



Universidade do Minho
Escola de Engenharia

Nuno Adriano Oliveira Maia

Tags Wi-Fi para Posicionamento
em Ambientes Interiores



Universidade do Minho
Escola de Engenharia

Nuno Adriano Oliveira Maia

Tags Wi-Fi para Posicionamento
em Ambientes Interiores

Dissertação de Mestrado
Ciclo de Estudos Integrados Conducentes ao
Grau de Mestre em Engenharia de Comunicações

Trabalho efetuado sob a orientação do
Professor Doutor Filipe Meneses
Professor Doutor Adriano Moreira

janeiro de 2015

DECLARAÇÃO

Nuno Adriano Oliveira Maia

Endereço electrónico: nunoaomaia@gmail.com

Telefone: 964888806

Número do Bilhete de Identidade: 12907664

Título dissertação :

Tags Wi-Fi para Posicionamento em Ambientes Interiores

Orientadores:

Filipe Miguel Lopes Meneses

Adriano Jorge Cardoso Moreira

Ano de conclusão: 2014_

Ciclo de Estudos Integrados Conducentes ao Grau de Mestre em Engenharia de Comunicações

Nos exemplares das teses de doutoramento ou de mestrado ou de outros trabalhos entregues para prestação de provas públicas nas universidades ou outros estabelecimentos de ensino, e dos quais é obrigatoriamente enviado um exemplar para depósito legal na Biblioteca Nacional e, pelo menos outro para a biblioteca da universidade respectiva, deve constar uma das seguintes declarações:

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Universidade do Minho, 09/01/2015

Assinatura: _____

"Sucesso é encontrar aquilo que se tenciona ser e depois fazer o que é necessário para isso."

EPICETETO

Agradecimentos

Perante a finalização de mais uma etapa da minha vida, não poderia deixar de agradecer àquelas pessoas que mais me ajudaram e apoiaram.

Começo por agradecer aos meus orientadores, Professor Doutor Filipe Meneses e Professor Doutor Adriano Moreira, pelo empenho, dedicação e disponibilidade que sempre demonstraram. Foi um privilégio realizar esta dissertação sob a vossa orientação.

Aos meus queridos pais e irmão que sempre me apoiaram, incentivaram e contribuíram para que fosse possível concluir todos os desafios com que fui confrontado. Agradeço acima de tudo, os diversos sacrifícios suportados com o objetivo de concluir mais uma etapa da minha vida. Agradeço também ao resto da minha família, pois sempre me apoiaram.

Quero agradecer a todos os meus amigos, principalmente àqueles que me acompanharam durante esta fase da minha vida, nomeadamente à Margarida, Adriana, Marta, ao Hugo, Xano, Tó e ao Luís pelos bons momentos, conselhos, apoio e ajuda sempre que necessário. Ao Hélder e Alexandrine por me fazerem sonhar e querer sempre mais.

À Vera pelo apoio incondicional, por fazermos um excelente grupo de trabalho, pela força, carinho, apoio e por tudo aquilo que significa para mim.

A todos vocês, Obrigado!

Resumo

Durante os últimos anos, inúmeros investigadores exploraram o uso de diferentes tecnologias com o objetivo de desenvolver soluções para posicionamento e navegação no interior de edifícios, com diferentes graus de precisão. As redes Wi-Fi podem ser encontradas em quase todo o lado e presentemente estão a ser amplamente exploradas para o posicionamento interior.

As redes Wi-Fi podem ser usadas de várias formas para implementar um sistema de posicionamento interior. Uma das formas mais usual é através da utilização de Wi-Fi *fingerprinting*. Neste modelo de funcionamento, um dispositivo (cliente Wi-Fi) lê o ambiente rádio circundante e fornece a leitura (lista de *access points* detetados e respetivo nível de sinal) a um serviço que estima a localização, comparando a leitura do dispositivo (*fingerprint*) com um mapa de rádio do edifício pré-construído.

Nesta dissertação é descrita a implementação de duas *tags* Wi-Fi, usando algum *hardware* disponível no mercado, de pequeno porte e baixo custo. Na primeira parte do documento, é apresentada a motivação, que se baseia principalmente na inexistência de *tags* Wi-Fi que possam ser usadas para investigação e prototipagem de soluções de localização e navegação interior. As *tags* Wi-Fi atualmente existentes, fazem parte de soluções comerciais completas e não podem ser facilmente utilizadas por sistemas de posicionamento interior de terceiros.

Foram desenvolvidas duas *tags* diferentes, baseadas em hardware existente no mercado. Este documento inclui uma descrição dos módulos de hardware e os algoritmos implementados na componente de software.

Nos últimos capítulos, é apresentada uma análise da autonomia e fiabilidade das *tags*. Para economizar bateria e melhorar a autonomia foram integrados acelerómetros, que foram combinados com *timers* fornecidos pelas APIs dos microprocessadores.

Abstract

During the last years, researchers explored the usage of different technologies to develop solutions for indoor positioning and navigation with different degrees of accuracy. Wi-Fi networks can be found almost everywhere and are currently being widely explored for indoor positioning.

Wi-Fi networks can be used in different ways to implement an indoor positioning system. One of the most common ways to implement it is to use of a Wi-Fi fingerprint. In this way, a Wi-Fi enabled device (Wi-Fi client) reads the surrounding radio environment and provides the fingerprints (list of access points and corresponding signal strength) to a network service that estimates the position, comparing the fingerprint with a prebuilt radio map of the building.

In this dissertation we describe the implementation of a Wi-Fi tag, using some small sized low cost off-the-shelf hardware. In the first section of the document we present our motivation, which is mostly based in the inexistence of open Wi-Fi tags that can be used for research and prototyping of indoor positioning and navigation solutions. The currently existent Wi-Fi tags are part of complete commercial solutions and cannot be easily used by third party indoor positioning systems.

We developed two different tags based on off-the-shelf hardware components. This document includes a description of the hardware modules and the algorithms implemented in the software component.

In the last sections it is presented an analysis of the tags' autonomy. To save battery and improve the autonomy, we integrated accelerometers, which were combined with the timers' functions provided by the microprocessors boards.

Índice

Agradecimentos.....	vii
Resumo.....	ix
Abstract.....	xi
Índice.....	xiii
Índice de Figuras.....	xvii
Índice de Tabelas.....	xix
Índice de Acrónimos e Abreviaturas.....	xxi
1. Introdução.....	1
1.1. Enquadramento.....	1
1.2. Objetivos.....	2
1.3. Abordagem.....	2
1.4. Estrutura da Dissertação.....	3
2. Estado da Arte.....	5
2.1. Técnicas de Localização Wi-Fi.....	5
2.2. Produtos Comerciais.....	7
2.2.1. Soluções Completas.....	8
2.3. Tags.....	22
2.3.1. Síntese.....	24
3. Estudo do Problema.....	25
3.1. Identificação do Problema.....	25
3.2. Requisitos.....	26
3.3. Recolha de Dados e Poupança de Energia.....	27

4.	Especificação do Sistema.....	29
4.1.	Arquitetura do Sistema.....	29
4.1.1.	Arquitetura Geral.....	29
4.1.2.	Arquitetura Específica.....	30
4.2.	Mensagens do Sistema.....	30
5.	Implementação.....	33
5.1.	Escolha do Hardware.....	33
5.2.	Construção da <i>Tag</i>	36
5.2.1.	Electric imp.....	36
5.2.2.	Openpicus FlyPort Wi-Fi.....	39
5.3.	Construção do Software.....	42
5.3.1.	Servidor.....	42
5.3.2.	Protocolo de Comunicação <i>Tag</i> - Acelerómetro.....	43
5.3.3.	Electric imp.....	45
5.3.4.	Openpicus Flyport Wi-Fi.....	54
6.	Medição dos Consumos e Análise da Autonomia.....	63
6.1.	Medição dos Consumos.....	63
6.1.1.	Medições Electric imp.....	63
6.1.2.	Medições Openpicus Flyport Wi-Fi.....	70
6.2.	Autonomia e Fiabilidade.....	76
6.2.1.	Determinação da Autonomia.....	78
7.	Conclusões e Trabalho Futuro.....	83
7.1.	Conclusões.....	83
7.1.1.	Recolha de Dados e Poupança de Energia.....	83
7.2.	Análise dos Resultados.....	84
7.3.	Trabalho Futuro.....	85

8. Bibliografia.....	87
A. Anexo: Script para o Cálculo da Autonomia.....	89

Índice de Figuras

Figura 2.1 - Visão geral do Ekahau RTLS (Ekahau, 2013)	12
Figura 2.2 - Implementação Ekahau RTLS (Ekahau, 2013)	13
Figura 2.3 - Ekahau A4 (Ekahau, 2013)	14
Figura 2.4 - Ekahau A4+ (Ekahau, 2013)	14
Figura 2.5 - Ekahau B4 (Ekahau, 2013)	14
Figura 2.6 - Ekahau W4 (Ekahau, 2013)	15
Figura 2.7 - Ekahau TS1 e TS2 respetivamente (Ekahau, 2013)	15
Figura 2.8 - Ekahau TS-c (Ekahau, 2013)	16
Figura 2.9 - Ekahau HS1 (Ekahau, 2013)	16
Figura 2.10 - Precisão e acuidade do Ekahau RTLS (Ekahau, 2013)	17
Figura 2.11 - Tags de Equipamentos (Centrak, 2013)	21
Figura 2.12 - Tags dos Funcionários (Centrak, 2013)	21
Figura 2.13 - Tags dos Pacientes (Centrak, 2013)	21
Figura 2.14 - Tags da Temperatura e Humidade respetivamente (Centrak, 2013)	22
Figura 3.1 - Diagrama de Estados da Tag.....	28
Figura 3.2 - Estratégia de recolha de dados controlada pelo acelerómetro	28
Figura 4.1 - Arquitetura Geral.....	29
Figura 4.2 - Arquitetura da tag	30
Figura 4.3 - Estrutura da mensagem enviada pela tag	31
Figura 4.4 - Conteúdo do campo Data da mensagem	31
Figura 5.1 - Cabo de carregamento da bateria Electric imp.....	38
Figura 5.2 - Montagem na breadboard para a implementação	38
Figura 5.3 - Placa PCB para a conexão do acelerómetro e Electric imp	39
Figura 5.4 - Tag Electric imp	39
Figura 5.5 - Flyport Wi-Fi conectado na placa Grove.....	41
Figura 5.6 - Tag Flyport Wi-Fi	41
Figura 5.7 - Excerto de um ficheiro txt guardado no servidor.....	42

Figura 5.8 - Protocolo SPI (Electric imp, 2014)	44
Figura 5.9 - Protocolo I ² C (Electric imp, 2014)	45
Figura 5.10 - Arquitetura Electric imp (Electric imp, 2014)	46
Figura 5.11 - Sensor Ótico do Electric imp (Electric imp, 2014)	47
Figura 5.12 - IDE Electric imp	48
Figura 5.13 - Envio dos dados para o Agent	49
Figura 5.14 - Amostragem periódica Electric imp	50
Figura 5.15 – Configuração Acelerómetro ADXL345.....	51
Figura 5.16 - Algoritmo detecção atividade / inatividade Electric imp.....	52
Figura 5.17 – Algoritmo final Electric imp.....	54
Figura 5.18 - IDE Openpicus Flyport.....	55
Figura 5.19 - Http Post Flyport Wi-Fi.....	57
Figura 5.20 - Algoritmo quinta fase	58
Figura 5.21 - Configuração acelerómetro MM7660FC	59
Figura 5.22 – Algoritmo detecção atividade / inatividade Flyport Wi-Fi.....	60
Figura 5.23 - Algoritmo Final FLYport Wi-Fi	62
Figura 6.1 - Medição 1 scan com LEDs e Prints	65
Figura 6.2 - Electric imp 1 scan	65
Figura 6.3 - Medição dois scans Electric imp ativados pelo acelerómetro	66
Figura 6.4 - Medição três scans ativados pelo timer	67
Figura 6.5 - Funcionamento geral Electric imp.....	68
Figura 6.6 - Medição um scan Flyport Wi-Fi ativado pelo acelerómetro com todos os canais.....	72
Figura 6.7 - Medição 1 scan acelerómetro 1 canal	73
Figura 6.8 - Medição 3 scans Timer Openpicus	73
Figura 6.9 - Funcionamento geral Flyport Wi-Fi	74
Figura 6.10 - Algoritmo autonomia	79
Figura 6.11 - Autonomia Electric imp	80
Figura 6.12 - Percentagem Electric imp.....	80
Figura 6.13 - Autonomia Flyport Wi-Fi.....	81
Figura 6.14 - Percentagem Flyport Wi-Fi.....	82

