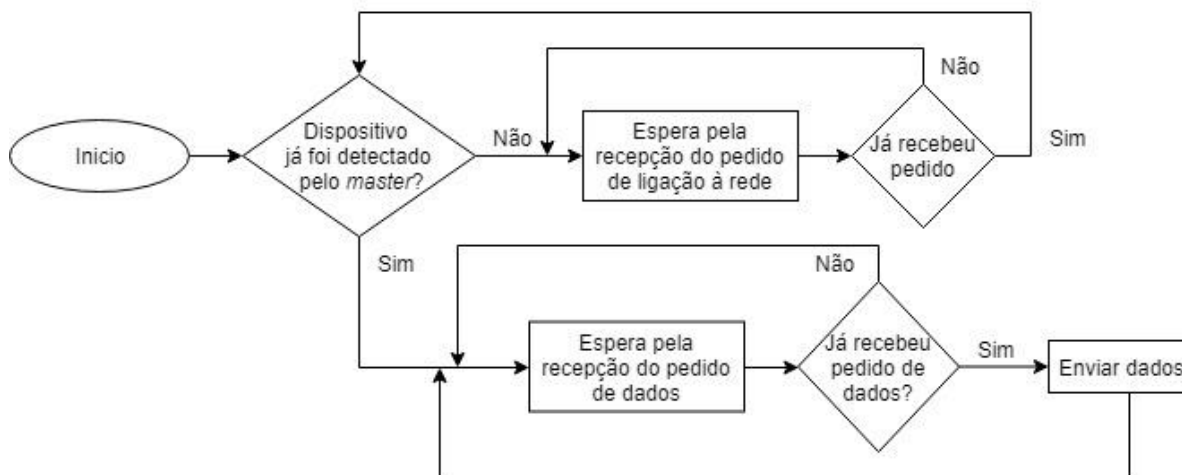


Figura 20 Rotina principal do nó

Esta tarefa foi concluída com sucesso sendo que foram acrescentadas algumas funcionalidades. Estas sub tarefas, de menor dimensão, tornaram a rede de teste mais robusta, flexível e melhoraram o seu desempenho. A primeira foi a utilizar as mensagens de pedido de dados como reconhecimento do estado dos clientes em rede, sendo que caso estes estivessem indisponíveis, ou seja, não respondiam aos pedidos, eram eliminados e procedia-se a uma verificação de novos clientes.

Ou seja, quando um cliente era eliminado, a rede procedia à verificação da existência de novos clientes. No caso de existirem, estes, eram adicionados à rede. Na situação contrária, a rede continuava o seu funcionamento normal, com os clientes em lista anteriores. Como tal, foi adicionado mais um nó à rede, de forma a testar o seu correto funcionamento.

Este último ponto é fundamental uma vez que numa aplicação prática, como a deste projeto, são necessários sistemas de monitorização capazes de permitir a total mobilidade dos equipamentos, sendo possível remover ou até adicionar elementos, no caso de estes serem necessários. Isto também permite detetar se elementos, que estão na rede, não estão a funcionar corretamente, permitindo a sua reparação rápida.

Este teste inicial, foi uma primeira abordagem prática a esta tecnologia, e serviu para tirar conclusões quanto ao tipo de arquitetura a implementar, modo de funcionamento e operacionalidade do LoRa. Uma vez que neste caso, apesar da rede utilizar esta tecnologia, não apresenta todas as suas vantagens, nomeadamente, em termos de consumos energéticos. Isto deve-se ao facto de que os equipamentos de fim de linha estarem sempre a

enviar os dados para o nó central, de forma a garantirem que a sua mensagem chega ao seu destinatário. Isto comporta desvantagens, uma vez que aumenta os gastos energéticos destes e dificulta a escalabilidade da rede.

Este último ponto é importante uma vez são necessários sistemas de monitorização capazes de adicionar elementos, no caso de estes serem necessários, ou também detetar se elementos, que estão na rede, não estão a funcionar corretamente, permitindo a sua reparação o mais rápido possível.

Numa segunda etapa, foi adicionada a esta comunicação, o envio de dados de teste de sensores, por parte dos nós, para o servidor, e a configuração das identificações dos vários equipamentos da rede, uma vez que era necessário encontrar uma forma para atribuir os números de série dos equipamentos à sua identificação na rede.

No que diz respeito aos dados de teste, foram utilizados sensores disponíveis no *kit* de desenvolvimento Dragino. Nesta fase, foi utilizado o sensor de luminosidade no envio de informações.

Uma vez concluídas estas duas etapas, ficou concluída a primeira sub tarefa do planeamento, “Criação de uma Rede Experimental LoRa”, todavia, ainda era necessário interligar o gestor da rede LoRa, com a base de dados e o servidor da empresa.

Por outro lado, nesta fase, conforme referido anteriormente, foi apenas utilizada a camada física do LoRa. O programa foi realizado através do ambiente de desenvolvimento Arduino IDE, e com o recurso às rotinas presentes na biblioteca da RadioHead, responsáveis pela interação com o *transceiver* da Semtech, SX1276, presente na Placa Yun disponibilizada no *Kit*.

Estas serviram para, configurar os parâmetros físicos da ligação, como são exemplo a frequência e a amplitude de envio, enviar e receber dados e também para verificar a disponibilidade do meio do módulo para o envio das mensagens.

3.3. REDE DE TESTE LORAWAN

3.3.1. CONTEXTO TEÓRICO

Concluído o passo inicial seguiu-se a interligação da rede com um servidor. No que diz respeito ao LoRa existe uma grande variedade de servidores *open-source* que implementam o protocolo LoRaWAN, pelo que foi escolhido o servidor disponibilizado pela The Things Network, que é um dos participantes do projeto LoRa Alliance.

A escolha deste servidor deveu-se ao facto de que a maioria dos servidores disponíveis, para além de outras limitações, apenas processam a informação. Já no caso deste servidor é possível aceder a diversas ferramentas de tratamento e armazenamento de dados, como por exemplo interligar a rede com serviço MQTT (*Message Queing Telemetry Transport*).

Para além desta opção temos também aplicações relacionadas com a localização e rastreio dos dispositivos, como o COLLOS e TTN (*The Things Network*) *mapper*, aplicativos de controlo e monitorização, Ubidots, TagoIO e OpenSensors. Todas estas opções estão disponíveis na secção *Integrations* no aplicativo criado.

No caso do utilizador não pretender utilizar nenhuma destas aplicações, é também possível implementar, através de *javascript*, rotinas de descodificação, conversão, validação e codificação dos dados recebidos, através da opção *PayloadFormats* (Figura 21).

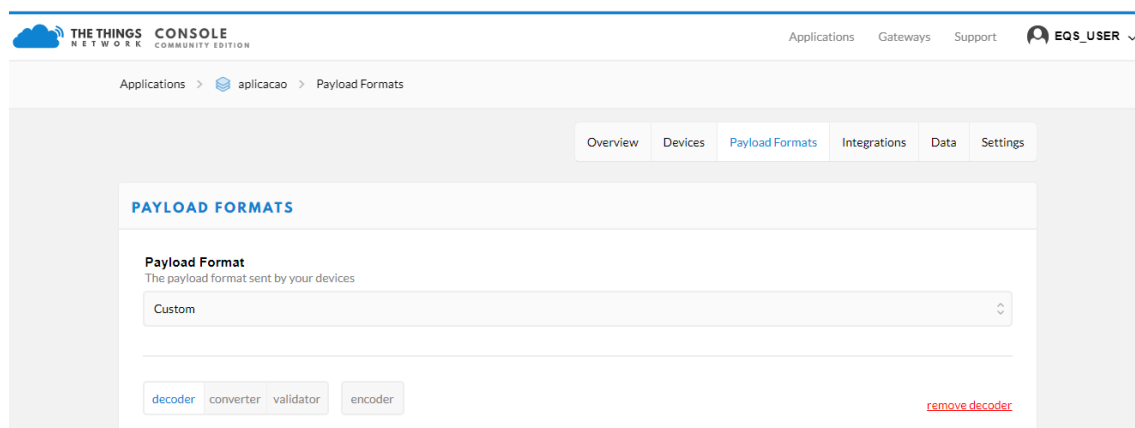


Figura 21 Formatos de dados

3.3.2. PROTOCOLO

Do lado do protocolo, este foi projetado pela LoRa Alliance, que é uma aliança de várias empresas com interesse na tecnologia e que contribuíram para a criação deste. O objetivo é a interligação de dispositivos de diferentes fabricantes que utilizam o LoRa criando assim um *standard* nesta área.

Suportado sobre a camada física do LoRa, o protocolo insere-se na camada MAC segundo o modelo OSI. Ou seja, corresponde à segunda e terceira camada do modelo. Todo o processamento e gestão da rede é efetuado pelo servidor. Desta forma, o processamento nos dispositivos de campo, quer sejam eles *gateways* ou nós, é reduzido ao máximo, sendo que estes apenas fazem a interação com o meio e/ou, no mínimo, o envio dos dados de forma ascendente, ou seja *uplink*. Isto entende o período de duração das baterias destes dispositivos, sendo que segundo a LoRa Alliance estes podem chegar até aos 10 anos.

Por outro lado, o protocolo em si implementa um mecanismo de segurança AES, 128 bits, que encripta as comunicações quer entre o dispositivo e o servidor quer entre o dispositivo e a aplicação. O que torna as comunicações bastante seguras. A Figura 22 apresenta a arquitetura do protocolo e a encriptação utilizada. Este também permite a utilização de *gateways* de outros proprietários para o envio das mensagens para o servidor, com máxima segurança, sem que este tenha acesso à informação.

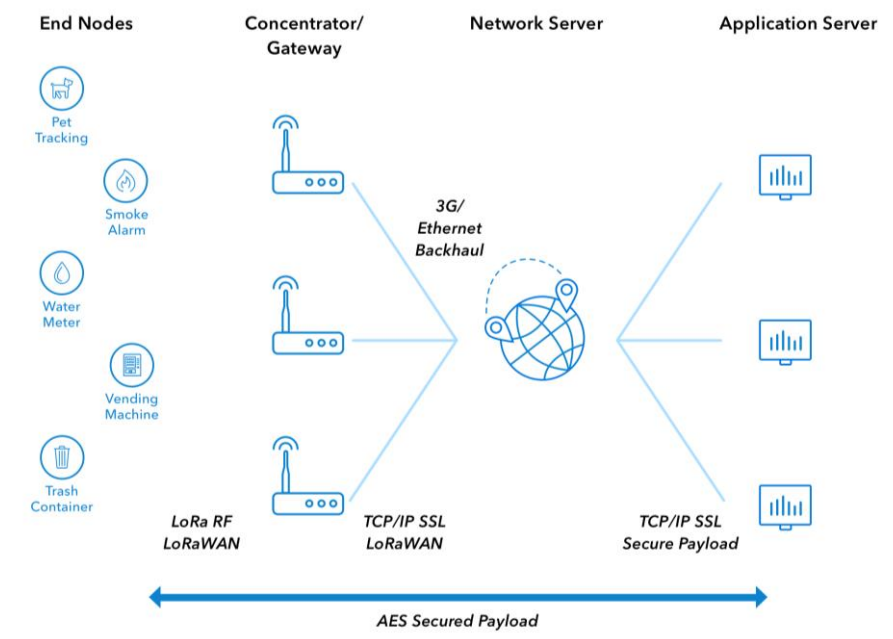


Figura 22 Arquitetura LoRaWAN[48]

De forma adicional, para atingir o máximo alcance possível, o servidor implementa mecanismos de adaptação das taxas de envio de dados, ADR (*Adaptative Data Rate*), através da manipulação do SF (*Spreading Factor*), da largura de banda e do potência de sinal. Por exemplo, no caso de termos um dispositivo que esteja a enviar informação com um SF de 12, que implica um *delay* no envio das mensagens, e o servidor verifique que existe margem para reduzir este valor, o equipamento vai receber esta indicação e vai diminuir o valor, passando assim a enviar dados de forma mais rápida.

Outra particularidade é a forma de ativação dos dispositivos. Existem duas formas possíveis a OTAA (*Over The Air Activation*) e a ABP (*Activation By Personalization*). No primeiro caso, os dispositivos efetuam um procedimento para aceder a rede no qual as chaves são negociadas, AppSKey (*Application Session Key*), NwkSKey (*Network Session Key*) e o endereço do dispositivo é atribuído (DevAddr) (Figura 24). Este é o método mais seguro, uma vez que estes processos são atribuídos de forma dinâmica, porém, têm como necessidade principal a existência de *downlink* na rede.

Por outro lado, no caso do ABP, para a ativação do dispositivo basta a criação de um endereço e das chaves, e posterior programação deste com estes valores. Este é um método menos seguro uma vez que as chaves são criadas antes do processo, podendo ser utilizadas por outro dispositivo sendo que também não há confirmação da correta correspondência das chaves com o dispositivo pelo que este é programado anteriormente [51][52].

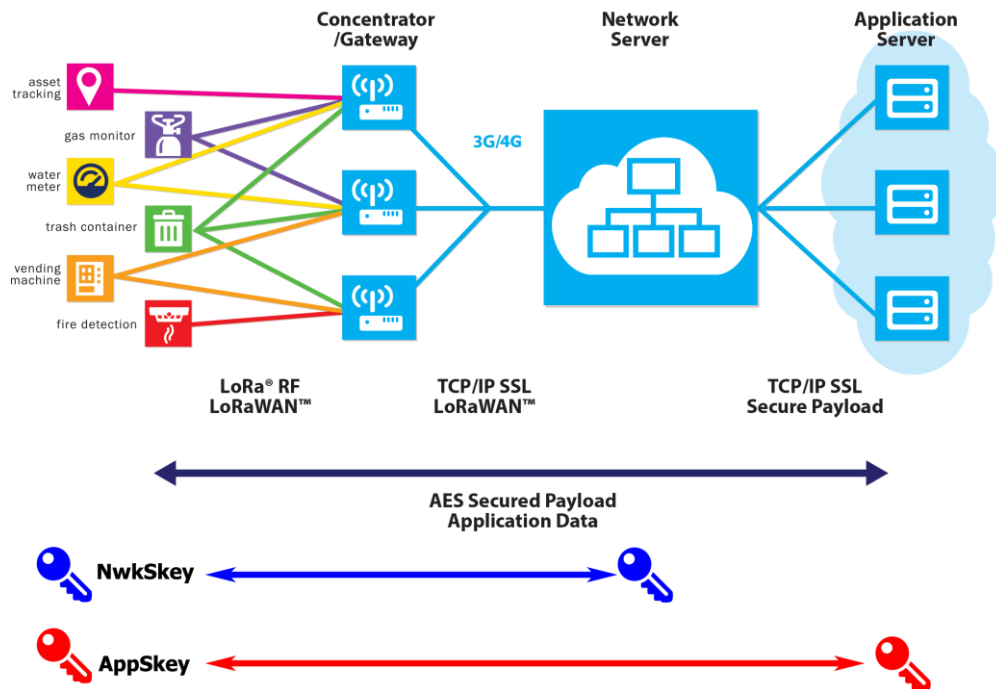


Figura 23 Arquitetura LoRaWAN [50]

3.3.3. SISTEMA

Neste caso, foi mais uma vez utilizado o *kit* de desenvolvimento Dragino para o efeito. Todavia, foi necessário programar os nós com o protocolo LoRaWAN, configurar o *gateway* para o encaminhamento dos pacotes, para o servidor, e o estabelecimento da ligação à Internet através do protocolo IP. De forma adicional, foi também necessária, o registo do *gateway* na plataforma TTN com a identificação do *gateway*, a frequência utilizada e o *router* associado. A Figura 25 apresenta o *gateway* criado no servidor.

Apesar do protocolo LoRaWAN, no continente Europeu, permitir a utilização de frequências entre 863-870 Mhz, foi apenas configurada a frequência de 868.1 com SF de 7 e largura de banda de 125, uma vez que o *gateway* apenas suporta um canal, que equivale a uma única frequência. Esta foi escolhida de entre 9 frequências disponíveis pelo servidor TTN, sendo que apenas 8 correspondem à tecnologia LoRa e a outra é a modulação FSK (*Frequency-Shift Keying*) [59].

GATEWAY OVERVIEW ⚙️ settings

Gateway ID eui-a840411b50f0ffff

Description LG01-P EQS GATEWAY

Owner EQS_USER [Transfer ownership](#)

Status ● connected

Frequency Plan Europe 868MHz

Router ttn-router-eu

Gateway Key base64

Last Seen 4 minutes ago

Received Messages 1011

Transmitted Messages 3

Figura 24 Gateway registado no servidor TTN

Conforme apresenta a Figura 25, o servidor fornece informações relativas ao número de mensagens por este recebidas, mensagens transmitidas, estado e configurações. Isto são tudo dados que ajudam a perceber o estado do sistema e também a detetar possíveis falhas.

O mesmo processo foi realizado para os nós, todavia, para estes foi necessário criar uma “aplicação”, para depois poder registar os dispositivos. Neste caso, foi necessário identificar o dispositivo e o método de ativação, ABP. Que é o único modo suportado por este *gateway*. Neste método são utilizados os parâmetros DevAddr, NwkSKey e AppSKey, que são criados antes da ligação do dispositivo e são atribuídos posteriormente. A Figura 26 apresenta a aplicação criada.

Também, do lado dos nós, é possível verificar os dados enviados, recebidos e qual a última vez que este enviou informações para o servidor, sendo esta uma funcionalidade interessante do servidor uma vez que permite verificar o estado do nó.

DEVICE OVERVIEW

Application ID aplicacao

Device ID arduino1

Activation Method ABP

Device EUI <> ≡ 11 27 27 13 82 84 18 84 📄

Application EUI <> ≡ 70 83 D5 7E D0 01 9D 91 📄

Device Address <> ≡ 26 01 14 68 📄

Network Session Key <> ≡ 🔒 41 91 06 13 50 2C E5 A7 EA 36 76 47 F6 91 B5 B9 📄

App Session Key <> ≡ 🔒 57 54 ED 54 64 BC 12 59 A0 4F 3D FB 2E 9E 48 87 📄

Figura 25 Nó registado no TTN

A Figura 27 apresenta a arquitetura do sistema de teste, utilizando o equipamento da Dragino. A rede é constituída por 2 nós, que vão fazer as leituras de sensores e enviar estes dados, através da tecnologia LoRa, sob o protocolo LoRaWAN, para o servidor The Things Network. A encaminhar os pacotes dos nós até ao servidor está o *gateway*, LG01.

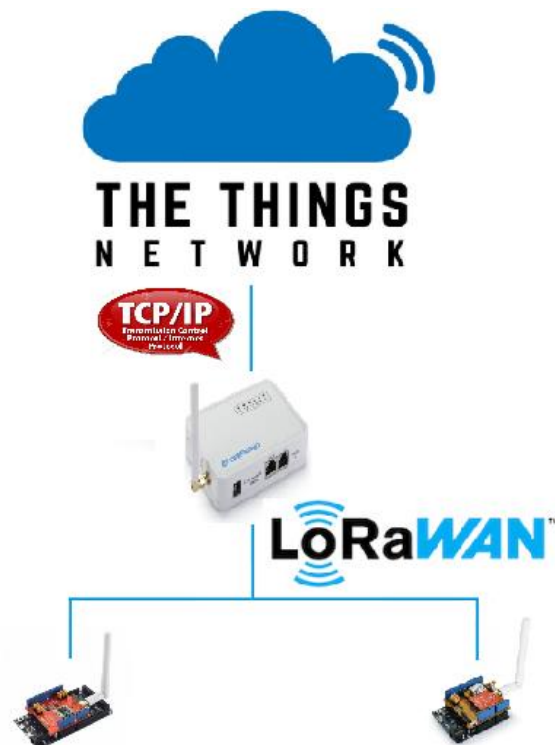


Figura 26 Arquitetura do sistema

3.3.4. DRAGINO

Por outro lado, foi necessário programar os nós de forma a que estes incluíssem o protocolo LoRaWAN e comunicassem com o *gateway*. Por outro lado, também foi necessário programar o *gateway* de forma a que este fosse capaz de fazer o *uplink* das mensagens recebidas da rede para o servidor e ainda configurar as especificações da comunicação rádio e o servidor LoRaWAN.

No que diz respeito ao *gateway*, para a configuração deste foi necessário, primeiramente, aceder à rede Wi-Fi criada quando este é ligado. Para depois ser possível proceder à configuração, através do *browser* utilizando o IP (*Internet Protocol*) predefinido do equipamento, 10.130.1.1.

Foi utilizado, para fazer o encaminhamento das mensagens para o servidor, o código disponibilizado pela Dragino que foi implementado no sistema Linux do *gateway*, através do acesso à secção Flash MCU. Sendo que foi também necessário configurar a ligação à Internet, por parte deste, para fazer o transporte dos pacotes para o servidor.

Radio Settings
Radio settings requires MCU side sketch support

TX Frequency	868100000
	Gateway's LoRa TX Frequency
RX Frequency	868100000
	Gateway's LoRa RX Frequency
Encryption Key	Encryption Key
Spreading Factor	SF7
Coding Rate	4/5
Signal Bandwidth	125 kHz
Preamble Length	8
	Length range: 6 ~ 65536

Figura 27 Configurações da rádio frequência

Já no que diz respeito à configuração dos parâmetros da ligação física, por rádio frequência, foi necessário aceder à página LoRaWAN, onde foram configuradas as frequências de transmissão, receção, *Spreading Factor*, *Coding Rate*, *Bandwidth* e *Preamble Length* (Figura 28). Também nesta página, foram configuradas as características

do servidor como o seu endereço, a Porta e o ID do *gateway*, conforme apresenta a Figura 29.

LoRa Gateway Settings

Configuration to communicate with LoRa devices and LoRaWAN server

LoRaWAN Server Settings

Server Address	router.eu.thethings.network
Server Port	1700
Gateway ID	a84041168f24fff
Mail Address	edwin@dragino.com
Latitude	Location Info
Longitude	Location Info

Figura 28 Configurações do protocolo

Por outro lado, no que diz respeito aos nós, uma vez que o módulo utilizado, SX1276, apresenta na placa Dragino não inclui a *stack* protocolar, é necessário incluí-la no microcontrolador. Como tal, foi necessário programar estes com o protocolo LoRaWAN, especificar o tipo de classe de dispositivo, as chaves definidas previamente na aplicação e ainda as características desejadas para a ligação rádio. Estas últimas têm de estar sincronizadas com as definições do *gateway*, uma vez que este não suporta a totalidade do protocolo.

Existem algumas bibliotecas disponíveis que são compatíveis com o protocolo LoRaWAN, sendo que a maioria utiliza como recurso a biblioteca LMIC (LoRaMAC-in-C) desenvolvida pelo laboratório de pesquisa de Zurique, IBM. Para o efeito foi utilizada esta ferramenta sendo que teve que ser adaptada para a utilização do ambiente de trabalho em questão.

A escolha desta ferramenta deveu-se ao facto de implementar as especificações do protocolo LoRaWAN, em linguagem C, e ser compatível com os módulos da Semtech, nomeadamente, o módulo SX1276 utilizado no *kit* Dragino. Para além disto foi utilizado o código fonte uma vez que permite uma maior flexibilidade e versatilidade do que outras ferramentas disponíveis que operam num nível de abstracção mais elevado.

Após análise e consulta do documento, com as especificações técnicas desta, foi possível entender o funcionamento da biblioteca e proceder aos ajustes necessários para criar uma solução operacional com o equipamento em questão.

Assim, uma das primeiras dificuldades foi a programação dos pinos do microcontrolador, em concordância com a placa utilizada. Este foi um ponto fulcral uma vez que qualquer divergência neste ponto impedia o correto funcionamento do equipamento.

