



Bernardo Manso Knoblich

Licenciado em Ciências da Engenharia Eletrotécnica
e de Computadores

Geolocalização e Monitorização de Equipamentos de Assistência a Aeronaves

Dissertação para obtenção do Grau de Mestre em
Engenharia Eletrotécnica e de Computadores

Orientador: Filipe de Carvalho Moutinho, Professor
Auxiliar, Faculdade de Ciências e Tecnologia
da Universidade Nova de Lisboa

Co-orientador: Francisco M S Gonçalves, Diretor da Direção
de Engenharia e Gestão de Equipamentos,
SPDH – Serviços Portugueses de Handling

Júri

Presidente: Doutor Paulo José Carrilho de Sousa Gil, FCT/UNL

Arguente: Doutor Rogério Alexandre Botelho Campos Rebelo,
FCT/UNL

Vogal: Doutor Filipe de Carvalho Moutinho, FCT/UNL

Geolocalização e Monitorização de Equipamentos de Assistência a Aeronaves

Copyright © Bernardo Manso Knoblich, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Agradecimentos

Esta dissertação fica marcada por importantes apoios e incentivos provenientes de inúmeras pessoas, sem as quais não teria sido realizada. Assim sendo, quero agradecer a todos aqueles que de certa forma se envolveram e ajudaram a alcançar este objetivo.

Em primeiro lugar, ao Professor Doutor Filipe Moutinho, ao Engenheiro Bruno Nunes e a toda a equipa de trabalho, que se demonstraram sempre disponíveis para ajudar, rever e orientar todo o trabalho desenvolvido nesta dissertação.

Ao Diretor da Direção de Engenharia e Gestão de Equipamentos, Francisco Gonçalves, que desde o início deste projeto acreditou nas minhas capacidades e se disponibilizou, aconselhando e orientando a atingir os objetivos.

Um agradecimento a todos os docentes e colegas que nesta nobre instituição, Faculdade de Ciências e Tecnologias da Universidade Nova de Lisboa, contribuíram para o meu percurso académico.

Por último, mas não menos importante, um agradecimento muito especial a todos os amigos e família que não só fizeram parte deste ciclo académico, mas que me ajudam, apoiam e sobretudo são a base da minha vida.

Resumo

Com a crescente evolução tecnológica na aviação civil, as atividades de assistência a aeronaves enfrentam um futuro marcado pela otimização operacional e desenvolvimento de processos de segurança e manutenção mais seguros e eficientes.

A presente dissertação propõe um sistema IoT (*Internet of Things*) de geolocalização e monitorização de equipamentos motorizados e não-motorizados de assistência a aeronaves. A ideia proposta visa otimizar os métodos utilizados na gestão operacional, através de informação obtida com a leitura de sensores. O sistema proposto permite monitorizar em tempo real os equipamentos, visualizar os percursos efetuados, bem como planear as suas manutenções preventivas.

Para analisar a viabilidade do sistema apresentado foi criado e testado um protótipo utilizando um módulo servidor e um módulo GSE (*Ground Support Equipment*) com comunicação LoRa, antena GPS e um sensor acelerómetro para detetar o funcionamento do equipamento. Por forma a que o utilizador visualize a informação, foi criada uma interface gráfica com recurso a mapas.

Palavras-chave: Equipamentos de assistência a aeronaves, IoT, Geolocalização, Monitorização, Sensores, GSE, LoRa, GPS, Acelerómetro, Interface gráfica.

Abstract

With the growing technological evolution in civil aviation, aircraft assistance activities face a future marked by operational optimization and the development of safer and more efficient maintenance tasks.

This thesis proposes an IoT (Internet of Things) system for geolocation and tracking of motorized and non-motorized ground support equipment. The overall idea is to optimize methods used in operational management, through information obtained with the readings of sensors. This system allows real-time equipment monitoring, visualization of the routes taken as well as a preventive maintenance planning.

To analyze the systems feasibility, a prototype was created and tested using both a server and a GSE (Ground Support Equipment) module with LoRa communication, GPS antenna and an accelerometer sensor to detect the operation of the equipment. So that the user visualizes the information, a graphical interface was created using maps.

Keywords: Ground Support Equipment, IoT, Geolocation, Tracking, Sensors, LoRa, GPS, Accelerometer, Graphical interface.

Índice de Matérias

Agradecimentos	V
Resumo	VII
Abstract	IX
Índice de Matérias	XI
Índice de Figuras	XIII
Índice de Tabelas	XV
Lista de Siglas	XVII
1 Introdução	1
1.1 Motivação.....	2
1.2 Objetivos e Contribuições	2
1.3 Estrutura da dissertação.....	3
2 Estado de Arte	5
2.1 Internet of Things	5
2.1.1 Aplicações	7
2.1.2 Arquitetura	8
2.1.3 Middleware.....	8
2.1.4 Segurança e Privacidade.....	10
2.2 Protocolos IoT	11
2.2.1 Protocolos de comunicação/rede IoT	11
2.2.1.1 Redes de curto-alcance	12
2.2.1.2 Redes de longo-alcance.....	14
2.2.2 Protocolos de dados IoT	19

2.3	Sensores de geolocalização/Módulo	21
2.4	Trabalhos Relacionados	22
3	Sistema Desenvolvido	27
3.1	Introdução ao Sistema	27
3.2	Módulo GSE.....	29
3.2.1	Descrição do módulo.....	30
3.2.2	Formato da Mensagem	32
3.2.3	Hardware	32
3.3	Módulo Servidor	36
3.3.1	Descrição do módulo.....	36
3.3.2	Hardware	37
3.4	Aplicação de Monitorização.....	39
4	Validação	41
4.1.	Testes ao Módulo GSE.....	41
4.2.	Testes ao Sistema	42
4.3.	Testes de Alcance.....	44
4.4.	Discussão de Resultados	46
5	Conclusões e Trabalhos Futuros	47
5.1	Conclusões	47
5.2	Trabalhos Futuros.....	48
	Referências Bibliográficas	51
	Anexos	55
	Anexo A – Código do módulo GSE.....	55
	Anexo B – Código do módulo Servidor	60

Índice de Figuras

FIGURA 2.1: ESQUEMA DA INTERNET DAS COISAS (ADAPTADA DE [5]).....	6
FIGURA 2.2: APLICAÇÕES IOT.....	7
FIGURA 2.3: GRÁFICO COMPARATIVO DAS COMUNICAÇÕES (RETIRADA DE [24]).....	12
FIGURA 2.4: TOPOLOGIAS ZIGBEE (ADAPTADA DE [28]).....	13
FIGURA 2.5: REDE LoRA.....	15
FIGURA 2.6: PROTOCOLO MQTT (ADAPTADA DE [33]).....	19
FIGURA 2.7: MÉTODO DE FUNCIONAMENTO CoAP.....	20
FIGURA 2.8: FM-ECO4 LIGHT+ S.....	21
FIGURA 2.9: MÓDULO MC60.....	22
FIGURA 3.1: ARQUITETURA DO SISTEMA DE GEOLOCALIZAÇÃO.....	28
FIGURA 3.2: DIAGRAMA DE SEQUÊNCIAS DO SISTEMA.....	29
FIGURA 3.3: PROTÓTIPO DO MÓDULO GSE.....	30
FIGURA 3.4: FLUXOGRAMA ALUSIVO AO FUNCIONAMENTO DO MÓDULO GSE.....	31
FIGURA 3.5: FORMATO DA MENSAGEM ENVIADA PELO MÓDULO GSE.....	32
FIGURA 3.6: ILUSTRAÇÃO DE UM ARDUINO UNO (RETIRADA DE [35]).....	32
FIGURA 3.7: DRAGINO LoRA/GPS SHIELD.....	34
FIGURA 3.8: SENSOR MPU9250-BMP280.....	35
FIGURA 3.9: PROTÓTIPO DO MÓDULO SERVIDOR.....	36
FIGURA 3.10: FLUXOGRAMA ALUSIVO AO FUNCIONAMENTO DO MÓDULO SERVIDOR.....	37
FIGURA 3.11: ILUSTRAÇÃO DE UM ARDUINO MEGA (RETIRADA DE [39]).....	37
FIGURA 3.12: DRAGINO LoRA SHIELD.....	38
FIGURA 3.13: ILUSTRAÇÃO DA INTERFACE GRÁFICA.....	39
FIGURA 4.1: MONITOR SERIAL DO MÓDULO GSE.....	41
FIGURA 4.2: INSTALAÇÃO DO SISTEMA DE GEOLOCALIZAÇÃO.....	42
FIGURA 4.3: REGISTO DAS MENSAGENS DURANTE A ROTA Nº15.....	43
FIGURA 4.4: PERCURSO EFETUADO NO PRIMEIRO TESTE.....	43
FIGURA 4.5: PERCURSO EFETUADO NO SEGUNDO TESTE.....	44
FIGURA 4.6: ALCANCE ATINGIDO NO PRIMEIRO TESTE.....	45
FIGURA 4.7: ALCANCE MÁXIMO ATINGIDO.....	45
FIGURA 4.8: DIAGRAMA DE ISHIKAWA, AVALIAÇÃO DAS PERDAS DE SINAL.....	46

Índice de Tabelas

TABELA 2.1: COMPARAÇÃO DAS TECNOLOGIAS [20],[22],[23],[25].	18
TABELA 3.1: CARACTERÍSTICAS DO ARDUINO UNO [35].	33
TABELA 3.2: LIGAÇÃO DO SENSOR AO ARDUINO UNO.	35
TABELA 3.3: CARACTERÍSTICAS DO ARDUINO MEGA 2560.	38
TABELA 3.4: CARACTERÍSTICAS DO <i>SHIELD</i> LORA.	39

Lista de Siglas

3GPP	3rd Generation Partnership Project
AC	Alternating Current
AeroMACS	Aeronautical Mobile Airport Communication System
AES	Advanced Encryption Standard
AMQP	Advanced Message Queuing Protocol
API	Application Programmable Interface
BLE	Bluetooth Low Energy
BW	Bandwidth
CAN	Controller Area Network
CEP	Circular Error Probable
CoAP	Constrained Application Protocol
CSS	Chirp Spread Spectrum
CSV	Comma-Separated Values
DC	Direct Current
DGPS	Differential Global Positioning System
DTLS	Datagram Transport Layer Security
EEPROM	Electrically-Erasable Programmable Read-Only Memory
FPGA	Field-Programmable Gate Array
GPRS	General Packet Radio Services
GPS	Global Position System
GSE	Ground Support Equipment
GSM	Global System for Mobile Communication
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol

I2C	Inter-Integrated Circuit
ICSP	In-Circuit Serial Programming
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IO	Input-Output
IoT	Internet of Things
IP	Internet Protocol
ISM	Industrial, Scientific and Medical
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
LPWAN	Low Power Wide Area Network
LTE	Long Term Evolution
M2M	Machine to Machine
MQTT	Message Queuing Telemetry Transport
MQTT-SN	Message Queuing Telemetry Transport for Sensor Networks
NB-IoT	Narrow Band Internet of Things
NFC	Near Field Communication
OTM	Outdoor Tracking Module
OBD	On-Board Diagnostic
OFDMA	Orthogonal Frequency Division Multiple Access
QoS	Quality of Service
QR	Quick Response
RFID	Radio-Frequency Identification
RTC	Real-Time Control
RTLS	Real-Time Locating System
SAP	Systems, Applications and Products in data Processing
SC-FDMA	Single Carrier Frequency Division Multiple Access
SSL	Secure Sockets Layer
SRAM	Static Random Access Memory
SQL	Structured Query Language
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UNB	Ultra Narrow Band
UDP	User Datagram Protocol

USB	Universal Serial Bus
VDSAT	Very Small Aperture Terminal
WCDMA	Wide-Band Code-Division Multiple Access
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network

Introdução

Com a globalização a dominar o mundo dos negócios, a facilidade de comunicação e a interdependência económica entre as nações, as empresas necessitam cada vez mais de uma política de atividade que vise o menor custo e uma maior produtividade, resultando numa alta competitividade. Esta realidade requer às empresas um processo contínuo de melhoria dos equipamentos e procedimentos nas várias etapas das suas atividades e, para tal, devem ser traçadas políticas de gestão com vista a melhorar a atividade de uma empresa. Tendo estes fatores em consideração, as empresas necessitam de métodos de previsão que sejam eficientes e confiáveis, de modo que os custos de operação e manutenção sejam reduzidos.

O avanço da tecnologia e o rápido desenvolvimento do mundo digital permite recriar objetos físicos e antigos em objetos inteligentes e otimizados, retirando o máximo das suas capacidades. Com o aparecimento da *Internet of Things* foi possível a conexão de um conjunto de dispositivos à internet. A proliferação de dispositivos inteligentes ajudou a criar uma infraestrutura que permite a sua interligação em rede, associados a um conjunto de serviços de análise de dados.

Os recentes desenvolvimentos nas tecnologias de comunicação permitem que os sistemas deixem de ser monitorizados e controlados manualmente, mas sim controlados automaticamente por um computador ou dispositivos remotos. As empresas com projeção no futuro optam por melhorias inovadoras e pela substituição dos sistemas clássicos por sistemas automáticos e programados.

1.1 Motivação

Atualmente, em Portugal, as empresas de *handling* encontram-se desatualizadas face à evolução tecnológica, estando limitadas a certos recursos informáticos. Como tal, verifica-se a necessidade de melhoria do sistema de gestão de frotas, sendo a Groundforce (empresa de Serviços Portugueses de *Handling*) pioneira em Portugal na procura e implementação de um sistema inteligente, que permita a localização e gestão dos seus equipamentos em tempo real.

A gestão e manutenção dos equipamentos de assistência às aeronaves é atualmente executada de maneira convencional. Toda a gestão da frota, sua localização e disponibilidade de uso, é efetuada manualmente, desperdiçando tempo e meios na procura dos equipamentos pela área aeroportuária. Atualmente, também o planeamento de manutenções preventivas dos equipamentos passa por um processo manual de verificação de horas de funcionamento e controlo dos níveis de consumo dos motores. Estas ações consistem em procedimentos morosos e sujeitos a erros de digitação na transposição para o *software* SAP (programa utilizado pela engenharia e gestão de equipamentos).

A introdução de um sistema de geolocalização de *Ground support equipment* (GSE) é de vital importância para a Groundforce, no entanto, este apresenta custos elevados de aquisição, instalação e funcionamento para a empresa.

1.2 Objetivos e Contribuições

A presente dissertação propõe um sistema de geolocalização baseado no conceito da *Internet of Things* (IoT), que permita monitorizar os equipamentos necessários aos serviços de *handling*. O sistema é constituído por módulos de instalação nos equipamentos motorizados e não-motorizados, por um servidor e por uma aplicação de monitorização. Os módulos têm a capacidade de obter dados através da leitura de sensores instalados nos equipamentos, nomeadamente um sensor de leitura da ignição para os equipamentos motorizados e um localizador de sinal GPS que, posteriormente, transmitem os dados obtidos para um servidor remoto. O servidor recebe e armazena esses dados, criando um historial dos equipamentos monitorizados. Por fim, a aplicação apresenta um mapa de monitorização e fornece ao utilizador a informação dos equipamentos, como a sua identificação, o trajeto realizado, a hora de início e fim do percurso, os quilómetros percorridos e as horas que o motor funcionou. Os dados obtidos são importantes para a monitorização e manutenção dos equipamentos da empresa. Adicionalmente, para uma melhor gestão das suas operações, pretende-se que o sistema executado seja integrado em soluções tecnológicas utilizadas na Groundforce, entre outras, o sistema de alocação RTC (*Real-Time Control*).

A empresa Groundforce dispõe de aproximadamente três mil equipamentos e uma atividade que depende fortemente da utilização eficiente dos mesmos. A introdução de uma

solução deste tipo à generalidade da frota oferece vantagens e melhorias operacionais em vários sectores da empresa.

1.3 Estrutura da dissertação

A dissertação está estruturada em cinco capítulos.

O primeiro capítulo inclui a introdução ao tema, a estrutura da dissertação, os objetivos e contribuições e ainda as motivações que resultaram na escolha do tema.

O segundo capítulo contém o levantamento do estado da arte do tema da dissertação. Para enquadrar o tema, neste capítulo são abordados alguns conceitos teóricos da *Internet of Things (IoT)*, de módulos de sensores e de protocolos de comunicação usados em aplicações IoT, tais como, comunicações RFID, tecnologias LoRa, *Bluetooth* e Wi-Fi. Adicionalmente são apresentadas aplicações já existentes com estas tecnologias, nos equipamentos em estudo e em diversas áreas.

O terceiro capítulo apresenta, numa visão geral, as capacidades do sistema desenvolvido e a implementação dos elementos usados durante a realização do projeto. Relativamente aos módulos de geolocalização, são introduzidos os vários componentes utilizados, bem como as especificações técnicas e as suas funcionalidades. Por último, é apresentada a aplicação desenvolvida para visualização e monitorização dos equipamentos.

O quarto capítulo destina-se à validação e apresentação dos resultados obtidos. São apresentados os testes efetuados ao módulo GSE e à aplicabilidade do sistema através da instalação de um módulo num equipamento. Nesta secção, são ainda mostrados os testes de alcance realizados para o estudo da eficiência da tecnologia de comunicação utilizada no protótipo. Como conclusão do capítulo, são discutidos os resultados obtidos nos testes realizados.

No quinto capítulo, referente às conclusões obtidas na realização do projeto desenvolvido, é apresentada uma reflexão sobre os resultados obtidos durante a implementação do sistema, bem como uma futura otimização e expansão para diversos equipamentos da empresa.