Índice de Tabelas

Tabela 5.1 - Tabela comparativa do Hardware disponível no mercado	34
Tabela 5.2 - Lista de compras.....	36
Tabela 6.1 - Comparação dos consumos anunciados e medidos Electric imp	70
Tabela 6.2 - Comparação dos consumos anunciados e medidos Flyport Wi-Fi.....	76

Índice de Acrónimos e Abreviaturas

AOA	<i>Angle Of Arrival</i>
AP	<i>Access Point</i>
API	<i>Application Programming Interface</i>
BSSID	<i>Basic Service Set IDentification</i>
CSV	<i>Comma-Separated Values</i>
FTDI	<i>Future Technology Devices International</i>
GPS	<i>Global Positioning System</i>
HTTP	<i>HyperText Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i>
I ² C	<i>Inter-Integrated Circuit</i>
ITS	<i>Indoor Triangulation System</i>
LAN	<i>Local Area Network</i>
LED	<i>Light Emitting Diode</i>
MAC	<i>Media Access Control</i>
MISO	<i>Master-Input Slave-Output</i>
MIT	<i>Massachusetts Institute of Technology</i>
MOSI	<i>Master-Output Slave-Input</i>
NTP	<i>Network Time Protocol</i>
PCB	<i>Printed Circuit Board</i>
PHP	<i>PHP: Hypertext Preprocessor</i>
RAM	<i>Random Access Memory</i>
RFID	<i>Radio-Frequency IDentification</i>
ROM	<i>Read Only Memory</i>
RSSI	<i>Received Signal Strength Indication</i>
RTLS	<i>Real Time Location System</i>
SCL	<i>Serial Clock Line</i>

SCLK	<i>Clock from Master</i>
SD	<i>Secure Digital</i>
SDA	<i>Serial Data Line</i>
SNTP	<i>Simple Network Time Protocol</i>
SPI	<i>Serial Peripheral Interface</i>
SS	<i>Slave-Select</i>
SSID	<i>Service Set Identifier</i>
TCP	<i>Transmission Control Protocol</i>
TDOA	<i>Time Difference Of Arrival</i>
WLAN	<i>Wireless Local Area Network</i>
USB	<i>Universal Serial Bus</i>
VBA	<i>Visual Basic for Applications</i>

1. Introdução

Neste primeiro capítulo é feito o enquadramento e abordagem ao tema da dissertação, bem como os objetivos que se pretendem alcançar com a realização da mesma. Na última secção é apresentada a estrutura adotada na elaboração deste documento.

1.1. Enquadramento

Depois de vários anos de investigação e desenvolvimento na área dos sistemas de posicionamento para ambientes interiores, começam agora a surgir várias aplicações destas tecnologias em diversas áreas, tais como monitorização de pessoas que vivem sozinhas, gestão de equipamentos em ambiente hospitalar, ou mesmo no rastreio de pessoas para efeitos de segurança.

Apesar da tecnologia de posicionamento em ambientes interiores ter evoluído bastante nos últimos anos, continua a verificar-se que existem poucos dispositivos móveis de posicionamento (vulgarmente designados por *tags*) disponíveis para integração noutros sistemas. Adicionalmente, verifica-se que as capacidades das *tags* existentes são reduzidas e acima de tudo, limitadas no que se refere à comunicação com sistemas abertos.

Ao longo dos anos, tem-se assistido a uma grande expansão das redes Wi-Fi, tornando-se assim na tecnologia de redes de área local sem fios mais utilizada atualmente. Cada vez mais os espaços públicos, empresas e habitações dispõem de redes Wi-Fi, tornando a utilização desta infraestrutura bastante apelativa para outras aplicações que não seja apenas o acesso à Internet ou a função de rede local.

O desenvolvimento de soluções de posicionamento em ambientes interiores pode ser feito através de várias tecnologias como, por exemplo, ultrassons, infravermelhos, ZigBee etc. No entanto, a poupança de tempo e o baixo custo de implantação proporcionada pelo Wi-Fi, torna os sistemas RTLS (*Real Time Location System*) Wi-Fi bastante atraentes em relação a sistemas RTLS que utilizam outras tecnologias.

Neste contexto, o objetivo deste projeto de dissertação é o desenvolvimento de uma *tag* para sistemas de posicionamento interior em redes Wi-Fi.

1.2. Objetivos

O objetivo deste trabalho é conceber uma *tag* Wi-Fi para sistemas de posicionamento interior em redes Wi-Fi, que possa ser integrada num motor de posicionamento já existente (e parcialmente desenvolvido no contexto de uma dissertação de mestrado do MIECOM). A *tag* a desenvolver deve cumprir um conjunto de requisitos: tamanho reduzido, possibilidade de ser utilizada em qualquer rede Wi-Fi, e possuir uma autonomia aceitável, o que implica um muito baixo consumo de energia.

1.3. Abordagem

A *tag* tem de ser capaz de realizar leituras do sinal Wi-Fi e enviar os dados recolhidos para um motor de posicionamento, para que este determine, a partir da leitura efetuada, a localização da *tag*. Tal como foi referido anteriormente, um dos principais requisitos da *tag* é a sua autonomia: deve ser a mais elevada possível, de forma que possa ser utilizada por longos períodos de tempo sem haver necessidade de frequentemente recarregar ou substituir a bateria que a alimenta. Para que se consiga obter um baixo consumo de energia foram consideradas e estudadas várias opções, incluindo o uso de acelerómetros e a definição do intervalo de tempo durante o qual a *tag* se encontra “adormecida”.

No hardware da *tag* foi considerado a utilização de um acelerómetro para detetar quando é que esta é movimentada. Não se detetando movimento, a *tag* pode ser ‘adormecida’ durante algum tempo, isto é, passar a efetuar leituras do ambiente rádio com uma frequência mais reduzida, diminuindo, de forma significativa, o consumo de energia. Existindo movimento (detetado pelo acelerómetro) então a *tag* necessita de reportar a sua posição com uma frequência superior, para que a sua posição atual seja sempre conhecida.

No sentido de facilitar o processo de desenvolvimento da *tag* Wi-Fi, realizou-se um estudo do hardware existente no mercado, de modo a identificar um *kit* base para o

desenvolvimento da *tag*. Para determinar a localização da *tag* foi usado um motor de posicionamento já existente, tal como referido anteriormente.

O motor de posicionamento existente funciona com base na leitura rádio do interface Wi-Fi dos dispositivos móveis. Com base no nível de sinal dos pontos de acesso observáveis num determinado local, é determinada a localização do dispositivo. O seu funcionamento baseia-se num complexo algoritmo de posicionamento e no conhecimento prévio que o motor possui do ambiente rádio de cada edifício. Assim, a sua utilização está restrita aos edifícios que foram previamente estudados e para os quais foram previamente elaborados mapas de leituras rádio.

1.4. Estrutura da Dissertação

A presente dissertação encontra-se dividida em sete capítulos. Neste primeiro capítulo é realizado o enquadramento e uma abordagem ao tema da dissertação, bem como os objetivos que se pretendem alcançar. Na última secção é apresentada a estrutura adotada na elaboração deste documento, finalizando desta forma o primeiro capítulo.

No segundo capítulo é apresentado o estudo do estado da arte relativo aos sistemas de posicionamento Wi-Fi interior, relativamente às técnicas mais utilizadas, aos vários produtos comerciais existentes, onde se destacam a oferta de soluções completas e *tags* existentes no mercado.

O terceiro capítulo incide no estudo do problema da dissertação, onde é feita a identificação do problema e o estudo dos requisitos da *tag* a desenvolver. Na última secção é feita a análise da recolha de dados efetuada pela *tag*, bem como a identificação de procedimentos que conduzam a uma poupança energética.

No quarto capítulo é apresentada a arquitetura de todo o sistema, explicando cada um dos componentes que o constituem, dando mais ênfase à *tag*. É ainda apresentada a estrutura das mensagens enviadas pela *tag* para o servidor.

No quinto capítulo são descritas as várias fases que dizem respeito à implementação da *tag*. Este capítulo é composto por três secções, onde inicialmente é descrita a fase da escolha do hardware e nas restantes são apresentadas as etapas necessárias para a construção da *tag*, com foco no hardware e software em secções distintas.

No capítulo seis é exposta a avaliação do sistema, apresentando-se os resultados obtidos nos testes efetuados.

Por fim, no último capítulo são apresentadas as conclusões do trabalho desenvolvido e são apresentadas algumas propostas para uma futura evolução e melhoria da solução desenvolvida.

2. Estado da Arte

Neste capítulo é descrito o estado da arte dos sistemas de posicionamento Wi-Fi interior, apresentando-se as técnicas mais usadas, os vários produtos comerciais existentes destacando-se as ofertas de soluções completas bem como algumas *tags* desenvolvidas.

2.1. Técnicas de Localização Wi-Fi

Nesta secção, serão apresentadas e descritas as técnicas mais relevantes usadas em posicionamento.

- TDOA (*Time Difference Of Arrival*)

Esta técnica consiste na determinação da posição da *tag* obtida pela diferença de tempo entre o momento da emissão do sinal e a sua chegada a múltiplos recetores (pontos de acesso) (Liu & Darabi, 2007). A distância entre a *tag* e o AP (*Access Point*) é diretamente proporcional ao tempo necessário para a deslocação do sinal entre os dois. Este método é menos utilizado devido ao facto de ser necessário sincronizar com precisão os relógios dos dispositivos dentro do sistema, todavia, quando implementado corretamente é um método preciso (Zegelin, 2003).

Esta técnica apresenta a desvantagem de sofrer do problema de múltiplos caminhos¹ e por isso fundamental um bom posicionamento dos APs (*Access Points*) para que a precisão seja a melhor possível (Yamasaki, Ogino, & Tamaki, 2005).

- AOA (*Angle Of Arrival*)

O fator mais importante para que seja possível calcular a localização das *tags* através desta técnica é a determinação da direção com que o sinal chega aos pontos de acesso. Para determinar a direção é necessário uma antena direcional bem como pelo menos dois pontos

¹ Múltiplos caminhos (*multipath*) é um problema que afeta a propagação dos sinais rádio e que se pode resumir no facto de os sinais rádio chegarem ao recetor por múltiplos caminhos. Por ser provado por várias causas, incluindo a reflexão dos sinais em corpos físicos como paredes, montanhas, etc.

distintos para determinar a localização da *tag* (são necessários pelo menos dois pontos de acesso).

Os pontos de acesso recebem o sinal enviado pela *tag* e calculam o ângulo no qual o sinal recebido é mais forte, assim torna-se possível calcular a localização da *tag* através da interseção das linhas definidas pelos ângulos.

Utilizando apenas esta técnica é difícil obter um bom nível de precisão no cálculo da localização interior, devido, entre outros, ao problema de múltiplos caminhos (Russell, 2003). Já em ambientes exteriores livres esta técnica funciona melhor. De forma a conseguir-se obter uma boa precisão, por vezes, implementam-se sistemas híbridos, constituído por várias tecnologias, desde o RFID - *Radio-Frequency Identification* até aos ultrassons.

- *Wi-Fi Fingerprinting*

Esta é a técnica que tem obtido maior destaque, sendo a mais utilizada nos sistemas de posicionamento Wi-Fi interior (Bahl & Padmanabhan, 2000).

A técnica Wi-Fi *fingerprinting* consiste em observar o ambiente rádio, obtendo-se o RSSI - *Received Signal Strength Indication* dos APs detetados pela interface Wi-Fi do equipamento. Por cada leitura do ambiente rádio é criado um *fingerprint* que é constituído pela lista dos APs visíveis e respetivos RSSI. O posicionamento com Wi-Fi *fingerprint* assume que existe uma base de dados, designada por mapa de rádio, onde se encontram registados *fingerprints* recolhidos nos vários locais onde se pretende dispor de um serviço de posicionamento. Se a leitura rádio efetuada pelo dispositivo for suficientemente parecida com uma ou mais leituras existentes na base de dados então considera-se que esse dispositivo se encontra no local físico associado ao registo da base de dados que é mais parecido com a *fingerprint*.