Resolvido este problema, através de várias tentativas e consulta de várias referências, foi possível encontrar o correto mapeamento dos pinos da placa Dragino.

Seguiu-se o entendimento e compreensão da biblioteca. Esta implementa a rotina `onEvent()`, que é responsável por fazer o despacho dos eventos do protocolo, contagens de tempo e manuseamento de interrupções. Por outro lado, todas as rotinas implementadas são executadas na rotina principal do programa, `os_runloop()`, responsável pelo agendamento destas dentro do *run-time* do sistema.

Para dar início ao funcionamento, é necessário executar a inicialização do sistema (Figura 30), chamar uma rotina inicial, para gerar um evento e por fim, chamar a rotina principal, `os_runloop()`, que necessita de um evento inicial para gerar os eventos seguintes [55].

```
void main () {
    osjob_t initjob;

    // initialize run-time env
    os_init();
    // setup initial job
    os_setCallback(&initjob, initfunc);
    // execute scheduled jobs and events
    os_runloop();
    // (not reached)
}
```

Figura 29 Rotina principal [55]

Os nós foram programados para operar na frequência EU-868 como dispositivos de classe A. No que diz respeito às janelas de recepção, *timers* e outros eventos que necessitem sincronismo, estes são geridos pela biblioteca implementada.

Por outro lado, foi necessário também, programá-los com as especificações desejadas para a camada física como o valor do SF, 7, e poder de transmissão, 14 dB. De forma adicional,

foi também configurada a parte de receção do módulo com um SF de 9, que é o valor utilizado para a janela de transmissão do servidor.

Em conformidade com o teste anterior, também neste, foram adicionados sensores para simular o funcionamento de um sistema real. Assim, foi adicionado o sensor de luminosidade a um dos nós e enviados os dados deste para o servidor.

Os resultados demonstram que este passo foi bem-sucedido, uma vez que quer do lado do *gateway*, os dados foram recebidos, quer do lado da API do servidor. No que diz respeito ao seu conteúdo, conforme apresenta o protocolo, só na parte das aplicações é possível fazer a descriptação completa, e ter acesso aos dados. Facto este que também se confirmou. De seguida são apresentados os resultados quer do lado do *gateway* (Figura 31) quer do lado da aplicação (Figura 32).

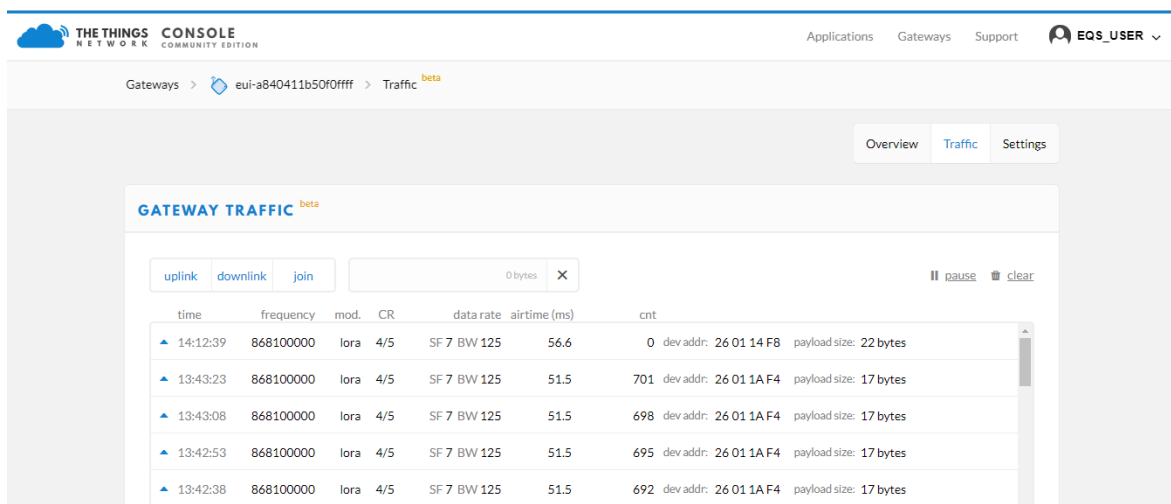


Figura 30 Tráfego de dados do gateway

Conforme apresenta a Figura 31, é possível verificar a chegada da mensagem do lado do *gateway*. Uma vez que este protocolo utiliza a encriptação AES, com duas chaves, não é possível fazer a recolha dos dados enviados pelo dispositivo, por questões de segurança. Estes dados só podem ser descriptados do lado da aplicação, conforme apresenta a Figura 32. Os dados apresentados em hexadecimal “6C 6F 72 61”, são a trama enviada pelo nó, com a *string* “lora”.

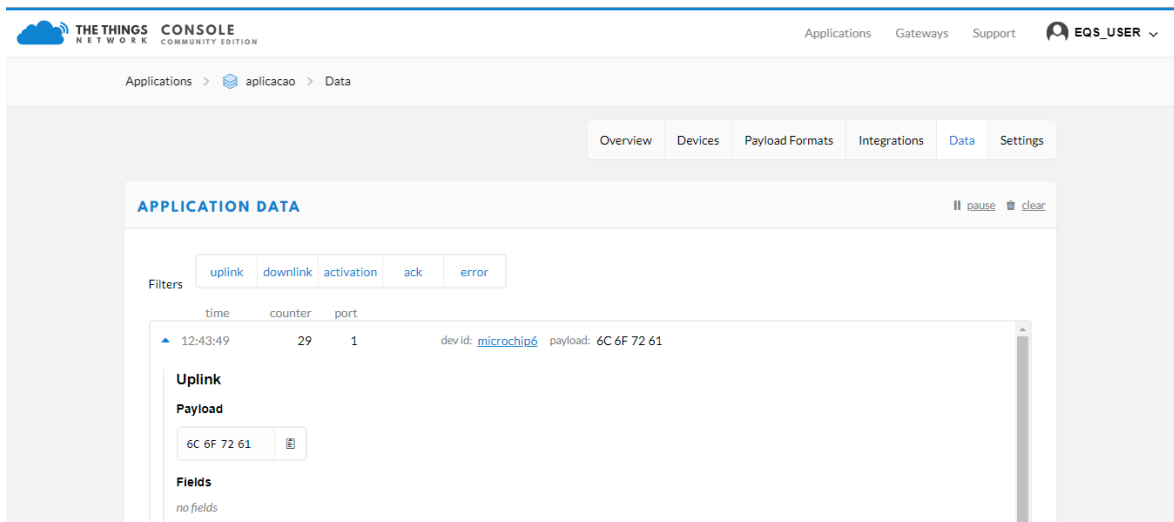


Figura 31 Dados recebidos pela aplicação

Após a interligação deste sensor, foi lançado um novo desafio, a interligação de mais um sensor neste nó. Neste caso, foi escolhido o sensor DTH11, um sensor de temperatura e humidade. A sua escolha deveu-se ao facto de ser um sensor mais complexo que o de luminosidade, uma vez que utiliza um protocolo com um único barramento para o envio de duas leituras e acrescentar mais duas grandezas físicas às leituras do equipamento (Figura 33).

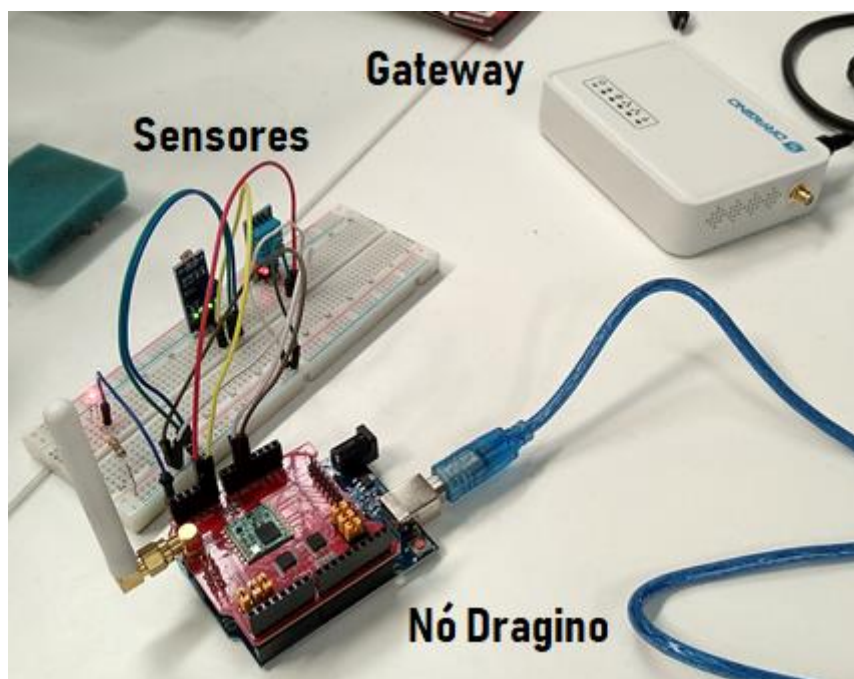


Figura 32 Sistema desenvolvido

Feita a interligação do componente com o nó, procedeu-se à criação do código para a leitura deste e acrescentar estes dados à trama de dados enviada anteriormente.

Paralelamente a esta parte do trabalho foi também descoberta uma funcionalidade do servidor que permitia descodificar a informação recebida, em hexadecimal, e através da criação de um código em *javascript*, descodificar esta e apresentá-la na forma desejada. A Figura 34 mostra a receção dos dados por parte do servidor e apresentação destes no formato desejado. A Figura 35 apresenta os dados enviados pelo nó.

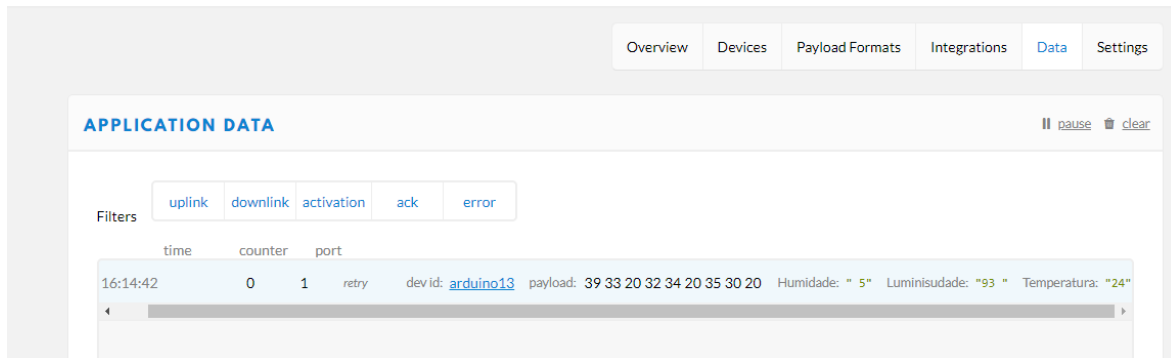


Figura 33 Dados recebidos pelo servidor

Do lado do *software*, nos nós, foram criadas rotinas que preenchem um vetor, *byte a byte* com os dados do sensor de luminosidade, temperatura e humidade, inserindo um espaço entre cada valor. Este é o formato de dados enviado.

No campo *payload*, é apresentada a informação recebida pelo servidor, no formato hexadecimal, pelo que é possível identificar a concordância entre os valores enviados e a trama de dados criada. De seguida são apresentados os campos criados pelo código desenvolvido na secção *PayloadFormats*. Este faz a identificação dos campos dos dados, converte estes para valores nas unidades do sistema internacional e apresenta-os.

Neste ponto surgiram os primeiros problemas, uma vez que após a adição destas linhas de código para a leitura de sensores, o sistema deixava de enviar os dados e proceder à sua normal operação, o envio de dados em intervalos de tempo de 20 segundos. Enviando apenas os dados uma primeira vez, deixando depois de fazer o ciclo de funcionamento normal.

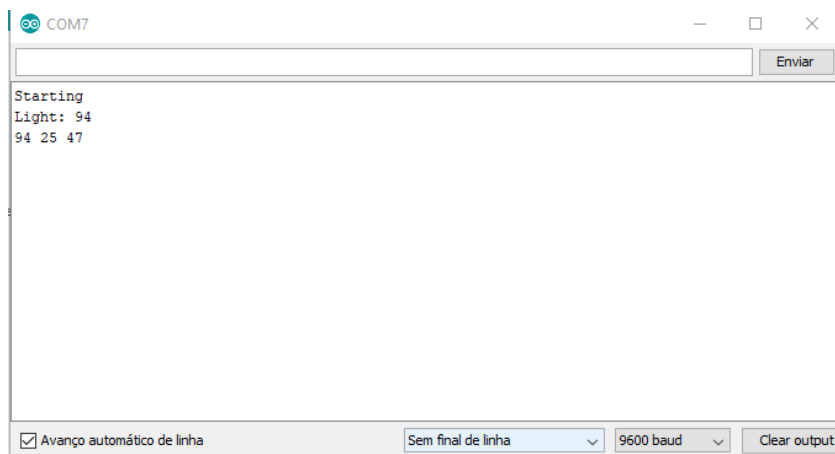


Figura 34 Dados enviados pelo nó

Assim, com desconhecimento da origem do problema, foi feita uma avaliação do código de forma metódica, dividindo este em partes de forma a tentar encontrar a fonte do problema.

Após várias tentativas e uma vasta análise do código, não foi possível compreender a origem deste problema, sendo que foi decidido fazer um retrocesso ao desenvolvimento e voltar a analisar a biblioteca.

Com os dados obtidos nos testes anteriores, reparou-se que o sistema conseguia enviar os dados do sensor de luminosidade, se apenas realizasse a sua leitura e apenas algumas instruções adicionais.

Assim, e em conformidade com a pesquisa efetuada, foram encontrados indícios que apontavam para um problema de *timings* do protocolo. Para este problema, existiam três possíveis soluções. Uma que era procurar fazer um relaxamento das janelas de envio de dados por parte do protocolo ou, por outro lado, diminuir as instruções de processamento e envio dos dados do sensor ou ainda, uma outra opção, que não resolvia o problema mas contornava-o, que era a alteração dos sensores.

Nesta parte optou-se por numa primeira fase, por otimizar o código uma vez que, a alteração dos *timings* da biblioteca poderia significar a dessincronização desta com as janelas de receção do servidor, impedindo assim o envio das mensagens.

Por outro lado, também se testou a leitura de um outro sensor analógico de forma a verificar se o sistema ficava operacional com a leitura de dois sensores analógicos, o sensor de luminosidade e o sensor de temperatura. Neste caso, foi utilizado o sensor de

comunicações e limitações no número de comunicações. Pelo que não foi encontrado nenhum indício que comprometesse as comunicações, ligado ao servidor [61].

Por último, uma vez que já tinham sido verificadas todas as alternativas possíveis para resolver o problema do envio dos dados dos sensores para o servidor, optou-se por voltar à primeira opção, a otimização do código de leitura dos sensores.

Desta vez, foi possível reparar que quer do lado de leitura dos valores quer do lado da sua escrita, existiam instruções que poderiam ser otimizadas. Primeiro, uma vez que o sensor DHT11 possui dois sensores, a rotina de leitura estava a ser chamada duas vezes. Em pesquisa, foi possível reparar que esta ação apenas era necessária uma vez. Por outro lado, a trama de dados enviada, convertia os dados recebidos do tipo *float*, para inteiro e retirava o carácter decimal. Também estas instruções foram alteradas e o processo simplificado.

Após a realização destes testes foi possível o envio dos dados dos sensores de forma cíclica. A Figura 38 apresenta as comunicações quer do lado da aplicação no servidor, quer do lado do dispositivo.

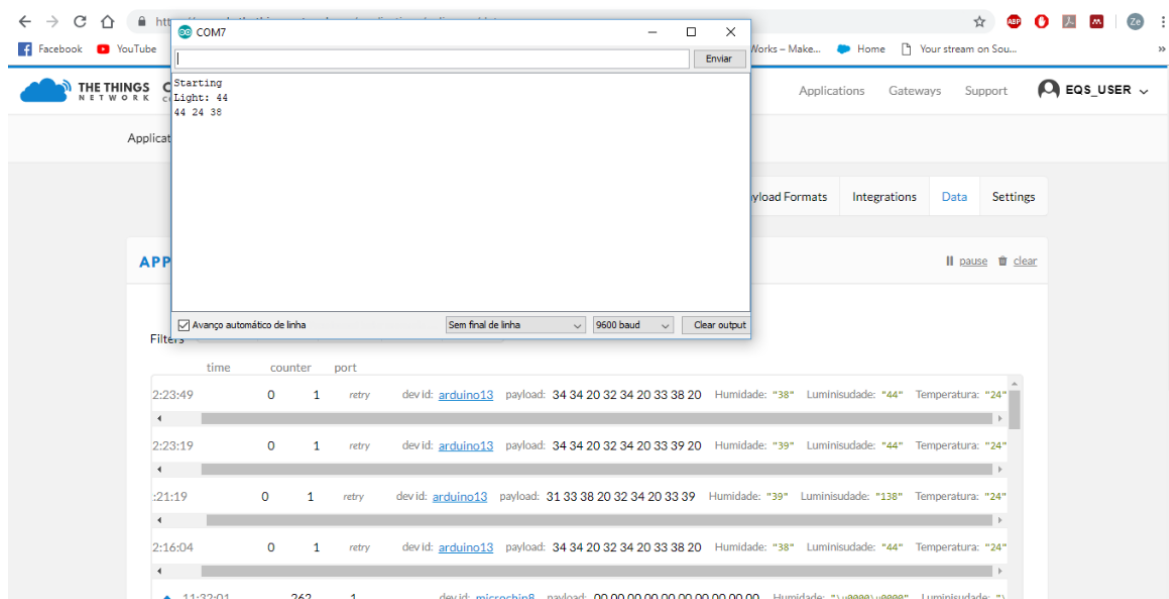


Figura 37 Comunicações do nó para o servidor

Mais uma vez, foi possível identificar outro ponto a melhorar no código. Desta vez, o problema prendeu-se com o facto de que a biblioteca estar concebida para o protocolo LoRaWAN. Ou seja, para ser utilizada num dispositivo de fim de linha que está conectado

a um *gateway full* LoRaWAN, um *gateway* com 8 canais que suporta todas as frequências disponíveis na região.

Este processo faz com que seja facilitado o envio de dados para os *gateways*, uma vez que os equipamentos podem estar ligados a diferentes canais. Por outro lado, também permite aliviar os canais e reduzir as colisões.

Por outro lado, nesta etapa o dispositivo estava a enviar dados, no tempo desejado, porém em frequências para as quais o *gateway* não estava programado, pelo que esta característica foi eliminada.

Assim, através da consulta do documento da biblioteca foi possível verificar a existência de uma rotina de cancelamento de canais, `LMIC_disableChannel()`. Na região europeia, estão disponíveis 8 canais, como tal, foi executada uma rotina para desligar todos os canais, com a exceção do canal 0, sendo que este foi utilizado para as comunicações. As figuras seguintes apresentam os resultados obtidos. Na Figura 39 as mensagens recebidas pela aplicação do servidor e o estado do nó. Por outro lado, na Figura 40, são apresentados e decodificados os dados recebidos por parte da aplicação.

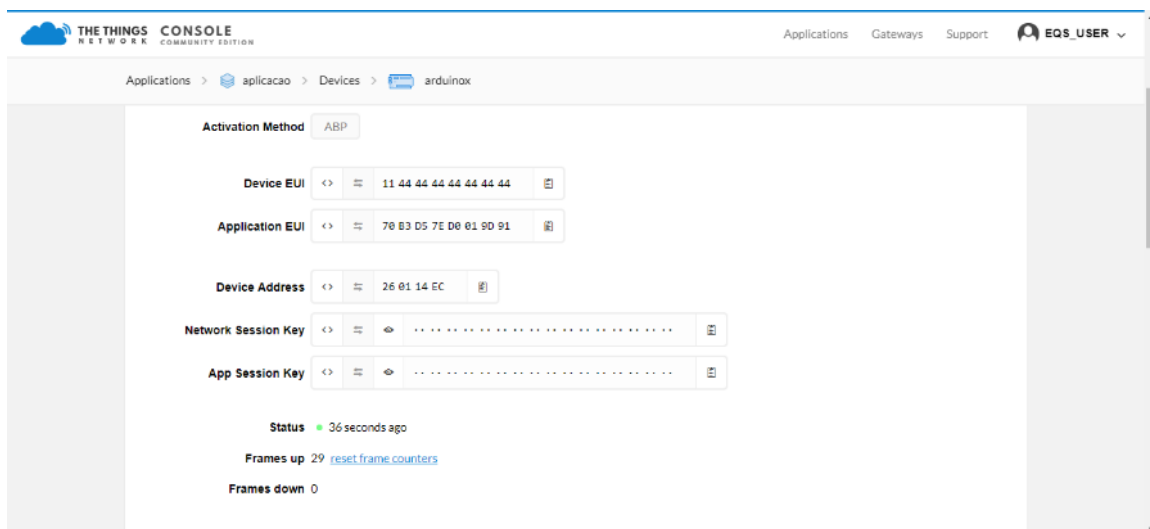


Figura 38 Registo de comunicações do servidor

The screenshot shows the 'Data' page in The Things Network Console for an application named 'aplicacao' and a device named 'arduino'. The data is filtered by 'uplink'. The table below represents the data shown in the screenshot.

time	counter	port	payload	Humidade	Luminisidade	Temperatura
16:54:16	29	1	payload: 34 30 38 20 32 31 20 37 32	"72"	"488"	"21"
16:49:15	28	1	payload: 33 36 35 20 32 31 20 37 32	"72"	"365"	"21"
16:44:15	27	1	payload: 33 36 30 20 32 31 20 37 32	"72"	"368"	"21"
16:39:00	26	1	payload: 33 36 38 20 32 31 20 37 32	"72"	"368"	"21"
16:33:59	25	1	payload: 33 36 39 20 32 31 20 37 33	"73"	"369"	"21"
16:28:59	24	1	payload: 33 36 36 20 32 31 20 37 32	"72"	"366"	"21"
16:23:59	23	1	payload: 33 38 30 20 32 31 20 37 33	"73"	"388"	"21"
16:18:44	22	1	payload: 33 39 38 20 32 31 20 37 33	"73"	"398"	"21"
16:13:43	21	1	payload: 34 31 30 20 32 31 20 37 32	"72"	"418"	"21"
14:27:07	0	1	retry: payload: 31 31 35 20 32 31 20 36 38	"68"	"115"	"21"

Figura 39 Registo de comunicações na aplicação do servidor

Os resultados apresentados na Figura 40 permitem afirmar que o objetivo foi concluído. O equipamento faz o envio de leituras de sensores, nomeadamente, humidade, luminosidade e temperatura no intervalo de tempo escolhido no código.

Durante todos os testes foi utilizado o equipamento da Microchip LoRa Mote, que fornece valores de temperatura e luminosidade, de forma a confirmar o correto funcionamento dos sensores.

Também nestes testes, o *gateway* adquirido, demonstrou não ser a solução adequada para este sistema uma vez que conforme a Figura 40 demonstra, com uma rede constituída por um único nó, as mensagens foram todas recebidas no intervalo de tempo desejado. Todavia, nos testes seguintes, com a interligação de mais um nó, o número de mensagens recebidas e os intervalos de tempo em que estas eram recebidos não condiziam com o programado, conforme apresentam as Figuras 41 e 42.