Por último, no final do documento são elencadas as referências consultadas e, em anexos, disponibilizados os excertos do código desenvolvido.

Estado de Arte

Neste capítulo é apresentada uma descrição do conceito da *Internet of Things* (IoT) e da sua importância para o progresso e evolução das empresas, com alguns exemplos de dispositivos IoT e suas aplicações práticas. São igualmente apresentados conceitos teóricos dos módulos, de uma forma fundamentada, dos módulos de geolocalização e dos protocolos de comunicação utilizados nestes módulos. Por último, são apresentados modelos e aplicações já implementadas com esta tecnologia em equipamentos de *handling*.

2.1 Internet of Things

O termo *Internet of Things* (IoT) foi proposto em setembro de 1999 por Kevin Ashton, especialista em inovação digital, ao apresentar uma nova solução do sistema RFID. A solução consistiu na etiquetagem eletrônica de produtos numa linha de produção, facilitando a logística por meio de identificadores de radiofrequência (RFID) e na qual as etiquetas RFID foram consideradas as ‘coisas’ [1]. Surge, desta forma, a primeira definição de IoT numa perspectiva orientada para as coisas. Segundo a comunidade RFID, a IoT pode ser definida como: “Uma rede mundial de objetos interligados exclusivamente endereçados, com base em protocolos de comunicação padrão” [12].

Em 1999, Kevin Ashton citou num artigo do RFID Journal [6]:

“If we had computers that knew everything there was to know about things – using data they gathered without any help from us – we would be able to track and count everything, and greatly reduce waste, loss and cost. We would know when things needed replacing, repairing or recalling, and whether they were fresh or past their best.”

A *Internet of Things* tornou-se um paradigma das tecnologias da informação e da comunicação digital [2]. O futuro será equipado com todo o tipo de objetos inteligentes com processadores e rádios, criando a capacidade de processar e transmitir informação. A

intercomunicação é realizada entre aparelhos, *machine to machine* (M2M) ou diretamente com o utilizador [3]. Esta solução tem reformulado as operações das empresas e a forma de trabalhar das pessoas devido à dimensão da aplicabilidade da tecnologia.

O termo IoT, não se refere apenas a uma Internet convencional, mas sim a todo o tipo de objetos e aparelhos com capacidade computacional e de ligação à *web*, tais como veículos, transportes públicos, sinais de trânsito, eletrodomésticos, detetores de condições ambientais, sensores de presença, sensores de estacionamento, dispositivos médicos, cidades inteligentes, entre muitos outros. Pela diversidade de objetos e serviços que a *Internet of Things* apresenta, torna-se mais prático dividi-la em áreas, como demonstra a figura 2.1.

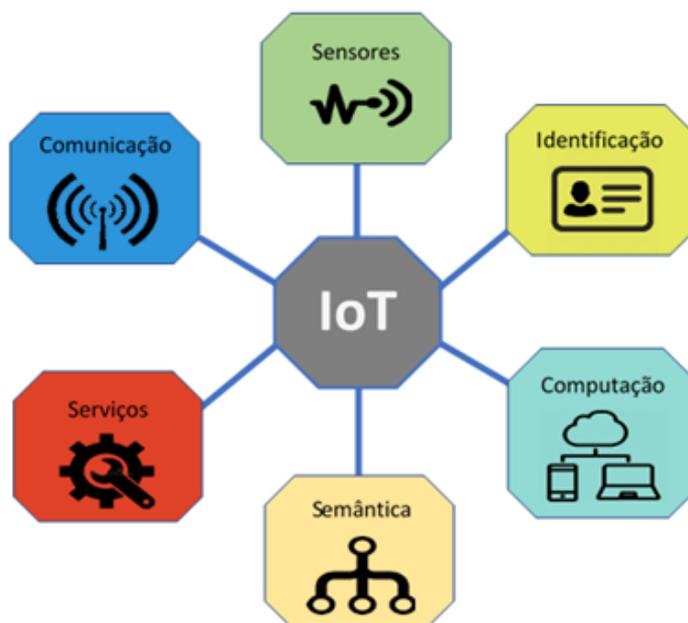


Figura 2.1:Esquema da Internet das coisas (adaptada de [5]).

A utilização de sensores permite obter dados referentes à utilização do objeto em causa. Posteriormente, todos esses dados são tratados e armazenados em servidores locais ou em nuvens. Os objetos podem conter identificação por meio de um código de barras ou de um código QR (código de barras bidimensional), cuja informação/dados podem ser obtidos após a sua leitura.

No campo da computação considera-se a capacidade de processamento por parte dos objetos, para tal, é necessário que na sua composição contenham elementos de processamento tais como, microcontroladores e FPGA.

No contexto da *Internet of Things*, a semântica é um atributo essencial para a comunicação dos diversos dispositivos por meio de uma linguagem computacional. A rede de dispositivos interligados está baseada em determinados protocolos de comunicação, como a tecnologia LoRa, *Bluetooth*, Wi-Fi, RFID e outras, que permitem a receção e o envio de dados entre os dispositivos para desenvolver o serviço pretendido.

A tecnologia IoT tem soluções para desenvolver e combater as necessidades existentes em várias áreas de serviços, como nas frotas das empresas, fornecendo informações em tempo real dos equipamentos, rotas percorridas, trânsito, acidentes e até os quilómetros dos seus

equipamentos. Esta tecnologia envolve, assim, novos modelos de negócio, possibilitando o desenvolvimento dos produtos e dos serviços prestados.

A *Internet of Things* é um conceito de computação que descreve um futuro onde os objetos físicos serão ligados à internet e serão capazes de se interligarem a outros dispositivos. A definição de coisa ou objeto é o elemento-chave na *Internet of things*, por se associar a palavra a vários sistemas de IoT, influenciando os processos que desencadeiam ações e serviços com ou sem intervenção humana direta [2].

“Objects are linked through wired and wireless networks, often using the same Internet Protocol (IP) that connects the Internet. These networks churn out huge volumes of data that flow to computers for analysis. When objects can both sense the environment and communicate, they become tools for understanding complexity and responding to it swiftly.”[4].

2.1.1 Aplicações

Cada vez mais empresas de diversos setores utilizam a IoT para operar com mais eficiência, entender melhor os clientes e oferecer um serviço de qualidade, melhorar a tomada de decisões e aumentar o valor dos negócios. A IoT suporta uma gama de aplicações infinita com diversidade de utilização. Muitos são os domínios em que se pode agrupar as aplicações, tal como mostra a figura 2.2, o domínio de transportes e logística, domínio da saúde, do ambiente inteligente e o domínio pessoal e social [3].



Figura 2.2: Aplicações IoT.

A tecnologia sem fio é ideal para permitir sistemas de localização em tempo real (RTLS) e conectar-se a outras sub-redes de IoT, melhorando o rastreamento e a gestão de equipamentos [9]. O sistema desenvolvido neste trabalho enquadra-se no domínio dos transportes e da logística, uma vez que se pretende visualizar em tempo real os equipamentos motorizados e não-motorizados da

empresa e gerir todas as informações obtidas. Todos os equipamentos, como automóveis, autocarros e tratores são equipados com *tags* e sensores que enviam informações importantes para controlo. Com esta tecnologia, é possível realizar uma otimização mais eficiente da rota, economizar energia, obter informação de circulação, bem como dos níveis de óleos, combustível e temperatura do equipamento.

2.1.2 Arquitetura

A implementação da *Internet of things* (IoT) no mundo real é possível apenas com a integração de várias tecnologias. Para gerir estas tecnologias é proposta uma divisão do conceito de IoT através de uma arquitetura técnica [8]. Sendo a IoT tão vasta e abrangente não existe uma arquitetura uniforme. De acordo com alguns investigadores [8], existem três camadas necessárias para a computação sem fio da *Internet of thing*:

- **Hardware** (infraestrutura) – camada física, composta por sensores, atuadores, câmaras IP e *hardware* de comunicação incorporado que deteta e fornece informação sobre o ambiente para as aplicações, passando ainda pela camada de *middleware*.
- **Middleware** – No ponto de vista da computação, o *middleware* fornece uma camada entre o *software* de aplicação e a camada de comunicação de rede, simplificando a complexidade do sistema ou do *hardware*.
- **Aplicação** – A camada de aplicação é responsável pelas interfaces de visualização e interpretação de todas as informações recebidas pelo *middleware*, facilitando a interação do utilizador com a rede. Define várias aplicações em que a *Internet of Things* pode ser implementada, por exemplo, em cidades inteligentes, escritórios/fábricas inteligentes, casas inteligentes e saúde inteligente [9].

2.1.3 Middleware

A solução adequada para as múltiplas aplicações existentes na IoT é possuir na sua arquitetura uma plataforma *middleware*, que resume os detalhes complexos das coisas para aplicações. Com base na plataforma referida, é elaborado um conjunto de requisitos para a arquitetura do *middleware* suportar a IoT [8], [10]:

1. **Interoperabilidade** – A interoperabilidade pode ser de três tipos: rede, semântica e sintática. Uma rede de dispositivos deve trocar informações através de diferentes redes, utilizando potencialmente diferentes tecnologias de comunicação. A interoperabilidade da rede trata desses protocolos de interface para a comunicação da rede. A interoperabilidade semântica trata da abstração do significado de informação dentro de um determinado domínio, permitindo a troca entre o conjunto dos dispositivos com todos os serviços. Por último, a interoperabilidade sintática permite que as aplicações sejam de diferentes formatos e estruturas codificadoras de dados trocados.

2. **Escalabilidade** – O *middleware* tem de gerir o crescimento da rede de dispositivos e aplicações ou serviços IoT para o sistema conseguir suportar.
3. **Tempo real** – O *middleware* pode fornecer serviços em tempo real quando o sistema implementado necessita da precisão de tempo em que é executada a operação. Como exemplo, os transportes ou cuidados de saúde contêm aplicações em tempo real para fornecer informações ou serviços fundamentais que, se não forem informações atualizadas, deixam de ser úteis para o utilizador.
4. **Análise e gestão de dados** – Para as aplicações IoT funcionarem é fundamental a recolha de dados que contenham as informações necessárias ao utilizador. Esses dados são recolhidos através de sensores ligados aos objetos, que normalmente são recolhidos em grandes quantidades. Por ser necessário analisar todos estes dados com grande detalhe, o *middleware* fornece serviços de gestão de dados, tais como, processamento de dados e armazenamento de dados.
5. **Serviços** – Um *middleware* baseado em serviços e que engloba um conjunto de serviços, como por exemplo, gestão de dados, fiabilidade e segurança, que proporcionam um ambiente flexível e fácil para o desenvolvimento de aplicações. Uma outra parte importante na implantação da IoT é o serviço em nuvem que analisa e armazena os dados dos sensores numa nuvem centralizada.
6. **Contexto** – O contexto é um requisito fundamental na construção de sistemas. A arquitetura do *middleware* tem de ser conhecedora do contexto do sistema, dos dispositivos e do ambiente, para poder oferecer serviços eficazes e essenciais aos utilizadores.
7. **Segurança e privacidade** – As aplicações IoT muitas vezes estão relacionadas com a vida pessoal dos utilizadores ou com indústrias, sendo a segurança fundamental para o seu perfeito funcionamento. Tal como a segurança, o *middleware* que utilize informações pessoais, precisa de preservar a privacidade do utilizador, como por exemplo, a sua localização.

Atualmente, já existem muitas soluções de *middleware* disponíveis para IoT, e todas procuram fornecer os requisitos mencionados. Todos os requisitos suportam a interoperabilidade e a abstração, que é a principal exigência do *middleware* [10].

Concluindo, o *middleware* fornece uma *Application Programming Interface* (API) para comunicação, gestão de dados, computação, segurança e privacidade. Na arquitetura IoT, o *middleware* permitirá a criação de um mapa dinâmico virtual do espaço real, utilizando uma elevada resolução temporal e espacial [9]. Juntamente a esse mapa, combina-se as características da rede de sensores e todos os objetos que estejam interligados.

2.1.4 Segurança e Privacidade

Atualmente, um dos problemas que ameaça, potencialmente, os dispositivos da *Internet of things* é a segurança e a privacidade dos dados recolhidos, que podem estar ligados a privacidade humana e a confidencialidade de processos empresariais. A *Internet of things* tem de ser construída de forma a garantir um controlo seguro e fácil do utilizador. Todos os consumidores precisam de confiança para aderir à IoT, a fim de usufruir dos seus potenciais benefícios e evitar quaisquer riscos para a sua segurança e privacidade [9]. Os desafios de segurança IoT estão relacionados com várias questões de gestão, confidencialidade, integridade, privacidade, autenticidade, identificação, entre muitas outras, sendo estas são as principais preocupações.

A procura de soluções de segurança para combater as ameaças à tecnologia IoT tem de ter em conta a sobrecarga dos recursos dos dispositivos. A segurança e privacidade das redes informáticas e dos sistemas de informação em geral consiste em prestar os seguintes serviços [11]:

- **Privacidade do utilizador** – Aumentar a privacidade e a segurança dos utilizadores em todas as camadas. Garantir que a identificação dos utilizadores não é identificável, nem o rastreio dos seus comportamentos e das ações realizadas no sistema.
- **Controlo de acesso** – Aplicação de um mecanismo de controlo de acesso para proteger os dados contra a utilização indevida e danos de informações privadas de um utilizador por outros. Apenas serão acessíveis os dados que pertencem ao utilizador, garantindo que os dados não são modificados por terceiros, de forma acidental ou intencional.
- **Gestão de identidade** – Garante a identificação de objetos utilizando diferentes técnicas num sistema e controla o acesso ao sistema a utilizadores legítimos.
- **Comunicações de dados** – Garante a comunicação de dados, com autenticação de objetos de comunicação, confidencialidade e integridade dos dados comunicados e a proteção de identidade dos objetos comunicados [14].
- **Resistência aos ataques** – Mecanismo para proteger e lutar contra diferentes ataques ao sistema.
- **Middleware seguro** – Os dados em IoT são recolhidos através da interação entre máquina-a-máquina ou máquina-utilizadores. Por essa razão, a criação de *middleware* é essencial para questões de segurança.

A segurança do sistema e a privacidade dos dados dos utilizadores é uma das principais preocupações da IoT e, para melhorar a sua privacidade existem algumas tecnologias que podem ser usadas, tais como [11]:

- **Encriptação** – A encriptação pode ser feita para a integridade dos dados durante o seu transporte. Sendo a maior parte dos dispositivos IoT operados por bateria, a

utilização de algoritmos de energia de baixo processamento é um dos desafios para tornar a encriptação mais rápida e menos consumidora de energia.

- **‘Virtual Private Network’** – Uma rede que pode ser acessada por parceiros próximos. Os parceiros podem acessar ao sistema, mas tem de ser mantida a integridade e a confidencialidade dos dados.
- **‘Onion routing’** – Uma técnica utilizada para melhoria da privacidade que encripta e funde o tráfego de internet de várias fontes.

2.2 Protocolos IoT

A implementação e eficácia deste sistema depende muito da tecnologia de comunicação sem fios para troca de dados entre os equipamentos e o utilizador. Para um bom funcionamento e para as aplicações IoT funcionarem de forma útil, as informações/dados recolhidos pelos dispositivos ou sensores têm de ser transportados corretamente e, para tal, existe uma série de protocolos.

A necessidade de elaborar novos protocolos, derivada do crescimento exponencial de objetos IoT e *middlewares*, gerou diversos desafios de interoperabilidade e comunicação. Estes protocolos e dispositivos são, de facto, a base de um modelo IoT eficiente e interoperável e, alcançando o compromisso e a compreensão entre eles, a conectividade total é atingida e o intercâmbio de informação pode tornar-se um êxito.

Esta secção, visa analisar as diferentes formas de transmissão de dados da *Internet of Things*, assim como protocolos e padrões de comunicação. Os protocolos e padrões de IoT estão classificados em duas categorias: Protocolos de comunicação, de curto e longo alcance, que cobrem alguns dos protocolos de rede e transporte utilizados em IoT. Protocolos de dados como o MQTT e CoAP são apresentados de maneira introdutória para enquadramento do sistema em estudo.

2.2.1 Protocolos de comunicação IoT

Os protocolos de comunicação IoT são usados para conectar dispositivos pela rede. Usando protocolos de comunicação IoT, é garantida a proteção e a segurança na comunicação de dados entre dispositivos conectados. Esta sub-secção apresenta algumas tecnologias de rede sem fio de curto e longo alcance, que atualmente são usadas em vários sistemas. São tecnologias viáveis, eficientes e adequadas ao uso de IoT, porém, como se pode verificar na figura 2.3, para além das suas capacidades todas têm vantagens e desvantagens que serão analisadas.

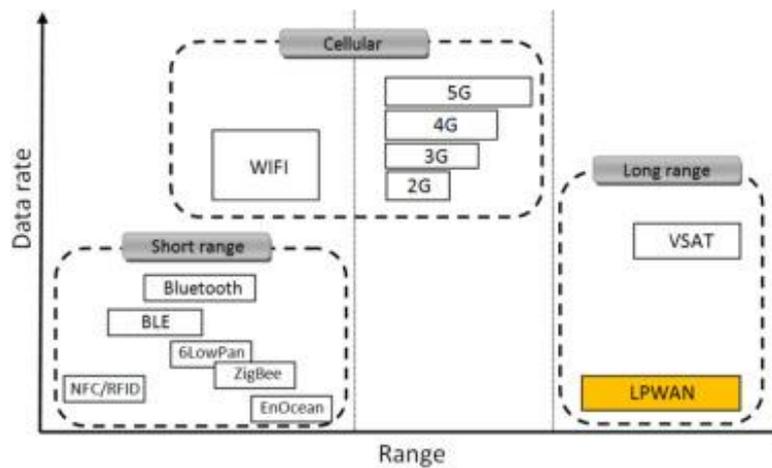


Figura 2.3: Gráfico comparativo das comunicações (retirada de [24]).