O primeiro passo na realização de posicionamento interior usando o Wi-Fi *fingerprinting* consiste na construção de um mapa de rádio para a área de posicionamento. Este contém um determinado número de amostras, que representam a cobertura Wi-Fi dentro da área de posicionamento (Eisa & Peixoto, 2013). Posteriormente, estes dados armazenados no mapa de rádio são utilizados para comparar as medições feitas pelo dispositivo (neste caso a *tag*) possibilitando assim a determinação da localização do dispositivo.

Esta técnica é constituída por duas fases: fase de calibração (offline) e a fase de posicionamento (online).

Fase de Calibração (offline):

Nesta fase é construído um mapa de rádio com os valores de RSSI associado aos pontos de acesso que se encontram nas proximidades do local (ou locais) em que se pretende dispor de um serviço de posicionamento. A estas medidas são associadas coordenadas físicas e/ou nomes. Os diversos pontos onde é feita a recolha chamam-se pontos de calibração. Este mapa de rádio será utilizado durante a fase de posicionamento (online) para determinar a posição do dispositivo (Eisa & Peixoto, 2013).

Dependendo do tamanho e da disposição do edifício são escolhidos um certo número de pontos. Para cada um destes pontos são feitas várias medições devido ao valor do RSSI ser afetado por vários fatores, incluindo pela posição do utilizador, e.g. se o corpo do utilizador está entre a *tag* e o ponto de acesso é provável que o RSSI seja menor do que quando o utilizador está virado para o lado oposto, pois o sinal sofre atenuação no corpo humano. A diferença entre estas posições foi determinada em cerca de 5 dB (Bahl & Padmanabhan, 2000). Geralmente estas medições são efetuadas em quatro direções diferentes (Retscher, Moser, Vredeveld, & Heberling, 2006).

Fase de Posicionamento (online):

Esta é a fase em que a *tag* obtém informações sobre os APs que se encontram na sua vizinhança e as envia para o servidor. A *tag* faz leituras Wi-Fi periodicamente e retira várias informações de cada AP que encontra na sua vizinhança. As informações retiradas são o SSID (*Service Set Identifier*), o BSSID (*Basic Service Set Identification*) e o RSSI correspondente a cada AP. A estas informações é adicionado um *timestamp* gerado no momento em que a *tag* realiza cada leitura Wi-Fi. De seguida os dados são empacotados e enviados para o servidor. Este recebe os dados e compara-os com os valores que constituem o mapa de rádio. Com o recurso a alguns cálculos é possível determinar o posicionamento da *tag* naquele instante (Mok & Retscher, 2007).

2.2. Produtos Comerciais

Nesta secção, serão apresentadas e descritas as soluções completas de posicionamento Wi-Fi interior existentes no mercado. Será possível verificar que alguns sistemas são constituídos por várias tecnologias.

2.2.1. Soluções Completas

Os sistemas de monitorização RTLS têm sofrido um enorme crescimento, tanto no desenvolvimento de hardware como no de software. O mercado está cada vez mais competitivo e começam a formar-se bastantes empresas com soluções nesta área. Estas soluções podem ser constituídas por uma vasta gama de tecnologias, desde o ZigBee passando pelos infravermelhos e pelo Wi-Fi.

Neste capítulo analisa-se algumas das soluções comerciais que estão disponíveis no mercado, com maior destaque para as soluções que utilizam a tecnologia Wi-Fi.

2.2.1.1. *AeroScout*

A *AeroScout* (Aeroscout, 2013) disponibiliza soluções para uma grande variedade de situações que vão desde a área da saúde até à educação, passando pela indústria em geral até às empresas de transporte e logística. Os sistemas desenvolvidos pela AeroScout podem ser integrados na maior parte das redes Wi-Fi, proporcionando assim uma implementação simples e um custo mais reduzido do que se fosse necessário uma infraestrutura dedicada.

As soluções desenvolvidas integram algumas tecnologias nomeadamente, Wi-Fi, RFID, GPS, *Low Frequency*, ultrassom, RFID passivo, sensores e telemetria². O sistema está preparado para ser utilizado tanto em ambientes interiores como exteriores e é composto por quatro componentes: o *MobileView*, as *tags*, a *Infrastructure* e; o *Location Engine*.

As *tags RFID* ativas e/ou os dispositivos do cliente enviam um sinal muito curto num intervalo regular para os pontos de acesso e de seguida estes dados são enviados para o *Location Engine*. Este, por sua vez, determina as coordenadas de localização através do RSSI ou do TDOA e de seguida envia a localização e estado para o *MobileView*. O *MobileView* fornece uma plataforma para todos os dados e permite ver o mapa, gerir os alertas, etc.

De seguida é explicado cada um destes componentes.

² Telemetria corresponde à medição da temperatura, humidade, exposição ao gás e movimento.

AeroScout MobileView

É a aplicação web onde o utilizador pode acompanhar graficamente em tempo real as *tags* e respetivo estado, gerir e configurar o sistema tendo em conta as opções disponibilizadas pela AeroScout tais como alertas e a possibilidade de trabalhar com aplicações de terceiros.

AeroScout Tags

Todas as *tags* Wi-Fi da *AeroScout* são de reduzidas dimensões, constituídas por sensores (de movimento e temperatura), botões de chamada, telemetria e caracterizam-se por disponibilizar uma longa duração e ser compatíveis com a maior parte das redes Wi-Fi. A AeroScout disponibiliza 4 modelos de *tags*, as T2, T3, T4 e as T5:

- **T2:** Estas *tags* contêm uma bateria de longa duração e utilizam a rede Wi-Fi para a comunicação. São ideais para monitorizar equipamentos, pessoas e são separadas em três modelos distintos, T2u, T2s e T2EB.

T2u: Esta *tag* é ideal para monitorizar pessoas, dispositivos médicos, contentores e veículos. É possível programar até 15 mensagens de 15 bytes cada uma; estas mensagens podem ser programadas pelo *Tag Management Suite* ou por um *Exciter* quando a *tag* se encontra na sua proximidade. Contém 3 LED (*Light Emitting Diode*), sensor de movimento e telemetria e disponibiliza uma autonomia máxima de 4 anos.

T2s: Esta *tag* pode ser usada por pacientes, funcionários e anexada a muitos equipamentos tais como dispositivos médicos. Possui um sensor de movimento, botão de alerta, recetor de baixa frequência e opcionalmente um recetor de ultrassom. O recetor de ultrassom é ideal para se obter uma boa precisão dentro de uma divisão dado que os sinais não ultrapassam paredes, fornecendo assim com precisão o local onde se localiza a pessoa ou o equipamento desejado. Disponibiliza uma autonomia máxima de 4 anos.

T2EB: Esta *tag* destina-se a ambientes e equipamentos industriais sendo bastante resistente aos choques. Possui uma bateria de longa duração, permite a programação de 10 mensagens de 10 bytes cada e contém sensor de movimento. Esta *tag* disponibiliza uma autonomia máxima de 8 anos (7,5 anos se transmitir a cada 5 minutos, 6 anos se transmitir a cada 1 minuto, 4,5 anos se transmitir a cada 30 segundos).

- **T3:** A *tag* T3 apresenta um formato de reduzidas dimensões, é constituída por uma bateria de longa duração, vários botões para alertas, sensor de movimento, indicadores de LED, telemetria e permite o funcionamento em praticamente qualquer ambiente, desde interior a exterior. Disponibiliza uma autonomia máxima de 4 anos.
- **T4:** A grande diferença desta *tag* em relação às outras, é o facto de possibilitar a comunicação bidirecional. Esta *tag* tem dois modelos distintos, um direcionado para a área da saúde (T4b) e outro para um ambiente industrial (T4p). Ambos os modelos têm uma bateria de longa duração que permite uma autonomia máxima de 4 anos.

T4b: Esta *tag* pode ser usada para determinar a localização e o estado de bens móveis e pessoas. As suas mensagens são recebidas e processadas pelos pontos de acesso Wi-Fi. Um exemplo da aplicação deste modelo pode ser um paciente num hospital enviar um alerta para solicitar assistência, utilizando o botão de chamada e de seguida receber o reconhecimento de que a ajuda está a caminho através de vibração ou de sinais emitidos pelos LEDs.

T4p: Um exemplo do funcionamento desta *tag* é um operário numa fábrica usar um botão de chamada para enviar um pedido de reabastecimento de peças e, posteriormente, recebe diferentes indicações visuais de que o pedido foi recebido e está em processamento. De seguida o trabalhador recebe atualizações de estado através dos LED intermitentes na *tag*, indicando que as peças estão a caminho. Quando as peças chegam ao operário é feito um *reset* na *tag*.

- **T5:** Esta *tag* foi desenvolvida com objetivo de monitorizar as temperaturas do local onde se encontram, caracterizando-se por uma maior resistência face às temperaturas. Um exemplo de aplicação desta *tag* é um *data center* constituído por uma grande quantidade de hardware, onde a temperatura ou a humidade podem ser fundamentais para que o equipamento não se danifique. Esta *tag* pode monitorizar remotamente as temperaturas e o nível de humidade verificando se estas se mantêm dentro da normalidade. É possível configurar o envio automático de mensagens num intervalo de tempo desde cinco segundos até três horas e meia. Disponibiliza uma autonomia máxima de 4 anos.

AeroScout Infrastructure

A principal infraestrutura utilizada pela *AeroScout* é qualquer rede Wi-Fi que hoje em dia é possível encontrar em praticamente qualquer lugar, e dependendo das necessidades dos clientes oferece uma grande variedade de opções para a complementar com uma infraestrutura adicional.

A *AeroScout* tem uma parceria com os principais fornecedores de equipamentos de WLAN – Wireless LAN para que a integração entre as *tags* e os produtos de *networking* seja simplificada. Os pontos de acesso têm uma função muito importante no sistema de localização, pois são utilizados como leitores das *tags*, tornando possível a localização destas pelo sistema e permitindo assim utilizar apenas uma rede para serviços de dados, voz e localização.

Como a precisão da localização depende das necessidades dos clientes e estas variam de caso para caso, a leitura das *tags* pode ser complementada com outros equipamentos, para que seja possível a adoção de outras técnicas de localização tais como a TDOA. Para isso, a *AeroScout* disponibiliza os *Exciters*, os *Location Receivers* e os *Location Points*.

Os *AeroScout Exciters* têm como função acionar imediatamente as *tags* quando estas passam num sítio previamente definido (e.g., um ponto de saída ou entrada de uma sala). Ao passar no local, as *tags* enviam uma mensagem para os pontos de acesso informando que se encontram naquele local.

Os *AeroScout Location Receivers* fornecem capacidades robustas e sofisticadas de medição para ambientes ao ar livre e mais adversos que requerem algoritmos de localização TDOA. Estes recebem transmissões das *tags*, de seguida executam medições de RSSI, realizam alguns cálculos e enviam estes dados para o *AeroScout Location Engine* para que este determine a localização da *tag*. Os *AeroScout Location Points* funcionam como ponto de acesso permitindo assim aumentar a cobertura da rede e funcionar ao mesmo tempo como recetor de RFID e como ponto de acesso.

A instalação destes equipamentos é facilitada pela comunicação com o *Location Engine* podendo ser realizada tanto por *Wireless* como por *Ethernet*, não necessitando de qualquer infraestrutura dedicada. Estes equipamentos conseguem realizar cerca de 300 medições de localização por segundo, o que os torna bastante capazes de satisfazer as aplicações mais exigentes.

AeroScout Location Engine

É o componente que determina a localização das *tags*. Este componente processa as informações recebidas dos pontos de acesso, de forma a produzir os dados de localização das *tags* ou de outros dispositivos equipados com interfaces Wi-Fi. A localização é determinada com recurso ao RSSI e a algoritmos de localização TDOA.