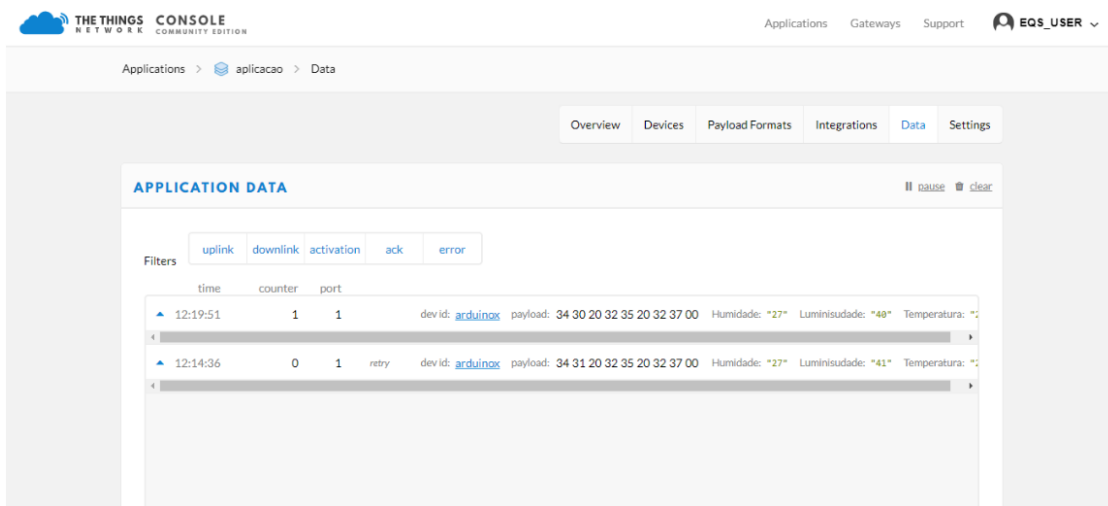


Figura 40 Registo das comunicações do nó com sensores

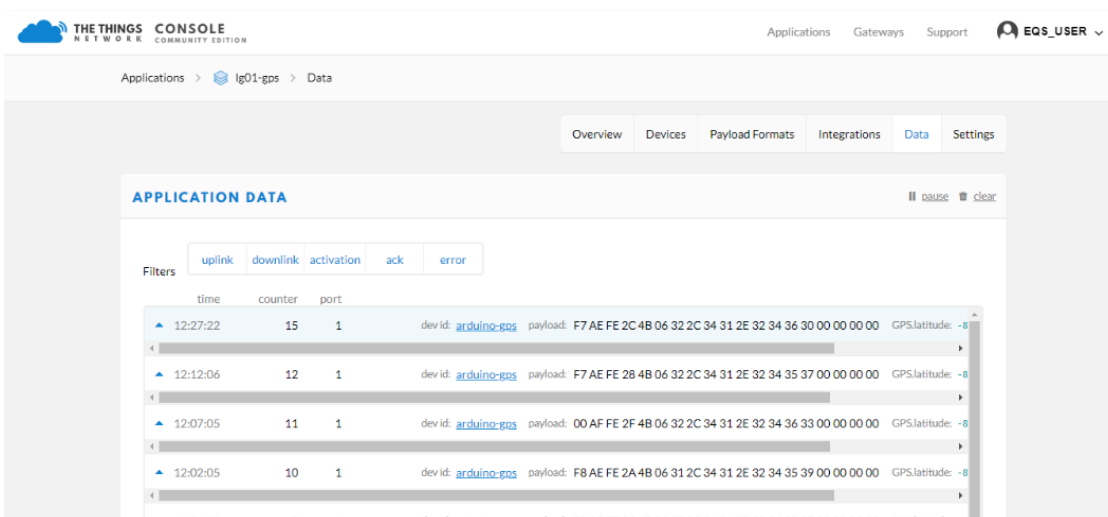


Figura 41 Registo das comunicações do nó com GPS

Chegou-se à conclusão de que estas não eram recebidas no servidor pois o canal encontrava-se ora ocupado com o nó 1 ora ocupado com o nó 2. Daí a disparidade de *timings* apresentadas. Ou seja, apesar dos nós estarem programados para enviar os dados de 5 em 5 minutos, os dados não eram recebidos no servidor, nestes intervalos de tempo, devido às restrições físicas do *gateway*.

Este foi o primeiro teste que permitiu verificar as limitações reais deste *gateway* numa situação prática.

3.3.5. DRAGINO GPS

Concluídos todos os passos anteriores, foi lançado o desafio de tornar possível a comunicação das coordenadas GPS dos nós para o servidor. Esta era uma ideia que desde o início do projeto podia ser abraçada, uma vez que o *kit* adquirido já incluía uma placa com módulo GPS.

Assim, o objetivo passou por utilizar o código desenvolvido anteriormente, criar as rotinas necessárias para a leitura do módulo GPS e aplicá-lo ao dispositivo.

O primeiro passo foi a visualização dos dados enviados pelo módulo para o MCU, através da utilização do USB-TTL (*Universal Serial Bus-Transistor to Transistor Logic*) e do *software* de visualização série, Termite. Para tal, foi consultado o *datasheet* do equipamento e o guia de comandos do protocolo.

Os comandos utilizados por este são os NMEA (*National Marine Electronic Association*), que é um *standard* nas comunicações de dispositivos de navegação, no qual está inserido o GPS.

A estrutura de mensagens enviadas (Figura 43) é composta por 1 *byte*, que identifica o início de mensagem, o tipo do dispositivo, no caso “GP” de GPS e o ID da mensagem NMEA, com 3 *bytes*. Segue-se o campo de dados, delimitado por vírgulas, o carácter de fim de dados, (*), o campo de confirmação, com 2 *bytes* e, por último, o carácter de fim de mensagem.

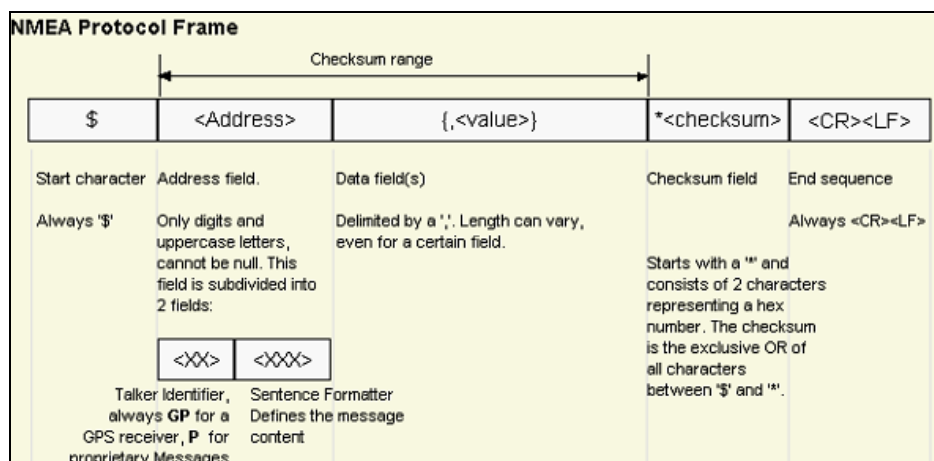


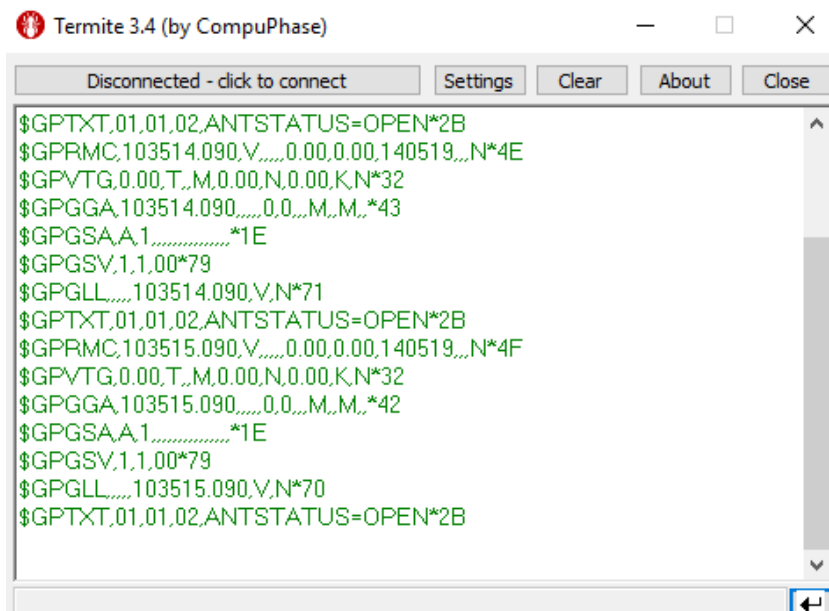
Figura 42 Estrutura de comandos NMEA [66]

De entre os vários campos enviados pelo módulo, GP-RMC (*Recommended minimum position data*), GP-VTG (*Track made good and ground speed*), GP-GGA (*Global positioning system fix data*), GP-GSA (*GNSS DOP and Active Satellites*), GP-GSV (*GNSS satellites in view*), GP-GLL (*Geographic Latitude and Longitude*) e GP-TXT, os campos de maior relevância são o GP-RMC e GP-GGA, que contém as informações da latitude, longitude e altitude, entre outros dados.

Seguindo a estrutura apresenta anteriormente dentro do campo de dados, a mensagem do tipo GPRMC envia o tempo no formato “hhmmss.sss”, seguido do campo “N/S”, longitude em graus e minutos “dddmm.mmmm”, seguido do campo “E/W”, velocidade, percurso sobre o solo em graus, data, variação magnética e a posição do dispositivo.

Por outro lado, a mensagem GPGGA apresenta o tempo, a latitude, longitude, número de satélites, altitude, altura acima do mar, entre outros dados.

Foi possível reparar que nos primeiros testes o GPS não apresentava qualquer valor nos campos de dados de interesse, latitude, longitude, altitude, data, tempo (Figura 44). Em pesquisa foi possível chegar à conclusão que este necessitava de tempo até sincronizar com os satélites e obter os dados, este tempo chama-se de TTFB (*Time To First Fix*). Isto porque o módulo nunca tinha sido utilizado e encontrava-se no modo *Cold Start* [62].



```
Termite 3.4 (by CompuPhase)
Disconnected - click to connect  Settings  Clear  About  Close
$GPTXT,01,01,02,ANTSTATUS=OPEN*2B
$GPRMC,103514.090,V,,,,,0.00,0.00,140519,,,N*4E
$GPVTG,0.00,T,,M,0.00,N,0.00,K,N*32
$GPGGA,103514.090,,,,,0.0,,M,,M,,*43
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPGSV,1,1,00*79
$GPGLL,,,,,103514.090,V,N*71
$GPTXT,01,01,02,ANTSTATUS=OPEN*2B
$GPRMC,103515.090,V,,,,,0.00,0.00,140519,,,N*4F
$GPVTG,0.00,T,,M,0.00,N,0.00,K,N*32
$GPGGA,103515.090,,,,,0.0,,M,,M,,*42
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPGSV,1,1,00*79
$GPGLL,,,,,103515.090,V,N*70
$GPTXT,01,01,02,ANTSTATUS=OPEN*2B
```

Figura 43 Dados enviados pelo módulo GPS

Esta foi a única razão que justificou o problema encontrado. Em consulta ao *datasheet*, este valor era no mínimo 12 minutos, todavia, no teste efetuado, o equipamento chegou a demorar duas horas para sincronizar, o que levou a concluir a fraca conectividade do local.

Na Figura 45 é possível verificar que o dispositivo já apresenta os valores do tempo e da data sincronizados, todavia, ainda não apresenta as coordenadas. A Figura 46 apresenta os dados enviados pelo equipamento após a sincronização.

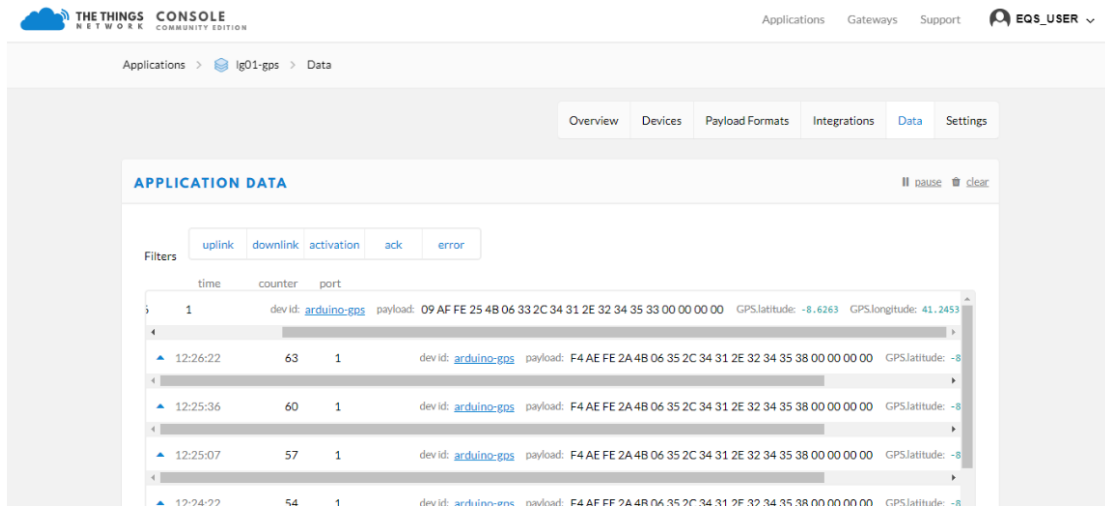


Figura 44 Dados enviados pelo nó com GPS

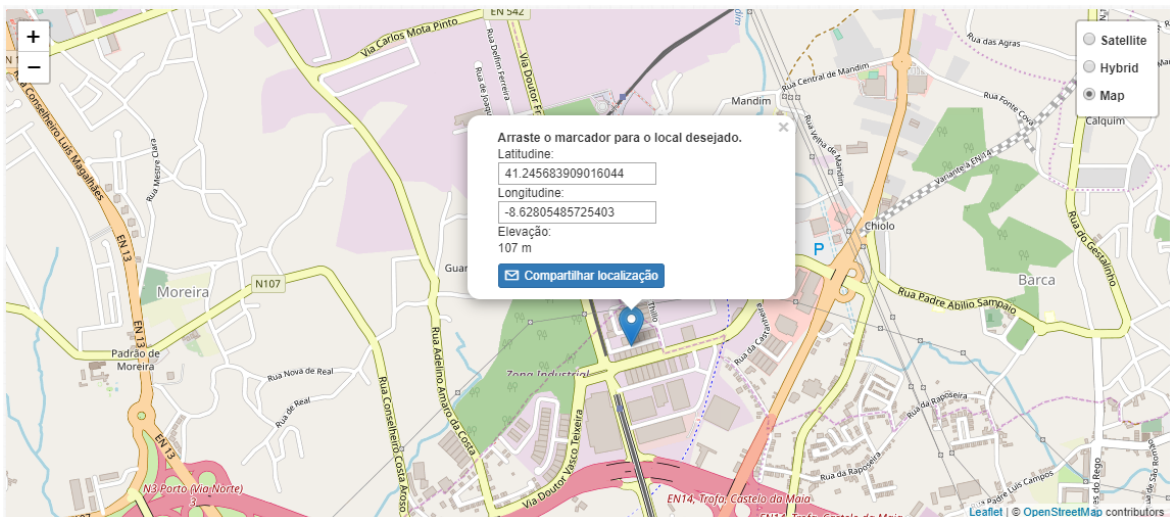


Figura 45 Localização do nó

Conforme é possível verificar, através da Figura 46, o dispositivo envia as coordenadas do local onde se encontra.

Após esta ação, o equipamento funcionou normalmente durante o resto dos testes, porém, após um período de desativação, este voltou a realizar o processo, sendo possível concluir que se encontrava no modo *warm*. Isto acontece quando o dispositivo se encontra desligado por volta de um dia e a informação que tem armazenada encontra-se fora de data [63].

3.3.6. TESTES

Por último, para tornar o sistema completamente portátil e realizar os testes relativos às distâncias para os quais este funciona, foi necessário desenvolver uma forma de alimentação para utilizar nos nós. Para o caso, uma vez que o objetivo é a portabilidade, é necessário a inclusão de uma bateria, sendo que a questão prende-se com a ligação desta ao equipamento.

A placa utilizada permite a operação com 5 V, no caso de estes serem estáveis, ou entre 7 a 12 V, no caso da alimentação através do pino Vin. No caso deste projeto, para simulação e testes foi utilizada uma bateria de 9 V.

Na consulta ao *datasheet* do equipamento, foram fornecidas informações que apontavam para alguma instabilidade relacionada com tensões abaixo dos 7 V, pelo que este não é o caso.

Nos testes realizados com o sistema, a ser alimentado pela bateria, não foram observados quaisquer problemas com a instabilidade dos *timers* ou funcionamento anormal do módulo que levassem à escolha de outro método de alimentação ou até a inclusão de um conversor *buck*. Todavia, posteriormente foi adquirida uma *Power bank* que também foi utilizada na alimentação da placa.

Seguiram-se os testes para verificar as distâncias máximas de operação do sistema. Segundo os criadores da tecnologia e do protocolo, estes suportam comunicações até 15 km em linha reta. Todavia, o contexto no qual estão inseridos, de difícil acesso no meio prático, são esperados resultados de alguns quilómetros. Neste caso foi utilizado o percurso presente na Figura 47.



Figura 46 Percurso de teste

Uma vez que esta era uma implementação simplificada do protocolo LoRaWAN, tendo em conta que o *gateway* utilizado não suportava o protocolo na totalidade, eram previstos resultados satisfatórios em termos de distâncias, porém eram esperados *delays* nas comunicações.

Isto porque uma das principais vantagens do LoRaWAN é o ADR, que permite um ajuste, enviado do servidor para os dispositivos de fim de linha, do SF e da largura de banda mais adequados para o dispositivo transmitir as informações. Permitindo que este opere em parâmetros mais apropriados para as condições em que se encontra.

Nos testes foi possível verificar que o sistema, configurado para enviar dados de minuto em minuto, assim que chegou ao ponto A, a cerca de 110 m, começou-se a verificar uma latência nas suas comunicações. Ou seja, estas, aparentemente, deixaram de ser recebidas pelo *gateway*, tendo depois, ao fim de algum tempo voltado a aparecer na aplicação do servidor, todavia, não nos períodos de tempo programados.

No ponto B, que se encontra a cerca de 428 m, a latência anteriormente verificada nas comunicações, acentuou-se, tendo sido necessário esperar cerca de meia hora para que estas voltassem a chegar até à aplicação. Assim que foram recebidos os primeiros dados, verificou-se que existiam perdas de pacotes de cerca de 50%.

Seguiu-se o teste no ponto C, situado a 410 m do *gateway*. Os resultados obtidos foram semelhantes aos obtidos no ponto anterior. Tal como seria esperado, dado a relativa proximidade das distâncias e semelhança nas estruturas envolventes nos dois casos.

Por último, no ponto D, a 90 m do equipamento, a latência nas comunicações diminuiu, assim como a perda de pacotes. O sistema passou relativamente pouco tempo, apresentou resultados satisfatórios, tendo os dados sido recebidos pela aplicação nos intervalos de tempo desejados.

Os resultados dos testes foram satisfatórios. De antemão já eram conhecidas as limitações desta tecnologia, uma vez que os pacotes são enviados, com a esperança de que sejam recebidos pelo *gateway* mais próximo, que no caso é o *gateway* criado.

Todavia, não era esperado o *delay* apresentado nas comunicações durante as deslocações, entre os diversos pontos, nem mesmo quando o sistema ficou imobilizado nos locais desejados.

3.3.7. CONSIDERAÇÕES FINAIS

Esta parte do trabalho serviu para tirar importantes conclusões sobre os equipamentos a utilizar, nomeadamente no que diz respeito ao servidor, *gateways* e dispositivos de fim de linha. Por outro lado, também foi possível compreender o funcionamento prático do protocolo, bem como entender certos problemas que poderiam surgir com a utilização deste.

Nesta fase do projeto prendia-se uma dúvida relativamente aos equipamentos adquiridos, pois era necessário saber se com a utilização do protocolo, estes funcionavam conforme desejado e se se adequavam à futura aplicação prática do projeto. Assim foi necessário responder a duas perguntas. Se estes nós e o *gateway*, com as suas limitações, se enquadravam ao propósito deste projeto.

No que diz respeito aos nós, estes cumprem os requisitos necessários para a aplicação. Por outro lado, ficaram bem patentes as limitações do *gateway*. Uma vez que este não suporta o método de ativação OTAA, implica um ponto fraco em termos de segurança, por outro lado, também não suporta *downlink*, devido à sua implementação física, uma vez que temos um microcontrolador ATMEGA 328p, a interligar o *transceiver* RFM95 e a placa

principal do Dragino. Esta ponte acrescenta um *delay* entre as comunicações, que numa aplicação como a do LoRaWAN, no qual as janelas de receção são fulcrais, não permite o seu funcionamento. A Figura 48 apresenta as interligações do equipamento.

LG01 System Overview:

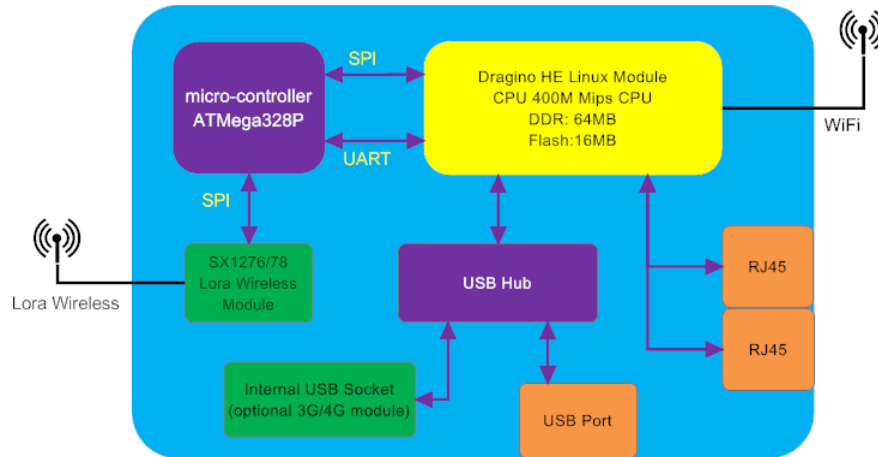


Figura 47 Gateway LG01-P Dragino [53]

Por outro lado, nos equipamentos de fim de linha temos alguns pontos fracos. Um deles é a necessidade de inclusão da *stack* protocolar do LoRaWAN no microcontrolador, uma vez que esta não está presente no *transceiver* utilizado na placa. Isto reduz a memória disponível para o programa. Outros aspetos negativos prendem-se com a reduzida flexibilidade permitida no código, através da utilização da biblioteca LoRaWAN, relacionadas com operações, interrupções e *timings* desta.

Esta é uma característica procurada para a arquitetura do projeto, uma vez que a comunicação com o nó é um ponto de interesse, permitindo a verificação do estado da ligação, o correto envio de mensagens, atualizações ou reparações ao equipamento, entre outras.

Por outro lado, a programação implementada nos equipamentos de fim de linha não permite flexibilidade de processamento para o envio de leituras de sensores, sem que a janela de transmissão do protocolo seja corrompida. Este facto condiciona o objetivo do projeto uma vez que impede o envio destes dados de forma cíclica, conforme é desejado.

3.4. MICROCHIP LORA MOTE

3.4.1. INTRODUÇÃO

Concluídos os primeiros testes com as redes experimentais utilizando o *kit* de desenvolvimento Dragino. A Criação da rede LoRa, numa primeira fase, e criação de uma rede LoRaWAN, numa segunda, surgiu um novo desafio.