2.2.1.1 Redes de curto-alcance

A presente subsecção evidência algumas tecnologias de rede sem fio de curto alcance que atualmente são usadas para suportar aplicações de comunicação *machine-to-machine* de baixa potência. São tecnologias viáveis e adequadas para utilizadores IoT, no entanto, podem não suportar aplicações do âmbito industrial ou outros com grandes dimensões que estejam além das suas capacidades e requisitos.

Bluetooth Low Energy (BLE)

A tecnologia *Bluetooth* é baseada no standard IEEE 802.15.1 para comunicações de área pessoal sem fios de curto alcance em configuração *Master-Slave*, onde um *Master* pode ter até sete *Slaves* ligados numa rede. Este conjunto de dispositivos *Bluetooth* que operam na frequência de 2,4 GHz e compartilham o mesmo canal de comunicação é denominado por *Piconet* [7]. Este protocolo de comunicação é seguro e ótimo para transmissões de curto-alcance, baixo consumo de energia e criado para substituir os fios em dispositivos móveis. Contudo, uma das desvantagens é a restrição de uma comunicação individual apenas entre dois dispositivos [15].

De modo a tornar concretizável a implementação das redes IoT, foi criada uma arquitetura *Bluetooth Low Energy* (BLE) versão 4.0, sendo uma versão de baixo consumo de energia do protocolo *Bluetooth*. Para além das múltiplas aplicações para redes de sensores, uma das mais interessantes é a capacidade que o BLE tem em detetar a proximidade de um *beacon* BLE. Ou seja, quando se instala um *beacon* num objeto e a função de localização espacial é ativada para desencadear um evento, consegue-se detetar a localização em tempo real e o estado de funcionamento do objeto [16].

Atualmente, a versão mais recente é o *Bluetooth* 5.0, com avanços importantes para a tecnologia, com melhoramentos no alcance, nas taxas de transmissão e na segurança. Este novo standard irá permitir uma gama mais ampla de cenários IoT [17].

Wi-Fi HaLow

O Wi-Fi é uma rede local sem fio (WLAN) que utiliza o padrão IEEE 802.11 para comunicações de alta largura de banda entre dispositivos localizados num curto alcance. Inicialmente, a rede apenas operava nas bandas de espectro ISM de 2.4 GHz e 5 GHz, mas tem vindo a adaptar-se à nova realidade tecnológica, criando o Wi-Fi HaLow com o padrão IEEE 802.11ah com o intuito de responder às necessidades e requisitos da tecnologia IoT [15].

Este novo padrão opera com frequências próximas de 900 MHz com melhor propagação do sinal através de obstáculos, com uma taxa de transmissão e alcance máximos de 78 Mbps e 1000 m, respetivamente. O Wi-Fi HaLow permite a conectividade de baixa energia necessária para aplicações IoT, incluindo redes de sensores.

ZigBee

O protocolo de comunicação de radiofrequência ZigBee foi criado pela ZigBee Alliance com base na norma IEEE 802.15.4. Este protocolo tem como características baixo custo, rede sem fios, rede de alcance inferior a 100 m, baixa taxa de transmissão de dados, escalabilidade, confiabilidade e design de protocolo flexível [20].

O ZigBee suporta três topologias diferentes, como mostra a figura 2.4, rede em malha (*mesh networks*), rede em estrela e rede em árvore. A rede em estrela contém um nó central, o controlador da rede e vários periféricos, enquanto que a topologia em árvore é constituída pelo nó principal do controlador da rede e várias ramificações com dois tipos de dispositivos, de funções completas ou de funções reduzidas. A vantagem desta última estrutura é a expansão de área de cobertura. Em contraste com a topologia em estrela, na rede em malha todos os dispositivos podem comunicar entre eles. Uma das vantagens da rede é a capacidade de ajuste para assegurar a transmissão correta dos dados caso um nó falhe [21].

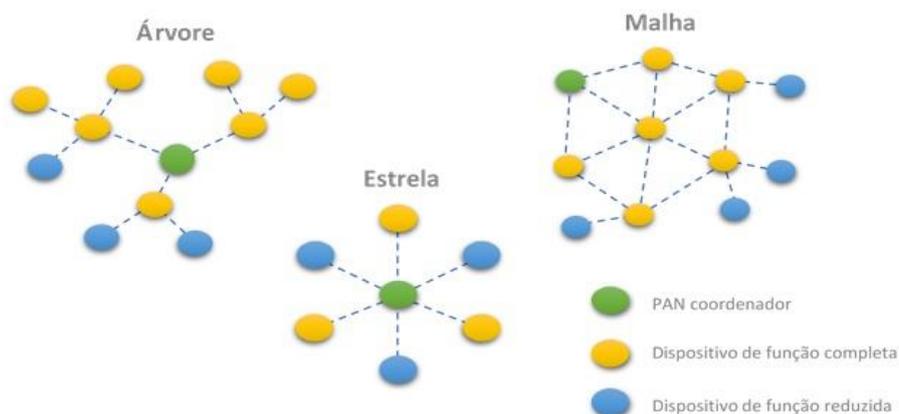


Figura 2.4: Topologias ZigBee (adaptada de [28]).

2.2.1.2 Redes de longo-alcance

Nesta subsecção são abordadas tecnologias de rede sem fio de longo alcance para soluções IoT de baixa potência. Serão analisadas algumas tecnologias LPWAN e outras como redes móveis 4G, que são tecnologias económicas para estabelecer redes privadas com requisitos específicos de qualidade e segurança.

LPWAN

Os requisitos dos aplicativos IoT impulsionaram o surgimento de um tipo de rede sem fio, *low-power wide-area network* (LPWAN).

A LPWAN é uma rede sem fios para comunicações de longo alcance constituída geralmente por quatro camadas: Aplicações, Servidores, *Gateways* e serviços. As principais características de uma LPWAN são o longo alcance, a baixa taxa de dados transmitidos, o baixo custo de dispositivos (sensores ou atuadores), o elevado número de dispositivos que se podem conectar à rede e, por último, o baixo consumo de energia, uma das principais preocupações da IoT. Esta característica deve-se à capacidade de os dispositivos estarem em modo suspenso até ser necessário comunicar com a *gateway*.

LoRa

LoRa é uma tecnologia da Semtech para redes IoT, com uma técnica de modulação de espalhamento espectral derivada da tecnologia existente *Chirp Spread Spectrum* (CSS) [22], que usa pulsos modulados em frequência linear de banda larga cuja frequência aumenta ou diminui com base nas informações codificadas, garantindo desta forma uma forte robustez em ambientes de rádio frequência adversos e resistência a interferências. Adicionalmente, consiste numa comunicação sem fios que transmite pequenas quantidades de dados a longas distâncias em redes regionais, nacionais e globais, com baixo custo de energia e baixo custo de implementação [23]. Estas são ótimas vantagens para um ambiente aeroportuário, que necessita de uma extensa área de cobertura de rede e de um vasto número de dispositivos ligados em constante movimento.

Esta tecnologia é constituída pelo nível físico LoRa e pelo protocolo de comunicação LoRaWAN (*Long Range Wide Area Network*) gerenciado pela LoRa Alliance e que utiliza a camada física. Este protocolo usa o espectro de rádio não licenciado na banda Industrial, Científica e Médica (ISM) sub-GHz fornecendo interoperabilidade contínua entre os dispositivos IoT [24].

A rede LoRa é estabelecida numa topologia em estrela, formada por *End Nodes* LoRa, sensores ou atuadores, ligados pela rede de baixa energia LoRaWAN a uma ou mais *Gateways*. As *Gateways* estabelecem a conexão entre os *End Nodes* e o *Network Server*. Por sua vez, a ligação de uma *gateway* ao *Network Server* é feita com redes de alta largura de banda como Wi-Fi, Ethernet ou redes móveis. Desta forma, as mensagens enviadas pelos dispositivos finais, depois de decodificadas corretamente pelas *Gateways*, são processadas e armazenadas pelo *Network Server*, que as encaminha para visualização na aplicação. No caso dos atuadores, o envio dos dados também é feito no sentido inverso do *uplink*, ou seja, desde o servidor até aos *End*

Nodes, chamado *downlink*. Com a ajuda do esquema da figura 2.5 adaptada de [22], consegue-se entender melhor a arquitetura de uma rede LoRa.

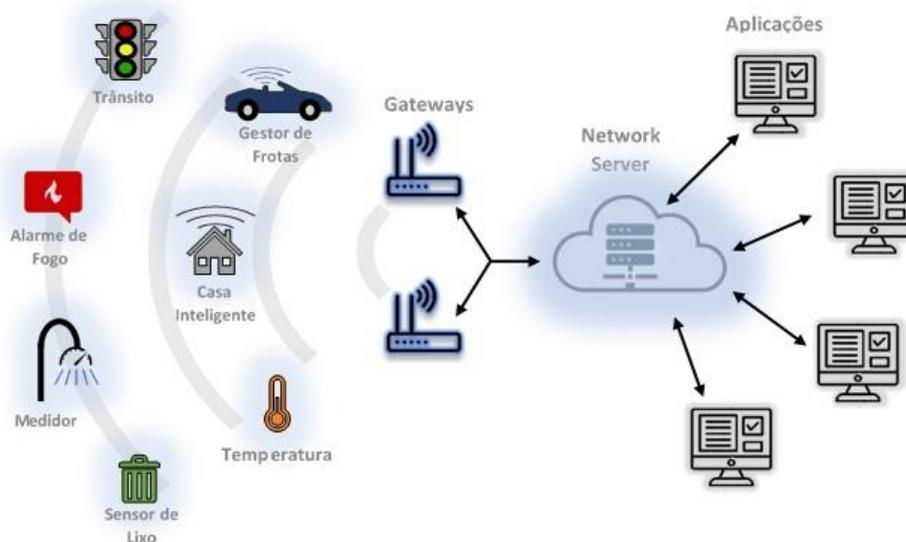


Figura 2.5: Rede LoRa.

Inicialmente, as mensagens transmitidas pelos dispositivos finais são recebidas por todas as *gateways* que estiverem dentro do alcance e, posteriormente, filtradas e enviadas para o servidor correspondente. Quando a rede de dispositivos fica bem estabelecida, as mensagens passam a ser enviadas para a *gateway* mais próxima [25].

Um dos pontos importantes para uma rede instalada num espaço aeroportuário é a sua segurança. A tecnologia LoRa possui um algoritmo de criptografia AES-128, mecanismo de segurança por meio de pares de chaves de 128 bits, garantindo uma proteção de integridade, confidencialidade e uma segura transmissão de dados ponto-a-ponto [23].

A taxa de dados suportada pela LoRa está entre 0.3 Kbps e 50 Kbps, dependendo do fator de difusão e da largura de banda do canal utilizado. No que respeita o comprimento máximo da mensagem, esta comunicação consegue enviar até 243 bytes por mensagem [24].

O protocolo LoRaWAN define três tipos de dispositivos finais através de classes para atender a diferentes requisitos de aplicativos IoT [24]:

Todos os dispositivos têm implementada a classe A, que suporta comunicações bidirecionais entre o dispositivo final e a *gateway*. Após uma transmissão de *uplink* pelo dispositivo final, são abertas duas janelas de recepção de *downlink* distanciadas por um segundo. Este sistema é destinado a aplicações de monitorização, onde os dados enviados pelos *End Nodes* devem ser recolhidos por um servidor. Esta classe é a mais económica, consumindo menos energia.

A classe B, para além de ter as especificações da classe A, os dispositivos abrem janelas de recepção no tempo programado pelo servidor, independentemente do tráfego de *uplink*. Esta classe

destina-se a dispositivos que necessitem de receber ordens de um controlador remoto, como os atuadores.

Por último, a classe C que tem as janelas de recepção continuamente abertas até ao momento de transmissão. O uso desta classe é para dispositivos que não têm restrições de energia.

Concluindo, a classe A é a que se adequa ao sistema que se pretende implementar, apropriada para sensores.

Sigfox

Sigfox foi a primeira tecnologia LPWAN projetada para atender aos requisitos de aplicativos IoT, com longa duração da bateria do dispositivo, baixo custo do dispositivo, baixa taxa de conectividade, alta capacidade de rede e longo alcance, como sensores e aplicações M2M [20]. A rede emprega uma modulação *Ultra Narrow Band* (UNB) e opera na banda 200 KHz disponível para troca de mensagens de rádio pelo ar. Cada mensagem tem uma largura de banda de 100 Hz e é transferida a uma taxa de dados de 100 ou 600 bits por segundo, dependendo da região [24].

Numa conexão em *uplink* as mensagens de rádio emitidas pelos dispositivos conectados são recebidas pelas estações base Sigfox e então transmitidas para a nuvem Sigfox e enviadas para a plataforma da aplicação. O serviço de *downlink* é orientado por dispositivo para minimizar o consumo de energia. Cabe ao dispositivo pedir à rede para obter uma mensagem de *downlink*. O padrão existente para comunicações Sigfox suporta por dia até 140 mensagens de *uplink* e 4 em *downlink*. Uma mensagem de *uplink* tem carga útil até 12 bytes e leva em média 2s para chegar às estações base que monitorizam o espectro em busca de sinais UNB para desmodulação. A tolerância de carga útil em mensagens de *downlink* é de 8 bytes [24].

NB-IoT

NB-IoT é uma tecnologia LPWAN baseada, como o nome indica, na tecnologia *Narrow Band* e é padronizada pelo projeto de parceria de 3ª geração (3GPP) para permitir o suporte para dispositivos IoT. O protocolo de comunicação NB-IoT é baseado no protocolo LTE, com capacidade de reduzir as funcionalidades do protocolo LTE e aperfeiçoando para aplicativos IoT [27]. O NB-IoT, além de fornecer suporte e cobertura para um grande número de dispositivos por célula, também oferece três modos flexíveis de operação [23] [24]:

- **Autónoma:** pode substituir a rede GSM, o que permite a recombinação eficiente da infraestrutura GSM para IoT.
- **De guarda:** utiliza novos blocos de recursos dentro da banda de guarda de uma portadora LTE
- **Banda:** utiliza blocos de recursos dentro de uma portadora LTE. Neste caso todos os canais de comunicação são compartilhados entre LTE e rede NB-IoT.

O NB-IoT concentra-se especificamente na cobertura interna, na longa duração da bateria na alta densidade de conexão e no baixo custo. Embora as redes NB-IoT tenham um maior custo de configuração, não requerem uma *gateway* o que torna a sua execução mais económica.

A taxa de dados é limitada a 200 Kbps para o *downlink* e a 20 Kbps para o *uplink*. O tamanho máximo da carga útil para cada mensagem é 1600 bytes. A tecnologia NB-IoT pode atingir os 10 anos de vida útil de uma bateria ao transmitir em média 200 bytes por dia [24].

4G-LTE

A comunicação móvel da quarta geração (4G), utiliza a tecnologia LTE (*Long Term Evolution*), capaz de fornecer uma conexão muito rápida e segura, com transmissão de dados em bandas ultra largas de 1.4 a 20 MHz [29]. A sua tecnologia foi criada com base no GSM e WCDMA, porém, LTE está focada na troca de dados, enquanto as anteriores gerações davam prioridade ao tráfego de voz.

Esta tecnologia, segundo [29], com interoperabilidade entre diversos padrões de redes, foi desenvolvida para recorrer a determinados requisitos que as outras tecnologias não conseguiam. A rede 4G proporciona uma otimização do uso do espectro, com troca de mensagens em ambiente IP, com uma taxa de transferência de 100 Mbps para *downlink* e *uplink* de 50 Mbps mantendo uma qualidade de serviço (QoS) de ponta a ponta com segurança, independentemente da localização. Para uma utilização eficiente do espectro, LTE incorpora modulações *Orthogonal Frequency Division Multiple Access* (OFDMA) em *downlink* e *Single Carrier Frequency Division Multiple Access* (SC-FDMA) em *uplink*.

Pela tabela 2.1, verifica-se que todas as tecnologias têm características interessantes dependendo do seu objetivo. As principais vantagens da LoRa em comparação com as outras redes é o baixo custo de implementação, com capacidade de longo alcance e troca de mensagens ilimitadas por dia.

Tabela 2.1: Comparação das tecnologias [20],[22],[23],[25].

Tecnologias	BLE	Wi-Fi Halow	ZigBee	LoRa	SigFox	NB-IoT	4G-LTE
Alcance	15 a 30 m	1000 m	100 m	Rural: 20 km Urbano: 5 km	Rural: 40 km Urbano: 10 km	Rural: 10 km Urbano: 1 km	Rural: 20 km Urbano: 2 km
Duração da bateria	Semanas	Meses	Anos	Anos	Anos	Anos	Semanas/ meses
Topologia	Estrela	Estrela	Malha, estrela e árvore	Estrela	Estrela	Estrela	Estrela
Taxa de transmissão	1 Mbps	78 Mbps	250 Kbps	50 Kbps	100 ou 600 bps	200 Kbps <i>downlink</i> 20 Kbps <i>uplink</i>	100 Mbps <i>downlink</i> 50 Kbps <i>uplink</i>
Tamanho máximo de dados	27 bytes	----	104 bytes	243 bytes	12 bytes <i>uplink</i> 8 bytes <i>downlink</i>	1600 bytes	----
Frequência	2.4 GHz	900 MHz	2.4 GHz	868 MHz na Europa	868 MHz na Europa	900 MHz	800; 1800; 2600 MHz
Largura de banda	1 MHz	26 MHz	2 MHz	125 KHz	100 Hz	200 KHz	1.4; 3; 5; 10; 15; 20 MHz
Permite rede privada	Sim	Sim	Sim	Sim	Não	Não	Sim
Autenticação e criptografia	AES-128	AES	AES	AES-128	Não	Sim	Sim
Máximo de mensagens por dia	Ilimitado	Ilimitado	----	Ilimitado	140 <i>uplink</i> 4 <i>downlink</i>	Ilimitado	Ilimitado
Custo de implementação	Baixo	Baixo	Médio	Baixo	Médio	Alto	Alto

2.2.2 Protocolos de dados IoT

Os protocolos de dados IoT são usados para comunicação M2M de baixa potência e adequados para *hardwares* com pouca capacidade de memória e microcontroladores com pouca capacidade de processamento. Dos quais, o *Message Queue Telemetry Transport* (MQTT), o *Constrained Application Protocol* (CoAP) e o *Advanced Message Queuing Protocol* (AMQP) frequentemente utilizados.