Possui ferramentas para a análise de precisão de localização, para a cobertura Wi-Fi, diagnóstico do sistema, calibração automática que proporciona uma melhoria da precisão de localização e possibilita o ajuste do sistema em tempo real tendo em conta as alterações físicas da área monitorizada. Permite localizar qualquer dispositivo que disponha de Wi-Fi sem a instalação de qualquer software no dispositivo.

2.2.1.2. Ekahau

A Ekahau (Ekahau, 2013) disponibiliza soluções para uma grande variedade de áreas como a educação e a indústria em geral, sendo que a sua principal área de negócio está centrada na área da saúde. As comunicações são feitas através das redes Wi-Fi existentes, proporcionando uma poupança e uma instalação mais simples do que aquela que ocorreria caso fosse necessário implementar uma infraestrutura específica para o RTLS.

Sistema Ekahau RTLS

As soluções desenvolvidas pela Ekahau podem ser divididas em três componentes, as *tags* Wi-Fi, a rede Wi-Fi e a aplicação Ekahau Vision, como é possível observar na Figura 2.1.

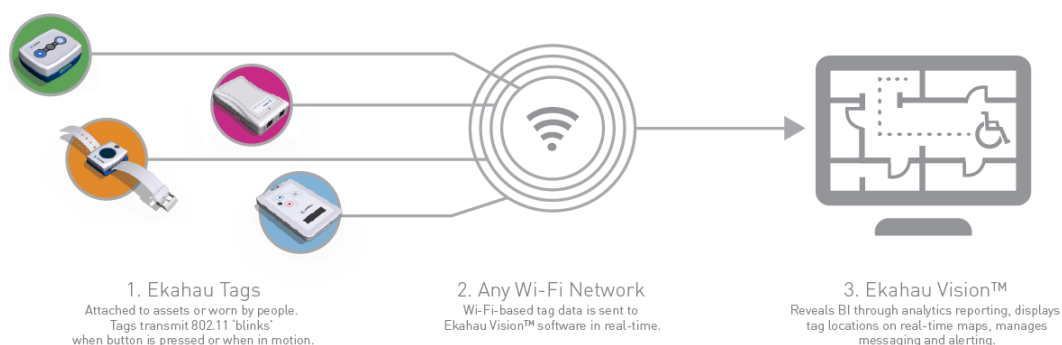


Figura 2.1 - Visão geral do Ekahau RTLS (Ekahau, 2013)

Cada componente desempenha várias tarefas. As *tags*, como qualquer outro cliente Wi-Fi (e.g. computador portátil), transmitem informação para os APs quando um botão é pressionado ou é detetada movimentação da mesma. Os APs, por sua vez, recebem estas informações bem como as medições do RSSI que as *tags* efetuam e enviam os pacotes com a informação do RSSI para a aplicação Ekahau Vision, de forma a que os algoritmos de localização proprietários da Ekahau determinem a localização da *tag*. Esta aplicação recebe, guarda e disponibiliza a informação recebida para que seja possível, entre outras coisas, realizar relatórios, visualizar em tempo real a localização das *tags*, e gerir mensagens e alertas.

O sistema não tem qualquer impacto no desempenho da rede, visto que as mensagens enviadas pelas *tags* têm um tamanho compreendido entre 60 e 120 bytes e a duração da comunicação entre as *tags* e os APs é inferior a 2 segundos. É caracterizado por uma fácil e relativamente rápida instalação (valor médio de 11 dias, como é possível observar na Figura 2.2) através de seis passos, onde o mais demorado é o processo de calibração.



Figura 2.2 - Implementação Ekahau RTLS (Ekahau, 2013)

Tags Ekahau

O sistema RTLS desenvolvido pela Ekahau disponibiliza 8 modelos distintos de tags e um emissor de infravermelhos caso seja necessária uma maior precisão na localização. Todos eles têm uma bateria de longa duração e comunicam através da tecnologia Wi-Fi.

Tag Ekahau A4: A Ekahau A4 é uma *tag* RFID ativa que oferece uma alta precisão de localização, utilizando a rede Wi-Fi 802.11b/g e um sensor integrado de infravermelhos. Esta *tag* contém dois botões configuráveis, usados para o envio de alarmes ou notificações. Pode ser configurada e gerida remotamente. É possível acionar remotamente os LEDs e a buzina da *tag* para identificar a *tag* correta ou alertar o utilizador desta. Disponibiliza uma autonomia máxima de 5 anos



Figura 2.3 - Ekahau A4 (Ekahau, 2013)

Tag Ekahau A4+: Para além das características da *tag* A4, a *tag* A4+ utiliza baterias externas ou fontes de energia, como carregadores USB que proporcionam uma maior autonomia.



Figura 2.4 - Ekahau A4+ (Ekahau, 2013)

Tag Ekahau B4: Esta *tag* possui sensor de movimento, um ecrã para leitura de mensagens de texto e três botões onde dois deles permitem a escrita de mensagens e o outro serve como botão de alarme.



Figura 2.5 - Ekahau B4 (Ekahau, 2013)

Tag Ekahau W4: A *tag* W4 é usada confortavelmente no pulso de uma pessoa como se de um relógio se tratasse. Esta *tag* é à prova de água, contém um led de duas cores, alarme vibratório, um botão de alarme personalizável para o envio de mensagens de emergência ou de estado e

um sensor de movimento. Disponibiliza uma autonomia máxima de 500h (com transmissão a cada 15s).



Figura 2.6 - Ekahau W4 (Ekahau, 2013)

Tag Ekahau TS1 e TS2: As *tags* TS1 e TS2 apenas diferem entre si no facto de a TS1 ser equipada com um sensor de temperatura enquanto a TS2 é equipada com dois. Ambas têm um LED que sinaliza o estado de alarme. Por defeito enviam mensagens com a informação sobre a temperatura a cada 15 segundos, mas este valor pode ser configurado. Disponibilizam uma autonomia máxima de 5 anos.



Figura 2.7 - Ekahau TS1 e TS2 respetivamente (Ekahau, 2013)

Tag Ekahau TS-c: Esta *tag* está equipada com um sensor de temperatura. A comunicação é feita através de Wi-Fi e envia alertas caso a temperatura esteja fora do limite configurado pelo utilizador. A cada 15 segundos são enviadas atualizações da temperatura, podendo este valor ser configurado pelo utilizador através da aplicação. Esta *tag* difere das anteriores nas temperaturas que consegue medir (mais críticas) e no facto de não ser necessário utilizar as “pontas de prova” pois o sensor está integrado na própria *tag*. Disponibiliza uma autonomia máxima de 2 anos.



Figura 2.8 - Ekahau TS-c (Ekahau, 2013)

Tag Ekahau HS1: A *tag* HS1 caracteriza-se pelos seus sensores de humidade, temperatura e pelo alarme em formato de som e LED a piscar. Por omissão envia mensagens com a informação da temperatura e humidade a cada 15 segundos, mas este valor pode ser configurado. Disponibiliza uma autonomia máxima de 5 anos.



Figura 2.9 - Ekahau HS1 (Ekahau, 2013)

Ekahau Vision

A plataforma Ekahau Vision, juntamente com as *tags*, oferece a possibilidade dos gestores e funcionários verificarem a localização, o estado e a condição dos equipamentos e pessoas através de um browser. A Ekahau Vision determina e apresenta, em tempo real, a localização das *tags* sendo esta calculada através de leituras *fingerprinting*. Os algoritmos da Ekahau calculam a localização através de métodos probabilísticos.

Quando esta plataforma recebe as leituras do RSSI enviadas pelas *tags*, compara-as com os valores armazenados numa base de dados e obtém uma localização provável num mapa virtual.

Em relação à acuidade, dependendo dos objetivos dos clientes, pode ser necessária a utilização de um sensor de infravermelhos para que se consiga uma acuidade de 1 até 3 metros como se pode observar na Figura 2.10. A solução Ekahau baseada apenas em Wi-Fi consegue uma acuidade de 1 a 3 metros.

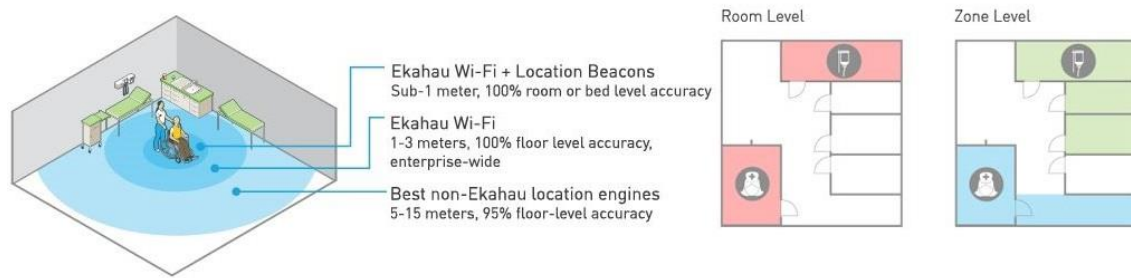


Figura 2.10 - Precisão e acuidade do Ekahau RTLS (Ekahau, 2013)

2.2.1.3. Navizon

O Navizon I.T.S. (*Indoor Triangulation System*) (Navizon, 2013) é um sistema de localização interior que fornece a localização de dispositivos com conectividade Wi-Fi, desde smartphones, tablets, computadores portáteis e *tags* Wi-Fi. Este permite localizar qualquer dispositivo que disponha de Wi-Fi sem a instalação de qualquer software no dispositivo.

Este sistema necessita da implantação de pequenos nós colocados a aproximadamente 100 metros de distância entre eles para cobrir um espaço tridimensional, que pode abranger um edifício de vários andares ou vários prédios num campus. Os nós não necessitam de fios, pois comunicam entre si por Wi-Fi. A localização dos dispositivos é obtida através do RSSI destes, detetada pelos nós do sistema. Posteriormente os dados são enviados para um sistema proprietário da Navizon na web.

Para ser possível verificar a localização dos dispositivos que se deseja controlar existem duas aplicações disponibilizadas pela Navizon: o Buddy Radar e o Floor Plan. Também é possível fazer uma integração com o Google Maps, de forma a poder verificar os dispositivos que se encontram numa determinada área.

O Buddy Radar é uma aplicação destinada a smartphones que oferece aos utilizadores uma visualização, em tempo real, da localização de outros dispositivos. O Floor Plan mostra uma vista aérea em duas dimensões dentro do edifício com a localização dos dispositivos.

Navizon Tags

As *tags* Wi-Fi disponibilizadas pela Navizon permitem monitorizar pessoas e equipamentos através do sistema RTLS Navizon I.T.S. Estas são constituídas por um transmissor de rádio, uma bateria de grande duração e podem ser anexadas a objetos ou transportadas por pessoas.

Existem dois modelos de *tags*. As compactas e as que se apresentam em formato de pulseira. Em qualquer uma delas a transmissão é feita através do IEEE 802.11b e a identificação de cada *tag* é conseguida pelo seu endereço MAC (48 bits).

As *tags* transmitem um sinal periódico que pode ser detetado por pontos de acesso Wi-Fi e a localização é determinada através de técnicas de triangulação baseada no RSSI.

Tag Compacta: As *tags* compactas possuem um botão que quando pressionado envia um sinal especial que o servidor pode interpretar como um alerta, por exemplo, para um paciente pedir auxílio, um técnico de uma sala de emergência chamar um assistente, entre outras. Estas *tags* possuem sensores de vibração que indicam quando a *tag* está em movimento e envia alertas automaticamente sempre que a bateria se encontra com carga insuficiente. Disponibiliza uma autonomia máxima de 2 meses.

Tag Pulseira: A *tag* pulseira é muito resistente e impermeável. Foi projetada para controlar pessoas, embora também possa ser utilizada para monitorização de animais e tem a capacidade de enviar alertas automaticamente sempre que a bateria se encontra com carga insuficiente. Disponibiliza uma autonomia máxima de 6 meses.

2.2.1.4. VIZZIA Technologies

A VIZZIA Technologies (VizziaTech, 2013) apresenta-se como uma especialista em soluções de tecnologia para a área da saúde e pretende ajudar os hospitais a cortar nos custos e a simplificar processos complicados, como a distribuição de equipamentos e a rapidez no atendimento dos utentes. Para isso propõe a utilização de RTLS e sistemas de monitorização RFID passivo.