Por parte da empresa foi adquirido um módulo da Microchip, DM164138 (Figura 49). Este incluía um módulo RN2483, capaz de suportar a modulação LoRaWAN, e um microcontrolador PIC. Este faz parte de outro *kit* de desenvolvimento, porém da Microchip. Como tal, incluí uma ferramenta que permitia a configuração da comunicação do módulo, para além de ser uma solução pronta a utilizar. Ou seja, este, já se encontrava programado com o protocolo LoRaWAN, como dispositivo de classe A onde era possível enviar valores de temperatura e luminosidade, mediante configuração dos parâmetros da rede.



Figura 48 Nó LoRa da Microchip (DM164138) [54]

Assim sendo, surgiu o desafio de interligar este dispositivo na rede LoRaWAN anteriormente criada e posteriormente, comparar as duas soluções e chegar a uma conclusão sobre qual seria a melhor opção a escolher quer em termos de *kit* de desenvolvimento quer na definição da solução a implementar como dispositivo de fim de linha.

3.4.2. DESENVOLVIMENTO

Conforme abordado anteriormente, o primeiro passo foi a instalação da ferramenta Lora Development Suite, que permitiu através da ligação USB (*Universal Serial Bus*) ao módulo, o envio de comandos para o módulo RN2483, para a configuração. Paralelamente a este processo foi criado o dispositivo no servidor TTN de forma a criar os campos necessários para a ativação. Através dos comandos foi necessário enviar os valores dos sensores para o servidor.

Durante este processo surgiu um problema, uma vez que nesta ferramenta, para além da possibilidade de configuração do módulo, entre outras funcionalidades, temos também a atualização do *firmware*. Por engano, esta tarefa foi executada, sem nenhum ficheiro, o que fez com que o equipamento deixasse de funcionar corretamente.

Este foi um ponto que dificultou a conclusão desta parte uma vez que em pesquisa foram encontradas poucas informações sobre o tema. Todavia foi possível tirar algumas conclusões sobre a melhor forma de resolver este problema.

Uma vez que a origem do erro deveu-se a uma errada atualização do *firmware*, foi necessário testar se o problema era do microcontrolador que controlava a placa, PIC18LF45K50, ou se era do microcontrolador presente no *transceiver*, PIC17LF46K22, conforme apresenta a Figura 50. Para tal, o primeiro passo foi a identificação do componente responsável pela origem do problema.

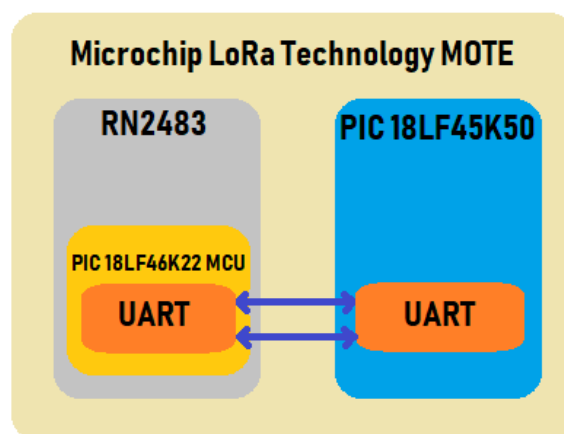


Figura 49 Diagrama de blocos do Microchip LoRa MOTE

Uma vez que o problema era do PIC que controlava a placa, apenas existia uma solução. Instalar o *firmware* de origem, todavia, este processo apenas podia ser realizado com o auxílio de um programador PICKit 3.

Este foi um ponto negativo identificado no componente, uma vez que para resolver um problema deste tipo era necessário adquirir um outro equipamento, com preço aproximado.

Assim que o equipamento foi disponibilizado pela empresa foi possível repor o *firmware* do módulo. Na placa, existem dois conectores para programador, o conector J2 e J5. Neste caso, foram utilizados os pinos do J2, uma vez que estes dão acesso ao módulo desejado, e foi feita a instalação, conforme apresenta a Figura 51.



Figura 50 PICKit3 e LorA Mote da Microchip

Com o recurso ao ficheiro referente ao *firmware* do módulo RN2483 (RN2483-v101.for.ICSP.Hex) foi feita a instalação, através do programa PICKit 3 (Figura 52), e este voltou a ficar operacional. Estando configurado para fazer a leitura da luminosidade e temperatura e enviar estes dados para o servidor.

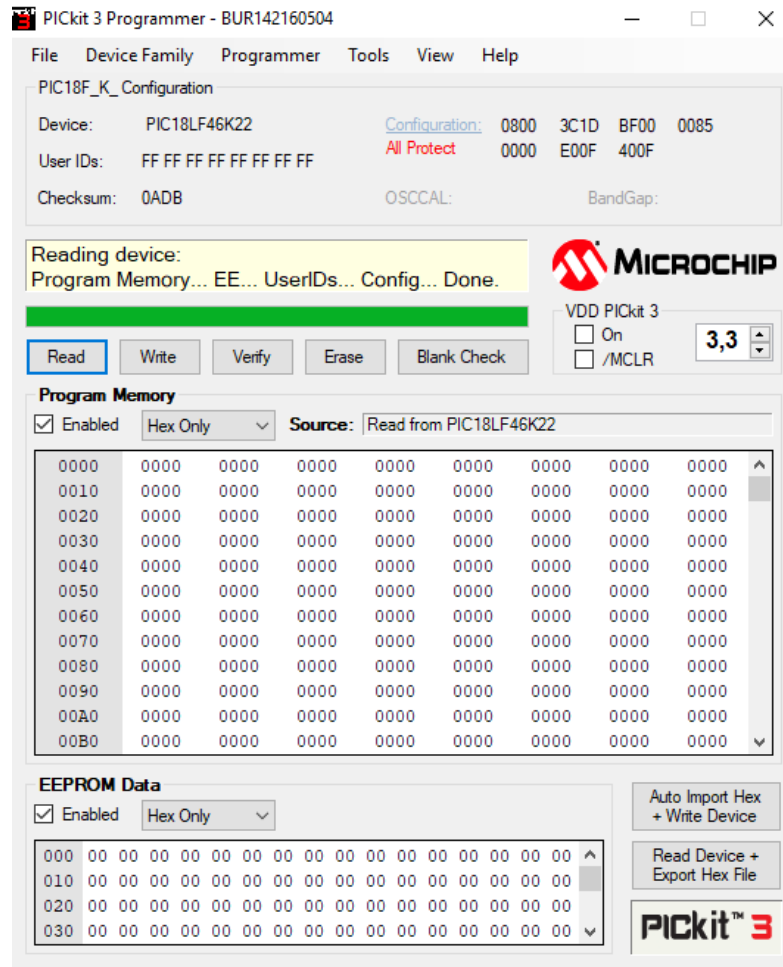


Figura 51 Ambiente de trabalho do PICKIT 3

3.4.3. INTERLIGAÇÃO À REDE LORAWAN

Feita a reposição do *firmware* do equipamento procedeu-se à interligação deste com a ferramenta Termite e foram enviados os comandos ASCII (*American Standard Code for Information Interchange*) necessários para a transmissão dos dados para o servidor.

Conforme o teste anterior, foi necessário proceder à criação do dispositivo no servidor TTN, no modo ABP, e utilizar as chaves criadas para a configuração deste, conforme apresenta a Figura 53.

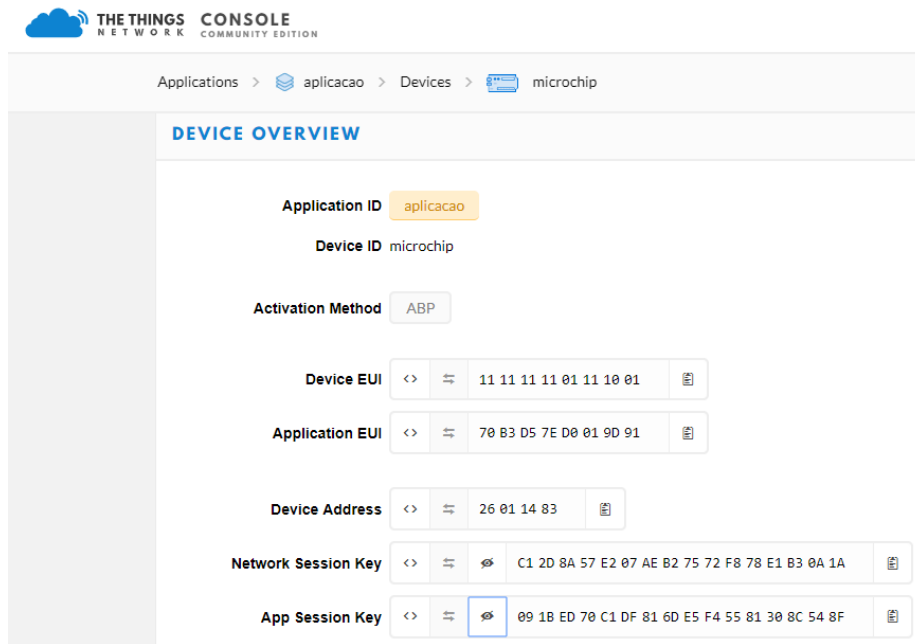


Figura 52 Dispositivo criado no servidor TTN

A ideia principal desta parte do trabalho era interligar este equipamento à rede criada anteriormente, com os outros dispositivos de fim de linha, e o *gateway* da Dragino, a fazer o encaminhamento dos pacotes, conforme apresenta a Figura 54.



Figura 53 Arquitetura da rede

Neste teste inicial, os resultados foram bastante satisfatórios tendo-se verificado o correto funcionamento do equipamento, bem como o correto envio dos dados e a sua receção por parte do servidor TTN.

Terminado este teste, passou-se à configuração do módulo através da ferramenta de desenvolvimento disponibilizada Lora Suite. Esta ferramenta permite a interligação ao

módulo através do cabo USB e também apresenta uma interface gráfica (Figura 55) que permite ao utilizar o envio dos comandos para o módulo e a sua configuração.

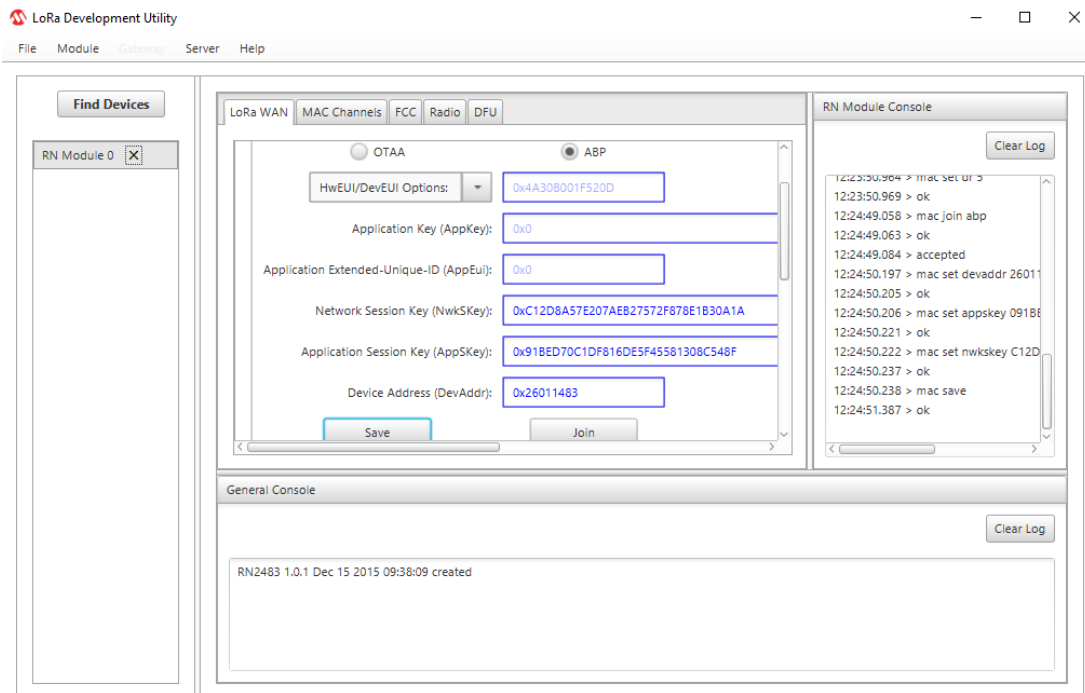
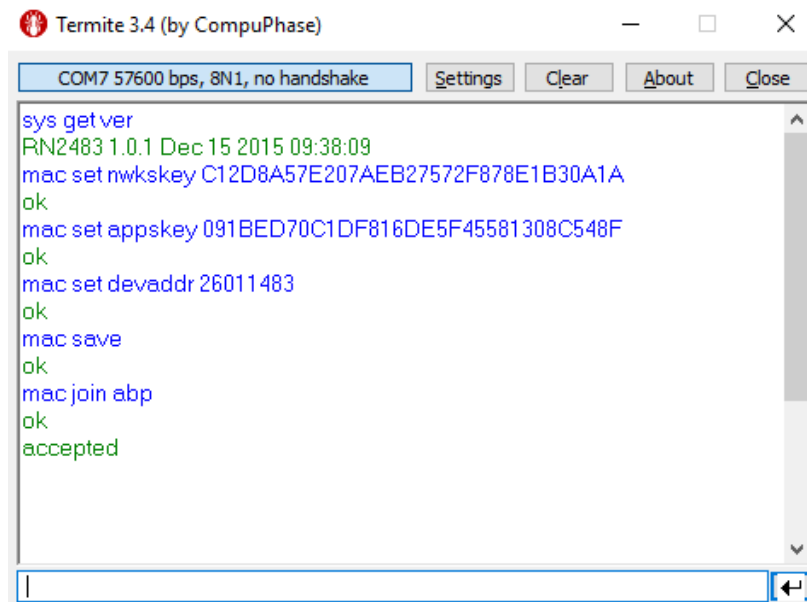


Figura 54 Interface gráfica Lora Suite

Nos testes efetuados com esta ferramenta, é necessário configurar todos os dados relativamente à ligação MAC, ou seja, escolha da frequência, SF, *Coding Rate*, *bandwidth*, e também o posterior envio das mensagens.

O sistema funcionou de acordo com o esperado e foram recebidos todos os dados enviados, por parte do servidor, inclusive recebidas mensagens de confirmação de algumas tramas. Assim, foi possível concluir que o dispositivo estava a operar conforme desejado.

Por outro lado, testou-se também a comunicação série, utilizando as ligações idênticas à Figura 51 e foram enviados comandos de forma a configurar a ligação e de seguida proceder ao envio das mensagens. A Figura 56 apresenta a sequência de comandos enviada, com as configurações do protocolo MAC.



```
Termitte 3.4 (by CompuPhase)
COM7 57600 bps, 8N1, no handshake  Settings  Clear  About  Close
sys get ver
RN2483 1.0.1 Dec 15 2015 09:38:09
mac set nwkskey C12D8A57E207AEB27572F878E1B30A1A
ok
mac set appskey 091BED70C1DF816DE5F45581308C548F
ok
mac set devaddr 26011483
ok
mac save
ok
mac join abp
ok
accepted
```

Figura 55 Configuração da camada MAC

Os comandos (mac), são direcionados para o protocolo LoRaWAN Classe A, sendo que o comando (set), acede e modifica os parâmetros deste, configurando assim a *Network Session Key*, *Application Session Key* e *Device Address*, necessários para estabelecer a ligação [47].

De seguida, foi necessário configurar os parâmetros da camada física (Figura 57), frequência, SF, *coding rate* e *bandwidth*, utilizando o comando (radio) e os restantes parâmetros.

```
radio set freq 869400000
ok
radio set sf sf7
ok
radio set cr 4/5
ok
radio set bw 125
ok
```

Figura 56 Configuração da camada física

Por fim, uma vez realizadas todas as configurações, procedeu-se ao envio de tramas de teste. Foi utilizado o método de envio sem confirmação, através do Porto 1. A Figura 58 apresenta o exemplo do envio da mensagem “12345678” com as características referidas.

```
mac tx uncnf 1 12345678  
ok  
mac_tx_ok
```

Figura 57 Transmissão sem confirmação

Com a realização desta fase do projeto chegou-se à conclusão que esta era também uma opção viável em alternativa aos nós apresentados anteriormente, da Dragino. O equipamento da Microchip, apresenta especificações semelhantes a este todavia, no que diz respeito a uma solução LoRaWAN, apresenta vantagens pois encontra-se implementado com o protocolo e é uma solução pronta a funcionar, que inclui dois tipos de sensores.

Por outro lado, no segundo teste, verificou-se também que este equipamento permite o envio de uma trama de dados maior que a anterior, ponto este que o destaca, uma vez que neste a biblioteca utilizada no equipamento da Dragino, não permite o envio de um campo de dados superior a 8 *bytes*.

3.4.4. LEITURA DE SENSORES

Por último, para tornar o módulo completamente funcional, procurou-se alterar o código do microcontrolador que controla a placa (PIC18LF45K50), para que este fosse capaz de controlar mais sensores, através dos pinos GPIO, e enviar estes dados de forma autónoma e cíclica para a rede. Como tal o objetivo nesta fase é o envio periódico das leituras de sensores, externos ao módulo, para a rede.

No que diz respeito ao programa a criar, a solução disponível era a alteração do código fonte do equipamento, fornecido pela Microchip. Optou-se por manter o código original do MOTE, e adicionar o modo de funcionamento, com o ciclo desejado.

Assim sendo, foi necessário realizar um complexo trabalho de análise a todos os ficheiros de forma a compreender, a fundo, o seu funcionamento, para depois, numa segunda parte, utilizar estes para as funcionalidades desejadas. O fluxograma da Figura 59 mostra o funcionamento do MOTE.

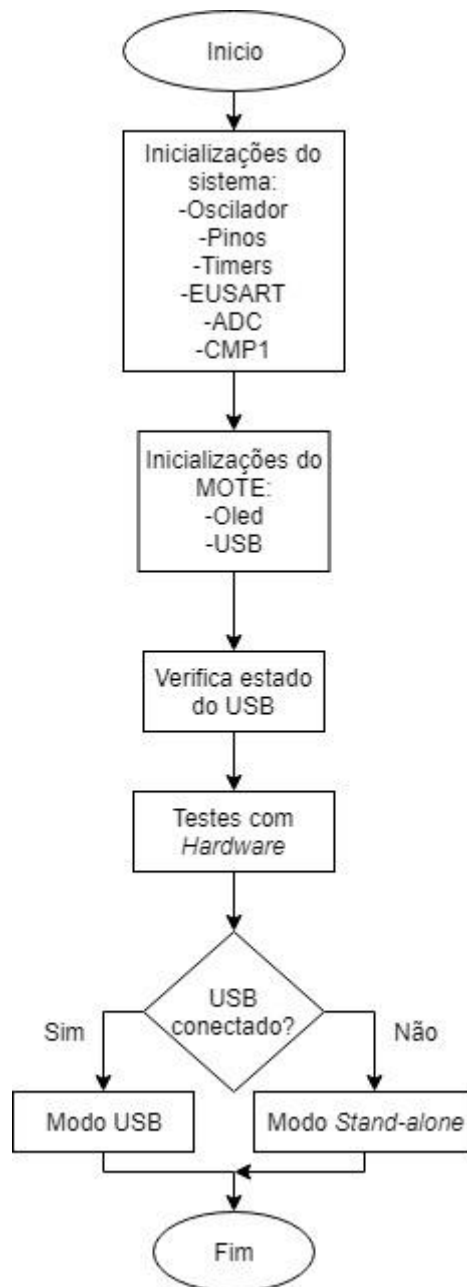


Figura 58 Fluxograma do funcionamento do MOTE

Através desta análise foi possível verificar que o funcionamento do programa era sob a forma de processos. Ou seja, o módulo operava sob a forma de estados, transitando entre estes, num ciclo infinito. Este facto dificultou a tarefa de compreensão, uma vez que é necessário identificar com clareza as dependências das rotinas, os diversos ciclos e as suas sequências.

Na rotina principal, eram executadas as inicializações, seguindo-se um ciclo no qual o programa verificava a conexão do cabo USB. Neste caso, o processo era atendido pela rotina do USB, no caso contrário, é atendida pelo mote.

Estes dois processos iniciais eram depois responsáveis por todo o funcionamento do mote, incluindo o funcionamento do LCD (*Liquid Crystal Display*), dos interruptores de escolha dos modos e ainda do funcionamento do módulo com comandos ASCII.

Através da análise do código fonte do módulo foi possível verificar que não era possível compilar este, uma vez que apresentava bastantes erros. Esta foi uma situação delicada, uma vez que este era fornecido pela Microchip. Assim, através de uma análise a este foi possível eliminar algumas falhas menores, porém não foi possível eliminar na totalidade estas e pôr o código a funcionar. Como tal, contactou-se a secção de *support* da Microchip, de forma a obter ajuda para solucionar este problema.

Durante o período de resolução do problema mencionado anteriormente, seguiram-se também avanços num outro objetivo proposto para concluir esta fase que era a utilização dos pinos disponíveis do equipamento, GPIO (*General Purpose Input/Output*) afetos ao RN2483, para fazer a leitura de outros sensores e o posterior envio destes para a rede.

Estes pinos podem ser configurados tanto como entradas ou saídas digitais ou como entradas analógicas. Estes comandos inserem-se nos comandos de sistema. E tanto podem ser do tipo de configuração como de verificação. A Figura 60 apresenta a estrutura destes comandos.

```
sys set pinmode <pinname> <pinmode>
```

Figura 59 Comandos GPIO [46]

O primeiro parâmetro faz referência ao pino a utilizar. Já o segundo indica o modo, quer este seja digital ou analógico [46].

Nesta fase do trabalho, foi utilizado o pino GPIO9, para interligar um sensor de luminosidade e o pino GPIO8, para ativar um LED (*Light Emitting Diode*). Como tal, este último foi programado como saída digital e o primeiro como entrada analógica. Na Figura 61 os dois primeiros comandos fazem referência às configurações de ambos os pinos,

sendo que o terceiro altera o estado do pino 8 para alto, ligando o LED, e o último comando retorna o valor lido pelo sensor de luminosidade.

```
sys set pinmode GPIO9 ana
ok
sys set pinmode GPIO8 digout
ok
sys set pindig GPIO8 1
ok
sys get pinana GPIO9
162
```

Figura 60 Comandos para controlar o sensor e LED

Os resultados obtidos nos testes, demonstram o correto funcionamento do módulo nesta fase pelo que deu-se por concluída a parte de aquisição de dados de sensores neste equipamento.

Quanto à programação do equipamento, em contacto com a Microchip, estes apesar de não fornecerem uma solução para o problema apresentado, forneceram, em alternativa, o código fonte do módulo do PIC18LF46K22.

Desta forma, a primeira opção de programação do equipamento foi descartada, uma vez que este era um problema do código fonte, fornecido pela Microchip. Problema este para o qual a empresa não arranhou solução.

Assim sendo, foi iniciado o trabalho de programação do equipamento, através da programação do módulo. Apesar desta opção dificultar a utilização de certos recursos da placa, como a porta USB, o LCD e os interruptores, ainda permite atingir o objetivo principal que é o envio das mensagens e leitura de sensores.

O primeiro passo foi a verificação do correto funcionamento deste e a sua compreensão. O programa foi obtido através da utilização da ferramenta do MPLAB X Code Configurator. Este chama a rotina de inicializações, configura o protocolo MAC e entra em ciclo, enviando a *string* “lora” para o servidor.