MQTT

O MQTT - *Message Queuing Telemetry Transport* é um protocolo simples, económico e de fácil implementação. Estas são características adequadas a dispositivos IoT, como sensores, detetores, aplicativos para mensagens de texto, que requerem pouca energia, processamento e memória do sistema.

Este protocolo utiliza um modelo *publish/subscribe*, baseado num servidor, *broker*, que é o responsável por enviar e receber mensagens, diferenciando as mensagens por tópicos. Os dispositivos *Publisher* enviam mensagens para o *Broker* com o devido assunto. Os *Subscribers* para poderem receber mensagens inscrevem inicialmente o tópico com interesse, para o *Broker* ter informação para onde enviar as mensagens [30].

É um protocolo executado em TCP (*Transmission Control Protocol*) que oferece suporte à troca de mensagens por meio de redes sem fio e usa como segurança o *Secure Sockets Layer* (SSL) ou o *Transport Layer Security* (TLS) [31].

As principais características do MQTT levaram à aplicação em soluções de baixo consumo de energia, pouca largura de banda ou conexão intermitente. Assim sendo, foi criada uma versão MQTT-SN, com foco em redes de sensores. O MQTT-SN foi desenvolvido para permitir o uso do MQTT em redes que não sejam TCP/IP, tais como 6LoWPAN e Zigbee, protocolos baseados no padrão IEEE 802.15.4 [31]. Na figura 2.6, é apresentada uma noção de como funciona este protocolo.

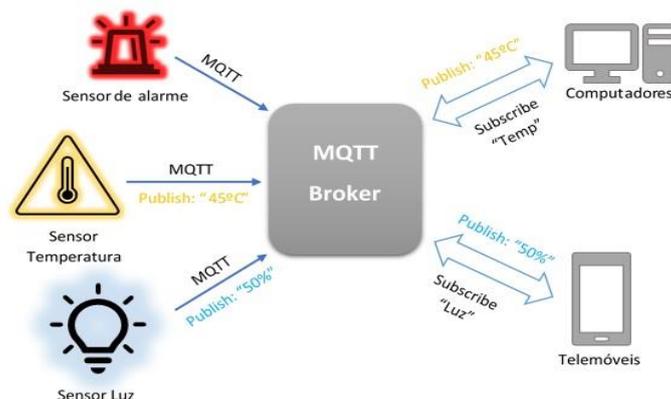


Figura 2.6: Protocolo MQTT (adaptada de [33]).

CoAP

O *Constrained Application Protocol* (CoAP) é um protocolo otimizado para dispositivos com potência e capacidade de processamento limitados e é geralmente aplicado em ambientes IoT. O protocolo é projetado para aplicações máquina a máquina, em dispositivos numa rede restrita e em nós gerais na Internet com recursos limitados, como por exemplo em nós de rede de sensores sem fio.

O CoAP foi desenvolvido para ser facilmente traduzido para HTTP, simplificando a integração com a *Web*. Além disso, contém requisitos importantes no que diz respeito à IoT e à comunicação M2M, como suporte *multicast*, sobrecarga muito baixa e simplicidade.

Baseado no protocolo de transporte UDP (*User Datagram Protocol*) com segurança de comunicação DTLS e arquitetura de *software* RESTful, o CoAP utiliza comunicação no modelo *Request/Response*, como se pode verificar na figura 2.7 adaptada de [34]. Neste modelo, existem dispositivos que requisitam informações (clientes) e dispositivos que respondem a esses pedidos, atuam como servidores. A comunicação acontece de forma assíncrona.

Além disso, o CoAP implementa a sua própria forma de confiabilidade para entrega de mensagens e um mecanismo de retransmissão. Há quatro tipos de mensagens: CON (Confirmável), NON (Não-Confirmável), ACK (Reconhecimento) e RESET.

Para diminuir ainda mais o tráfego de rede, o protocolo permite o envio de mensagens em *piggyback* utilizando *MessageID* e *Token* para garantir o envio correto da informação.

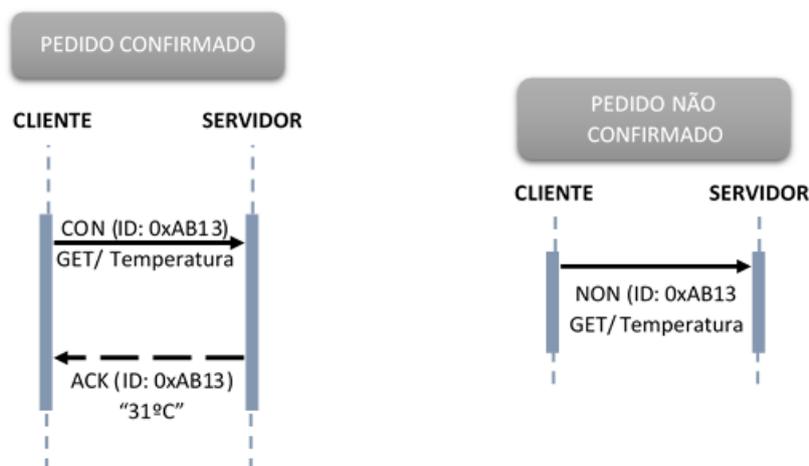


Figura 2.7: Método de funcionamento CoAP.

AMQP

O AMQP (*Advanced Message Queuing Protocol*) é um protocolo de padrão aberto para ambiente de *middleware*, baseado em mensagens que fornecem roteamento e enfileiramento. Para além destes dois recursos, este protocolo foi projetado tendo em consideração a segurança, a orientação de mensagens, a confiabilidade e a interoperabilidade [31].

No protocolo AMQP, os aplicativos enviam mensagens para os *brokers* AMQP, que por sua vez distribuem para os consumidores, aplicativos que processam mensagens.

Este protocolo é usado para uma conexão ponto a ponto segura e suporta a troca contínua de dados entre os dispositivos conectados e a *cloud*. Segundo [32], a estrutura do AMQP consiste em três componentes separados: *Exchange*, *Message Queue* e *Binding*. Todos os componentes garantem uma troca e um armazenamento seguro e bem-sucedido de mensagens. Sempre que uma mensagem é enviada por um servidor, o protocolo rastreia a mensagem até que a mensagem chegue ao destino pretendido sem falhas.

2.3 Sensores de geolocalização/Módulo

Para avaliar os impactos e os benefícios que esta solução poderá oferecer às operações de *handling*, a empresa analisou e testou uma solução de geolocalização com módulos Ruptela.

A Ruptela é uma empresa de soluções tecnológicas para gestão de frotas e rastreamento por GPS, tendo uma variadíssima gama de *hardware* adequado a todas as configurações pedidas pelos clientes. A Groundforce para implementar um sistema de geolocalização, primeiramente efetuou testes com um módulo **FM-Eco4 light+ S** para equipamentos motorizados, como ilustrado na figura 2.8.



Figura 2.8: FM-Eco4 light+ S.

FM-Eco4 light+ S é um módulo compacto projetado para o rastreamento de veículos por GPS. Este dispositivo contém uma antena interna GPS e GSM, com entradas analógicas e digitais configuráveis para a instalação do dispositivo ao equipamento com uma tensão de entrada de 6V DC, memória interna de 4 MB e uma avaliação IP67.

As suas principais características são a recolha de dados em tempo real, a monitorização do equipamento e do condutor, recolha de informação sobre o equipamento e as possíveis configurações ou adições de sensores.

Esta solução de geolocalização abrange muitas tecnologias e comunicações disponíveis, incumbindo à empresa de desenvolvimento de *hardware* utilizar a que considera mais adequada à sua aplicação. Para a Ruptela, a comunicação GSM e GPS oferece um bom resultado para o rastreio dos equipamentos. De um modo mais específico, para a comunicação GSM foi instalado um módulo Quectel MC60, idêntico ao da figura 2.9, com uma antena interna GSM *Quad-band* que funciona nas bandas de frequência 850, 900, 1800 e 1900 MHz. O mesmo módulo Quectel MC60 integra a tecnologia GNSS - *Global Navigation Satellite System* com antena interna para navegação por GPS ou Galileo com frequência 1575,42 MHz e GLONASS na frequência 1601,71 MHz.



Figura 2.9: Módulo MC60.

2.4 Trabalhos Relacionados

Ao longo desta secção, abordam-se projetos e trabalhos relacionados com o tema desta dissertação. Todos os projetos são de empresas estrangeiras e nenhuma tem implementada esta solução em Portugal. Com o aumento da aviação em espaço nacional, torna-se necessário aumentar o número de equipamentos de suporte, o que dificulta o controlo, manutenção e a gestão de todos os GSE.

Em reunião com algumas das empresas estrangeiras que estão dentro da área da aviação, foram recolhidos dados sobre as tecnologias, sensores e comunicações usadas por cada uma. As informações obtidas são muito importantes para o estudo e desenvolvimento do projeto.

Primeiramente, será apresentada uma empresa de aluguer de GSE que oferece soluções inteligentes de gestão e monitoramento dos equipamentos através do módulo de geolocalização instalados em todos os GSE. De seguida, serão analisadas duas empresas com principais soluções tecnológicas na área da aviação, as quais criaram um sistema inovador de monitorização e gestão de *Ground Support Equipment*. Para conclusão do capítulo, será feita uma análise resumida de todos os projetos discutidos e da tecnologia mais adequada para o desenvolvimento do projeto.

HiSERV

HiSERV, empresa de aluguer de equipamentos de suporte a aeronaves, em parceria com a Trackunit (fornecedora dos dispositivos de geolocalização), criou uma solução de geolocalização para toda a frota da empresa. Desta forma, a HiSERV pode monitorizar e gerir os equipamentos alugados a todas as empresas de *handling* e verificar quais são os equipamentos que necessitam de manutenção, fornecendo dados confiáveis para otimizar a utilização e os custos.

De acordo com a HiSERV, o Trackunit Raw é um dispositivo robusto e versátil, que fornece uma conectividade estável, segura e confiável. Este torna a sua frota digital, visível em tempo real, comunicando e fornecendo dados que são importantes para a gestão dos seus equipamentos.

Trackunit Raw é um dispositivo de geolocalização para GSE motorizados com ligação à bateria do equipamento. Trackunit Raw fornece dados detalhados do equipamento por meio da interface de barramento CAN. Algumas das comunicações utilizadas, LTE-M, NB-IoT, 2G/3G e GSM. Para visualização dos equipamentos em tempo real utiliza, GPS, GLONASS e Galileo. Esta solução também utiliza comunicação *Bluetooth* 5.0 (BLE) em situações de curto alcance.

Para equipamentos não motorizados, a Trackunit desenvolveu o Trackunit Spot, alimentado por bateria, *firmware* inteligente para ativar o Trackunit Spot apenas quando necessário, para menores consumos de energia e rede 2G/3G. Adicionalmente, tem a possibilidade de incorporar um leitor de cartões RFID Trackunit para acesso ao equipamento a utilizadores autorizados.

TrackIT

EnTrackGSE é um sistema desenvolvido pela TrackIT, empresa de soluções tecnológicas direcionada para o setor da aviação. Este sistema de gestão de equipamentos pode ser implementado em material de suporte aéreo motorizados e não motorizados.

O controlo de GSE é feito principalmente por meio de GPS, o método mais fácil com uma precisão de 2.5 m e CEP (*circular error probable*) de 50%. No entanto, também é possível uma leitura de localização altamente precisa (menos de 1 m) com sistemas baseados em DGPS. Para isso, é instalado um dispositivo de localização no GSE motorizado. No entanto, a aplicação deve ter a capacidade de se vincular a um componente de mapa. Idealmente, o mapa deve ser específico para o aeroporto relevante e os pontos de interesse devem ser claramente identificados juntamente com a infraestrutura.

Para os equipamentos não motorizados, a localização pode ser obtida por vários métodos. Tal inclui a instalação de *beacons* de localização baseados em BLE no GSE não motorizado que transmite sinais para recetores BLE. Esses recetores BLE podem ser posicionados no GSE motorizado ou em pontos estratégicos do aeroporto.

Além da geolocalização e rastreio, os dispositivos também têm recursos tecnológicos avançados com um *firmware* robusto para lidar com as várias combinações lógicas.

Esses recursos permitem obter informações do motor e o *status* operacional do GSE, os níveis de combustível, óleo, entre outros. Além disso, estes dispositivos também podem ser conectados às portas CANBUS ou OBD no GSE, o que permite a captura de vários conjuntos de dados como horas do equipamento, quilômetros ou consumos de combustível. O sistema também permite a monitorização da velocidade e práticas seguras de condução. Os dispositivos contêm acelerómetros que permitem a deteção de forças G e ajudam a identificar acelerações e travagens bruscas.

Através de *geofencing* é possível definir zonas e limites geográficos, garantindo uma operação segura e protegida.

Relativamente à comunicação, existem vários modos de comunicação de dados possíveis, estes incluem Wi-Fi, GPRS e *Bluetooth*. Geralmente, a conectividade Wi-Fi no espaço aeroportuário tem pontos cegos e, portanto, o GPRS é mais comum para esta prática. Para gerenciar espaços cobertos, são utilizados *beacons* BLE para comunicação.

Além dos modos de comunicação mencionados acima, os protocolos de comunicação LoRa e Sigfox também são utilizados, dependendo do aeroporto e do país em causa.

ADVEEZ

A ADVEEZ, é uma empresa de produtos eletrónicos com sede em França e U.S., com uma rede de equipamentos monitorizados de cerca de 7 mil e com sistema instalado em mais de 50 aeroportos em todo o mundo.

A ADVEEZ está envolvida na conceção de todas as fases de uma solução tecnológica, desde o design do *hardware*, o desenvolvimento e arquitetura do *software*, como a plataforma *Web*, até à visualização das informações, administrando todo o processo para apoio aos seus clientes. Estes processos incluem a recolha dos dados através de sensores e, o tratamento e gestão de dados recolhidos.

FAMA é um projeto usado para equipamentos motorizados que recolhe dados técnicos dos veículos, bem como a sua posição. Este dispositivo é conectado à bateria do equipamento e não contém bateria própria, para uma maior durabilidade do sistema. Algumas das tecnologias consideradas permitem a localização exata, através do posicionamento baseado em GPS (*global positioning system*). Além disso, obtém-se toda a informação do veículo através de CANBUS ou entradas analógicas e digitais, como controlo da velocidade e de impactos. Todos estes dados recolhidos pelos sensores são enviados para a nuvem através de 4G/LTE e GSM maioritariamente. Em zonas sem cobertura de sinal, são implementadas antenas LoRa no espaço aeroportuário e usam LoRaWAN para enviar a informação para a nuvem. Para segurança e controlo de acesso aos equipamentos, a ADVEEZ criou uma tecnologia de mãos livres, que usa tecnologia RFID com *tags* e leitor de cartões.

Para equipamentos não motorizados, a empresa arquitetou o OTM (*Outdoor Tracking Module*), com bateria interna de longa duração, de 3 a 5 anos, que pode ser substituída reduzindo o custo operacional e minimizando o impacto ambiental. Neste módulo é apenas usada a tecnologia LoRa, devido ao baixo consumo de energia que proporciona.

Estes dois produtos criados pela ADVEEZ estão ligados a um *software* adequado para processar os dados e gerar informação que possa ser utilizada para fornecer avisos em tempo real aos utilizadores. A escolha do *software* pode ser alternativa, mas é proposto ao cliente o *software* Localeez, plataforma *Web* criada pela empresa para gerir os equipamentos.

A aplicação Localeez pode ser acedida por um número ilimitado de utilizadores, disponível em IOS e Android com inúmeras características. Toda a informação sobre os equipamentos é disponibilizada, como acidentes, lembretes de manutenção, reportes, níveis de baterias, trajetórias e *status* de GSE.

Em espaços fechados e de curtas distâncias, como o terminal de bagagem, a ADVEEZ utiliza tecnologia *Bluetooth* 5.0 (BLE) e UWB para não existir perdas de sinal. Esta tecnologia necessita de instalação de *beacons* BLE em pontos estratégicos para captar toda a área monitorizada.

Síntese comparativa das soluções existentes

Como se pode verificar, todos estes projetos apresentam semelhanças, tanto no modelo que implementam, como na utilização de tecnologias sem fios para estabelecer a comunicação entre os diferentes elementos dos sistemas, tendo cada um as suas vantagens e desvantagens.

Quanto às tecnologias de comunicação, a tecnologia que obteve mais destaque foi a LoRa. Esta é a tecnologia que mais benefícios traz a um sistema IoT, com o seu baixo consumo de bateria. No entanto, nenhum destes projetos estudados disponibiliza informação específica sobre o *hardware* implementado nas suas soluções.