Disponibiliza aos clientes bastantes tecnologias de forma a que o RTLS seja o mais adequado possível. Todos os sistemas RTLS dependem de radiação eletromagnética, sendo que as mais comuns de radiação eletromagnética são a RFID e Wi-Fi.

O erro na precisão da localização através do Wi-Fi ou do RFID é na ordem dos metros, no entanto a comunicação com as *tags* e a transmissão dos dados é feita sobre a infraestrutura Wi-Fi já existente, tornando os sistemas baseados nestas tecnologias bastante apetecíveis em termos económicos. Este erro pode repercutir-se numa falsa informação de localização, dando a

indicação de que um equipamento se encontra num determinado espaço (e.g. sala) quando na realidade se encontra noutra (fisicamente próximo).

De forma a evitar este problema e dependendo dos requisitos impostos pelos clientes, a VIZZIA também implementa sistemas híbridos. Estes sistemas combinam infravermelhos ou ultrassom com RFID ou Wi-Fi.

O sistema RTLS Wi-Fi funciona sobre uma infraestrutura Wi-Fi existente. As *tags* comunicam com os pontos de acesso enviando-lhes o seu número de série e estes calculam a sua localização através de triangulação. Posteriormente estes dados são enviados para um servidor que os armazena numa base de dados para que possam ser analisados/consultados.

Tag Wi-Fi

Os modelos utilizados pela VIZZIA são fornecidos por vários fabricantes de sistemas e componentes de Wi-Fi, incluindo a AeroScout (Wi-Fi) e a Aginova (RFID). Algumas delas são constituídas por sensores de temperatura e movimento. Disponibilizam uma autonomia máxima de 5 anos.

Na área da saúde, as *tags* Wi-Fi são tipicamente aplicadas em equipamentos médicos (e.g., bombas de infusão, cadeiras de rodas, etc.) e utilizadas pelos pacientes e pessoal clínico para que seja possível monitorizar a sua localização e respetivos movimentos.

Pontos de Acesso Wi-Fi

Como já foi referido, o sistema RTLS Wi-Fi utiliza a infraestrutura Wi-Fi existente. O sistema pode ser integrado com os pontos de acesso existentes mas, dependendo da precisão desejada e do número de *tags* em uso pode ser necessário instalar mais pontos de acesso. A VIZZIA disponibiliza também soluções híbridas tendo em conta a precisão desejada pelos clientes.

Como cada *tag* monitorizada é um “cliente” adicional na rede Wi-Fi então em determinadas circunstâncias (elevado número de *tags*) pode ser necessários adicionar pontos de acesso adicionais para evitar a sobrecarga da rede existente.

A VIZZIA Technologies disponibiliza em termos de software, tendo em conta diferentes finalidades, várias soluções. Todo o sistema se centra no VixPort.

VixPort

O VixPort é a solução de software proposta pela Vizzia que, dependendo das necessidades dos clientes, oferece a possibilidade de gestão de tags, monitorização de temperatura e a otimização da rapidez do atendimento dos pacientes. De seguida é apresentada a solução de gestão de equipamentos, visto ser a solução que melhor se enquadra no tema da dissertação.

Gestão de Equipamentos (Asset Management)

O Asset Management disponibiliza uma interface onde é possível realizar bastantes tarefas, das quais se destacam:

- A apresentação de um mapa interativo onde é possível verificar a localização de cada *tag*,
- Permite informar/avisar os utilizadores quando uma *tag* (e.g., equipamento equipado com *tag*) sai para o exterior do edifício em que se encontra ou quando sai para fora de uma determinada área. Quando ocorre um destes casos, são gerados e enviados alertas por email ou por SMS;
- Função de pesquisa permite aos utilizadores procurar por equipamentos específicos, equipamentos limpos ou sujos, equipamentos em movimento, etc;
- Os utilizadores podem gerar relatórios para visualizar as taxas de utilização de equipamentos, entre outros dados.

2.2.1.5. Centrak

A Centrak (Centrak, 2013) disponibiliza soluções para a área da saúde e combina várias tecnologias. Destas soluções destacam-se a monitorização e gestão de equipamentos, a monitorização do ambiente, o controlo da higienização das mãos e a rapidez no atendimento dos pacientes.

Monitorização e gestão de equipamentos

A monitorização e gestão de equipamentos têm uma ligação direta com a produtividade dos funcionários. Com este serviço é possível determinar a localização de um determinado equipamento (e.g., equipamento de medição da tensão arterial) e se este se encontra em utilização.

De forma a monitorização ser possível, é necessário que cada equipamento esteja equipado com uma *tag*. As *tags* apresentadas na Figura 2.11 comunicam através de Wi-Fi e são constituídas por um botão personalizável e um LED. A sua localização é determinada através de Wi-Fi mas dependendo do nível de precisão desejada, pode ser auxiliada pelo Gen2IR (infravermelhos).



Figura 2.11 - Tags de Equipamentos (Centrak, 2013)

Staff Tags: As *tags* observadas na Figura 2.12 destinadas aos funcionários diferem das *tags* de equipamentos no tamanho (mais finas e leves), e no facto de serem constituídas por 3 botões personalizáveis.



Figura 2.12 - Tags dos Funcionários (Centrak, 2013)

Patient Tags: As *tags* observadas na Figura 2.13 destinadas aos pacientes diferem da dos funcionários apenas no facto de disponibilizarem um único botão.



Figura 2.13 - Tags dos Pacientes (Centrak, 2013)

Temperature & Humidity Tags: As *tags* apresentadas na Figura 2.14 para a monitorização da temperatura e da humidade diferem das anteriores no facto de serem constituídas por sensores de temperatura e de humidade.



Figura 2.14 - Tags da Temperatura e Humidade respetivamente (Centrak, 2013)

Multi-Mode Technology

A Centrak Multi-Mode Technology é uma solução híbrida que combina localização Wi-Fi, infravermelhos (Gen2IR) e baixa frequência. Dependendo da precisão necessária, é possível posicionar dispositivos Gen2IR para que a localização seja a mais acertada possível independentemente dos locais onde são necessários, incluindo quartos, corredores, entre outros. Também suporta o uso de dados de localização de clientes Wi-Fi, incluindo computadores, *smartphones*, etc.

2.3. Tags

Depois da análise feita às soluções comerciais, destaca-se agora o hardware. O grande interesse e foco das empresas concentram-se na apresentação de soluções completas (hardware e software), com o objetivo de fornecer uma solução completa e, eventualmente, obter um lucro maior. No entanto há alguns produtores que apenas se focam no hardware (*tags*), e é a estes que é feita referência nesta secção.

As empresas que irão ser analisadas em seguida são a Redpine Signals e a Vestac. A Aeroscout também produz *tags*, mas estas já foram apresentadas na secção anterior.

Redpine Signals

A Redpine Signals (RedPineSignals, 2013) disponibiliza duas *tags* (WM1-20 e WM1-50) que diferem entre si apenas na banda de frequências em que funcionam e disponibiliza também um configurador (WMO).

WM1-20: Além da localização, a *tag* fornece informações de telemetria. É constituída por um botão de emergência que pode ser utilizado para configurar o sistema de alarmes e notificações. O consumo energético pode ser otimizado, configurando a frequência da troca de mensagens quando a *tag* se encontra em repouso. A autonomia máxima desta *tag* é de 5 anos (com transmissão a cada 5 minutos).

WM1-50: As características do WM1-50 são as mesmas do WM1-20, acrescentando a compatibilidade com banda de frequências 5GHz. A autonomia máxima desta *tag* é de 5 anos (com transmissão a cada 5 minutos).

WMO – Configurador: O WMO permite a configuração sem fios das *tags* quando estas se encontram a cerca de 2 metros. Com o *firmware* adequado, permite que o WMO seja utilizado como um detetor de *chokepoint*, fazendo com que as *tags* enviem mensagens Wi-Fi quando estão ao seu alcance. Com isto é possível localizar as *tags* com uma maior precisão.

Vestac

A Vestac (Vestac, 2013) disponibiliza uma *tag* (220Bi) que comunica através de Wi-Fi e que dependendo da necessidade dos clientes é possível juntar um emissor de infravermelhos para que a precisão seja melhor e se consiga também uma poupança energética.

220Bi AssetTag: A *tag* Vestac 220Bi disponibiliza uma ampla variedade de configurações com vista à poupança de energia:

- Hibernar depois de um determinado tempo parado;
- Ajustar a periodicidade da transmissão de mensagens;
- Selecionar a potência da transmissão (alta ou baixa).

Juntamente com estas configurações, é possível combinar um dispositivo de infravermelhos que, para além de melhorar a precisão, ajuda na poupança energética. A Vestac alega que a autonomia máxima da *tag* 220Bi é de 5 anos.

330XI Infravermelhos: Este emissor necessita de ser colocado em sítios estratégicos para que se consiga tirar maior partido dele. Em conjunto com a *tag* 220Bi, consegue-se níveis de precisão bastantes bons e também uma poupança de energia.

2.3.1. Síntese

Apresentadas as *tags*, é possível concluir-se que todas elas possuem conectividade Wi-Fi, bateria de longa duração que varia entre 2 meses até 5 anos e são na generalidade constituídas por sensores, tendo em conta o objetivo de cada *tag*.

O mais comum é encontrar *tags* com sensor de movimento, temperatura e humidade. Há ainda algumas que são constituídas com RFID, infravermelhos e/ou ultrassom com o objetivo de melhorar a precisão na determinação do posicionamento da *tag*.

Algumas *tags* permitem que se possa configurar a frequência com que esta envia as mensagens ao servidor. Esta configuração é feita com base em intervalos de tempo pré-definidos pelo fabricante.

Por último há *tags* que possuem botões configuráveis e LEDs. Os primeiros permitem o envio de mensagens e o envio de alertas e os segundos têm o intuito de alertar o utilizador de uma forma mais eficaz. Refira-se ainda que existem *tags* (e.g. Ekahau B4) constituídas por ecrãs que permitem transmitir ao utilizador mensagens de texto enviadas por outras *tags* ou pelo servidor.

3. Estudo do Problema

Neste capítulo proceder-se-á à apresentação do problema, sendo feita a identificação deste e o estudo dos requisitos da *tag* a desenvolver. Posteriormente é feita a análise da recolha de dados efetuada pela *tag*, bem como a identificação de procedimentos que conduzam a uma poupança energética.

3.1. Identificação do Problema

Como foi possível verificar no capítulo 2, a oferta ao nível de *tags* é bastante reduzida devido ao facto dos fabricantes desejarem sempre proceder à venda de soluções completas e que certamente lhes permite obter um maior lucro. O objetivo desta dissertação passa por conceber uma *tag* Wi-Fi para sistemas de posicionamento interior em redes Wi-Fi, que possa ser integrada no motor de posicionamento já existente e para efeitos de investigação.

Tendo em mente alguns dos requisitos pretendidos logo à partida, era possível seguir vários caminhos com vista à construção da *tag*, e.g. construção quase de raiz, comprando chips separadamente; construção com base em módulos Wi-Fi existentes; construção com base em placas que integram interface Wi-Fi e microprocessador; aquisição de uma *tag* configurável (e.g. RedPine).

Devido à boa oferta no mercado, às potencialidades e à possibilidade de programação através de API (*Application Programming Interface*) bem definidas, a escolha recaiu na aquisição de placas com microprocessador. Na secção 5.1 serão discutidos e explicados os fatores que estiveram na base desta decisão, apresentando as alternativas existentes no mercado.

Nas secções que se seguem é feita uma análise de requisitos da *tag* e estudada a estratégia de recolha de dados e poupança de energia.

3.2. Requisitos

Na secção anterior foi apresentado o problema e com base nisso é possível perceber quais os requisitos necessários para a conceção da *tag*.

A *tag* que se pretende desenvolver serve para localizar objetos ou pessoas, necessitando por isso, que esta disponha de conectividade Wi-Fi, seja leve e de dimensões reduzidas. Caso seja necessário despoletar ações ou obter acesso a determinadas informações, pode também ser necessário o uso de botões, displays ou até mesmo vários tipos de sensores.

Para determinar a localização da *tag*, esta tem de ser capaz de fazer uma leitura do sinal Wi-Fi e enviar os dados para o motor de posicionamento.