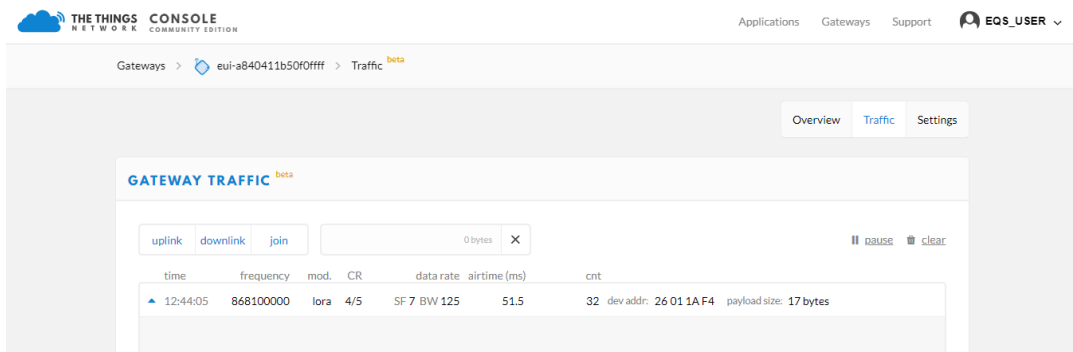


Figura 61 Dados recebidos pelo gateway

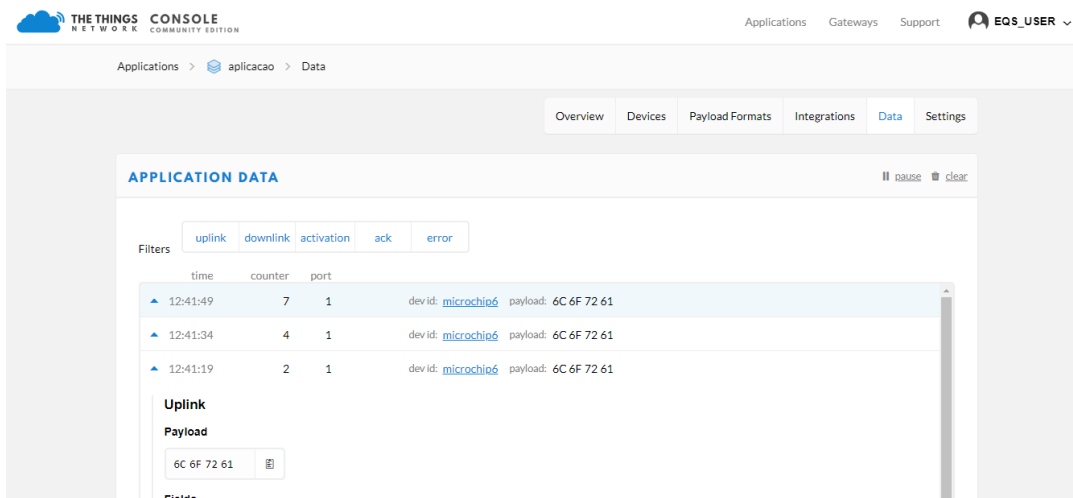


Figura 62 Dados recebidos pela aplicação

Conforme apresentam as Figuras 62 e 63, quer do lado do servidor quer do lado da aplicação as transmissões foram todas recebidas, os dados foram decodificados e o sistema enviava estes de forma cíclica, não apresentando qualquer tipo de *delay*.

Uma vez verificado o funcionamento deste seguiu-se a identificação das conexões internas entre o PIC e o *Transceiver SX1276* (Figura 64). Esta análise permitiu entender se existiam alguns outros recursos da placa disponíveis através desta, bem como verificar quais os pinos disponíveis para posterior utilização para entrada de sensores.

Conforme mencionado anteriormente o objetivo era manter o funcionamento do equipamento, todavia, adicionar uma opção de funcionamento nova, que permite o envio de leituras de sensores de forma periódica. O fluxograma seguinte (Figura 65) apresenta o funcionamento do *software* inserido no MOTE.

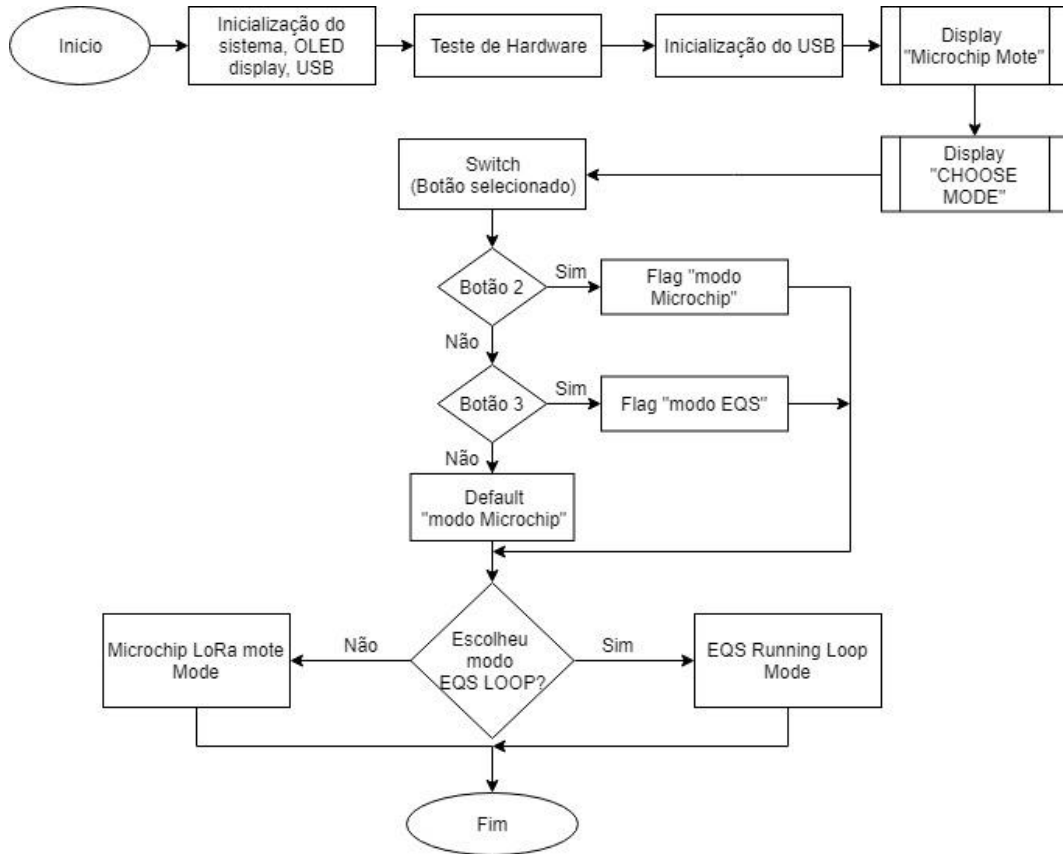


Figura 64 Funcionamento do novo software do Mote

O programa procede à inicialização dos vários componentes do MOTE, do protocolo LoRaWAN e do USB, testa o *hardware* e consulta o utilizador para este escolher o modo. No que diz respeito ao “EQS Running Loop Mode” este chama a rotina de envio. Nesta, é feito o ciclo de envio de dados para o servidor, que começa com a verificação do estado do módulo, seguida pela configuração do protocolo, e envio dos dados. Nesta fase, os conhecimentos adquiridos no sub-subcapítulo 3.4 foram fulcrais na medida em que agilizaram todo o processo. O fluxograma da Figura 66 descreve o funcionamento das rotinas principal criado no *software* do Mote.

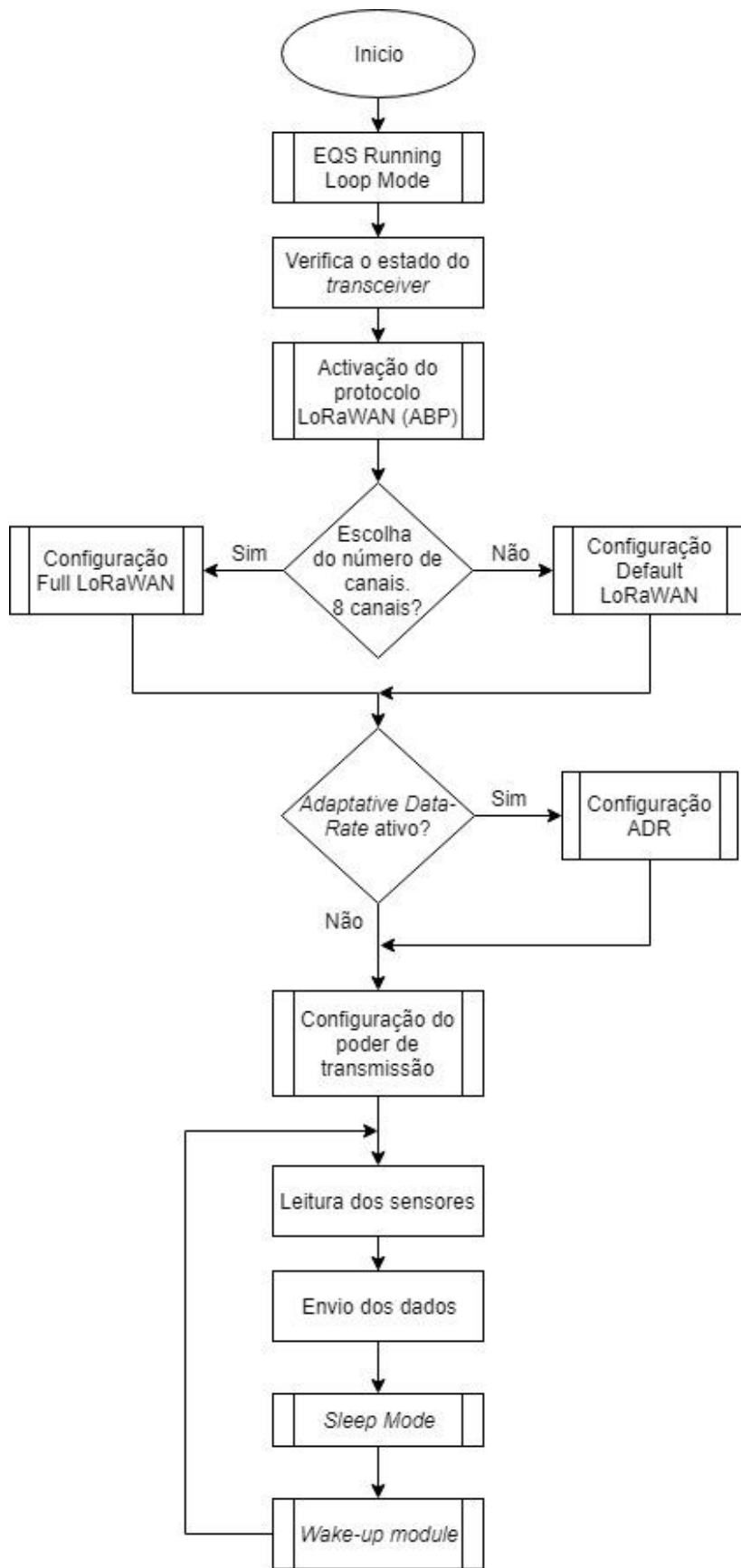


Figura 65 Fluxograma da rotina principal

3.4.5. CONSIDERAÇÕES FINAIS

Concluído este passo, dá-se por terminada a fase de programação do LoRa MOTE da Microchip. O código desenvolvido permite ao equipamento configurar os vários pinos disponíveis para a interligação de sensores e enviar estes de forma cíclica, no intervalo de tempo desejado.

Foram tidos em consideração os gastos energéticos do módulo sendo que foram utilizadas rotinas que reduzem estes consumos sempre que possível. O código criado também permite a utilização do programa disponibilizado pela Microchip como *kit* de desenvolvimento.

De forma adicional acrescentou-se a flexibilidade necessária ao código para que este além de permitir a utilização das funcionalidades do MOTE, permitisse a configuração de características do protocolo e envio dos dados, através dos periféricos disponíveis.

3.5. GATEWAY

Nos primeiros trabalhos de pesquisa efetuados foi possível compreender, desde cedo, que o *kit* de desenvolvimento da Dragino, que era uma possível solução para o projeto, poderia não ser a mais adequada para o trabalho a desenvolver.

Todavia, era necessário testar este e verificar se realmente isto se verificava. Ao longo do tempo, todas as questões relativas aos nós, foram sendo esclarecidas, tendo chegado à conclusão que estes eram adequados à funcionalidade pretendida, quer da Dragino, quer da Microchip. Porém, o mesmo não aconteceu com o *gateway* (Figura 67).



Figura 66 Gateway Dragino

Com a criação da primeira rede de equipamentos, na qual o *gateway* foi utilizado como gestor da rede LoRa *Raw*, não houve qualquer tipo de problema. Todavia, foi necessário ajustar o tempo de reconhecimento de dispositivos na rede, de forma a contornar o facto de só ser possível interligar equipamentos numa frequência. Assim, foi possível interligar até 2 dispositivos nesta rede, sendo possível, mediante posterior ajuste das tentativas de reconhecimento, adicionar mais.

Tomada a decisão de utilizar o protocolo LoRaWAN, foi necessário criar uma nova rede, utilizando os mesmos equipamentos, porém, alterando a sua programação e configuração. O *gateway* neste caso, foi interligado com servidor TTN, mediante a configuração adequada e os nós foram programados com o protocolo LoRaWAN, para enviar os dados de sensores e GPS.

Neste ponto as fragilidades do *gateway* foram expostas de forma mais acentuada. Nas especificações este não apresenta o suporte total do protocolo, apenas apresentando comunicações num sentido, *uplink*, e num único parâmetro, configurado previamente. Por outro lado, com apenas um canal, não era possível apresentar uma solução com a escalabilidade pretendida, sendo que com a utilização de apenas 2 equipamentos de fim de linha a latência apresentada não era adequada à aplicação.

Por outro lado, em consulta ao fórum do servidor TTN, chegou-se à conclusão que este tipo de *gateways* não era uma solução com a qual estes pretendiam trabalhar, nem era oficialmente suportada. Uma das razões é que estes equipamentos poluem o espectro, apenas enviando comunicações em uma frequência. De forma adicional, devido ao facto de não permitirem *downlink*, apresentam problemas para o servidor, relacionados com mensagens de confirmação, enviadas pelo servidor, que nunca chegam aos nós.

Assim, tendo em base o referido anteriormente, apenas é permitida a utilização destes *gateways* durante algum tempo, de forma a facilitar a adaptação e conhecimento por parte do utilizador, da ferramenta, sem a necessidade de investimento num *gateway full* LoRaWAN. *Gateway* este que comporta custos elevados e em alguns casos necessidade de programação e conhecimentos da tecnologia profundos.

Como tal, passados cerca de 2 meses e meio este deixou de aparecer conectado ao servidor, não sendo possível o envio dos dados para o servidor, conforme acontecia anteriormente.

Este foi um ponto fulcral na decisão de investimento num novo equipamento, uma vez que era necessário avançar com o projeto [64][65].

A solução seguinte passava pela escolha ou montagem de um novo *gateway*. Neste ponto existiam algumas soluções, pelo que foi necessário definir com clareza a opção a tomar.

De entre várias soluções distinguiram-se duas, o *gateway* constituído por uma placa RAK381 e um Raspberry Pi 3, ou o *gateway* da Microchip disponível no *kit* de desenvolvimento. Estes são duas opções bastante distintas uma vez que no primeiro caso, o *hardware* é *open-source* e necessita de programação de *firmware*. Por outro lado, a segunda opção é um *gateway* pronta a utilizar que faz parte do *kit* de desenvolvimento.

Ambas as escolhas estavam em cima da mesa, uma vez que existiam boas referências de ambos os equipamentos e estavam dentro do orçamento proposto para este equipamento.

Em conversações com a empresa, foi tomada a decisão de seguir com o *gateway* da Microchip (Figura 68). Em termos técnicos, ambos apresentam como módulo principal o *concentrator* LoRa SX1301, que é um módulo de 8 canais que permite a escuta de vários dispositivos ao mesmo tempo, com diferentes SF e larguras de banda. Para além disto têm também dois *chips* SX1257, que permitem o envio de mensagens *downlink*, em simultâneo com o concentrador.



Figura 67 Gateway Microchip

Em termos de fiabilidade e confiança, o equipamento da Microchip apresenta melhores indicações, sendo produzido por uma marca de confiança e referência no mercado.

Em suma, para além do preço, o *kit* é uma solução completa, sendo que em comparação com outras ofertas no mercado é a que mais se enquadra aos objetivos do projeto e da empresa.

Para a interligação do *gateway* com o servidor TTN foi utilizada a ferramenta LoRa suite da Microchip. Através da interface gráfica, foram configuradas as definições do equipamento e o *setup* do servidor.

Uma vez que este não se conectou ao servidor após as configurações, foi necessário recorrer à funcionalidade de *polling*, que faz um *ping* para o servidor, para forçar este a conectar-se. A Figura 69 apresenta o estado do *gateway* após conexão.

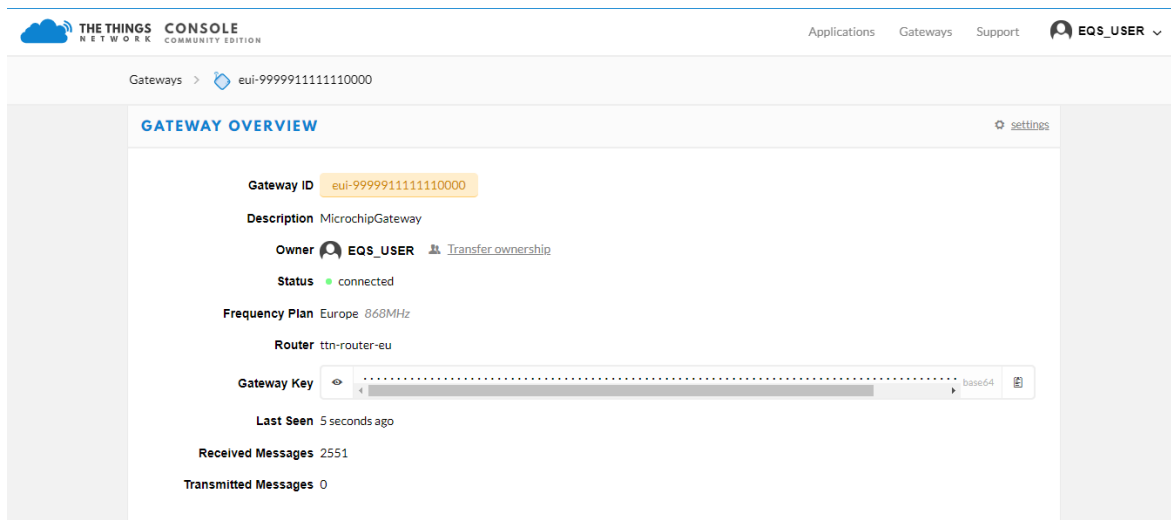


Figura 68 Estado do gateway no servidor

3.6. CONCLUSÃO

Realizado todo o trabalho com os equipamentos disponibilizados, *kit* de desenvolvimento da Dragino e *kit* de desenvolvimento da Microchip foi possível tirar ilações quanto à melhor solução a utilizar na arquitetura do projeto.

Para a definição da arquitetura, pendiam-se questões relativamente aos nós, compostos por um *transceiver* e microcontrolador, *gateway* e servidor a utilizar, para além da definição do *kit* a utilizar.

No que diz respeito aos nós, nos testes efetuados ficou bem vincada a superioridade do equipamento da Microchip, face ao da Dragino. Ambos fazem o envio de dados de sensores para o servidor, todavia, o LoRa MOTE, apresenta um *transceiver* que já inclui o protocolo LoRaWAN, dispõe de maior número de pinos programáveis, e o *kit* inclui uma antena melhor. Para além destas vantagens é uma solução pronta a usar num ambiente industrial, sendo que este foi tido como a solução a implementar.

Quanto ao *gateway* a utilizar, a possibilidade de utilização do dispositivo da Dragino foi fragilizada assim que foi tomada a decisão de seguir com o protocolo LoRaWAN. Este é um protocolo que requer a escuta de uma vasta gama de frequências, com diferentes fatores, em simultâneo, característica esta que não era suportada por este equipamento devido às suas limitações de *hardware*.

Assim, foi tomada a opção de comprar o *gateway* da Microchip, apenas disponível através do seu *kit* de desenvolvimento. Esta foi uma boa escolha, tendo em conta o preço de outros *gateways* no mercado, o seu *hardware* e ainda a inclusão de dois nós no *kit*.

No que diz respeito ao servidor optou-se por utilizar o servidor TTN. Tendo em conta que o objetivo do trabalho é a comunicação LoRaWAN, esta ferramenta, que permite a visualização dos dados e até fornece outras opções de tratamento de parâmetros, algumas pagas, adequa-se na perfeição a este projeto. Todavia, algumas funcionalidades de maior interesse não são fornecidas, como por exemplo o armazenamento dos dados. Como tal, em futuros desenvolvimentos, adequa-se a utilização de um servidor outro tipo de servidor, como por exemplo o Loraserver.io ou o TTI (*The Things Industry*).

No primeiro caso temos uma solução *open source*, que pode ser interessante no caso de se pretender alterar partes do *software*. Por outro lado, o TTI oferece uma solução similar ao TTN, mas orientada para a indústria.

4. ARQUITETURA

Com base nas informações recolhidas, quer através da realização do trabalho de pesquisa do Estado da Arte, quer através dos testes no Capítulo do Protótipo foi possível, primeiramente, compreender as características de interesse a integrar no projeto, bem como escolher os equipamentos que melhor forneciam estas características.

Assim, foi definida uma solução que abrange as especificações procuradas, tendo em conta os equipamentos disponibilizados. Este capítulo tem como objetivo a interligação entre a visão inicial, a conceção dos vários protótipos de teste e a implementação final deste projeto.

4.1. INTRODUÇÃO

Para a criação de uma rede de comunicações, sem fios, LoRa, para monitorizar sensores na indústria petroquímica a solução deve incluir os elementos de deteção de parâmetros, sensores, equipamentos computacionais, responsáveis pelas leituras e interpretação de dados, e posterior envio, os *gateways*, responsáveis pela gestão da rede e encaminhamento dos dados, e por fim, o servidor acompanhado da base de dados, na qual os valores vão ser armazenados para monitorização. A Figura 70 apresenta o conceito deste projeto.

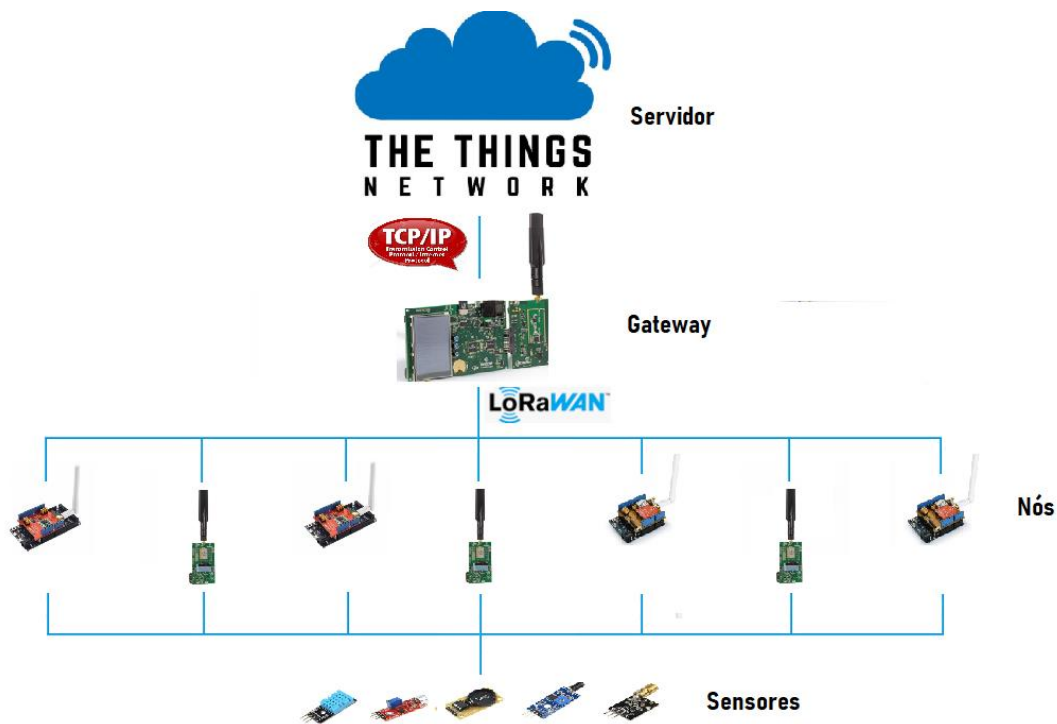


Figura 69 Arquitetura do sistema

Neste caso, a ideia é a interligação de diversos nós na rede, todavia, no âmbito prático deste projeto apenas vão ser interligados os nós disponíveis. Contudo, o funcionamento da rede deve permitir a interligação de vários dispositivos.