A LoRa já foi testada em muitos países em evolução tecnológica, como a Holanda e a Coreia, que instalaram uma rede que cobre todo o país [26]. A rede LoRa é usada para medidores de água, gás, portas, rastreio de tráfego, aparelhos inteligentes com muitas finalidades tanto na saúde como em aeroportos inteligentes. É o caso do novo aeroporto de Istambul, onde a Skysens criou uma infraestrutura de rede IoT, que cobre todos os locais internos e externos com implementação de 10 mil sensores LoRa. O aeroporto aproveita os sensores de geolocalização para visualizar a localização dos veículos, pessoas e bagagem.

Contudo, para o uso da tecnologia LoRa, em países sem cobertura, será necessário implementar *gateways* dentro do espaço aeroportuário, o que implica autorizações por parte do aeroporto. Porém, sendo esta tecnologia já utilizada em aeroportos no espaço europeu, como França, Holanda e Bélgica, tratar-se-á de uma tecnologia segura e que não causa interferências com as aeronaves.

Por ter esse inconveniente, quando não há autorização ou cobertura no país, a tecnologia LoRa é substituída pela rede 2G/3G. Atualmente, para certas empresas a utilização do 2G/3G já foi descartada pela tecnologia 4G/LTE, com mais velocidade, capacidade e uma segurança melhor que as antigas gerações.

O *Bluetooth* 5.0, mais especificamente *Bluetooth Low Energy*, é uma tecnologia bastante utilizada para este sistema, embora tenha algumas restrições de uso, derivadas do seu fraco

alcance. No entanto, é uma tecnologia utilizada em espaços onde as outras tecnologias não têm cobertura de rede, como espaços fechados.

Algumas empresas para enriquecer as suas soluções de *tracking* de GSE, aplicaram a tecnologia RFID para controlo de utilização dos equipamentos.

Apesar de existirem muitas soluções no mercado, a ADVEEZ apresenta um módulo robusto, adaptável a vários requisitos e com tecnologia avançada de última geração comparado com algumas empresas do mesmo setor.

Com base nos trabalhos existentes e nas tecnologias IoT, este projeto apresentará uma rede IoT cujos dispositivos são comunicados através de um protocolo LoRa.

Sistema Desenvolvido

Neste capítulo é apresentado o sistema desenvolvido para geolocalização e monitorização dos equipamentos de *handling*. A primeira secção descreve os requisitos considerados para o desenvolvimento do sistema, bem como a comunicação entre o módulo GSE e o módulo Servidor. Posteriormente, na secção 3.2 são expostos os componentes e a tecnologia utilizada na elaboração do módulo GSE. As duas secções finais apresentam o módulo Servidor, com todos os seus componentes e, por último, a interface gráfica de monitorização e visualização dos equipamentos.

3.1 Introdução ao Sistema

No sistema proposto nesta dissertação foram analisados e estipulados determinados requisitos para o correto funcionamento da monitorização dos equipamentos:

- O módulo GSE tem de ser autónomo, capaz de receber informação dos satélites, enviar automaticamente mensagens com informação lida pelos sensores e com baixo consumo de energia;
- Um módulo robusto, com resistência ao choque e às condições meteorológicas;
- Comunicação eficaz, sem perdas de sinal assinaláveis numa zona urbana e sujeita a muitas interferências;
- A base de dados necessita de possuir memória suficiente para comportar todos os dados recebidos e fornecê-los corretamente à aplicação de monitorização em execução;
- A aplicação de monitorização necessita de apresentar, num mapa, os percursos efetuados, bem como outras informações sobre os equipamentos.

A fim de clarificar o sistema de geolocalização desenvolvido, são apresentados na figura 3.1 os processos que possibilitam a monitorização e gestão de equipamentos na interface gráfica.

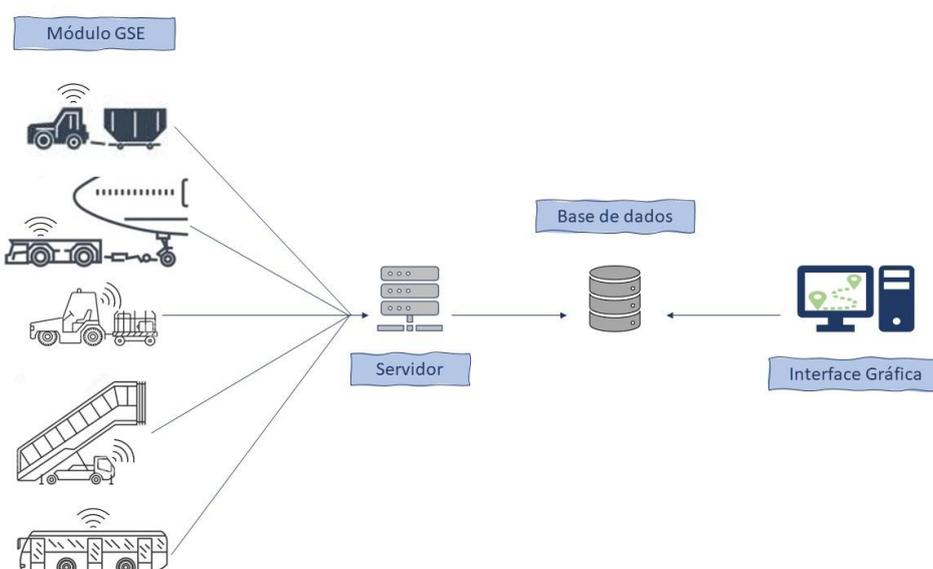


Figura 3.1: Arquitetura do Sistema de Geolocalização.

A ideia principal é desenvolver um sistema com capacidade de localização em tempo real, através dos dados GPS e do sensor acelerómetro para, posteriormente, armazenar em base de dados e o utilizador poder visualizar a informação numa interface gráfica.

Nesta perspetiva, é necessário definir a estrutura do módulo GSE que, para além da tecnologia, é constituído por vários componentes. Um *End Node* é composto por sensores, um microcontrolador para processar os dados dos sensores e uma antena para transmitir esses dados ao servidor. Deste modo, o sistema terá de realizar um conjunto de tarefas, sendo de destacar: a capacidade de iniciar ou parar de acordo com a ignição do equipamento, comunicar automaticamente com o servidor e ler sinais GPS através da comunicação com satélites. Adicionalmente, este sistema deverá ter a capacidade de instalar novos sensores e/ou atuadores, consoante a informação que o utilizador pretender adquirir.

A escolha da comunicação usada na interação dos módulos é essencial para que o sistema funcione corretamente no ambiente em que estiver inserido e para que os dados sejam enviados sem falhas ou interrupções para o servidor. Os dados recebidos são guardados em base de dados para, posteriormente, visualização na interface gráfica ou caso se pretenda aceder ao historial dos equipamentos.

O utilizador somente na parte final do sistema visualiza a informação anteriormente captada e processada, em mapa, podendo retirar as suas conclusões, consoante os dados e as localizações dos equipamentos. O sistema desenvolvido permite ao utilizador gerir com eficiência o uso dos equipamentos no local de trabalho, através da visualização em tempo real dos equipamentos, otimizando recursos ao alocar os operadores ao GSE mais próximo.

Como exemplificado na figura 3.2, a comunicação entre os dois módulos inicia-se com o envio de dados através do módulo GSE para o servidor. Este, por sua vez está sempre à espera de uma mensagem que, após receber, grava numa base de dados. A interface obtém toda a

informação que necessita da base de dados, para expor num mapa. Este último processo ocorre apenas quando o equipamento tem rotas concluídas no seu histórico ou termina a rota atual. Quando estes parâmetros são cumpridos, o utilizador pode visualizar os dados que o módulo consegue captar do equipamento monitorizado.

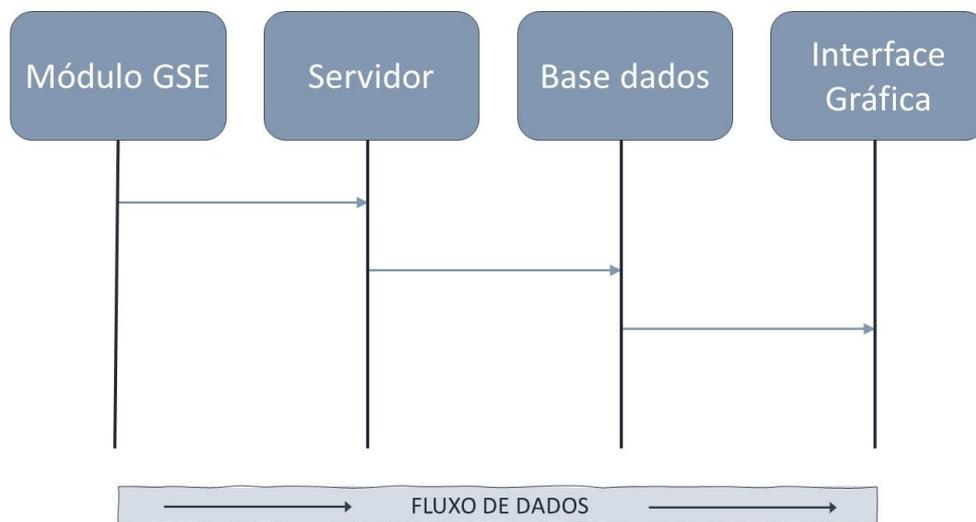


Figura 3.2: Diagrama de sequência do Sistema.

Após uma análise sobre tecnologias de comunicação para suportar a interação entre os módulos GSE e o servidor, foi ponderada a utilização de comunicação GSM, visto ser uma tecnologia bastante desenvolvida e que melhor se adequa num ambiente com interferências de muitas outras comunicações. Contudo, esta tecnologia trazia custos elevadíssimos para a empresa e foi desconsiderada.

Posteriormente, analisou-se a tecnologia LoRa, que já é utilizada em aeroportos para suportar sistemas de IoT, como por exemplo, o aeroporto de Istambul. Esta permite a transmissão de dados a longas distâncias e com baixo consumo de energia, utilizando pulsos modulados em frequência linear de banda larga, para garantir resistência a interferências, como analisado anteriormente. Visto ser uma tecnologia viável, de fácil implementação e com custos dentro dos parâmetros aceitáveis, optou-se por utilizar esta tecnologia.

3.2 Módulo GSE

De forma a validar o sistema apresentado, foi desenvolvido um protótipo de um módulo GSE e um módulo servidor, que garantissem os requisitos mínimos especificados. Nas subsecções seguintes será apresentado o funcionamento do módulo GSE, assim como a sua composição *hardware* e *software*. Numa primeira instância, analisam-se os requisitos e o formato necessários para a composição da mensagem enviada para o servidor. De seguida, na subsecção 3.2.3 são apresentados os diferentes componentes integrados no protótipo final do módulo de

geolocalização. Seguidamente, a última subsecção descreve os métodos utilizados para a implementação do *software*.

3.2.1 Descrição do módulo

Na construção do sistema de monitorização de GSE, foram analisadas algumas tecnologias-chave dos componentes integrantes do protótipo relativo ao módulo GSE. Apesar da escolha dos componentes para o protótipo ser limitada, os seguintes requisitos são os pretendidos num sistema de geolocalização:

- **Eficiência** do sistema;
- **Longo-Alcance** na comunicação entre os módulos;
- **Autonomia** do módulo GSE;
- **Consumo** mínimo de energia;
- **Localização** em tempo real;
- **Integração** de vários sensores ao módulo GSE;
- **Sistema aberto** para que os dados possam ser integrados nos próprios sistemas de gestão da empresa;
- **Custos** mínimos de implementação e manutenção.

Para o módulo GSE foi usado um microcontrolador, um *shield* LoRa/GPS, uma antena GPS e um sensor acelerómetro para detetar a ignição do equipamento. Conforme se pode verificar na figura 3.3, (a) ilustra os componentes utilizados no protótipo do módulo GSE e (b) o protótipo final com caixas elaboradas em 3D para instalação num equipamento.

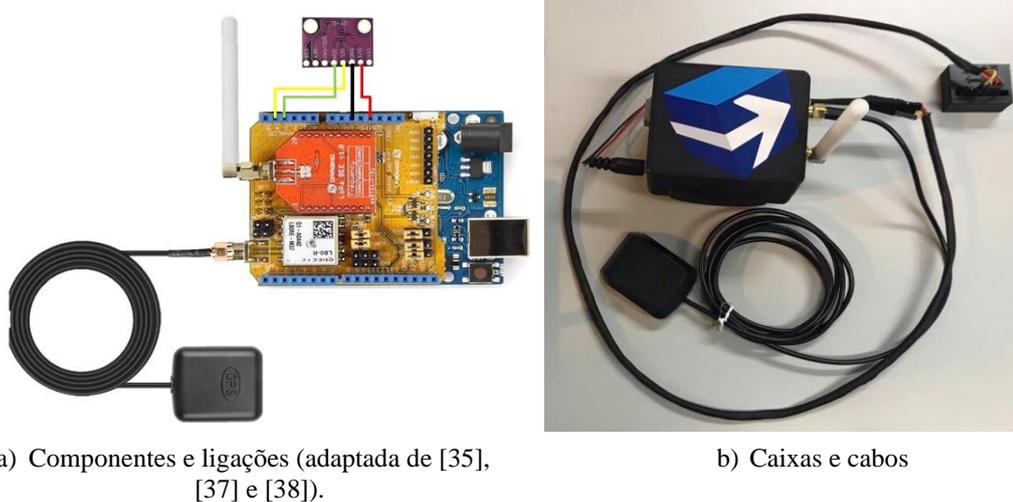


Figura 3.3: Protótipo do módulo GSE.

O comportamento do módulo GSE pode ser mais facilmente compreendido com o auxílio do fluxograma apresentado na figura 3.4. O módulo inicializa-se com a seleção da frequência para 915 MHz e, de seguida, verifica a ligação e o correto funcionamento do acelerómetro. Após a configuração do módulo ser bem sucedida, o programa entra em ciclo infinito. Neste ciclo, o

programa está constantemente a pedir leituras ao acelerómetro que, através de intervalos de calibração, é definido se o equipamento está a trabalhar ou desligado. A leitura do sensor, para ambos os casos, é verificada dez vezes consecutivas, de forma a que o módulo não apresente valores incorretos, por choques ou vibrações erróneas. Caso se verifique que o equipamento está ligado, o módulo ativa o GPS e envia uma mensagem ao servidor com a informação da localização e das horas atualizadas. No caso de o equipamento estar desligado, o módulo apenas verifica as horas e envia a mensagem com a última localização obtida. Após o envio da mensagem, o módulo conclui um ciclo e fica apto para repetir o procedimento novamente.

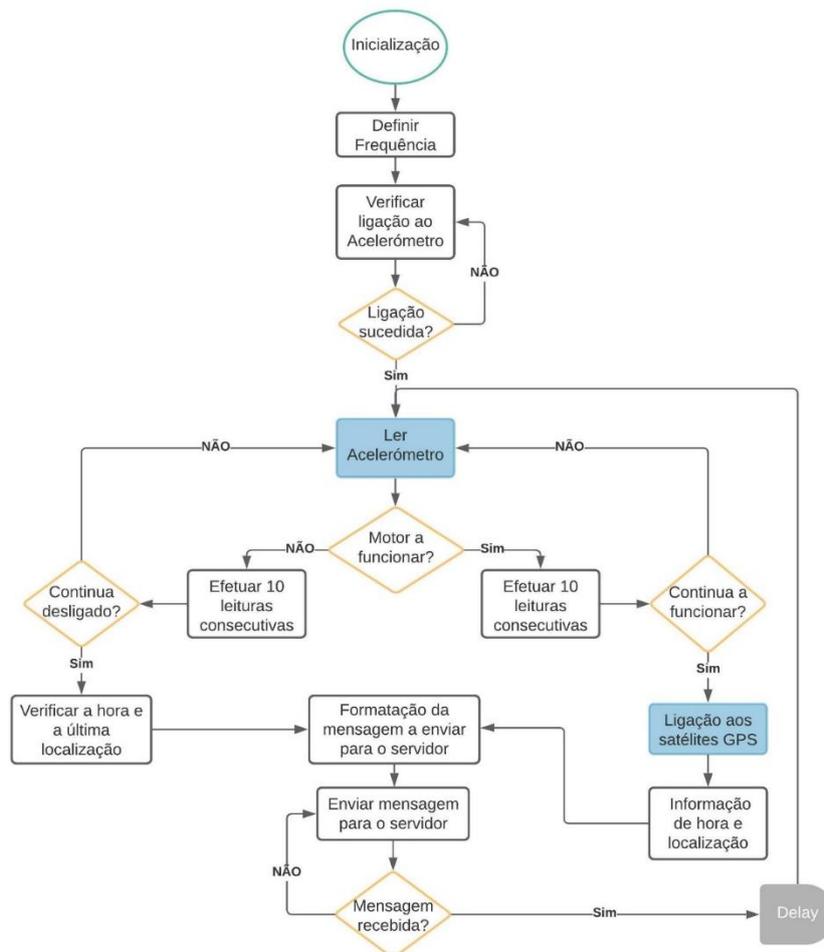


Figura 3.4: Fluxograma alusivo ao funcionamento do módulo GSE.

O módulo beneficia de um conjunto de funcionalidades implementadas para disponibilizar ao utilizador a informação pretendida. Após a instalação completa de todos os componentes, iniciou-se a fase de recolha e tratamento dos dados. Tendo em consideração que este sistema é constituído por dois módulos, a integração e conexão requer que estes sejam configurados antes de se proceder ao tratamento dos dados obtidos. Os módulos foram configurados recorrendo ao ambiente de programação Arduino IDE, um ambiente de desenvolvimento acessível para microcontroladores. No caso do módulo GSE, o código do Arduino UNO utiliza as bibliotecas *MPU9250*, *SoftwareSerial*, *TinyGPS* e *RH_RF95*, conforme se pode verificar no Anexo A.