Como foi referido anteriormente, um dos principais requisitos da *tag* é a sua autonomia, obrigando a um cuidado especial nos consumos energéticos. Para que se consiga obter um baixo consumo de energia há estratégias que têm de ser estudadas e decisões que têm de ser tomadas, nomeadamente o tempo durante o qual a *tag* fica adormecida, explorando eventualmente a utilização de acelerómetros. De seguida são apresentados os requisitos tidos em conta na implementação da *tag*.

Requisitos Funcionais:

- Realizar a recolha dos dados (SSID, BSSID e RSSI) referentes aos APs que se encontram na vizinhança utilizando a interface Wi-Fi.
- Enviar os dados juntamente com o *timestamp* e o endereço MAC (*Media Access Control*) da própria *tag* para o servidor.
- Fazer a receção, processamento e armazenamento dos dados no servidor.

Requisitos não Funcionais:

- Possuir uma autonomia razoável que permita o uso sem constantes trocas da bateria - minimizar o consumo de energia da *tag*.
- Minimizar o tamanho e peso da *tag* para que o seu transporte não seja incómodo.

3.3. Recolha de Dados e Poupança de Energia

Um dos fatores de maior importância nos dispositivos móveis é a autonomia da bateria. Tem-se verificado que a evolução das baterias não tem sido proporcional à dos processadores, chips Wi-Fi e restante hardware, assistindo-se assim a uma otimização dos sistemas de forma a consumir a menor quantidade de energia possível.

No contexto deste trabalho, o gasto energético varia com o nível de utilização da *tag*, nomeadamente ao nível do processamento, utilização da interface Wi-Fi, entre outras, tornando-se crítico e essencial definir uma estratégia para a recolha e envio dos dados para o servidor de forma a maximizar a duração da *tag*.

Tendo em conta resultados verificados em alguns estudos, onde se refere que a percentagem média de movimento de uma pessoa durante o dia é inferior a 20% (J. Chon, 2011) (Y. Chon, Talipov, Shin, & Cha, 2011), foi decidido procurar maximizar a poupança energética considerando os movimentos das pessoas e objetos. Assim, tornou-se evidente que deveria existir uma forma de garantir que a *tag* era usada de forma eficiente, ativando-se quando ocorre movimento e diminuindo ou minimizando a sua atividade quando não existe movimento. Adicionou-se à *tag* um acelerómetro com o objetivo de detetar movimento conseguindo-se, desta forma, reduzir o tempo de atividade da *tag*, reduzindo-se assim o consumo energético.

Com a utilização do acelerómetro é possível definir uma estratégia de recolha de dados específica e otimizada para os dois estados diferentes (Figura 3.1): quando a *tag* se encontra estática e quando se encontra em movimento. Nos momentos em que a *tag* se encontra em repouso (estado *sleep*), esta pode ser adormecida de forma a garantir um menor consumo de bateria do que se estivesse ativa de forma contínua, realizando apenas recolha de informação de Y em Y segundos. Quando é detetada movimentação, a *tag* passa para o estado ativo e realiza recolha de informação, no entanto é necessário verificar se a *tag* continua em movimento. Para isso, pode ser definido um *timer* (quando a *tag* se encontra ativa) com uma periodicidade de X em X segundos para verificar se a *tag* se encontra em movimento, quando deteta inatividade é colocada no estado *sleep*.

Tendo em conta que durante 80% do dia uma pessoa se encontra parada (J. Chon, 2011)(Y. Chon et al., 2011), torna-se possível adormecer a *tag* e potenciar a utilização de temporizadores (*timers*) para otimizar a recolha de dados sem gastar muita energia.

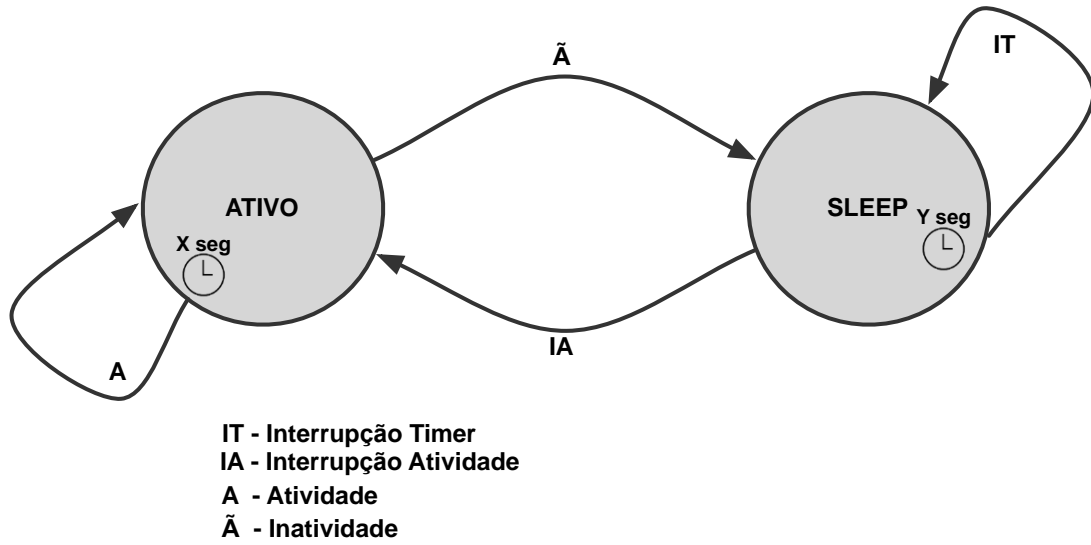


Figura 3.1 - Diagrama de Estados da Tag

Deste modo conclui-se que, com uma estratégia de recolha de dados periódica e com o auxílio do acelerómetro, é possível potenciar a autonomia da *tag* (Figura 3.2).

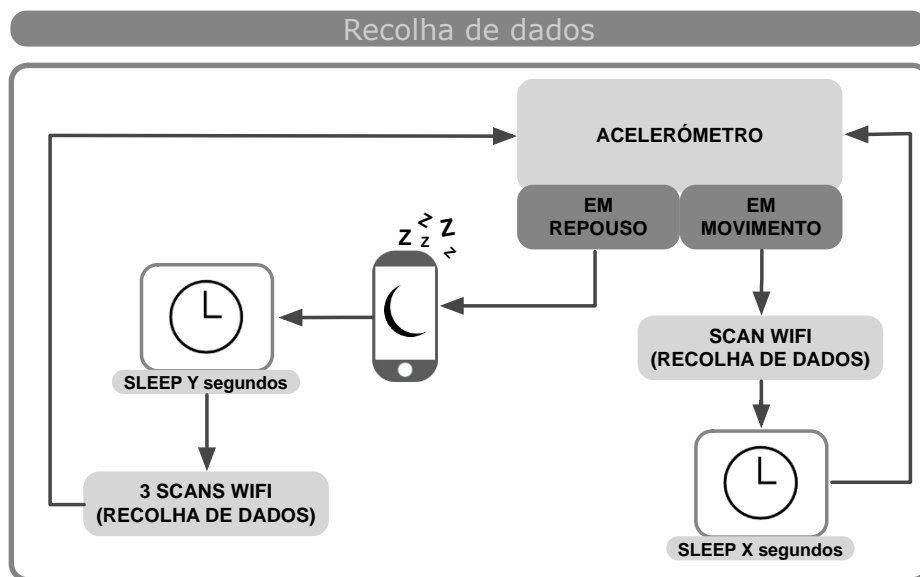


Figura 3.2 - Estratégia de recolha de dados controlada pelo acelerómetro

4. Especificação do Sistema

Neste capítulo será apresentada a arquitetura do sistema, bem como todos os componentes que o constituem. Dada a sua importância será também apresentada a composição das mensagens envolvidas neste sistema.

4.1. Arquitetura do Sistema

Nesta secção é descrita a arquitetura geral do sistema e a arquitetura da *tag*.

4.1.1. Arquitetura Geral

O sistema é composto por uma ou mais *tags*, pontos de acesso e por um servidor. Os pontos de acesso têm de ser colocados em sítios estratégicos para melhorar a precisão do sistema.

Como referido na secção 2.1 a técnica de *fingerprinting* é composta por duas fases, a fase de calibração e a fase online. Assim, o servidor tem de estar preparado para lidar com os dados provenientes de cada fase.

Na fase de calibração, o servidor recebe os dados enviados pela *tag* e armazena-os no mapa de rádio para que posteriormente (na fase online), o motor de posicionamento os possa utilizar com o objetivo de realizar o cálculo da localização da *tag*. Este disponibiliza ainda uma interface gráfica a partir da qual é possível verificar em tempo real a localização das *tags* que estão a ser monitorizadas.

A Figura 4.1 ilustra a arquitetura geral do sistema.

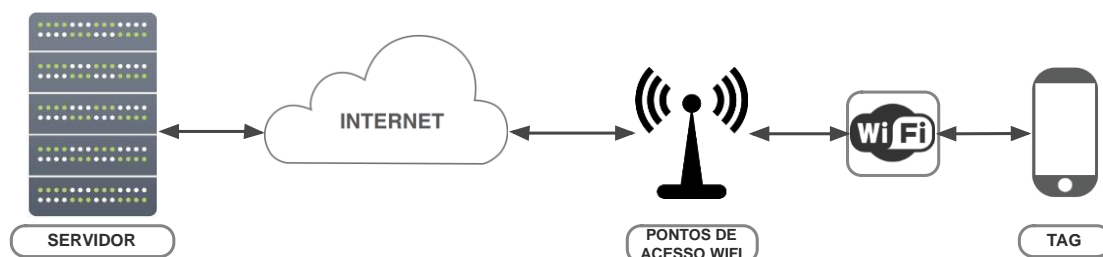


Figura 4.1 - Arquitetura Geral

4.1.2. Arquitetura Específica

A *tag* efetua um *scan* e recolhe o RSSI dos APs através da interface Wi-Fi, obtém o seu próprio endereço *MAC* para a identificação de cada *tag* e um *timestamp*. Em seguida, envia estas informações para o servidor com o objetivo deste posteriormente calcular a localização da *tag*. Com esta finalidade, a *tag* é constituída por vários componentes e pode ser dividida em duas áreas diferentes, hardware e software.

A nível de hardware a *tag* tem de ser constituída por uma antena e um chip (modula e desmodula o sinal) Wi-Fi para possibilitar a conetividade, por um microprocessador, RAM e ROM de forma a possibilitar a sua programação. Tem ainda de possuir uma bateria para permitir a sua mobilidade e um acelerómetro, de forma a diminuir o consumo energético através da deteção de movimento, como referido na secção 3.3.

A nível de software a implementação pode ser vista por camadas, desde a aplicação passando pela camada de transporte, pelos drivers permitindo o acesso ao hardware e pela camada física.

Na Figura 4.2 é possível verificar a arquitetura da *tag* com a respetiva separação das áreas referidas.

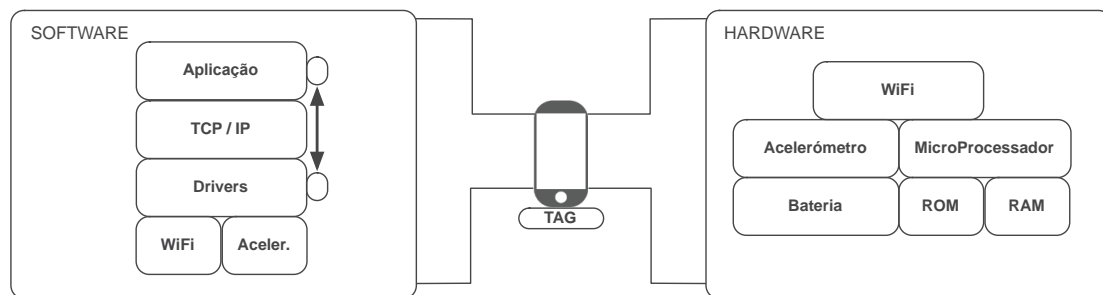


Figura 4.2 - Arquitetura da tag

4.2. Mensagens do Sistema

O envio, para o servidor, dos dados recolhidos é feito através de uma mensagem com a composição definida na Figura 4.3.