Seguindo a arquitetura definida, a rede a criar segue a topologia em estrela, suportada pelo protocolo LoRaWAN. Este é responsável pela camada MAC, gerindo as comunicações entre os equipamentos de fim de linha e o *gateway*. O protocolo está suportado na camada física, ou seja, na modulação de rádio frequência LoRa.

As leituras dos sensores são feitas pelos microcontroladores nos nós e enviadas para o *gateway*. De seguida este faz o encaminhamento destas, através do protocolo TCP/IP para o servidor, sendo depois feita a descriptação dos pacotes e recebidos os dados.

Outro ponto vantajoso deste sistema é o facto de permitir a inclusão de mais *gateways* ou nós no sistema, sendo que o protocolo é flexível neste aspeto. Por outro lado, no caso de termos pacotes duplicados, ou outro tipo de problemas associados à captação do mesmo pacote por diferentes *gateways*, estes são resolvidos sempre pelo servidor.

4.2. DESENVOLVIMENTO

Com as informações adquiridas nos capítulos anteriores, destacando-se o Capítulo Protótipo, na qual foram realizados testes com diferentes sistemas, foi possível chegar à definição dos equipamentos que se adequavam à arquitetura final do projeto.

Deste modo ficaram definidos os equipamentos de fim de linha, *gateway* e servidor a utilizar. Por outro lado, também foi tomada a decisão de utilizar as antenas incluídas nos *kits* uma vez que era necessário primeiramente, provar o conceito, para depois se proceder a um investimento na tecnologia.

4.2.1. GATEWAY

Começando pelo equipamento central da rede, temos o *gateway* da Microchip, composto por duas placas, a placa rádio, onde se encontram os *transceivers*, o *concentrator* LoRa e a placa central, responsável por receber os pacotes da placa radio e encaminhar estes, através do protocolo TCP/IP, para o servidor. Este foi adquirido através do *kit* de avaliação da Microchip.

Em termos de *hardware* este é composto por dois módulos. Do lado da placa rádio (Figura 71) temos à disposição dois *transceivers* SX1257, que funcionam como *Front-End*, encaminhando as mensagens para o *concentrator* SX1301. Este é o componente principal da placa de rádio, uma vez que dispõe de 8 canais de alto desempenho, que permitem receber e transmitir pacotes em simultâneo com diferentes SF. Só a inclusão de um *chip* destes permite o total suporte do protocolo LoRaWAN, que se estende a estas 8 frequências.

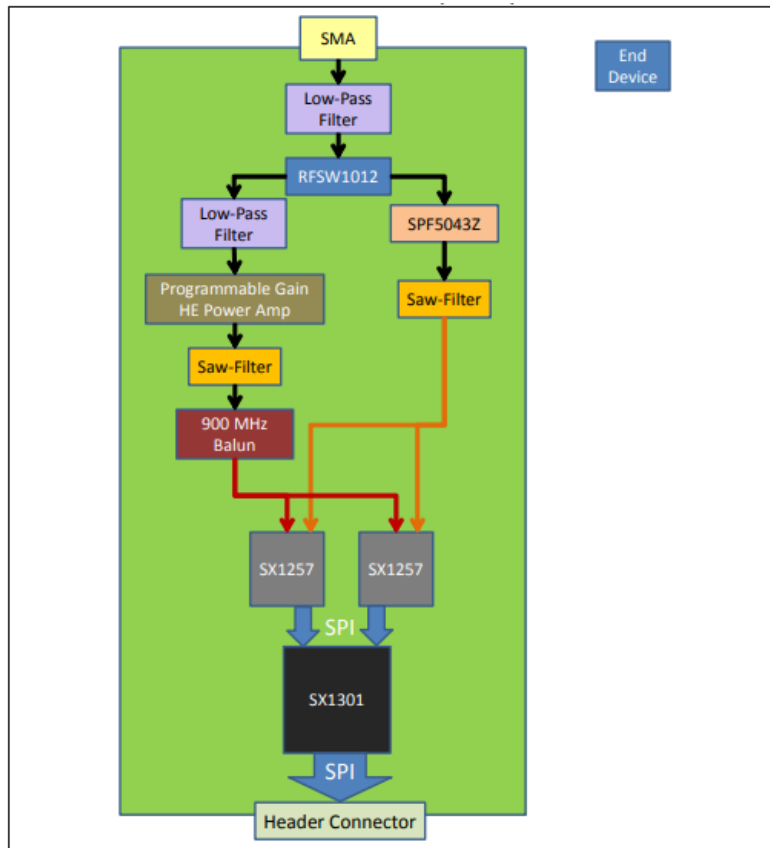


Figura 70 Diagrama de funcionamento da placa rádio [68]

Ao contrário de outras soluções, o *concentrator* permite a recepção em simultâneo de diferentes frequências em diferentes SF, sem que este tenha conhecimento destas configurações utilizados nos nós.

Os *transceivers* neste caso para além de desempenharem um papel de *Front-End*, também aumentam a sensibilidade do *concentrator*, melhorando a *performance* global do equipamento [68].

Por outro lado, a placa de central do *gateway* (Figura 72) é responsável por encaminhar os pacotes recebidos pela placa de rádio, e enviar estes para a Internet. O microcontrolador presente na placa, PIC24EP512GU810, controla a comunicação USB-Série, os periféricos da placa, e a comunicação I2C, entre as duas placas.

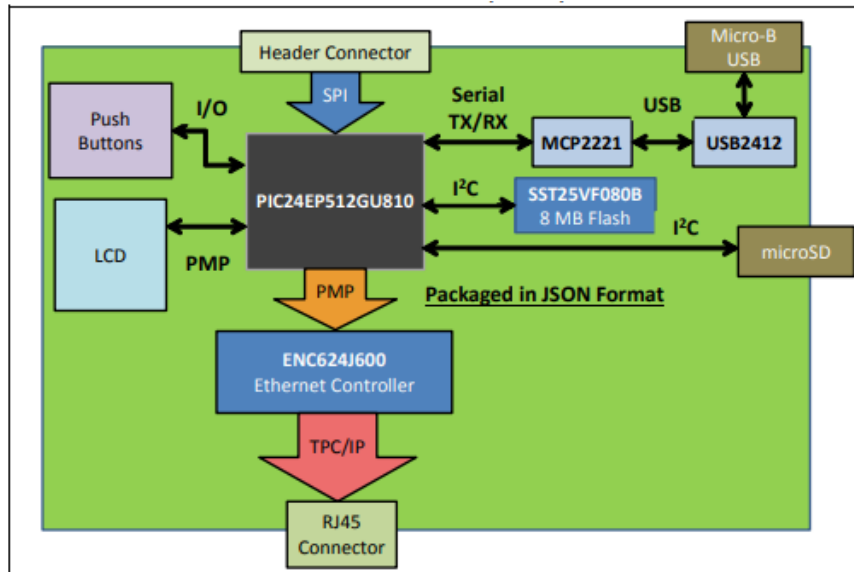


Figura 71 Diagrama de funcionamento da placa central do gateway

4.2.2. Nós

No que diz respeito aos equipamentos de fim de linha, no capítulo anterior foram testados dois dispositivos de duas marcas diferentes, sendo que as diferenças entre ambos não se prendiam tanto com o *transceiver* que utilizavam, mas sim com outros componentes da sua arquitetura como o microcontrolador e a antena.

No que diz respeito ao equipamento da Dragino, este apresenta o microcontrolador Atmega 328p, e uma antena de baixo custo. Estes componentes comparados aos utilizados pelo equipamento da Microchip, que possui dois processadores da gama PIC18LF, um dedicado exclusivamente à ligação rádio e ao protocolo LoRaWAN, e outro responsável pelo processamento dos dados e das ferramentas do Mote, oferecem uma maior capacidade de processamento de informação que o nó da Dragino, e uma antena bastante mais robusta, representando uma melhor opção.

Apesar das diferenças entre os dois equipamentos ambos podem ser incluídos no projeto uma vez que dentro da rede vamos ter dispositivos a monitorizar parâmetros pouco complexos e com baixas taxas de transmissão, não sendo necessário um alto poder de processamento, como são exemplo os parâmetros de temperatura ou o nível de líquidos, e também, colocados a curtas distâncias do *gateway*.

Assim sendo, podemos distribuir e organizar os equipamentos na rede conforme as suas características e o objetivo a concretizar.

No que diz respeito ao equipamento da Dragino, todo o trabalho desenvolvido no capítulo do protótipo foi fundamental para a inclusão deste na arquitetura do projeto. Conforme abordado anteriormente este é utilizado para fazer a monitorização de parâmetros com baixa taxa de transmissão e relativamente perto do *gateway*.

Já no que diz respeito aos equipamentos da Microchip estes tanto fazem a monitorização de parâmetros mais complexos como vibrações nas tubagens ou medições energéticas, ou temperaturas, sendo que a distância a que estes pontos de medida se encontram do *gateway* vai definir se é necessário o uso destes equipamentos ou dos da Dragino.

- Dragino

Na rede estão incluídos dois nós da Dragino. Um faz a monitorização da temperatura, luminosidade e humidade. Por outro lado, o segundo nó inclui um módulo GPS, e envia estas coordenadas para o servidor.

Por outro lado, os parâmetros a monitorizar, são parâmetros de teste, todavia exemplificam a situação de alta latência na perfeição, uma vez que tirando o parâmetro da luminosidade, todos os outros parâmetros, incluindo a localização GPS que é um parâmetro que é suposto manter-se fixo, são indicadores com alguma latência, que não necessitam de uma alta taxa de transmissão nem processamento. O Anexo A apresenta o esquemático das ligações do equipamento.

No que diz respeito ao funcionamento destes, estes foram programados para interligar-se ao servidor através do método ABP. Assim que o sistema é iniciado, os nós procedem ao envio dos respetivos dados dos sensores, de forma cíclica, no intervalo de tempo desejado. Entre o tempo de envio estes permanecem num estado de redução dos consumos energéticos.

Conforme é apresentado pelo protocolo LoRaWAN, os nós para enviarem os dados começam pela primeira frequência da gama programada. De seguida é verificada, de forma autónoma, a disponibilidade da frequência em termos de *duty-cycle*. Para manter o nó dentro das limitações impostas, no caso de ter o canal próximo da limitação, que é 1%, é utilizada a frequência seguinte e assim sucessivamente.

Os valores de frequência a utilizar, em concordância com o servidor utilizado são 868.1, 868.3, 868.5, 867.1, 867.3, 867.5, 867.7, 867.9 MHz. Sendo estes os valores programados no equipamento. Assim, é possível afirmar que estes nós, estão programados para operar nos 8 canais disponibilizados pelo protocolo. O Fluxograma da Figura 73 apresenta o funcionamento do equipamento.

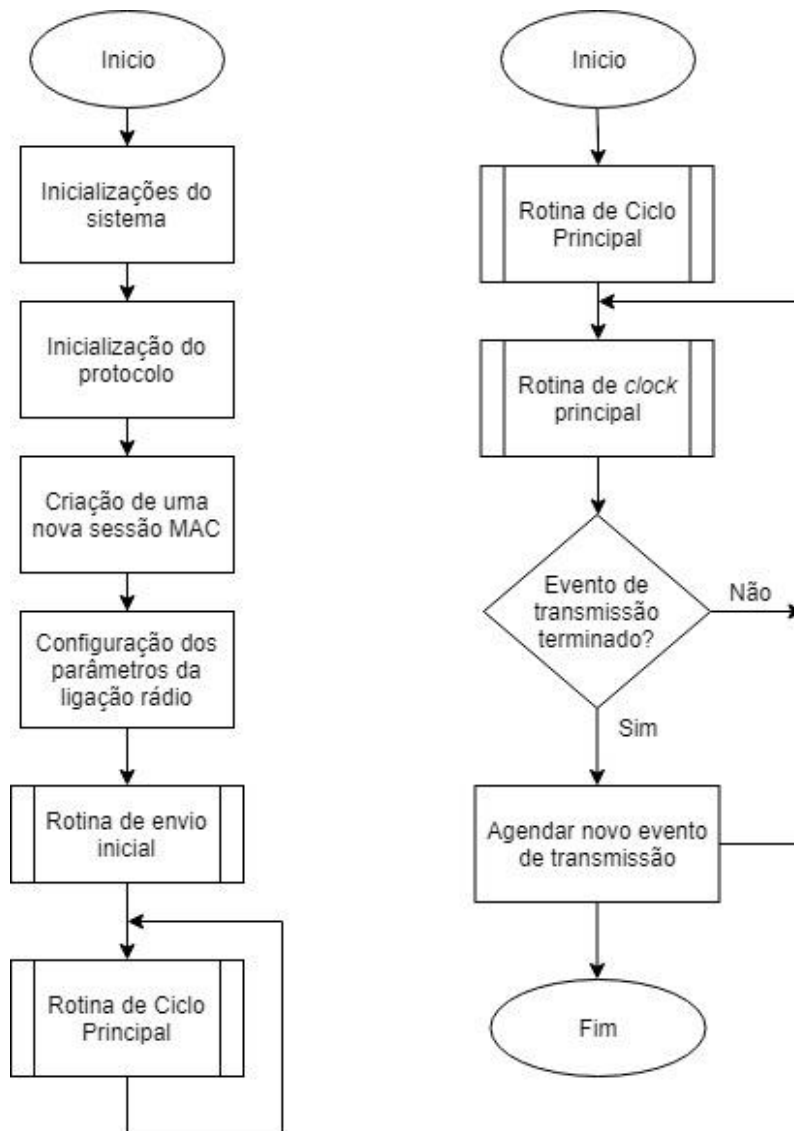


Figura 72 Fluxograma do funcionamento do nó Dragino

Uma vez que o *Transceiver* utilizado nestes nós apenas inclui as funcionalidades rádio do LoRa, a *stack* protocolar foi programada através de *software*, no microcontrolador. Este facto levou a que a tarefa de otimização da energia consumida pelo equipamento, através da inclusão de um modo de poupança de energia, entre o envio das mensagens, fosse dificultada.

- Microchip

Quanto aos nós da Microchip, estes foram programados para ligar à rede através do método ABP, como dispositivos de classe A, e enviam para o servidor os dados capturados nos sensores.

Após inicialização do sistema esta pergunta ao utilizador o modo a utilizar. Programados estão dois modos distintos. O modo “Microchip”, que é o modo programado no sistema de fábrica, e o modo “EQS”.

Feita a seleção deste último, o sistema procede às inicializações necessárias para o protocolo LoRaWAN. Configura as chaves de ligação, pergunta ao utilizador se pretende ligar a funcionalidade ADR, configura o poder de transmissão e entra no ciclo de envio, conforme apresenta o Fluxograma da Figura 74.

No ciclo de envio, este faz a leitura dos sensores, prepara a trama de dados a enviar, preenche-a com as leituras e faz o envio dos dados para o servidor. Todavia, foi também necessário incluir rotinas para configuração dos pinos de ligação dos sensores, uma vez que quando o equipamento entra no modo de *standby*, nos testes verificou-se que é necessário reprogramar estes novamente.

No *buffer* de envio dos dados os campos são enviados com o formato passível de decodificação do lado do servidor, onde foi criada uma rotina para este efeito. Sendo são enviados os dados do sensor de luminosidade seguidos pelos dados do sensor de temperatura.

Para a interligação destes, foi utilizado um divisor de tensão, com uma resistência com valor aproximado ao valor da resistência máxima dos sensores. Assim, deste modo, ambos os componentes apresentam a mesma influência, em percentagem, na queda de tensão.

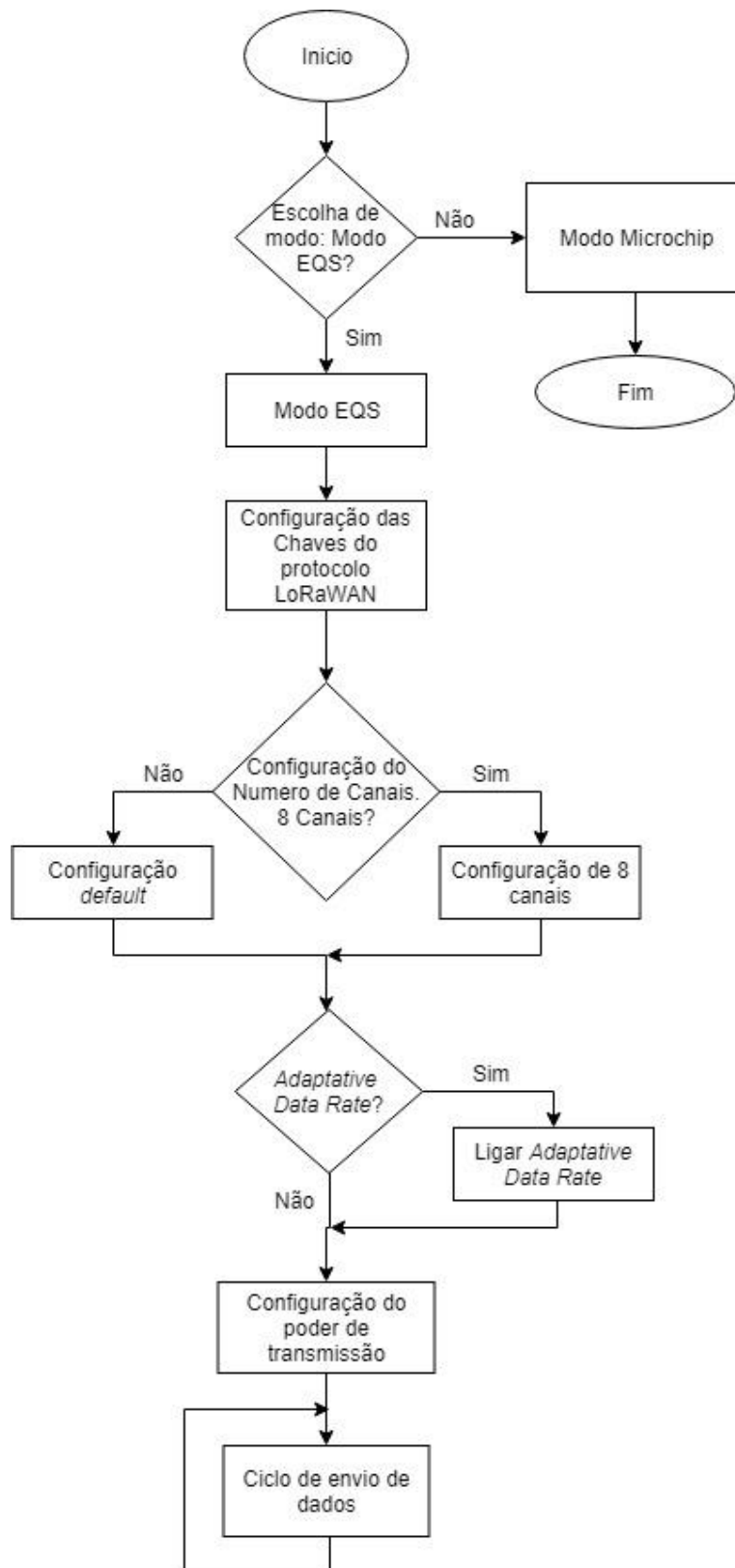


Figura 73 Fluxograma do nó da Microchip

5. SOFTWARE

Ao nível do *software* desenvolvido foi necessário programar os nós, para enviarem as informações dos sensores, sob o protocolo LoRaWAN, para o servidor. Foi também necessário configurar o *gateway*, para que este reencaminhasse os dados para o servidor escolhido. Também do lado do servidor procedeu-se à sua interligação com o *gateway* e os nós, e criação de *software* de decodificação dos dados recebidos.

5.1. DRAGINO

No que diz respeito aos nós, foi necessário programar o equipamento da Dragino e o da Microchip. No primeiro equipamento, uma vez que o *transceiver* utilizado neste não contém o protocolo LoRaWAN, foi necessário programar estes por *software*, através do auxílio da biblioteca LMIC.h (LoRaWAN-MAC-In-C). Esta é responsável pelo processamento protocolar aliado à gestão dos *timings* de acordo com o servidor. Por outro lado, foi também necessário adicionar as rotinas para a leitura dos sensores e posterior envio destas para a rede.

No que diz respeito às configurações, o sistema começa por fazer a inicialização do protocolo, através da rotina `os_init()` que é a rotina de eventos principal e inicialização dos

pinos. No ambiente de execução esta rotina, periodicamente, chama a rotina `os_runloop()`, responsável pelo agendamento das rotinas no ciclo principal.

De seguida é feito o *reset* do protocolo, de forma a fechar e eliminar os dados da sessão anterior, chamada a rotina `LMIC_setClockError()`, para relaxar as janelas protocolares é iniciada uma nova sessão com novas chaves.

No que diz respeito às configurações especiais, foi desabilitada a confirmação de pacotes, configurada a janela de receção 2 com SF 9, escolhido o SF 7 com um poder de transmissão 14.

De forma a agilizar a inicialização do protocolo é necessário fazer o agendamento de uma rotina inicial. De seguida o sistema entra em ciclo, na rotina `os_runloop()`, que faz o processamento dos eventos, e agenda futuros processos relacionados com o protocolo. Assim, os parâmetros retornados na rotina inicial, servem para acionar novos eventos, nesta rotina. A Figura 75 apresenta o funcionamento do *software* do nó.



Figura 74 Fluxograma de funcionamento do nó Dragino

Nesta, assim que uma mensagem é transmitida, é agendado um novo envio, no intervalo de tempo desejado, e assim sucessivamente.

A rotina de envio (Figura 76) faz uma verificação da disponibilidade de transmissão, para depois fazer a leitura dos dados, quer no caso do GPS quer no caso dos sensores, e procede ao agendamento do envio destes assim que haja disponibilidade. Ou seja, assim que os canais de transmissão estejam livres.



Figura 75 Rotina de envio de dados

Uma vez que um dos nós da Dragino inclui na sua placa um módulo GPS, este envia as suas coordenadas para o servidor. Este é um dado importante numa rede de longo alcance como a deste projeto, visto que a localização dos dispositivos pode ser um dado importante. Tanto mais que o sistema também pode ser utilizado num ambiente móvel. Por outro lado, o outro nó envia os dados dos sensores.

Assim sendo, no primeiro caso foram criadas as rotinas para a interpretação dos pacotes recebidos pelo GPS. Este comunica com o microcontrolador através da ligação série, com comandos NMEA, conforme foi apresentado no subcapítulo 3.3.5 deste relatório.

Com recurso à biblioteca TinyGPS, criada por Mikal Hart, foi feita a leitura da latitude e longitude do equipamento. De seguida os dados são validados e processados para depois serem recebidos pelo servidor. O fluxograma da Figura 76 representa o funcionamento destas rotinas

Quanto à leitura dos sensores, no caso do outro nó, foram incluídos dois sensores, o DHT11 e um LDR. No caso do DHT11 este é um sensor de temperatura e humidade, que comunica através de um protocolo série do tipo *single-wire two-way*. Este transmite uma trama de dados até um máximo de 40 bits, sendo que os bits mais elevados são transmitidos primeiro.