3.2.2 Formato da Mensagem

A mensagem, em formato *String*, enviada pelo módulo GSE é estruturada na seguinte ordem da figura 3.5.

Rota_ID	State	GSE_ID	Message_ID	Latitude	Longitude	Time
---------	-------	--------	------------	----------	-----------	------

Figura 3.5: Formato da mensagem enviada pelo módulo GSE.

Inicialmente, o primeiro parâmetro é o número do percurso que o equipamento efetuou. Como segundo parâmetro, o estado da ignição, ON ou OFF. Em terceiro lugar é indicada a identificação da matrícula e, de seguida, são apresentadas as coordenadas de localização do GSE, latitude e longitude. Por último, é indicada a hora em que é captada a localização e enviada a mensagem para o servidor. Como exemplo, é apresentada a seguinte mensagem com todos os parâmetros exigidos separados por vírgulas:

14,ON,70087,92,38.764465,-9.133225,17:23:52

3.2.3 Hardware

Arduino UNO

Conforme já descrito, para captar e enviar os dados da localização em tempo real, é fundamental utilizar um processador suficientemente rápido. Atualmente, existe no mercado uma gama variada de placas de desenvolvimento, que proporciona uma escolha adequada e garante todos os requisitos exigidos.

Um fator importante e decisivo para o sucesso de um projeto IoT, é o dispositivo ter baixo consumo de energia. O microcontrolador tem um papel central no processo de integração dos diversos componentes, tendo a escolha recaído no Arduino Uno (ver figura 3.6). Esta deve-se à simplicidade de utilização, ao baixo consumo de energia, à possibilidade de ligar um *shield* loRa/GPS, à documentação disponível, como aplicações e bibliotecas e, também decisivo para um projeto de grande dimensão, o custo unitário do Arduino.



Figura 3.6: Ilustração de um Arduino UNO (retirada de [35]).

Arduino é uma plataforma *open-source*, baseada em *hardware* e *software* fácil de usar, flexível e suportada por uma comunidade de desenvolvimento de bibliotecas de funções, que permitem aos programadores aceder às potencialidades dos microcontroladores e integrarem os mais variados *shields* e sensores.

O Arduino Uno é uma placa que integra o microprocessador *ATmega328P*. A sua arquitetura possui 14 pinos de entrada ou saída digital, 6 entradas analógicas, um cristal de quartzo de 16 MHz, uma conexão USB, um conector de alimentação, um conector ICSP e um botão de *reset*. Para criar um módulo autónomo com base no Arduino Uno, o microcontrolador pode ser ligado a um adaptador AC-DC ou a uma bateria [35]. A seguinte tabela confere as suas características:

Tabela 3.1: Características do Arduino Uno [35].

Microcontrolador	ATmega328P
Tensão operacional	5 V
Tensão de entrada (recomendada)	7-12 V
Tensão de entrada (limite)	6-20 V
Digital I/O pins	14 (dos quais 6 fornecem saída PWM)
Pinos de entrada analógicos	6
Corrente DC por Pin I/O	20 mA
Corrente DC para 3.3V Pin	50 mA
Memória <i>Flash</i>	32 KB
SRAM	2 KB
EEPROM	1 KB
Velocidade do relógio	16 MHz

***Shield* Dragino LoRa/GPS**

As placas de expansão, *shields*, podem ser incorporadas no Arduino, acrescentando funcionalidades ao módulo. Para a escolha de um módulo de transmissão de dados, foi necessário avaliar a cobertura das redes disponíveis no local de utilização, o consumo de energia e ter em consideração as características dos módulos.

O ***Shield* Dragino LoRa/GPS**, não só apresenta todas as características pretendidas, como também permite incorporar um *modem* LoRa para a comunicação e de GPS, simplificando a arquitetura e reduzindo os custos.



Figura 3.7: Dragino LoRa/GPS *Shield*.

O Dragino, ilustrado na figura 3.7 (adaptada de [36]), é composto pela placa-mãe LoRa/GPS *Shield* e LoRa BEE com uma antena de 915 MHz.

No que concerne à comunicação LoRa, o **LoRa BEE** é composto por um transceptor SX1276 / SX1278 que fornece comunicação de espectro disperso de alcance ultralongo, bandas de frequências de 868 / 433 / 915 MHz e resistência a interferências, minimizando a corrente de consumo [37]. Contudo, o desempenho deste módulo pode ser prejudicado com a ocorrência de altas interferências de banda ISM 868 MHz por outras tecnologias ou se o ambiente envolver muitos obstáculos.

No que compete ao GPS, o Dragino inclui o **modelo GPS L80-R** da QUECTEL, baseado no MTK MT3339, projetado para aplicações que requerem localizações ou tempo. Teoricamente, o módulo GPS, para obter uma localização precisa, tem de se conectar a pelo menos três satélites. Uma grande vantagem do GPS L80-R é a rapidez de atuação que, usando a informação da órbita emitida pelos satélites (efemérides) até três dias armazenada na memória flash, calcula e prevê órbitas automaticamente, enquanto o módulo fixa a posição com informação atualizada [37].

Na implementação da placa com o Arduino, é necessário remover o *jumper* GPS_RXD e GPS_TXD do *shield* e fazer a ligação nas portas digitais D4 e D3, respetivamente.

Sensor MPU9250-BMP280

Inicialmente, o projeto estava focado na comunicação LoRa e na visualização da localização. Posteriormente, numa melhoria do projeto, foi elaborada uma forma de ler coordenadas apenas quando o equipamento é ligado, de modo a definir os percursos que este realiza entre o instante em que é ligado até ao momento que se desliga. Adicionalmente, obtém informação sobre as horas de funcionamento do motor. Esta melhoria permite a realização das manutenções preventivas dentro do tempo recomendado pelos fabricantes dos equipamentos. Para a leitura dessa informação, foi proposta a utilização de um acelerómetro, sensor que deteta e monitoriza as vibrações de um sistema mecânico.

O **MPU9250-BMP280** integra dois sensores numa microplaca única, um MPU9250 da InvenSense e um sensor Bosch BMP280, como observado na figura 3.8 (adaptada de [38]).

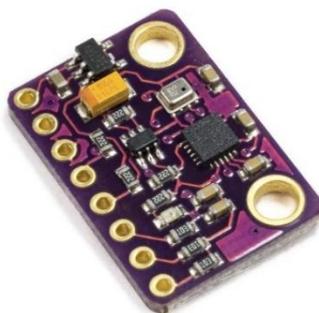


Figura 3.8: Sensor MPU9250-BMP280.

O **MPU-9250** consiste num módulo giroscópio de 3 eixos (x, y e z), um acelerómetro de 3 eixos e um magnetómetro AK8963 com bússola digital de 3 eixos, tudo num *microchip* de dimensões 3x3x1mm e um consumo de energia de apenas 9.3 μ A [39].

O dispositivo **BMP280** é um sensor de pressão barométrica otimizado em termos de consumo de energia, que foi projetado para aplicações móveis e de implementação em dispositivos alimentados por baterias [40].

Este sensor é de fácil utilização e integração com o Arduino, disponibilizando bibliotecas para a sua implementação. No *hardware*, o módulo inclui barramento I2C e um regulador de 3.3 V na placa. A tabela 3.2, apresenta a ligação necessária do sensor ao Arduino para a montagem do circuito apresentado na figura 3.3 (a).

Tabela 3.2: Ligação do sensor ao Arduino UNO.

<u>Arduino</u>		<u>Sensor</u>
3.3V	→	VDD
GND	→	GND
A5	→	SCL (Clock I2C)
A4	→	SDA (entrada de dados I2C)

A calibração inicial é feita sempre que um módulo é instalado num equipamento, visto estar sujeita às características do equipamento e do local junto do motor onde é possível fazer a instalação. Depois de instalado, é efetuado um conjunto de testes, ligando e desligando o equipamento várias vezes e verificando os intervalos de vibração. Os testes são efetuados deste modo porque os níveis de vibração variam de equipamento para equipamento.

3.3 Módulo Servidor

Analogamente ao *End Node*, foi necessário criar um módulo Servidor preparado para comunicar e receber os dados obtidos pelo módulo GSE. Nas subsecções seguintes será apresentado o funcionamento do módulo, bem como a sua composição.

3.3.1 Descrição do módulo

O módulo Servidor, como se pode ver na figura 3.9 (a) é composto por um Arduino Mega e um *shield* LoRa. Na figura 3.9 (b) é apresentado o protótipo final do módulo Servidor com uma caixa em impressão 3D.



a) Placas (Adaptada de [41] e [42]).



b) Caixa do módulo.

Figura 3.9: Protótipo do módulo Servidor.

Na inicialização do módulo Servidor, é definida a frequência de comunicação de 915 MHz, igualmente utilizada pelo módulo GSE. De seguida, o microprocessador inicia um ciclo aguardando por mensagens do módulo GSE e, após receção, guarda os dados obtidos num ficheiro CSV de base de dados, repetindo o ciclo novamente. A figura 3.10 apresenta um fluxograma do funcionamento do módulo Servidor.



Figura 3.10: Fluxograma alusivo ao funcionamento do módulo Servidor.

De forma a realizar o processo de receção das mensagens do módulo GSE, foram desenvolvidos dois subsistemas para o servidor. O primeiro numa placa Arduino (código apresentado no Anexo B) e, o segundo, num computador, para receção dos dados do Arduino e seu armazenamento. Para o segundo subsistema, foi elaborado um código em *Python* que guarda os dados num ficheiro CSV.

3.3.2 Hardware

Arduino Mega

A decisão do microcontrolador a utilizar ficou limitada pelos componentes disponíveis para o protótipo. Assim sendo, o Arduino Mega foi seleccionado como o microcontrolador do módulo Servidor. Contudo, um dos requisitos necessários na selecção do Arduino, foi a capacidade da placa servir de base para o *Shield* LoRa, imprescindível na comunicação com o *End Node*. A outra particularidade que levou à escolha deste *hardware* é a memória *flash* de 256 KB. A figura 3.11 mostra a placa utilizada no desenvolvimento do Servidor.

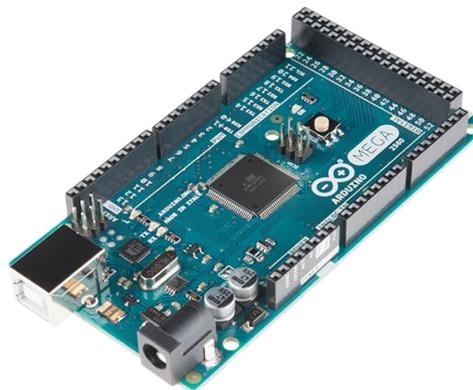


Figura 3.11: Ilustração de um Arduino Mega (retirada de [39]).

A placa Arduino Mega contém um microcontrolador *ATmega2560* da *Atmel* de 8 bits e outras especificações que podem ser analisadas na tabela 3.3 [41].

Tabela 3.3: Características do Arduino Mega 2560.

Microcontrolador	ATmega2560
Tensão operacional	5V
Tensão de entrada (recomendada)	7-12V
Tensão de entrada (limite)	6-20V
Digital I/O pins	54 (dos quais 14 fornecem saída PWM)
Pinos de entrada analógicos	16
Corrente DC por Pin I/O	40 mA
Corrente DC para 3.3V Pin	50 mA
Memória <i>Flash</i>	256 KB
SRAM	8 KB
EEPROM	4 KB
Velocidade do relógio	16 MHz

A placa Mega, tal como a placa UNO, são facilmente programadas através de um cabo USB ao computador e com o *software* IDE, desenvolvido para programação de microcontroladores genéricos em C e C++.

Shield LoRa

O *Shield* Dragino LoRa, representado na figura 3.12 [42], utiliza como base o *chip* Semtech SX1276 / SX1278. Esta placa permite ao utilizador receber ou enviar dados e alcançar intervalos longos com baixa taxa. Tendo as mesmas características e um código de programação semelhante ao *Shield* Dragino LoRa/GPS, optou-se pela sua utilização no módulo Servidor.



Figura 3.12: Dragino LoRa *Shield*.

Na tabela 3.4, pode-se observar as principais características do *Shield LoRa* [42].

Tabela 3.4: Características do *Shield LoRa*.

Tensão de alimentação	3.3 – 5V
Taxa de dados	Até 300 kbps
Banda de frequência	433 MHz / 868 MHz / 915 MHz
Sensibilidade	Até -148 dBm
Antena externa utilizada	915 MHz

3.4 Aplicação de Monitorização

A aplicação desenvolvida tem o intuito de permitir ao utilizador visualizar os percursos efetuados pelos equipamentos. Assim sendo, foram consideradas diversas possibilidades para o desenvolvimento de mapas. A biblioteca *folium*, desenvolvida numa linguagem de programação de alto nível, o *Python*, enquadrou-se com os requisitos pretendidos.

O programa inicializa-se com a leitura dos dados gravados no ficheiro de *database* antecipadamente criado pelo servidor. De seguida, os dados são divididos de acordo com as rotas realizadas, de forma a facilitar a monitorização ao utilizador. Por último, depois de executada a separação, o programa introduz as coordenadas de cada percurso para visualização no mapa. Mediante o exposto, foi usada a biblioteca *csv* para ler/escrever e criar ficheiros e a biblioteca *folium* para criar os mapas em *HTML*. No programa elaborado, todas as bibliotecas se interligam para apresentar uma interface gráfica igual à da figura 3.13.

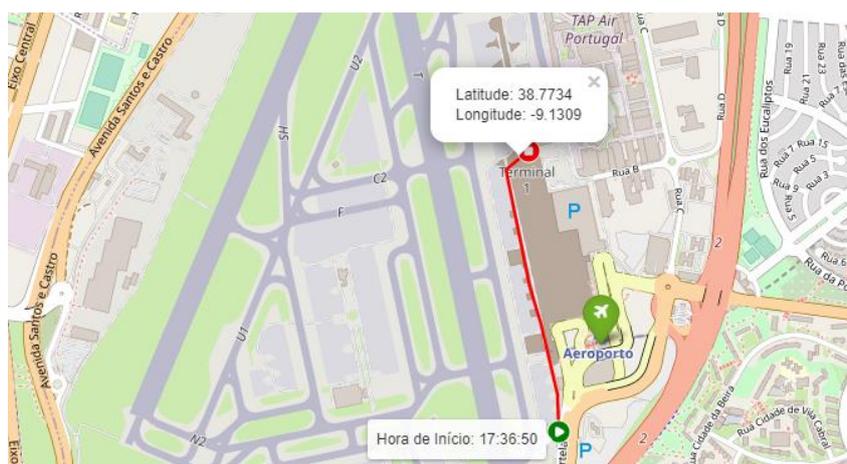


Figura 3.13: Ilustração da Interface Gráfica.

O código foi elaborado de forma a que, ao ser aberto o ficheiro *HTML* pelo utilizador, seja apresentado o mapa do Aeroporto de Lisboa. Neste, podem ser disponibilizadas várias informações ao utilizador, como o número de **rota**, o número de **identificação do equipamento**, a **hora** de início e fim de percurso, tudo através de *pop-ups* e *tooltips*.

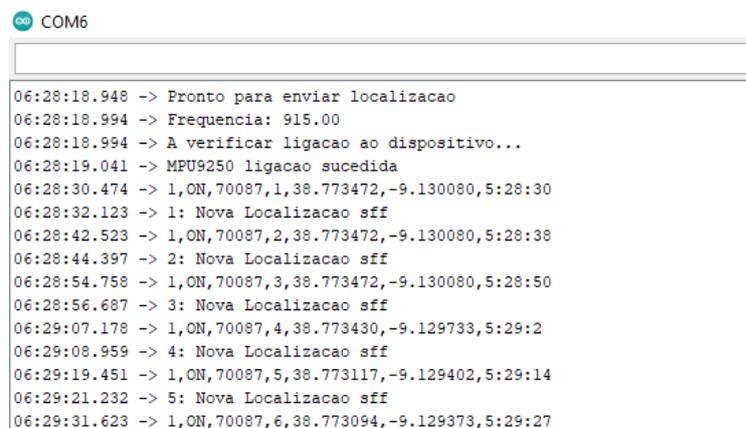
Validação

Neste capítulo pretende-se apresentar o método utilizado para validar o sistema proposto e apresentam-se os resultados obtidos com a implementação do sistema de geolocalização.

O protótipo foi testado durante o seu desenvolvimento e, após a aprovação dos testes iniciais, foram realizadas diferentes experiências, no âmbito da visualização na aplicação e do alcance da comunicação no espaço aeroportuário.

4.1. Testes ao Módulo GSE

O módulo GSE foi testado por fases. Primeiramente, testou-se a ligação do módulo aos satélites GPS. Concluída a primeira fase, testou-se a comunicação utilizada e a sua frequência. Por último, com a implementação do acelerómetro, foram efetuados testes gerais ao módulo GSE. A figura 4.1 apresenta o monitor serial do módulo GSE em funcionamento, com as verificações aos componentes do módulo antes de iniciar a leitura do sensor e do GPS.



```
COM6
06:28:18.948 -> Pronto para enviar localizacao
06:28:18.994 -> Frequencia: 915.00
06:28:18.994 -> A verificar ligacao ao dispositivo...
06:28:19.041 -> MPU9250 ligacao sucedida
06:28:30.474 -> 1,ON,70087,1,38.773472,-9.130080,5:28:30
06:28:32.123 -> 1: Nova Localizacao sff
06:28:42.523 -> 1,ON,70087,2,38.773472,-9.130080,5:28:38
06:28:44.397 -> 2: Nova Localizacao sff
06:28:54.758 -> 1,ON,70087,3,38.773472,-9.130080,5:28:50
06:28:56.687 -> 3: Nova Localizacao sff
06:29:07.178 -> 1,ON,70087,4,38.773430,-9.129733,5:29:2
06:29:08.959 -> 4: Nova Localizacao sff
06:29:19.451 -> 1,ON,70087,5,38.773117,-9.129402,5:29:14
06:29:21.232 -> 5: Nova Localizacao sff
06:29:31.623 -> 1,ON,70087,6,38.773094,-9.129373,5:29:27
```

Figura 4.1: Monitor Serial do Módulo GSE.