Figura 4.3 - Estrutura da mensagem enviada pela tag

- *MAC Address Tag*: É o endereço MAC da interface Wi-Fi da *tag*, através da qual foi recolhida a amostra e que serve como identificador da *tag*.
- *Timestamp*: É a data e hora da recolha da amostra no formato Unix.
- *Flag*: Identifica se a amostra foi recolhida em movimento (*flag* a 1) ou se a *tag* foi acordada pelo *timer* (*flag* a 0).
- *Data*: É composto pelas informações obtidas pelos *scans* feitos pela *tag* nomeadamente pelo, SSID, BSSID e RSSI referentes aos vários APs que a *tag* deteta (Figura 4.4).

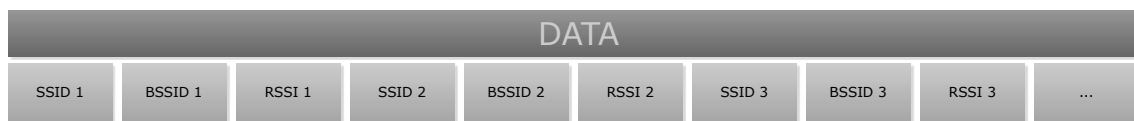


Figura 4.4 - Conteúdo do campo Data da mensagem

O tamanho deste campo é variável e está dependente do número de APs detetados pela *tag*.

5. Implementação

Neste capítulo são descritas as várias fases que dizem respeito à implementação da *tag*. Na secção 5.1 é descrita a fase da escolha do hardware com a análise e explicação das várias alternativas que foram encontradas no mercado. Na secção 5.2 são apresentadas as etapas necessárias à construção da *tag*, com um maior foco na parte do hardware. Por fim, na secção 5.3 são expostos os procedimentos necessários ao desenvolvimento do software.

5.1. Escolha do Hardware

Como referido no capítulo 3, este projeto impunha logo à partida alguns requisitos que a *tag* teria de cumprir, tornando-se assim possível iniciar uma pesquisa sobre o hardware existente no mercado que respeitasse os requisitos mais básicos.

Os requisitos mais importantes no que diz respeito ao hardware são, o baixo consumo energético, a conectividade Wi-Fi, tamanho e peso reduzidos e a possibilidade de conexão de dispositivos que pudessem vir a ser necessários como por exemplo um acelerómetro (como referido na secção 3.3).

Após uma pesquisa inicial foi possível elaborar uma lista, com algumas possibilidades que poderiam servir como base para a construção da *tag*. Nessa lista constam, o *Raspberry Pi*, o *BeagleBone*, o *Electric Imp*, o *SparkCore*, o *Openpicus FlyPort* Wi-Fi e o *Arduino Uno R3*, como se pode verificar na Tabela 5.1.

Hardware Disponível no Mercado						
	Raspberry Pi B	Beaglebone	Arduino Uno R3	Spark Core	Flyport WiFi	Electric imp
Processador (MHz)	700	720	16	72	32	120
RAM	512 MB	256 MB	2 KB	20 KB	16 KB	128 KB
WiFi	NÃO	NÃO	NÃO	SIM	SIM	SIM
Acelerómetro	NÃO	NÃO	NÃO	NÃO	NÃO	NÃO
Dimensões (mm)	85,6x56x21	76,2x76,2x16	68,6x53,3	37,3x20,3x12	35x48x15	51x32x11
Peso (g)	45	40	28	14	11	14

Tabela 5.1 - Tabela comparativa do Hardware disponível no mercado³

Depois de analisadas todas as características e opções, foi possível reduzir a lista para três placas: o *Electric Imp*, o *Openpicus FlyPort Wi-Fi* e o *Arduino Uno R3*. A lista final ficou reduzida a estas três opções uma vez que, tanto o *Raspberry Pi* como o *BeagleBone* são constituídos por um processador demasiado potente para aquilo que é pretendido, fazendo com que o consumo energético fosse, logo à partida, mais elevado que o desejado. Para além disso, estes dispositivos não dispõem nativamente de conectividade Wi-Fi, o que implicaria a compra de um dispositivo USB com conectividade Wi-Fi (*dongle* Wi-Fi). O *SparkCore* é um dispositivo bastante atrativo, pelo facto de cumprir alguns requisitos nomeadamente, ser pequeno, leve e possuir uma interface Wi-Fi. No entanto, este dispositivo foi eliminado da lista pelo facto de ainda estar num ponto de desenvolvimento bastante embrionário em relação à API e ao próprio dispositivo.

O *Electric Imp* foi um dos dispositivos que desde o início despertou bastante interesse, quer por ser pequeno e bastante leve, quer pela potencialidade de ser programado através da conectividade Wi-Fi oferecida nativamente pelo cartão SD Wi-Fi. Este dispositivo necessita apenas de uma placa (e.g. *April Board*) para alimentar o cartão SD e para proporcionar a possibilidade de se conectar dispositivos (e.g. acelerómetro, etc.). Como referido na secção 3.3, com o objetivo de reduzir o consumo energético decidiu-se adquirir um acelerómetro pelo que, após uma pesquisa no mercado, chegou-se à conclusão que o acelerómetro (ADXL345) da empresa

³ Informações retiradas dos respetivos sites.

Analog seria uma boa opção. A versão adquirida do acelerómetro ADXL345 foi a versão comercializada pela *Adafruit*⁴.

O *Openpicus FlyPort* Wi-Fi, por sua vez despertou atenção devido ao tamanho reduzido, a um grande universo de acessórios que a *Openpicus* disponibiliza, bem como, o facto de proporcionar conectividade Wi-Fi nativamente e pela API bem definida. A *Openpicus* disponibiliza a *Grove Board* que permite que a conexão de outros dispositivos (e.g. sensores, etc.) seja bastante simples, bastando apenas conectá-los nas entradas, utilizando os cabos apropriados e comercializados pela *Openpicus*, não necessitando de soldas. Esta placa permite ainda a programação do *FlyPort* Wi-Fi através de micro USB e também a conexão de uma bateria (*Lithium* 3,7V 1060mAh) que a própria *Openpicus* comercializa, bem como um acelerómetro (MMA7660FC) produzido pela *freescale*⁵.

Um dispositivo incontornável nesta área das pequenas placas, é sem dúvida o *Arduino*. Numa primeira abordagem, colocou-se em dúvida a existência de algum que cumprisse os requisitos. Dentro do universo *Arduino*, a escolha recaiu sobre o *Arduino Uno R3*, pelo fator preço e pela API existente. Após a escolha, houve duas características que fizeram com que o *Arduino* não se tornasse numa escolha viável: o seu tamanho e a necessidade de comprar um *shield* Wi-Fi motivado pelo facto de este modelo não possuir conectividade Wi-Fi por defeito. Existia a possibilidade de comprar o *shield* Wi-Fi, mas devido ao tamanho já significativo apenas do próprio *Arduino*, acrescentando o *shield*, ficaria ainda maior. Devido a este facto a hipótese de compra do *Arduino* foi descartada.

Inicialmente a ideia era desenvolver apenas uma *tag* e realizar todos os testes possíveis. No entanto com estes dois dispositivos tão interessantes e acessíveis disponíveis no mercado, optou-se por comprar o *Electric Imp* e o *Openpicus FlyPort*. Na Tabela 5.2 pode ser visualizada a lista de compras, onde constam os dispositivos e respetivos acessórios.

⁴ Site da Adafruit com informação do acelerómetro: <http://www.adafruit.com/products/1231>

⁵ Site com informação do acelerómetro:

http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MMA7660FC

LISTA DE COMPRAS		
	Flyport WiFi	Electric imp
Dispositivos	Grove Kit WiFi	imp + April Board
Acelerómetro	MMA7660FC	ADXL345
Bateria	1060mAh	1500mAh
Carregador	Carregador 5V	
Acessórios	Cabos Conexão Dispositivos	

Tabela 5.2 - Lista de compras

Com isto, o objetivo deste trabalho não passou pela construção de uma, mas sim duas *tags*, testá-las e comparar os resultados, verificando assim qual se adequa mais ao problema proposto nesta dissertação.

5.2. Construção da *Tag*

Após a análise das várias alternativas de hardware disponíveis no mercado e consequente escolha do hardware utilizado, segue-se a explicação das várias fases para a construção de cada uma das *tags* desenvolvidas. A construção do hardware consistiu na montagem dos dispositivos, soldas e construção de uma placa PCB (*Printed Circuit Board*) para o dispositivo baseado no *Electric imp*.

5.2.1. Electric imp

O *Electric imp* apresenta-se como um dispositivo de pequenas dimensões e centra-se num processador e antena Wi-Fi com o formato de um cartão SD designado por *imp*. Para o alimentar e conectar a outros dispositivos é necessário uma placa. A escolha passou pela placa *April*, uma placa constituída apenas pelos componentes necessários ao funcionamento do *imp* e que se caracteriza pela sua reduzida dimensão. A placa *April* possui um circuito de alimentação

onde é possível escolher de que forma é que o *imp* é alimentado. Este pode ser alimentado pela porta mini USB da placa ou através de uma bateria que tem de ser soldada nos pinos V+ e V-. É ainda constituída por um integrado que codifica e cifra as informações que são enviadas para o servidor.

Depois da compra do material, a primeira fase para a construção da *tag* correspondeu à montagem do hardware e à sua configuração inicial. Esta fase requereu relativamente pouco tempo pois revelou-se bastante simples e intuitiva, onde a maior dificuldade esteve no estudo dos respetivos *datasheets*, para que a montagem fosse feita sem percalços. Inicialmente, a configuração com vista à conexão à internet revelou-se trabalhosa pois o *imp* não respondia da melhor forma à configuração adotada, mas rapidamente ultrapassada, uma vez que o site do *Electric imp* está bem documentado e ajudou na solução deste percalço.

A necessidade de utilização de um acelerómetro, foi devidamente justificada na secção 3.3. Para além da poupança de energia, o acelerómetro permite aumentar a eficácia, uma vez que apenas desperta a *tag* quando esta se encontra em movimento. Desta forma, é possível realizar um acompanhamento do movimento da pessoa ou objeto ao qual a *tag* está anexada.

A montagem dos vários componentes tiveram como suporte uma *breadboard* para que a conexão entre o *Electric imp* e o acelerómetro fosse possível. Na parte da conexão do acelerómetro foi necessário o estudo do *datasheet* da placa *April*, do *imp*⁶ e do próprio acelerómetro, dada a necessidade de saber quais os pinos que seriam utilizados. Esta questão está diretamente relacionada com o protocolo de comunicação a ser utilizado entre o acelerómetro e o *Electric imp*.

O protocolo escolhido para a comunicação entre o *Electric imp* e o acelerómetro foi o I²C. A comunicação podia ser estabelecida utilizando a combinação dos pinos 1 (*clock*) e 2 (dados) ou dos pinos 8 (*clock*) e 9 (dados). Optou-se pela utilização dos pinos 8 e 9 para que a construção da *tag* ficasse mais compacta. A escolha do protocolo utilizado será devidamente explicada na secção 5.3.

Relativamente à bateria, esta foi conectada apenas quando começaram a ser efetuados os testes à *tag*. Durante o desenvolvimento, a *tag* foi sempre alimentada pelo cabo USB. Como a

⁶ Informações placa *April*: <https://electricimp.com/docs/hardware/resources/reference-designs/april/>

⁷ *Datasheet imp*: https://electricimp.com/docs/attachments/hardware/datasheets/imp001_specification.pdf

placa *April* não possui um circuito que permita o carregamento da bateria, foi montado um cabo (Figura 5.1) para a conectar no circuito de carregamento da bateria do *Openpicus Flyport Wi-Fi*.



Figura 5.1 - Cabo de carregamento da bateria Electric imp

Após a realização das medições de consumos efetuadas recorrendo a um osciloscópio e multímetro e conseqüentemente os primeiros testes efetuados à *tag*, foi decidido passar a montagem que estava feita na *breadboard* (Figura 5.2) para uma placa PCB (Figura 5.3) com o objetivo de tornar a *tag* mais apelativa, fiável e transportável. Para isso foi criada uma placa que permitiu que a *tag* ficasse bastante compacta e pudesse ser transportada sem qualquer preocupação de desmontagem de algum componente ou fio que colocasse em causa a sua fiabilidade. A posição definitiva dos componentes foi pensada de forma que a antena Wi-Fi do *imp* não fosse afetada, a bateria ficasse num local acessível para quando seja necessário realizar um carregamento e de modo a que a *tag* ficasse com um tamanho reduzido. A montagem final pode ser visualizada na Figura 5.4.