O formato dos dados é composto por 5 campos de 8 bits, na qual os dois primeiros são referentes à humidade relativa, os seguintes à temperatura, e o último é um campo de confirmação.

Após a leitura dos sensores, é chamada a rotina de processamento dos dados que os organiza para serem depois carregados para o *buffer*, sob um determinado formato reconhecido pelo servidor.

5.2. MIRCROCHIP MOTE

Já no que diz respeito ao LoRa Microchip Mote, este apresenta o *chip* RN2483, que é responsável pela manipulação rádio e também pelo protocolo. A comunicação com este é feita através do protocolo série, onde são enviados comandos para manipular o módulo rádio e o LoRaWAN.

O equipamento foi programado através do microcontrolador presente na placa, PIC18LF45K50. Este procede à inicialização de todos os periféricos, bem como do protocolo LoRaWAN. De seguida o utilizador escolhe o modo de funcionamento.

Os dois modos disponíveis são o modo “Microchip” que é o modo programado de origem no equipamento, e o modo “EQS”. No caso de seleção deste último, o equipamento procede à interligação à rede, através da configuração das chaves criadas no servidor com o método ABP. De seguida o equipamento questiona o utilizador sobre ao número de canais usar e se pretende utilizar a funcionalidade ADR. Por último, antes de iniciar a rotina de envio dos dados do sensor, o equipamento faz a configuração, com o auxílio do utilizador do poder de transmissão adequado ao local. O fluxograma da Figura 77 representa o funcionamento do código.

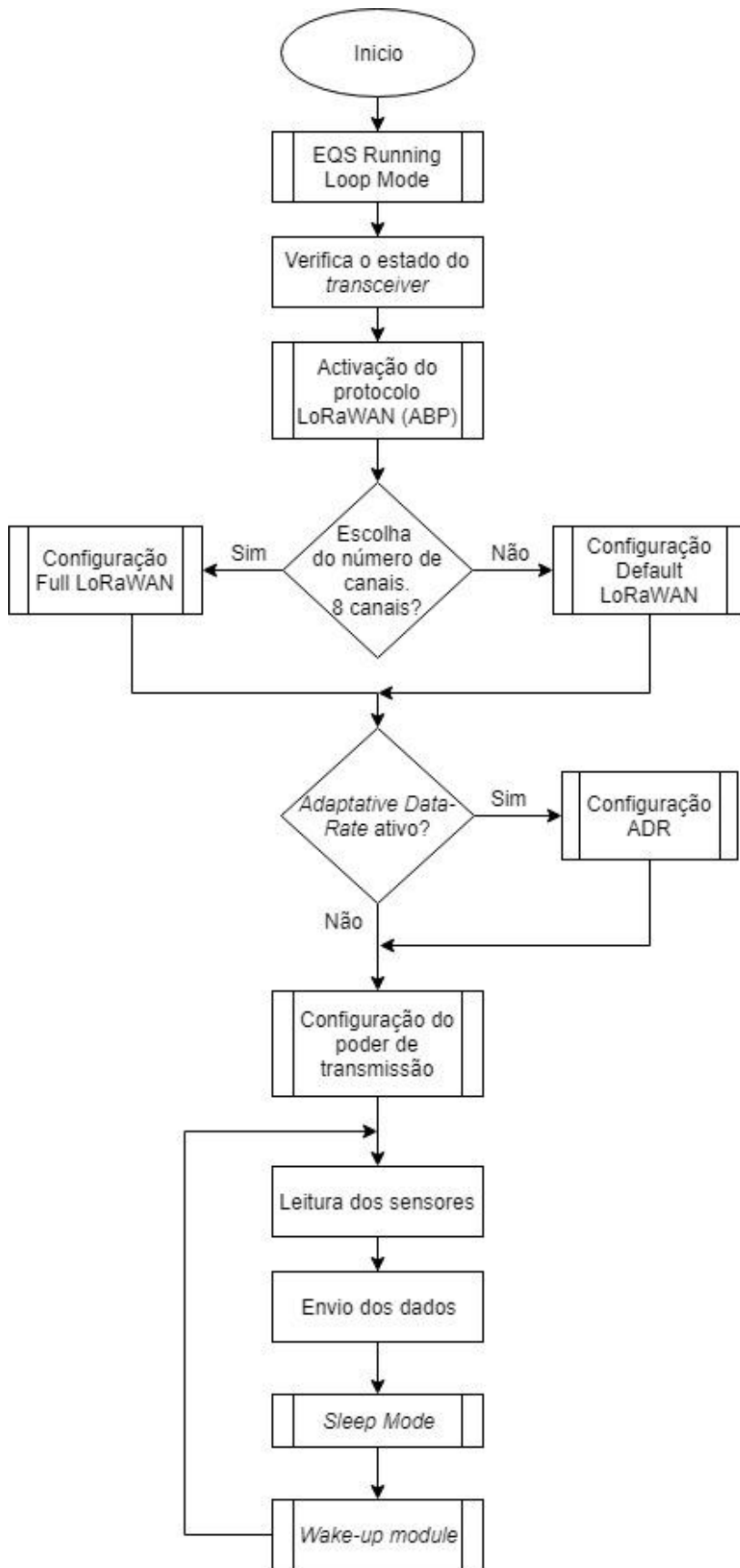


Figura 76 Fluxograma de funcionamento do nó Microchip

Após as configurações iniciais, o sistema entra em ciclo. Para além da rotina de leitura e envio de dados são chamadas duas rotinas. Uma que “adormece” o módulo, desligando o *transceiver*, os *timers* e os periféricos, de forma a otimizar os consumos energéticos, e outra que inicializa as funcionalidades necessárias apenas para o envio da informação.

Para a leitura dos sensores foram criadas rotinas para a configuração dos pinos analógicos como de leitura. A rotina de configuração é sempre chamada, antes da rotina de leitura, uma vez que como o módulo é “desligado”, é necessário reconfigurar estas ligações. No caso da rotina de configuração dos pinos esta faz o envio da mensagem de configuração, aguarda pela resposta, e verifica se o processo foi concluído, informando o utilizador, através do LCD, do sucedido.

Por outro lado, para a rotina de leitura foi criada uma rotina auxiliar, que apenas é utilizada para a leitura dos valores dos sensores. Esta verifica a receção dos dados, *byte a byte*, enviados pelo módulo, e armazena-os num *buffer*.

No que diz respeito ao envio dos dados é preparado um *buffer* com um numero aleatório, entre 0 e 255, que representa o porto de envio, seguido dos dados dos sensores a enviar, devidamente separados. Segue-se a alocação deste *buffer* com a mensagem de transmissão de dados. Após esta, o sistema espera pela resposta do módulo a confirmar a correta transmissão da trama, conforme apresenta o fluxograma da Figura 78.



Figura 77 Fluxograma de envio dos dados

Na rotina de `wake_up()` (Figura 79) foi necessário fazer a reinicialização da comunicação EUSART, para depois proceder ao envio dos comandos para o *Transceiver*. Uma vez que foi efetuada uma queda nas comunicações é necessário fazer o ciclo de restabelecimento e sincronização. Como tal, conforme apresenta o *datasheet* do microcontrolador [70], é necessário fazer o envio de um caracter especial, para o *buffer* de transmissão, que é o *byte* de sincronismo do *Auto-baud*. Todavia, antes é necessário habilitar o funcionamento da EUSART, do seu circuito de transmissão e configurar o modo de operação.

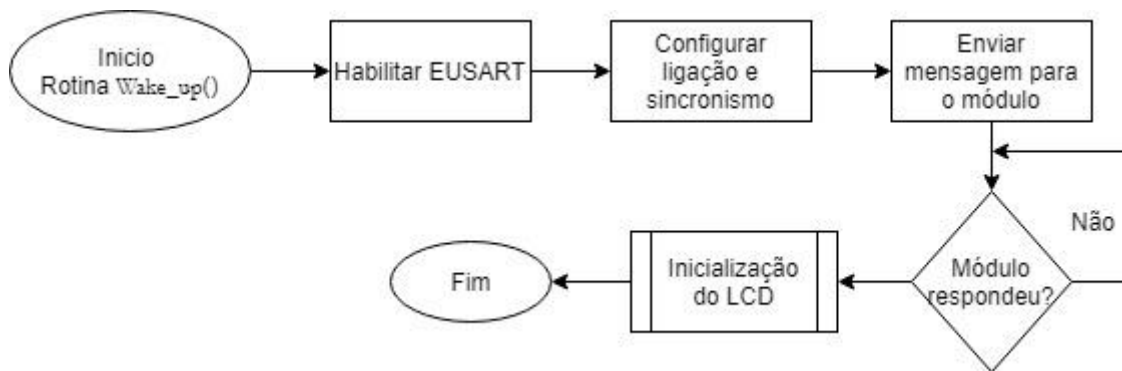


Figura 78 Fluxograma de funcionamento da rotina wake_up()

De seguida, tendo as comunicações sido reativadas, é enviada uma mensagem para o módulo para desencadear uma resposta, e após a receção desta, é chamada a rotina de inicialização do LCD.

6. TESTES

O objetivo deste Capítulo é a apresentação dos vários ensaios realizados ao sistema e o esclarecimento da origem destes, mediante a exposição das características a testar. Por último são expostas as conclusões tiradas nos ensaios. Faz também parte deste capítulo o esclarecimento das soluções encontradas para contornar os problemas verificados na rede criada.

6.1. DISTÂNCIA

Um dos parâmetros mais importantes a testar era a alcance do sistema. Isto porque este era a principal vantagem desta tecnologia face às demais. Assim, optou-se por realizar diversos testes no sentido de verificar, numa aplicação prática, estes valores e compreender se se adequam ao âmbito de aplicação da rede.

De antemão eram conhecidos alguns fatores que iam condicionar o desempenho da rede tais como o meio envolvente, o posicionamento dos nós e *gateway*, as antenas utilizadas e o poder de transmissão. Todavia, foi necessário fazer o levantamento de dados recolhidos em situações reais para posterior avaliação.

O objetivo destes testes é a verificação do alcance máximo da rede bem como a compreensão da influência de cada uma das características anteriores neste parâmetro.

Uma vez que nos testes realizados no capítulo do protótipo, com outras redes criadas, o valor máximo de distância obtido foi de 1 km, este foi um valor de referência para os testes. Deste modo foi possível ter uma sensibilidade e percepção inicial para os valores a esperar.

A Figura 80 apresenta a localização do *gateway* e as várias posições do nó. Nestes testes optou-se por utilizar apenas um nó para facilitar a leitura dos dados do lado do servidor.

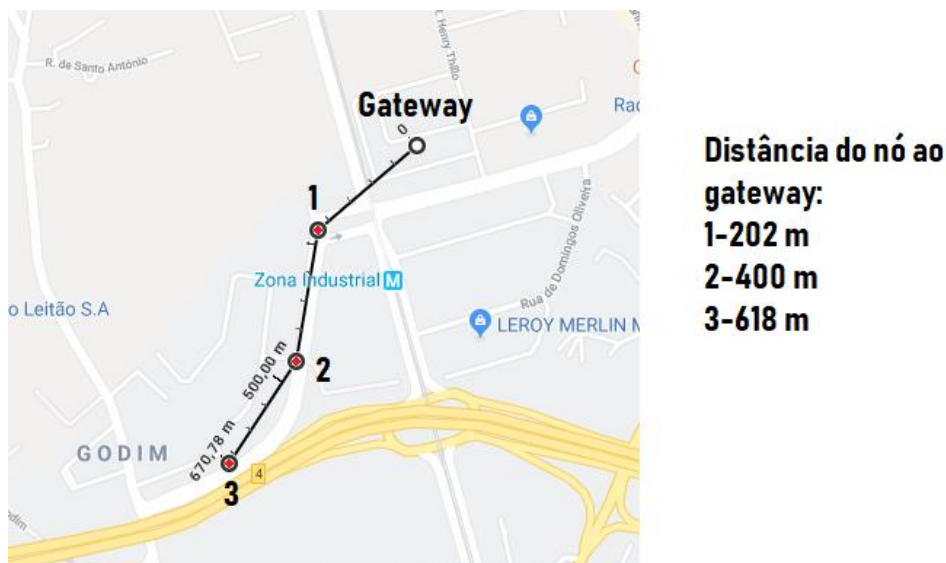


Figura 79 Mapeamento dos locais de teste

O teste consistiu em enviar leituras do sensor de minuto em minuto, durante 10 minutos, no mesmo local. Mediante a receção destes pacotes, por parte do servidor e *gateway*, a distância era aumentada. Isto até que estes deixassem de ser recebidos ou tivessem uma latência que impedisse que em 10 pacotes nenhum destes fosse recebido.

Na posição 1, a cerca de 200 m do *gateway*, todos os pacotes enviados foram recebidos. A 400 m, na posição 2, verificou-se uma perda de pacotes de 50%, uma vez que em 10 pacotes, apenas 5 chegaram ao seu destino. Todavia, a indicação era de que 2 em cada 3 pacotes era perdido, verificando-se uma latência de 2 pacotes. Por último, na posição 3, nenhum dos pacotes enviados foi recebido.

Os resultados deste teste não foram como o esperado. Uma vez que em testes anteriores, foram obtidos valores de alcance bastante superiores a estes.

Após análise ao problema foram identificadas três possíveis razões para os resultados obtidos. Uma delas prende-se com o ambiente envolvente, uma vez este é situado numa zona industrial, com uma forte presença de infraestruturas que prejudicam a desempenho do sistema. Para além deste dado, a altitude a que se encontra o *gateway* e os nós, vai decrescendo ao longo do percurso, fator este que condiciona a ligação. Por outro lado, também foi possível identificar duas melhorias a fazer ao sistema. Uma delas é a alteração do posicionamento do *gateway* para um local mais elevado ou até mesmo *outdoor*. Também, no que diz respeito ao poder de transmissão, tomou-se a opção de desenvolver e incluir uma rotina que incrementa este parâmetro para melhores resultados, uma vez que este valor pode ser alterado entre -3 a 15 dBm.

Assim, incluída esta rotina, seguiram-se mais testes de distância (Figura 81), todavia em locais diferentes.

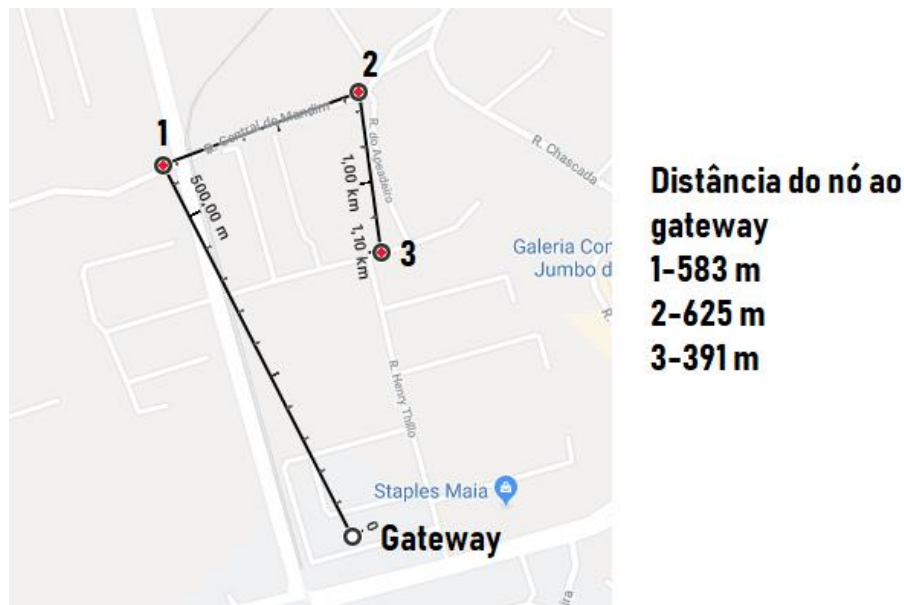


Figura 80 Mapeamento da rede

Numa primeira fase optou-se por utilizar o valor mínimo do poder de transmissão e realizar os testes, para depois, utilizar a rotina de ajuste deste parâmetro e verificar as diferenças.

Assim, no primeiro teste, a 583 m de distância apenas 30% dos pacotes foram recebidos. Na posição 2, a 625 m, nenhum dos 10 pacotes enviados foi recebido. Por fim, a 391 m, apenas um pacote foi perdido.

Este ensaio confirmou a dificuldade verificada no teste anterior, isto porque a zona envolvente tem uma forte presença de infraestruturas e outras variáveis que condicionam as comunicações. Deste modo, optou-se por um outro tipo de teste, para tentar perceber a influências destas infraestruturas.

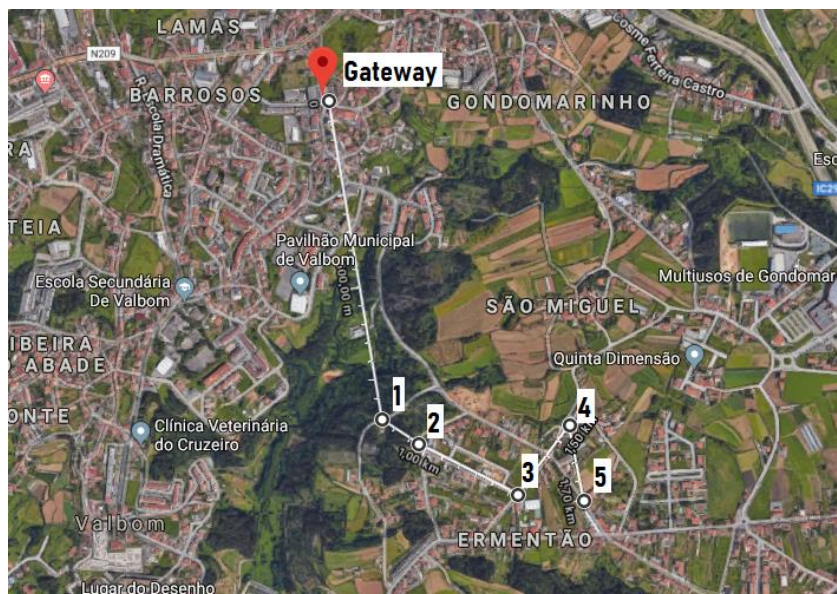
Como tal, foi realizado um outro teste no qual a distância entre os dois locais mantinha-se relativamente inalterada, todavia, não temos mais construções entre o *gateway* e o nó.

Assim, a Figura 82 apresenta os dois locais analisados e os resultados comprovam que a presença destas construções, condiciona a comunicação, uma vez que na posição 1, todos os pacotes foram recebidos. Facto este que não se verifica na posição 2, uma vez que os pacotes foram todos perdidos.



Figura 81 Mapeamento dos novos locais de teste

Tendo em conta os resultados obtidos, optou-se por testar o sistema num local com condições mais favoráveis que as anteriores. Assim, procurou-se um local mais elevado, com menos infraestruturas envolventes de onde fosse possível localizar visualmente o local dos testes. A Figura 83 apresenta a localização do *gateway* e dos nós neste teste.



Distância dos nós ao gateway:
1-855 m
2-949m
3-1,16 km
4-1,08 km
5-1,24 km

Figura 82 Mapeamento da rede

Numa primeira fase o objetivo era verificar o correto funcionamento na posição 1. Isto porque entre a localização do *gateway* e este, não existia nenhuma infraestrutura. Após o envio de 5 pacotes de dados, todos os dados foram recebidos pelo *gateway*. Como tal, optou-se por verificar a distância máxima possível nesta localização. Na posição 5, a 1,24 km de distância foi obtido o maior alcance do sistema, tendo sido recebidos os 5 pacotes de teste.

6.2. QUALIDADE DE SINAL

Para além dos testes de medição da distância máxima atingida, optou-se também por utilizar os dados fornecidos pelo servidor, sobre a intensidade de sinal e relação sinal-ruído, para ter uma melhor perceção do desempenho da tecnologia, conforme apresenta a Figura 84.

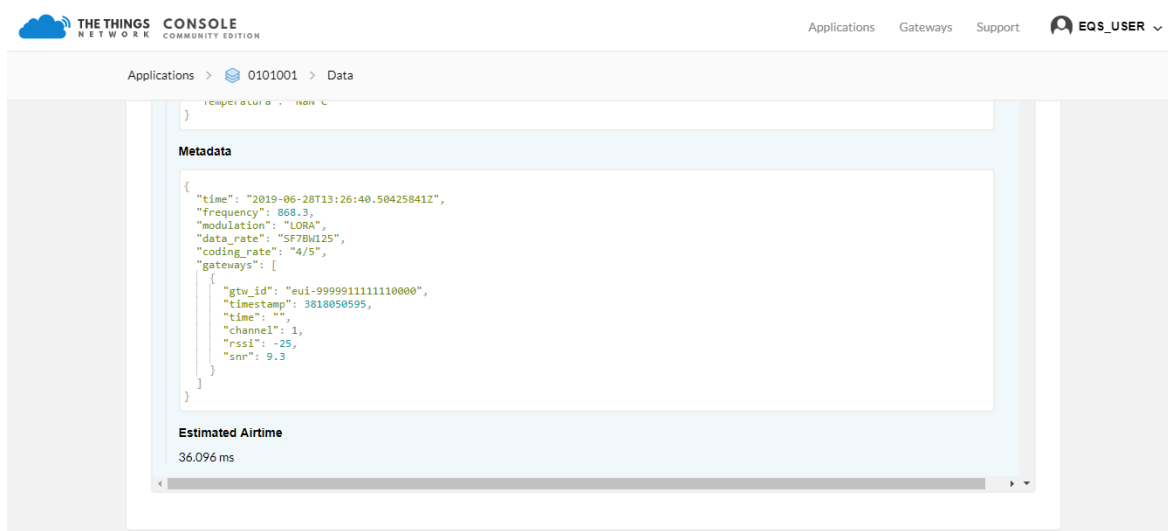


Figura 83 Parâmetros da comunicação

Nos testes de distância existem várias variáveis que são incontroláveis e que influenciam o desempenho da rede. Por exemplo, diferenças de altitude, edifícios, posicionamento do *gateway*, entre outros.

Deste modo, é possível ter uma melhor percepção da *performance* do sistema, através da utilização de parâmetros conhecidos e uniformes. Assim, foram utilizados quer os nós da Dragino, quer os nós da Microchip para este teste.

O valor do RSSI (*Received Signal Strength Indicator*) serve para indicar a intensidade do sinal recebido pelo *gateway*, por outro lado o SNR (*Signal-To-Noise Ratio*) representa a relação sinal em função do ruído. Com estes parâmetros é possível fazer estimativas das distâncias a obter.

O ensaio consistiu em fazer a transmissão de 10 tramas de dados e fazer a média dos valores lidos no servidor. Os dados apresentados na Tabela 4 são referentes a estes testes.

Através do primeiro valor é possível ter uma percepção da qualidade do sinal que chega ao recetor, e pela análise da tabela, conforme era esperado, o equipamento da Microchip apresenta uma vantagem em relação ao da Dragino.

Tabela 4 Dados adquiridos no teste

Parâmetro	RSSI (dBm médio)	SNR (médio)	Localização (m)
Microchip Mote	-92	9,72	1
	-94	8,72	90
	101,6	7,3	215
	108,4	-6,66	418
Dragino	-94,6	9,14	1
	-98	8,42	90
	-95,6	9,12	215
	-112,4	-4,28	418

Por outro lado, partindo destes valores, é possível também ter uma perceção das distâncias a obter. Apesar de os valores de RSSI serem bastante instáveis e dependentes do ambiente, estes cálculos servem para fornecer uma ideia da ordem dos valores.