4.2. Testes ao Sistema

Para os testes aqui apresentados, foi necessário instalar o módulo GSE num equipamento da empresa apto a circular no espaço aeroportuário. Sempre que o módulo é instalado num equipamento, é fundamental definir a zona de instalação junto do motor e, posteriormente, calibrar o sensor acelerómetro como explicado anteriormente. A figura 4.2 mostra o módulo instalado no equipamento alvo de testes.



a) Módulo GSE junto do motor.

b) Antena GPS no exterior do equipamento.

Figura 4.2: Instalação do Sistema de Geolocalização.

Concluída a instalação, iniciaram-se os testes na área restrita do Aeroporto Humberto Delgado. Contudo, é de salientar que durante estes testes o módulo Servidor estava dentro do veículo.

O **primeiro teste** foi realizado numa zona descoberta, dedicada ao estacionamento de aeronaves. Apesar de, o módulo Servidor ter estado sempre ao alcance do módulo GSE, as interferências de comunicações na área do aeroporto são muito elevadas, o que limita o sinal de comunicação entre os módulos e com os satélites GPS. A figura 4.3 ilustra as mensagens que o módulo Servidor recebeu durante a rota nº15.

```
IDLE Shell 3.9.7
File Edit Shell Debug Options Window Help
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Asus\Desktop\Tese - Prática\Testes\GSE_0787 (teste 3 27.08.2021)\criar_rotas&mapas.py
15,ON,70087,95,38.765007,-9.133443,17:24:59
15,ON,70087,96,38.765003,-9.132843,17:25:14
15,ON,70087,97,38.764130,-9.133020,17:25:28
15,ON,70087,98,38.763943,-9.134232,17:25:42
15,ON,70087,99,38.764210,-9.135167,17:25:55
15,ON,70087,100,38.764423,-9.134820,17:26:10
15,ON,70087,101,38.765575,-9.135102,17:26:24
15,ON,70087,102,38.766579,-9.135995,17:26:38
15,ON,70087,103,38.767883,-9.136310,17:26:53
15,ON,70087,104,38.768944,-9.136575,17:27:6
15,ON,70087,105,38.769737,-9.136622,17:27:20
15,ON,70087,106,38.770538,-9.136827,17:27:34
15,ON,70087,107,38.770725,-9.137293,17:27:48
15,ON,70087,108,38.770176,-9.137192,17:28:2
15,ON,70087,109,38.769527,-9.136943,17:28:15
15,OFF,70087,110,38.769527,-9.136943,17:28:29
```

Figura 4.3: Registo das mensagens durante a rota nº15.

Ao analisar os dados, verifica-se que são enviadas mensagens em intervalos consecutivos de (14 ± 1) segundos e, através da hora de início e fim, calculamos que o equipamento realizou o percurso em três minutos e trinta segundos. A figura 4.4 apresenta, na interface gráfica, o percurso realizado durante o primeiro teste.

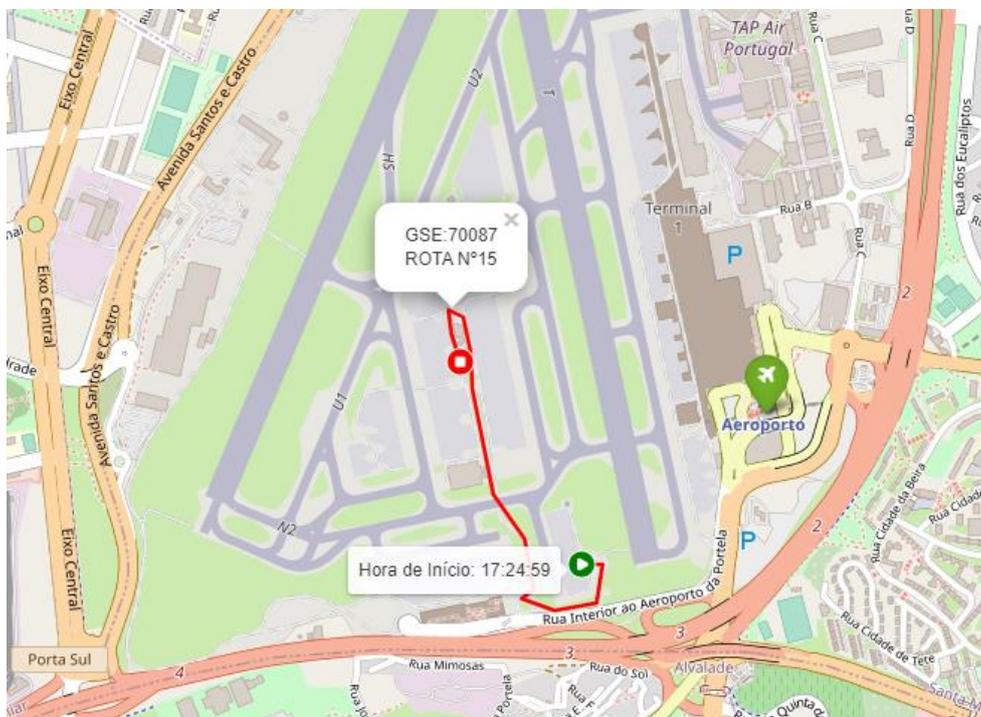


Figura 4.4: Percurso efetuado no primeiro teste.

O GSE iniciou a rota nº15 às dezassete horas, vinte e quatro minutos e cinquenta e nove segundos, percorrendo 1.38 Km no espaço aeroportuário.

No **segundo teste** optou-se por analisar uma zona do aeroporto mais condicionada em termos de edifícios e de interferências. Adicionalmente, no segundo percurso aumentou-se a distância percorrida para 2.74 Km, como se pode observar na figura 4.5.

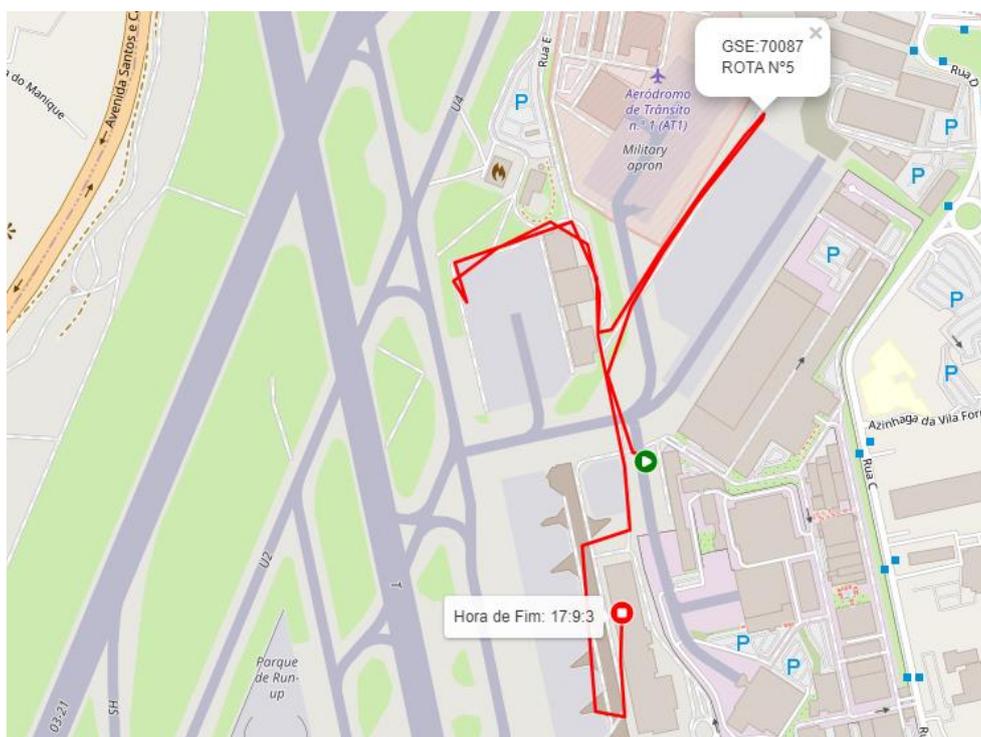


Figura 4.5: Percurso efetuado no segundo teste.

Neste teste, o equipamento demorou doze minutos e vinte e um segundos a completar o percurso. Os dados extraídos de cada percurso são importantes para o historial do equipamento, para prever a realização das manutenções preventivas, bem como a sua monitorização.

4.3. Testes de Alcance

De forma a testar a distância de comunicação entre o servidor e o equipamento, executaram-se alguns testes em zonas opostas do aeroporto. O módulo Servidor conectado a um computador foi posicionado numa sala de um dos edifícios pertencentes à empresa, enquanto o equipamento sujeito a testes, circulava no aeroporto.

No **primeiro teste** de alcance, como se pode observar pela figura 4.6, a distância percorrida dentro do alcance de comunicação foi de apenas 204 m, após essa distância a comunicação falhava e perdia-se mensagens, o que limita a monitorização dos equipamentos no espaço aeroportuário.



Figura 4.6: Alcance atingido no primeiro teste.

Como se pode verificar, o percurso do primeiro teste de alcance está bastante condicionado por edifícios e obstáculos.

No **segundo teste** de alcance, explorou-se uma zona de visão ampla entre o servidor e o equipamento monitorizado. A distância, como se pode verificar na figura 4.7, aumentou significativamente em relação ao primeiro teste.



Figura 4.7: Alcance máximo atingido.

4.4. Discussão de Resultados

Como se pode verificar, o módulo GSE funcionou como esperado, no entanto, para o alcance do sinal, a realidade não corresponde à expectativa do alcance máximo. No diagrama de Ishikawa da figura 4.8, é exemplificada a metodologia utilizada avaliando as principais causas para o problema de perda de sinal.

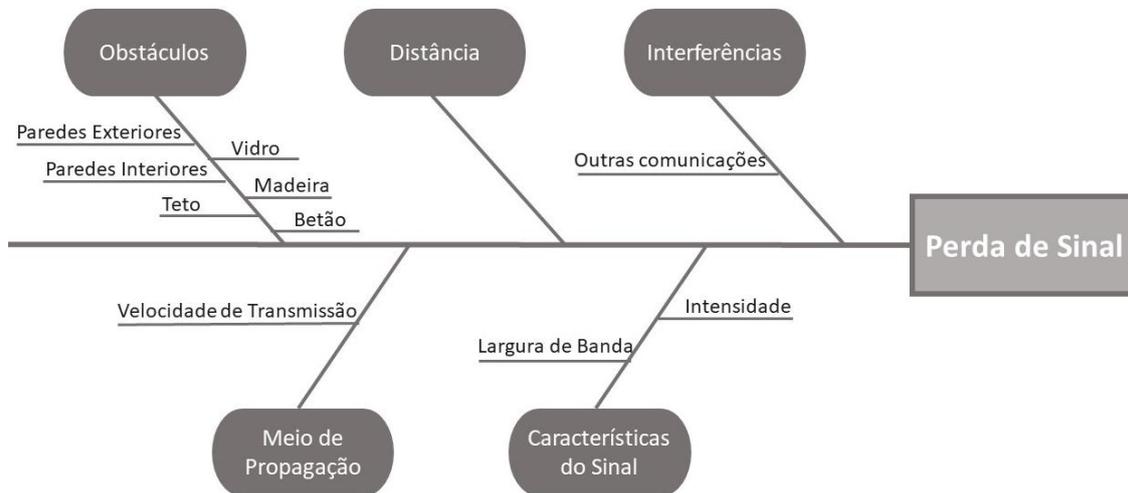


Figura 4.8: Diagrama de Ishikawa, avaliação das perdas de sinal.

A perda de sinal causada por obstáculos influencia a recepção do sinal e diminui consideravelmente o alcance. No caso da comunicação LoRa, pode-se verificar que existem muitos fatores que influenciam a atenuação de onda, tais como, uma parede exterior com 60 cm de espessura capaz de atenuar o sinal cerca de 35 dB, ou, uma janela que reduz entre 0.8 a 30 dB ao sinal transmitido [43].

Nos dois testes de alcance efetuados foram considerados todos os fatores, devido ao ambiente onde está inserido o sistema desenvolvido e ao módulo Servidor estar localizado dentro de um edifício.

Considerando que a tecnologia LoRa é utilizada no aeroporto de Istambul para sistemas instalados em locais internos e externos, as antenas necessitam de uma elevada potência e robustez, com capacidade de cobertura de toda a área. Tendo como base estas características, as antenas LoRa utilizadas no protótipo não são as mais eficazes para um sistema com a dimensão desejada. Contudo, a tecnologia é funcional e viável para a utilização em espaço aeroportuário, sendo também utilizada em aeroportos europeus como referido no estado de arte.

Conclusões e Trabalhos Futuros

Neste capítulo são abordadas as conclusões relativas ao projeto desenvolvido na presente dissertação, bem como o cumprimento dos objetivos iniciais. Posteriormente, são apresentados os trabalhos futuros considerados relevantes para o melhoramento do sistema.

5.1 Conclusões

Nos últimos anos, assistiu-se a uma aposta das companhias aéreas e das empresas de *handling* em novas tecnologias, explorando e otimizando todos os processos, na tentativa de revolucionar cada vez mais o transporte aéreo. Neste sentido, esta dissertação propõe um sistema de geolocalização para monitorização dos equipamentos de assistência a aeronaves com comunicação LoRa.

De acordo com o trabalho realizado, conclui-se que a tecnologia LoRa e o protocolo de comunicação LoRaWAN, de longo alcance e baixo consumo de energia, poderão ser adequados para suportar a interação entre os equipamentos aeroportuários e o servidor.

Considerando esta tecnologia, foram construídos dois protótipos, o módulo GSE e o módulo Servidor para validar o sistema proposto. Após a realização de diferentes testes, conseguiu-se verificar que todo o sistema apresentado é funcional, isto é, permite a monitorização dos equipamentos através de sensores e a visualização dos percursos efetuados numa interface gráfica.

Durante o processo de desenvolvimento e de acordo com os testes realizados, verificou-se que as antenas utilizadas nos módulos, não estão bem posicionadas e não têm capacidade de alcance suficiente para abranger todo o espaço de circulação dos equipamentos. Portanto, para a comunicação de rede ser concebível em longas distâncias e sem perdas de dados, é necessário que do ponto de vista do módulo GSE até ao módulo Servidor exista um campo de visão aberto.

Apesar das distâncias obtidas durante os testes não serem as mesmas que as estipuladas para a tecnologia LoRa, os valores são toleráveis, considerando os inúmeros obstáculos e interferências no local dos testes. Importa salientar que o foco deste projeto é a monitorização de equipamentos, o qual foi conseguido com sucesso.

O trabalho desenvolvido no âmbito de um sistema de geolocalização de *Ground Support Equipment* atingiu os objetivos propostos inicialmente, sendo de referir ainda que este sistema tem capacidade para ser integrado no sistema de gestão operacional da empresa.

Para implementação do sistema, terá de ser autorizada a instalação de antenas LoRa pela empresa gestora do aeroporto de Lisboa ou a alteração do sistema de comunicação para uma tecnologia já utilizada no espaço aeroportuário, como por exemplo, a tecnologia AeroMACS. Esta tecnologia opera na faixa de espectro de aviação protegida e licenciada de 5091MHz a 5150 MHz. Com base no padrão WiMAX, é uma tecnologia de banda larga sem fio que suporta a crescente necessidade de comunicação de dados e partilha de informações na superfície do aeroporto para aplicações fixas e móveis.

5.2 Trabalhos Futuros

O sistema desenvolvido foi pensado e criado para de futuro ser facilmente ampliável, acrescentando algumas funcionalidades e valorizando o sistema de monitorização.

Como proposta para trabalho futuro, reconhece-se algumas melhorias e um maior investimento nos dispositivos utilizados, nomeadamente um processador mais rápido, uma antena GPS com atualizações rápidas da posição, antenas de comunicação com maior alcance e otimização do código, mitigando falhas de ligação ou perdas de informação no envio dos dados dos equipamentos.

Como trabalho futuro pretende-se adicionar ao módulo GSE, sensores capazes de obter informação acerca dos níveis do motor, óleo, água e combustível, como também questões de avarias, nomeadamente em equipamentos antigos. Para equipamentos mais recentes com sensores controlados por uma centralina, o módulo GSE seria ligado ao *On-Board Diagnostic* para obter a informação dos sensores. O envio da informação para o sistema de gestão da empresa possibilitaria a rápida intervenção de uma equipa de manutenção ao equipamento avariado.

Em alternativa à utilização do módulo Servidor que, no presente protótipo apenas permite uma ligação em cada instante, terão de ser utilizadas *gateways* LoRa, permitindo a ligação de mais dispositivos, criando assim uma rede de monitorização e gestão dos equipamentos da empresa.

De forma a otimizar os dados recebidos dos equipamentos, sugere-se ainda a inclusão no sistema de uma base de dados SQL, em detrimento do armazenamento de dados em ficheiro. Desta forma, será possível ter uma plataforma eficiente em armazenamento e na disponibilização dos dados recolhidos.

Como neste momento não é possível a instalação de uma rede LoRa no aeroporto de Lisboa, será ponderado o desenvolvimento de um novo protótipo com a utilização do protocolo AeroMACS. Contudo, a utilização desta tecnologia implicaria custos de utilização avultados.

Referências Bibliográficas

- [1] “O que é a Internet das Coisas”. [Online]. Disponível em: www.wattson.pt [Acedido: março de 2020]
- [2] Gubbi, J., Buyya, R., Marusic, S., et al., (2013). “*Internet of things (IoT): A vision, architectural elements, and future directions*”. *Future Generation Computer Systems*, 29(7), pp.1645–1660.
- [3] L. Atzori, A. Iera, & G. Morabito, (Out 2010). “*The Internet of things: A survey*”. *Computer Networks*, vol. 54, no. 15, pp. 2787–2805.
- [4] Chui, Michael, Löffler, Markus & Roberts, Roger, (mar 2010). [Online]. “*The Internet of Things*”. Disponível em: <https://www.mckinsey.com/>. [Acedido: março de 2020]
- [5] “Virtualização A representação digital do processo produtivo, ou seja, do conjunto de máquinas, dispositivos e robôs, simulando o processo, suas interferências”. [Online]. Disponível em: slideplayer.com.br/slide/17094285 [Acedido: março de 2020]
- [6] Janssen, C., (2011). “*Internet of things (IoT)*”. Techopedia [Online]. Disponível em: www.techopedia.com/definition/28247/internet-of-things-iot [Acedido: abril de 2020]
- [7] Madakam, S., Ramaswamy, R. & Tripathi, S., (2015). “*Internet of things (IoT): A Literature Review*”. *Journal of Computer and Communications*, vol. 3, 164-173.
- [8] Razzaque, M.A. et al., (2015). “*Middleware for Internet of things: a Survey*”. *Internet of things Journal*, IEEE, pp 99, p.1.
- [9] Sundmaeker, Harald & Guillemin, Patrick & Friess, Peter & Woelfflé, Sylvie., (2010). “*Vision and Challenges for Realizing the Internet of Things.*”. *Cluster of European Research Projects on the Internet of Things, European Commission*.
- [10] Sethi, Pallavi & Sarangi, Smruti., (2017). “*Internet of Things: Architectures, Protocols, and Applications*”. *Journal of Electrical and Computer Engineering*, pp 1-25.

- [11] Rehman, Aqeel-ur & Rehman, Sadiq Ur & Khan, Iqbal & Moiz, M. & Hasan, S., (2016). "Security and privacy issues in IoT". *International Journal of Communication Networks and Information Security*, vol. 8, no. 3, pp 147-157.
- [12] Gubbi, Jayavardhana & Buyya, Rajkumar & Marusic, Slaven & Palaniswami, Marimuthu., (2012). "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions". *Future Generation Computer Systems*.
- [14] Babar, Sachin, Parikshit Mahalle, Antonietta Stango, Neeli Prasad, & Ramjee Prasad., (2010). "Proposed security model and threat taxonomy for the Internet of things (IoT)". *Recent Trends in Network Security and Applications*, pp. 420-429.
- [15] G. A. Akpakwu, B. J. Silva, G. P. Hancke & A. M. Abu-Mahfouz, (2018). "A Survey on 5G Networks for the Internet of things: Communication Technologies and Challenges". *IEEE*. vol. 6, pp. 3619-3647.
- [16] K. E. Jeon, J. She, P. Soonsawad, & P. C. Ng, (2018). "BLE Beacons for Internet of things Applications: Survey, Challenges and Opportunities". *IEEE Internet Things J.*, pp. 1–1.
- [17] "Bluetooth 5/ Go Faster. Go Further." [Online]. Disponível em: <https://www.bluetooth.com/bluetooth-resources>. [Acedido: julho de 2020]
- [18] "Low power, long range Wi-Fi® for IoT". [Online]. Disponível em: <https://www.wi-fi.org/discover-wi-fi/wi-fi-halow> [Acedido: julho de 2020].
- [19] R. Want, (mar 2006). "An introduction to RFID technology". *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 25-33.
- [20] S. Al-Sarawi, M. Anbar, K. Alieyan & M. Alzubaidi, (2017). "Internet of things (IoT) communication protocols: Review". *8th International Conference on Information Technology (ICIT)*, Amman, pp. 685-690.
- [21] S. C. Ergen, (2004). "ZigBee/IEEE 802.15.4 Summary". U.C. Berkeley, pp. 1-37.
- [22] Semtech, (Mai 2018). "Migrating an Internet of things (IoT) Sensor Design to LoRaWAN®", pp. 1-16. Disponível em: <https://www.semtech.com> [Acedido: julho de 2020]
- [23] Song, Yonghua & Lin, Jin & Tang, Ming & Dong, Shufeng, (2017). "An Internet of Energy Things Based on Wireless LPWAN". *Engineering*. vol. 3. pp. 460-466.
- [24] Mekki, Kais & Bajic, Eddy & Chaxel, Frédéric & Meyer, Fernand, (2019). "A comparative study of LPWAN technologies for large-scale IoT deployment". *ICT Express*, vol. 5, pp. 1-7.
- [25] Vangelista, Lorenzo & Zanella, Andrea & Zorzi, Michele, (2015). "Long-Range IoT Technologies: The Dawn of LoRa™". *XXXVII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, pp. 51-58.

- [26] J. Bardyn, T. Melly, O. Seller & N. Sornin, (2016). "IoT: The era of LPWAN is starting now". *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*, Lausanne, pp. 25-30.
- [27] S. S. Basu, A. K. Sultania, J. Famaey & J. Hoebeke, (2019). "Experimental Performance Evaluation of NB-IoT". *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Barcelona, Espanha, pp. 1-6.
- [28] Isikdag, Umit & Zlatanova, Sisi, (mai 2011). "Sensor services for buildings: A framework and opportunities". *GeoInformation for Disaster Management*, Antalya, Turquia.
- [29] "Análise de Cobertura e Capacidade em Redes Móveis LTE de Quarta Geração (4G)" [Online]. Disponível em: www.anacom.pt [Acedido: setembro de 2020].
- [30] Cosmi, Arthur & Mota, Vinícius, (2019). "Uma Análise dos Protocolos de Comunicação para Internet das Coisas". Workshop de computação urbana (courb). Porto Alegre: Sociedade Brasileira de Computação. pp. 153-166.
- [31] Frigieri, Daniel & Parreira, Luís. "Protocolos M2M para Ambientes Limitados no Contexto do IoT: Uma Comparação de Abordagens" [Online]. Disponível em: <https://www.inatel.br/smartcampus/imgs/protocolos-para-iot-pt> [Acedido: setembro de 2020].
- [32] H. Subramoni, G. Marsh, S. Narravula, Ping Lai & D. K. Panda, (2008) "Design and evaluation of benchmarks for financial applications using Advanced Message Queuing Protocol (AMQP) over InfiniBand". *Workshop on High Performance Computational Finance*, Austin, TX, pp. 1-8.
- [33] "What is MQTT? | Protocol for the Internet of Things (IoT)". [Online]. Disponível em: <https://www.opc-router.com/what-is-mqtt/> [Acedido: setembro de 2020].
- [34] "Internet of Things (IoT) protocols COAP MQTT". [Online]. Disponível em: <https://pt.slideshare.net/vgholkar/iot-protocolsoscon2014> [Acedido: setembro de 2020].
- [35] "Arduino Uno Rev3". [Online]. Disponível em: <https://www.elektor.com> [Acedido: agosto de 2021].
- [36] "What is a GPS navigator and how it works". [Online]. Disponível em: <http://ipkey.com.ua/uk/faq/1087-what-is-gps-navigator.html> [Acedido: agosto de 2021].
- [37] "Arduino Shield featuring GPS and LoRa® technology". [Online]. Disponível em: www.dragino.com/products/lora/item/108-lora-gps-shield.html [Acedido: agosto de 2021].
- [38] "MPU9250-BMP280 - Giroscópio, Acelerómetro, Magnetómetro e Barómetro". [Online]. Disponível em: <http://www.vladcontrol.com.br> [Acedido: julho de 2021].
- [39] "MPU-9250 | TDK". [Online]. Disponível em: <https://invensense.tdk.com> [Acedido: agosto de 2021].

- [40] “*Pressure Sensor BMP280 | Bosch Sensortec*”. [Online]. Disponível em: <https://www.bosch-sensortec.com/products> [Acedido: agosto de 2021].
- [41] “*Arduino Mega 2560 Original*”. [Online]. Disponível em: <https://www.electrofun.pt> [Acedido: setembro de 2021].
- [42] “*Lora Shield - Wiki for Dragino Project*”. [Online]. Disponível em: https://wiki.dragino.com/index.php?title=Lora_Shield [Acedido: maio de 2021].
- [43] C. Penha, (2019). “Arquitetura de Suporte ao Desenvolvimento de Sistemas Auto-configuráveis para Redes de Sensores sem Fios”. Inst. Politécnico Beja.

Anexos

Anexo A – Código do módulo GSE

```
//Bibliotecas
#include <MPU9250.h>
#include <SoftwareSerial.h>
#include <TinyGPS.h>
#include <RH_RF95.h>

//Variáveis Globais
#define RF95_FREQ 915.0
#define RFM95_CS 10
#define RFM95_INT 2
#define AXMAX -14900
#define AXMIN -16000
#define AYMAX -5500
#define AYMIN -6500
#define LOOPON 10
#define LOOPOFF 10
#define matricula 70087

//Declara uma variável estática rf95, do tipo RH_RF95 com dois parâmetros
RH_RF95 rf95(RFM95_CS, RFM95_INT);
MPU9250 ignicao;
TinyGPS gps;
SoftwareSerial mySS(4,3);

int ano;
byte mes, dia, hora, minuto, segundo, centesimo;
unsigned long age;

int16_t ax, ay, az;
int16_t gx, gy, gz;
int16_t mx, my, mz;
int i=0, z=0, j=0, x=1, rota=1;
int message=1;
```

```

float flat, flon;
float lastlat, lastlong;

char estado[4] = {"OFF\0"};
char lastla[15] = {"\0"};
char lastlo[15] = {"\0"};
char tempo[13] = {"\0"};
char coord[70] = {"\0"};

static void smartdelay (unsigned long ms);
static void print_float (float val, float invalid, int len, int prec);
char * stringtosend (int rota, float flat, float flon);
char * timecoordenadas (char * timer, byte hora, byte minuto, byte segundo);
char * remove_espacos (char * out, const char * in );

/***** Inicialização *****/
*****/
void setup()
{
  Wire.begin();
  Serial.begin(9600);
  mySS.begin(9600);
  if (!rf95.init())
    Serial.println("Falha ao iniciar");
  Serial.println("Pronto para enviar localizacao");

  if (!rf95.setFrequency(RF95_FREQ))
  {
    Serial.println("Falha na frequência");
    while (1);
  }
  Serial.print("Frequencia: ");
  Serial.println(RF95_FREQ);

  rf95.setTxPower(23, false);

  ignicao.initialize(); //Verificar conexão
  Serial.println("A verificar ligacao ao dispositivo...");
  Serial.println(ignicao.testConnection() ? "MPU9250 ligacao sucedida" : "MPU9250 falha
na ligacao");

  delay(1000);
}

/***** Ciclo *****/
*****/
void loop()
{
  ignicao.getMotion9(&ax, &ay, &az, &gx, &gy, &gz, &mx, &my, &mz); //le o sensor
timecoordenadas(tempo, hora, minuto, segundo);
  if ((ax<AXMIN || ax>AXMAX) && (ay<AYMIN || ay>AYMAX)) //intervalo de calibração
  {

```

```

i++;
if(i>=LOOPON)
{ //retira a hipótese de leituras precipitadas e.x choque ou vibrações
j=0;
gps.f_get_position(&flat, &flon, &age); //dá a posição

if(flat!=1000.00 && flon!=1000.00) //retira a primeira leitura errada
{
strcpy(estados,"ON\0");
//imprime um float de latitude e outro de longitude com 6 casas decimais e um total de 11
caracteres
print_float(flat, TinyGPS::GPS_INVALID_F_ANGLE, 11, 6);
print_float(flou, TinyGPS::GPS_INVALID_F_ANGLE, 11, 6);

stringtosend(rota, message, flat, flon);

rf95.send((uint8_t*)coord, sizeof(coord)+10); //envia os dados para o servidor
rf95.waitPacketSent();
message++;

// espera por uma resposta do servidor
uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
uint8_t len = sizeof(buf);

if (rf95.waitAvailableTimeout(3000))
{ // recepção da mensagem de resposta
if (rf95.recv(buf, &len))
{ //contagem das mensagens de resposta
Serial.print(x);
Serial.print(" ");
Serial.println((char*)buf);
x++;
}
else
{
Serial.println("Falha no recibo");
}
}
else
{ delay(50);
// se não receber resposta.
Serial.println("Sem resposta do servidor");
}
i=0;
lastlat = flat;
lastlong = flon;
}
smartdelay(5000);
}
}
else
{
z++;
if(z>=LOOPOFF)
{

```

```

if(strcmp(estado,"OFF\0")==0 && j==0) //caso deixe de trabalhar e passe ao estado OFF
{
if(lastlat!=0.0 && lastlong!=0.0)
{
stringtosend (rota, message, lastlat, lastlong);
rf95.send((uint8_t*)coord, sizeof(coord)+10); //envia os dados para o servidor
rf95.waitPacketSent();
rota++;
message++;
}
j=1;
z=0;
}
}
}
delay(500);
strcpy(estado,"OFF\0");
}

/***** Conjunto de Funções *****/
*****/
static void smartdelay(unsigned long ms)
{
unsigned long start = millis();
do {
while (mySS.available()){
gps.encode(mySS.read());
}
} while (millis() - start < ms);
}

/***** Determina as coordenadas no formato pretendido *****/
static void print_float(float val, float invalid, int len, int prec)
{
int vi = abs((int)val);
int flen = prec + (val < 0.0 ? 2 : 1); // '!' e '-'
flen += vi >= 1000 ? 4 : vi >= 100 ? 3 : vi >= 10 ? 2 : 1;
smartdelay(0);
}

/***** Formata a String a enviar *****/
char * stringtosend (int rota, int message, float flat, float flon)
{
char GSEID[6]={"\0"};
char messageID[4]={"\0"};
char rotas[4]={"\0"};
char gps_lat[15]={"\0"};
char gps_lon[15]={"\0"};
char coordenadas[70];

gps.crack_datetime(&ano, &mes, &dia, &hora, &minuto, &segundo, &centesimo, &age);
timecoordenadas(tempo, hora, minuto, segundo);
//torna um float numa string
dtostrf(rota, 3, 0, rotas);
dtostrf(matricula, 5, 0, GSEID);

```

```

dtostrf(message, 3, 0, messageID);
dtostrf(lat, 12, 6, gps_lat);
dtostrf(lon, 12, 6, gps_lon);

//colocar a string coordenadas no seguinte formato: IDRota, estado, GSEID, messageID,
latitude, longitude
strcat(strcpy(coordenadas,rotas),",");
strcat(strcat(coordenadas,estado),",");
strcat(strcat(coordenadas,GSEID),",");
strcat(strcat(coordenadas,messageID),",");
strcat(strcat(coordenadas,gps_lat),",");
strcat(strcat(coordenadas,gps_lon),",");
strcat(coordenadas,tempo);
Serial.println(remove_espacos(coord,coordenadas));
}

/***** Determina a hora GMT *****/
char * timecoordenadas(char * timer, byte hora, byte minuto, byte segundo)
{
    char hor[3], mins[3], seg[3];
    dtostrf(hora, 2, 0, hor);
    dtostrf(minuto, 2, 0, mins);
    dtostrf(segundo, 2, 0, seg);
    strcat(strcat(strcat(strcat(strcpy(timer,hor),":"),mins),":"),seg);
    return timer;
}

/***** Remover espaços da String *****/
char * remove_espacos( char * out, const char * in )
{
    const char * p = in;
    int i = 0;

    while( *p ) {
        if( !isspace(*p) )
            out[i++] = *p;
        p++;
    }
    out[i] = 0;
    return out;
}

```

Anexo B – Código do módulo Servidor

```
//Biblioteca
#include <RH_RF95.h>

//Variáveis Globais
#define RFM95_CS 10
#define RFM95_RST 7
#define RFM95_INT 2
#define RF95_FREQ 915.0

//Declara uma variável estática rf95, do tipo RH_RF95 com dois parâmetros
RH_RF95 rf95(RFM95_CS, RFM95_INT);

void setup()
{
  pinMode(RFM95_RST, OUTPUT);
  digitalWrite(RFM95_RST, HIGH);

  while (!Serial);
  Serial.begin(9600);
  delay(100);

  //reset manual
  digitalWrite(RFM95_RST, LOW);
  delay(10);
  digitalWrite(RFM95_RST, HIGH);
  delay(10);

  while (!rf95.init()) {
    while (1);
  }
  if (!rf95.setFrequency(RF95_FREQ)) {
    //Falha na frequência
    while (1);
  }
  rf95.setTxPower(23, false);
}

void loop()
{
  if (rf95.available())
  {
    //Recebe mensagem
    uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
    uint8_t len = sizeof(buf);

    if (rf95.recv(buf, &len))
    { //imprime a mensagem
      Serial.println((char*)buf);
      delay(200);

      //Envia resposta
    }
  }
}
```

```
uint8_t data[] = "Nova Localizacao sff";  
rf95.send(data, sizeof(data));  
rf95.waitPacketSent();  
}  
}  
delay(3000);  
}
```