Como tal, para o cálculo da distância foi utilizada a fórmula presente na Figura 85.

$$RSSI \text{ (dBm)} = -10 * (\text{Propagation Constant } n) * \log_{10}(\text{distance}) + (\text{dBm @ 1 metre})$$

- Propagation Constant (Path Loss) in free space = 2
- dBm @ 1m: Reference Transmit Power at Source = **-60.0** for RPi3

Figura 84 Equação de cálculo da distância com base no RSSI [69]

Através da manipulação da equação, em função da distância, e substituindo os valores obtidos do RSSI para 1 metro temos que quer para o equipamento da Dragino quer para o da Microchip, a distância máxima é de cerca de 8 mil metros.

Este foi um resultado satisfatório, uma vez que aponta para valores de distância próximos dos anunciados pelos detentores da tecnologia.

7. CONCLUSÕES

Ao longo deste trabalho foram sendo desenvolvidos conhecimentos ao nível das comunicações sem fios *wireless*, em particular da tecnologia LoRa, bem como dos outros protocolos de comunicação. Foi estudada e avaliada uma solução para o problema proposto, bem como realizado todo o trabalho associado ao estado da arte para escolher os equipamentos a utilizar.

No que diz respeito ao objetivo deste trabalho podemos concluir que este foi atingido, uma vez que partindo dos equipamentos fornecidos, foi criado um sistema de monitorização *wireless* que envia as leituras de sensores para o servidor, com recurso ao protocolo LoRaWAN. Estas podem ser consultadas pelo utilizador através da interface gráfica do servidor TTN, onde foi criado um decodificador para as diferentes mensagens de cada nó.

Para além disto, a rede privada criada, permite a interligação de mais nós no sistema, bem como de *gateways*, permitindo assim a escalabilidade da mesma. Posto isto, também temos as vantagens associadas ao protocolo, como o envio das tramas de forma segura, com baixos consumos energéticos e elevado alcance, desde os dispositivos de campo até ao servidor.

Quanto aos testes efetuados ao sistema, podemos concluir que alguns dos dados obtidos não foram de acordo com o esperado. Mais especificamente nos primeiros testes de distância efetuados, onde eram esperados outros valores. Todavia, até então era desconhecida a medida de influência de parâmetros como a localização do *gateway*, dos nós e infraestruturas envolventes, na qualidade do sinal. Por outro lado, no ultimo teste realizado com o sistema, com o *gateway* numa posição mais favorável, foram obtidos resultados bastante satisfatórios, dentro das especificações apresentadas para a tecnologia. Assim, foi possível validar o conceito do projeto e comprovar as potencialidades do LoRa como tecnologia de comunicação de longo alcance.

De forma adicional, nos testes efetuados com recurso aos dados fornecidos pelo servidor em relação à qualidade de sinal (RSSI e SNR) foi possível verificar quer a correta operação dos equipamentos quer obter uma medida da intensidade de sinal e poder tirar conclusões quanto às distâncias a esperar do equipamento, a operar em condições melhores. Tendo estes resultados apontado para distâncias de máximas de 8 km.


Assim, no que toca à implementação de um sistema de monitorização na indústria petroquímica, podemos concluir que a localização do *gateway* é uma característica importantíssima nas comunicações. Estes devem estar posicionados o mais alto possível, no exterior e com o menor número possível de infraestruturas nas suas proximidades.

Em suma, a escolha do LoRa e do protocolo LoRaWAN como tecnologias a utilizar neste projeto é adequada, devido às necessidades de segurança, em conformidade com criação de rede privada, custo, longo alcance e consumos energéticos, exigidas. Por outro lado, os equipamentos utilizados são indicados à solução a implementar uma vez que cumprem com os requisitos para esta aplicação.

Como futuras ideias a implementar para melhorarem o sistema temos a inclusão de uma rotina que faz a leitura do nível da bateria dos nós e envia um alerta para o servidor, no caso de esta estar abaixo de um certo nível. Para tal, uma solução seria conectar a alimentação a um conversor analógico-digital, e por *software* verificar este valor. No caso do equipamento da Microchip, uma vez que este utiliza uma bateria de 3 V, esta ligação pode ser direta. Por outro lado, no caso dos nós da Dragino, é necessário a inclusão de um divisor de tensão, uma vez que esta é de 9 V, e o conversor A/D apenas suporta até 5 V. Por outro lado, outra melhoria a incluir é a criação de um protocolo de comunicação, com

criptação para utilizar na rede privada, tornando esta uma solução adequada ao meio industrial.

Referências Documentais

- [1] eqs-servicos-de-engenharia-qualidade-e-seguranca_20170221151153.jpg (600×200). (n.d.). Retrieved March 3, 2019, from https://www.jobmatching.pt/uploads/logos/eqs-servicos-de-engenharia-qualidade-e-seguranca_20170221151153.jpg
- [2] Web Exclusive: Analysis of wireless industrial automation standards: ISA-100.11a and WirelessHART - ISA. (n.d.). Retrieved March 2, 2019, from <https://isajobs.isa.org/standards-publications/isa-publications/intech-magazine/2012/december/web-exclusive-analysis-wireless-industrial-automation-standards-isa-100-11a-wirelesshart/>
- [3] Petroquímica - Química e Petróleo - InfoEscola. (n.d.). Retrieved March 4, 2019, from <https://www.infoescola.com/quimica/petroquimica/>
- [4] Conno, C. De. (n.d.). GPSA Engineering Data Book [Gas Processing] 12th ed_2004. Retrieved from https://www.academia.edu/11912717/GPSA_Engineering_Data_Book_Gas_Processing_12th_ed_2004
- [5] M2M Technology to be Worth Above \$1 Trillion by 2020 - World Industrial Reporter. (n.d.). Retrieved March 4, 2019, from <https://worldindustrialreporter.com/m2m-technology-to-be-worth-above-1-trillion-by-2020/>
- [6] M2M Technology to be Worth Above \$1 Trillion by 2020 - World Industrial Reporter. (n.d.). Retrieved March 4, 2019, from <https://worldindustrialreporter.com/m2m-technology-to-be-worth-above-1-trillion-by-2020>
- [7] O protocolo BACnet. (n.d.). Retrieved March 5, 2019, from http://webcache.googleusercontent.com/search?q=cache:http://www.apirac.pt/documentos/comunicacoes_2_inst/c_05.pdf
- [8] Types of Network Architecture - DEV Community . (n.d.). Retrieved March 5, 2019, from https://dev.to/cadellsinghh_25/types-of-network-architecture-56i7
- [9] Wireless Sensor Network (WSN) Architecture And Applications. (n.d.). Retrieved March 5, 2019, from <https://www.elprocus.com/architecture-of-wireless-sensor-network-and-applications/>

- [10] Joaquim, D., Rodrigues, A., Laura, A., De, T., & Moreira, S. (2016). *Antenas para Redes de Sensores sem Fios*. Retrieved from <https://digituma.uma.pt/bitstream/10400.13/1590/1/MestradoLauraMoreira.pdf>
- [11] Figura-5-Topologias-suportadas-pelo-padrao-IEEE-802154-Alguns-padroes-para-camadas.png (850×343). (n.d.). Retrieved March 10, 2019, from https://www.researchgate.net/profile/Ruan_Gomes/publication/270137615/figure/fig3/AS:392039956008960@1470481017056/Figura-5-Topologias-suportadas-pelo-padrao-IEEE-802154-Alguns-padroes-para-camadas.png
- [12] Industrial LAN/WAN Networks | Vector InfoTech. (n.d.). Retrieved March 10, 2019, from <https://www.vectorinfotech.com/solutions/industrial-lan-wan-networks/>
- [13] Lu, C., Saifullah, A., Li, B., Sha, M., Gonzalez, H., Gunatilaka, D., ... Chen, Y. (2016). Real-Time Wireless Sensor-Actuator Networks for Industrial Cyber-Physical Systems. *Proceedings of the IEEE*, 104(5), 1013–1024. <https://doi.org/10.1109/JPROC.2015.2497161>
- [14] Kumar S., A. A., Ovsthus, K., & Kristensen., L. M. (2014). An Industrial Perspective on Wireless Sensor Networks — A Survey of Requirements, Protocols, and Challenges. *IEEE Communications Surveys & Tutorials*, 16(3), 1391–1412. <https://doi.org/10.1109/SURV.2014.012114.00058>
- [15] What is WSN (wireless sensor and actuator network)? - Definition from WhatIs.com. (n.d.). Retrieved March 11, 2019, from <https://internetofthingsagenda.techtarget.com/definition/WSAN-wireless-sensor-and-actuator-network>
- [16] Web Exclusive: Analysis of wireless industrial automation standards: ISA-100.11a and WirelessHART - ISA. (n.d.). Retrieved March 12, 2019, from <https://isajobs.isa.org/standards-publications/isa-publications/intech-magazine/2012/december/web-exclusive-analysis-wireless-industrial-automation-standards-isa-100-11a-wirelesshart/>
- [17] 5---Fig5_ISA100_tunneling.jpg (450×267). (n.d.). Retrieved March 12, 2019, from https://www.isa.org/uploadedImages/Content/Standards_and_Publications/ISA_Publications/InTech_Magazine/2012/December/5---Fig5_ISA100_tunneling.jpg
- [18] Song, J., Han, S., Mok, A., Chen, D., Lucas, M., Nixon, M., & Pratt, W. (2008). WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control. In *2008 IEEE Real-Time and Embedded Technology and Applications Symposium* (pp. 377–386). IEEE. <https://doi.org/10.1109/RTAS.2008.15>
- [19] Tudo pronto para abraçar a NB-IoT? - Blog PT Empresas. (n.d.). Retrieved March 13, 2019, from <https://blog.ptempresas.pt/tudo-pronto-para-abracar-a-nb-iot-73078>
- [20] NarrowBand-IoT- | Altice Portugal. (n.d.). Retrieved March 14, 2019, from <https://www.telecom.pt/pt-pt/media/noticias/Paginas/2018/novembro/NarrowBand-IoT-.aspx>

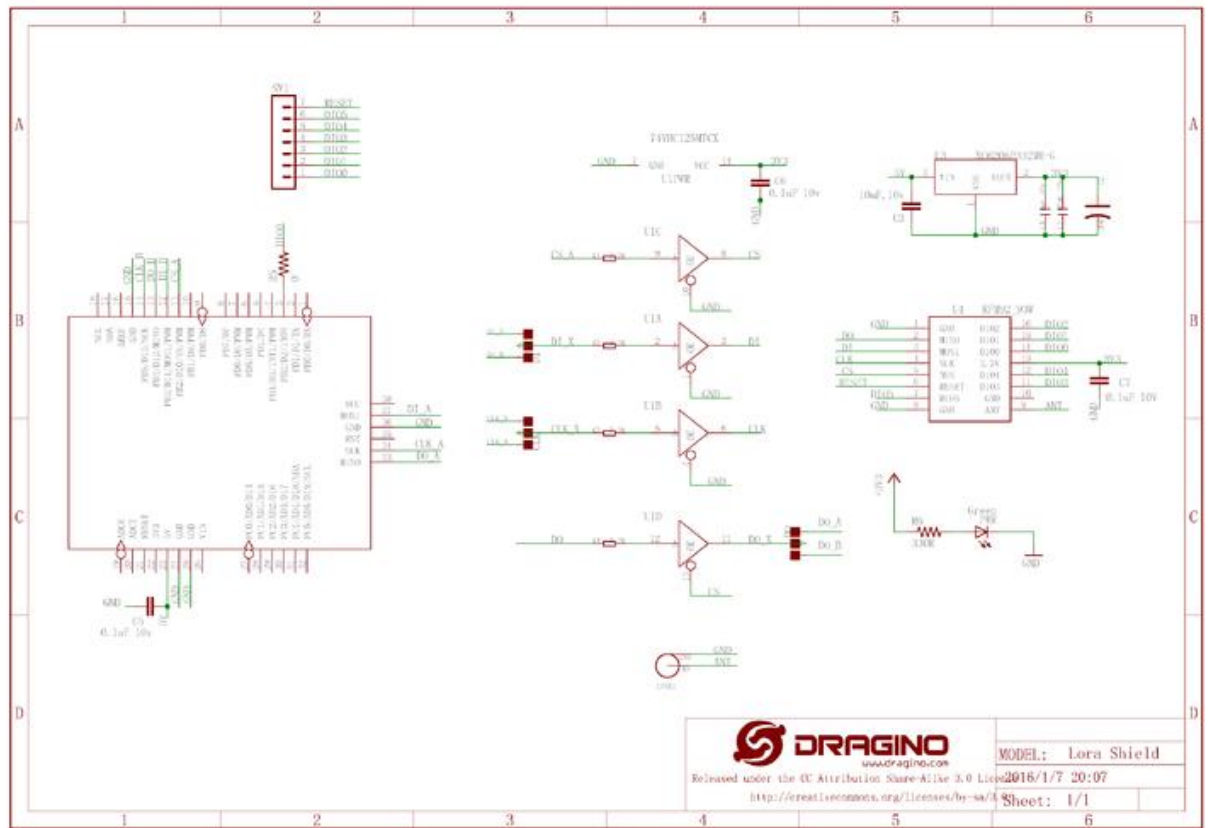
- [21] What is NB-IoT? - everything RF. (n.d.). Retrieved March 14, 2019, from <https://www.everythingrf.com/community/what-is-nb-iot>
- [22] 4019.1519798933.jpg (2000×1125). (n.d.). Retrieved March 14, 2019, from <https://devopedia.org/images/article/59/4019.1519798933.jpg>
- [23] image15.png (737×342). (n.d.). Retrieved March 12, 2019, from <https://www.embarcados.com.br/wp-content/uploads/2018/02/image15.png>
- [24] LoRaWAN-Overview.png (2000×1500). (n.d.). Retrieved March 14, 2019, from <https://www.thethingsnetwork.org/docs/lorawan/LoRaWAN-Overview.png>
- [25] About LoRaWAN™ | LoRa Alliance™. (n.d.). Retrieved March 15, 2019, from <https://lora-alliance.org/about-lorawan>
- [26] LoRaWAN | The Things Network. (n.d.). Retrieved March 15, 2019, from <https://www.thethingsnetwork.org/docs/lorawan/>
- [27] Limitations | The Things Network. (n.d.). Retrieved March 16, 2019, from <https://www.thethingsnetwork.org/docs/lorawan/limitations.html>
- [28] Wixted, A. J., Kinnaird, P., Larijani, H., Tait, A., Ahmadi, A., & Strachan, N. (2016). Evaluation of LoRa and LoRaWAN for wireless sensor networks. In *2016 IEEE SENSORS* (pp. 1–3). IEEE. <https://doi.org/10.1109/ICSENS.2016.7808712>
- [29] Different-LoRaWAN-classes.ppm (644×350). (n.d.). Retrieved March 21, 2019, from <https://www.researchgate.net/publication/326134076/figure/fig18/AS:668398971285525@1536370141790/Different-LoRaWAN-classes.ppm>
- [30] The authors reserve the right to change specifications without notice. LoRaWAN™ 1.1 Specification 2 NOTICE OF USE AND DISCLOSURE 5. (2017). Retrieved from https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-_v1.1.pdf
- [31] Barbosa, R. M. R. (2017). Sistema de monitorização de consumo de água utilizando tecnologia Sigfox. Retrieved from <https://repositorio-aberto.up.pt/handle/10216/106909>
- [32] Uma visão técnica da Rede Sigfox e seus princípios - Embarcados. (n.d.). Retrieved March 18, 2019, from <https://www.embarcados.com.br/uma-visao-tecnica-da-rede-sigfox/>
- [33] visão-tecnica-Rede-Sigfox-11.png (724×423). (n.d.). Retrieved March 17, 2019, from <https://www.embarcados.com.br/wp-content/uploads/2017/08/visão-tecnica-Rede-Sigfox-11.png>
- [34] Design and Analysis of Wireless Networks for Petroleum Refineries. (n.d.). Retrieved March 17, 2019, from <https://prism.ucalgary.ca/handle/11023/4114>

- [35] Uma visão técnica da Rede Sigfox e seus princípios - Embarcados. (n.d.). Retrieved March 21, 2019, from <https://www.embarcados.com.br/uma-visao-tecnica-da-rede-sigfox/>
- [36] SigFox Vs. LoRa: A Comparison Between Technologies & Business Models. (n.d.). Retrieved March 20, 2019, from <https://www.link-labs.com/blog/sigfox-vs-lora>
- [37] What is RPMA Technology? - everything RF. (n.d.). Retrieved March 17, 2019, from <https://www.everythingrf.com/community/what-is-rpma-technology>
- [38] Finnegan, J., & Brown, S. (n.d.). *A Comparative Survey of LPWA Networking*. Retrieved from <https://arxiv.org/pdf/1802.04222.pdf>
- [39] Raza, U., Kulkarni, P., & Sooriyabandara, M. (n.d.). *Low Power Wide Area Networks: An Overview*. Retrieved from <https://arxiv.org/pdf/1606.07360.pdf>
- [40] Peer to Peer (P2P) Networks Explained. (n.d.). Retrieved April 12, 2019, from <https://youtubemusicsucks.com/peer-to-peer-p2p-networks-explained/>
- [41] WTF is peer-to-peer? – Albert Hu – Medium. (n.d.). Retrieved April 12, 2019, from <https://medium.com/@alberthu/wtf-is-peer-to-peer-4b865d29e44e>
- [42] 2017-11-10-lcom-figure-1.jpg (1200×663). (n.d.). Retrieved April 12, 2019, from <https://eenews.cdnartwhere.eu/sites/default/files/images/01-picture-library/JPJoosting/2017/2017-11-10-lcom-figure-1.jpg>
- [43] Nixon, M. (n.d.). Title: A Comparison of WirelessHART and ISA100.11a A Comparison of WirelessHART™ and ISA100.11a Title: A Comparison of WirelessHART and ISA100.11a PRELIMINARY A. Retrieved from <https://www.controlglobal.com/assets/12WPpdf/120904-emerson-wirelesshart-isa.pdf>
- [44] WirelessHART. (n.d.). Retrieved April 12, 2019, from <https://pt.slideshare.net/eawareTech/wirelesshart>
- [45] Ingenu Webinar: How RPMA Works - YouTube. (n.d.). Retrieved April 12, 2019, from <https://www.youtube.com/watch?v=4beoZapuBXw>
- [46] How to Use/Configure GPIO Pins on an RN2483 Device - Developer Help. (n.d.). Retrieved April 17, 2019, from <http://microchipdeveloper.com/faq:1088>
- [47] *LoRa® Mote User's Guide*. (2015). Retrieved from [https://ww1.microchip.com/downloads/en/DeviceDoc/LoRa Mote Users Guide.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/LoRa%20Mote%20Users%20Guide.pdf)
- [48] LoRaWAN-Overview.png (2000×1500). (n.d.). Retrieved April 17, 2019, from <https://www.thethingsnetwork.org/docs/lorawan/LoRaWAN-Overview.png>
- [49] Lora-wan-class.png (989×444). (n.d.). Retrieved April 14, 2019, from <https://witekio.com/wp-content/uploads/2018/01/Lora-wan-class.png>
- [50] Schema-Security.png (2280×2280). (n.d.). Retrieved April 15, 2019, from <https://www.resiot.io/wp-content/uploads/2018/01/Schema-Security.png>

- [51] What is the difference between OTAA and ABP Devices - End Devices (Nodes) - The Things Network. (n.d.). Retrieved April 17, 2019, from <https://www.thethingsnetwork.org/forum/t/what-is-the-difference-between-otaa-and-abp-devices/2723>
- [52] LoRaWAN | The Things Network. (n.d.). Retrieved April 15, 2019, from <https://www.thethingsnetwork.org/docs/lorawan/>
- [53] dragino-gateway-au915mhz-as923mhz-non-screw-version-no-cellular-module-lora-openwrt-iot-gateway-lg01-26158274899_1024x1024.png (746×435). (n.d.). Retrieved April 16, 2019, from https://cdn.shopify.com/s/files/1/1386/3791/products/dragino-gateway-au915mhz-as923mhz-non-screw-version-no-cellular-module-lora-openwrt-iot-gateway-lg01-26158274899_1024x1024.png?v=1550475591
- [54] DM164138 Microchip Technology | RF/IF and RFID | DigiKey. (n.d.). Retrieved April 16, 2019, from <https://www.digikey.com/product-detail/en/microchip-technology/DM164138/DM164138-ND/5639433>
- [55] *IBM Long Range Signaling and Control IBM LoRaWAN Modem (IMST WiMOD Edition)*. (2015). Retrieved from <https://net.zmartnet.com/shared/IBM-LoRaWAN-Modem-v1.1.pdf>
- [56] Quickstart Debian / Ubuntu - LoRa Server, open-source LoRaWAN network-server. (n.d.). Retrieved April 10, 2019, from <https://www.loraserver.io/guides/debian-ubuntu/>
- [57] ce419ff8046a064da5b57d995e351d565d60f631.png (2000×1188). (n.d.). Retrieved May 1, 2019, from <https://www.thethingsnetwork.org/forum/uploads/default/original/2X/c/ce419ff8046a064da5b57d995e351d565d60f631.png>
- [58] schema-del-processo-di-industria-di-produzione-del-gas-di-olio-per-l-insegna-del-sito-web-94229509.jpg (1300×871). (n.d.). Retrieved May 2, 2019, from <https://thumbs.dreamstime.com/z/schema-del-processo-di-industria-di-produzione-del-gas-di-olio-per-l-insegna-del-sito-web-94229509.jpg>
- [59] Overview | The Things Network. (n.d.). Retrieved May 12, 2019, from <https://www.thethingsnetwork.org/docs/lorawan/frequency-plans.html>
- [60] LM35 Precision Centigrade Temperature Sensors. (1999). Retrieved from www.ti.com
- [61] Limitations: data rate/packet size, 30 sec up and 10 messages down p/d Fair Access - End Devices (Nodes) - The Things Network. (n.d.). Retrieved May 2, 2019, from <https://www.thethingsnetwork.org/forum/t/limitations-data-rate-packet-size-30-sec-up-and-10-messages-down-p-d-fair-access/1300>
- [62] *L80 Hardware Design GPS Module Series*. (2013). Retrieved from www.quectel.com

- [63] GPS Time to First Fix (TTFF) Defined. (n.d.). Retrieved May 12, 2019, from <https://www.lifewire.com/time-to-first-fix-ttff-1683313>
- [64] The future of single-channel gateways - Gateways / Single Channel Gateway - The Things Network. (n.d.). Retrieved May 14, 2019, from <https://www.thethingsnetwork.org/forum/t/the-future-of-single-channel-gateways/6590>
- [65] Single Channel Gateway shows "not connected" - Gateways / Single Channel Gateway - The Things Network. (n.d.). Retrieved May 14, 2019, from <https://www.thethingsnetwork.org/forum/t/single-channel-gateway-shows-not-connected/6189/39>
- [66] pic15.png (611×374). (n.d.). Retrieved May 25, 2019, from <http://globodaq.com/wp-content/uploads/2014/11/pic15.png>
- [67] Mekki, K., Bajic, E., Chaxel, F., & Meyer, F. (2019). A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express*, 5(1), 1–7. <https://doi.org/10.1016/J.ICTE.2017.12.005>
- [68] *LoRa® Technology Gateway User's Guide*. (2016). Retrieved from <http://ww1.microchip.com/downloads/en/DeviceDoc/40001827A.pdf>
- [69] BLE Localiser (Full) for iOS Dev Scout. (n.d.). Retrieved June 29, 2019, from <https://www.slideshare.net/yeokm1/ble-localiser-full-for-ios-dev-scout>
- [70] *PIC18(L)F2X/45K50*. (n.d.). Retrieved from <http://ww1.microchip.com/downloads/en/devicedoc/30000684B.pdf>

Anexo A. Esquemático dos nós Dragino



Anexo B. Sensor de temperatura LM35