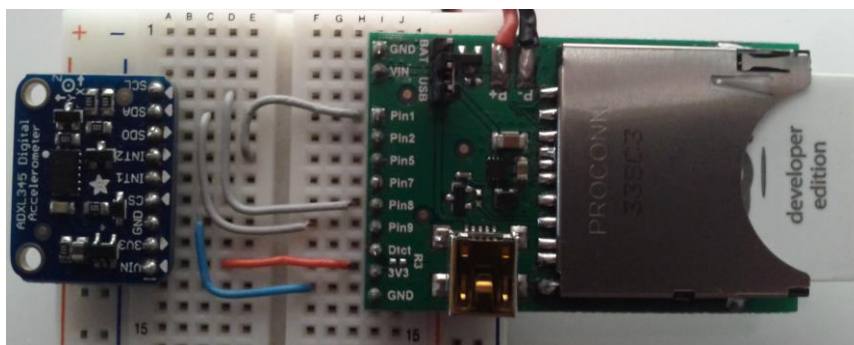


Figura 5.2 - Montagem na breadboard para a implementação

Na Figura 5.3 é apresentada a placa PCB construída para a conexão entre o acelerómetro e o *Electric imp*.



Figura 5.3 - Placa PCB para a conexão do acelerómetro e *Electric imp*

De seguida é possível visualizar a versão final da *tag*.



Figura 5.4 - *Tag Electric imp*

5.2.2. Openpicus FlyPort Wi-Fi

Na secção 5.1 foram referidos os componentes adquiridos para funcionar em conjunto com o *Openpicus Flyport Wi-Fi*.

O *Flyport Wi-Fi* apresenta-se como um dispositivo de pequenas dimensões com conectividade Wi-Fi nativa. Para o programar, utiliza-se uma placa denominada de *Grove Board*, onde se encaixa o *Flyport Wi-Fi* e onde é possível conectar todo o tipo de componentes que

sejam necessários. Para além de servir de interface entre o computador e o *Flyport* Wi-Fi, a placa *Grove* possibilita o carregamento da bateria.

Depois da compra do material, a primeira fase para a construção da *tag* correspondeu à compreensão da forma de conexão do *Flyport* Wi-Fi à placa *Grove* e à sua configuração inicial. Esta fase requereu o estudo do *datasheet* do *Flyport* Wi-Fi⁸ e da placa *Grove*⁹ para que a montagem se realizasse sem qualquer percalço.

O *Flyport* Wi-Fi pode ser alimentado através da placa *Grove* utilizando a bateria, a porta micro USB ou por um transformador externo, sendo que, para o programar é obrigatoriamente necessária a conexão USB ao computador.

A montagem dos vários componentes, tiveram como suporte a placa *Grove* quer para a programação do *Flyport* Wi-Fi, quer para a conexão entre o acelerómetro e o *Flyport* Wi-Fi. Este pode ser conectado através do protocolo I²C ou SPI. Tal como no *Electric imp*, a escolha recaiu sobre o I²C, pela razão apresentada na secção 5.3 onde é possível analisar também a explicação de ambos os protocolos. Ao contrário do *Electric imp*, a conexão do acelerómetro não necessita da definição dos pinos para a utilização do protocolo I²C, visto que a placa *Grove* contém portas com ligações específicas e a conexão é feita através de um cabo que encaixa apenas de uma única forma.

Inicialmente, a alimentação utilizada foi através da porta micro USB, mas para a realização dos testes procedeu-se à utilização da bateria.

Após a realização das medições de consumos através de um osciloscópio e multímetro e consequentemente os primeiros testes efetuados à *tag*, foi decidido retirar a placa *Grove* (Figura 5.5) e realizar a montagem dos dispositivos diretamente no *Flyport* Wi-Fi com o objetivo de tornar a *tag* mais leve, transportável e com um design mais apelativo.

Como a placa *Grove* acrescenta um tamanho dispensável, para o protótipo final, foi necessário um estudo aprofundado do *datasheet* do *Flyport* Wi-Fi para determinar se havia possibilidade de o alimentar sem a utilização da placa *Grove* e se o acelerómetro podia ser conectado diretamente ao *Flyport* Wi-Fi. Após o estudo do *datasheet*, foi possível chegar à conclusão que seria possível alimentar o *Flyport* Wi-Fi diretamente nos pinos 16 (VCC) e 22

⁸ *Datasheet* placa *Grove*: http://space.openpicus.com/u/ftp/datasheet/datasheet_grove_board.pdf

⁹ *Datasheet* *Flyport* Wi-Fi: http://space.openpicus.com/u/ftp/datasheet/datasheet_flyport_Wi-Fi_g.pdf

(GND) e conectar o acelerômetro através do protocolo I²C usando os pinos 1 (*clock*), 3 (dados) e 5 (interrupção).

A posição definitiva dos componentes foi pensada de forma que, a bateria ficasse num sítio acessível para quando é necessário realizar um carregamento e de modo a que a *tag* ficasse mais compacta e conseqüentemente com um tamanho mais reduzido. A montagem final pode ser visualizada na Figura 5.6.



Figura 5.5 - Flyport Wi-Fi conectado na placa Grove

A montagem final da *tag* baseada no Flyport Wi-Fi pode ser visualizada na Figura 5.6. Ao contrário do *Electric imp*, não foi desenvolvida uma placa PCB pois a bateria já é constituída por uma placa e o Flyport Wi-Fi é possui entradas específicas, não sendo necessário criar caminhos para a conexão entre os dispositivos.

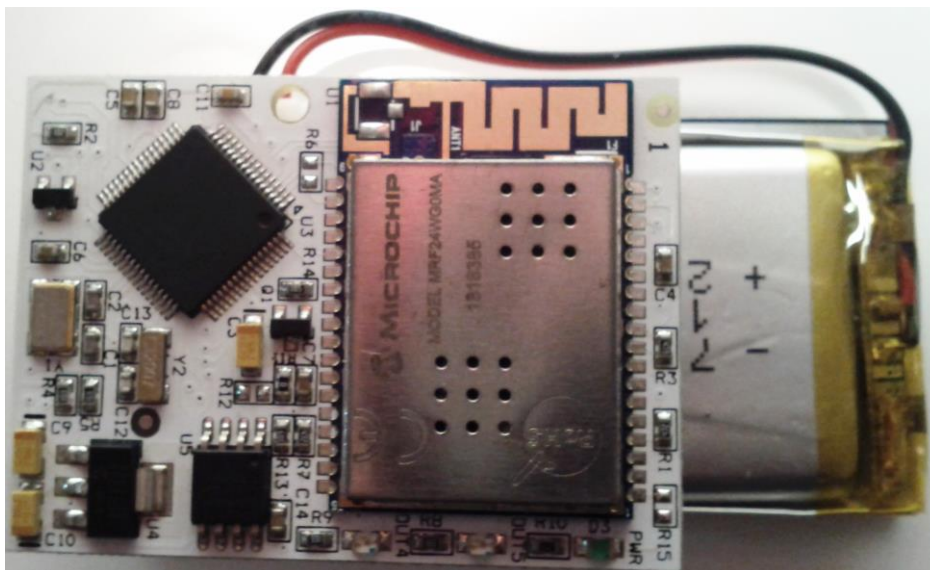


Figura 5.6 - Tag Flyport Wi-Fi

5.3. Construção do Software

Os dispositivos adquiridos são constituídos por microprocessadores, no entanto estes vêm sem qualquer código inserido na sua memória. Deste modo, para que os dispositivos realizassem o que era desejável, todo o software implementado foi desenvolvido de raiz.

Antes de iniciar a explicação de cada fase relativa ao desenvolvimento do software das respetivas *tags*, são apresentados o servidor e o protocolo de comunicação entre as *tags* e respetivo acelerómetro.

5.3.1. Servidor

Como referido na subsecção 4.1.1 o sistema é constituído por um servidor, pelas *tags* e pelos APs. O servidor do sistema tem o objetivo de receber os dados enviados pelas *tags* e determinar a localização destas. Apesar deste servidor já se encontrar desenvolvido, foi necessária a implementação de um servidor mais simples para o apoio ao desenvolvimento das *tags*.

Com o objetivo de receber os dados enviados pelas *tags*, foi desenvolvido um servidor, tendo por base um código PHP preparado apenas para receber pedidos *HTTP POST* e *GET*. As informações recebidas pelo servidor são armazenadas em ficheiros “txt” diários e no seu conteúdo é colocada a informação recebida no servidor em conjunto com a data e hora em que os dados foram recebidos. Na Figura 5.7 é possível verificar um excerto de um ficheiro “txt” guardado no servidor.

```
2014-07-22 09:58:31,0c2a69000b04,1406019505,0,wireMaia,002417d85828,-36,Lecom_Squash_1,00022d677405,-51,Lecom_Squash_2,00022d75df4e,-60,wire
2014-07-22 10:28:36,0c2a69000b04,1406021311,0,wireMaia,002417d85828,-36,Lecom_Squash_1,00022d677405,-52,Lecom_Squash_2,00022d75df4e,-60,wire
2014-07-22 10:58:42,0c2a69000b04,1406023117,0,wireMaia,002417d85828,-36,Lecom_Squash_1,00022d677405,-52,Lecom_Squash_2,00022d75df4e,-60,wire
2014-07-22 11:08:13,0c2a69000b04,1406023692,1,wireMaia,002417d85828,-27,Lecom_Squash_1,00022d677405,-60,Lecom_Squash_2,00022d75df4e,-75,
2014-07-22 11:08:16,0c2a69000b04,1406023694,1,wireMaia,002417d85828,-42,Lecom_Squash_1,00022d677405,-56,Lecom_Squash_2,00022d75df4e,-63,
2014-07-22 11:08:18,0c2a69000b04,1406023697,1,wireMaia,002417d85828,-42,Lecom_Squash_1,00022d677405,-49,Lecom_Squash_2,00022d75df4e,-60,
2014-07-22 11:08:21,0c2a69000b04,1406023699,1,wireMaia,002417d85828,-53,Lecom_Squash_1,00022d677405,-45,Lecom_Squash_2,00022d75df4e,-55,
2014-07-22 11:08:23,0c2a69000b04,1406023702,1,wireMaia,002417d85828,-62,Lecom_Squash_1,00022d677405,-51,Lecom_Squash_2,00022d75df4e,-51,
2014-07-22 11:08:25,0c2a69000b04,1406023704,1,wireMaia,002417d85828,-49,Lecom_Squash_1,00022d677405,-52,Lecom_Squash_2,00022d75df4e,-53,
2014-07-22 11:08:28,0c2a69000b04,1406023706,1,wireMaia,002417d85828,-66,Lecom_Squash_1,00022d677405,-62,Lecom_Squash_2,00022d75df4e,-38,
```

Figura 5.7 - Excerto de um ficheiro txt guardado no servidor

Estes ficheiros servem como base aos cálculos da autonomia das *tags* que irão ser apresentados na secção 6.1.

5.3.2. Protocolo de Comunicação *Tag*- Acelerómetro

Ambos os acelerómetros e *tags* disponibilizam a possibilidade de conexão de dispositivos através dos protocolos I²C (*Inter-Integrated Circuit*) e SPI (*Serial Peripheral Interface*). Os protocolos I²C e SPI foram desenvolvidos por marcas diferentes embora com o mesmo objetivo, a comunicação entre chips. Estes tornaram-se nos protocolos mais utilizados para comunicação entre chips, permitindo o suporte deste por ambas as *tags*.

O protocolo SPI foi desenvolvido pela Motorola com o objetivo de permitir que os seus chips comunicassem uns com os outros. Este protocolo consiste num barramento de interface utilizado para trocar dados entre microcontroladores e pequenos periféricos (e.g. sensores, cartões SD etc.). Utiliza um *clock* que mantém ambos os lados em perfeita sincronização e canais de dados separados, juntamente com um canal de seleção que permite a escolha do dispositivo com o qual quer comunicar.

O SPI tem algumas vantagens sobre I²C, nomeadamente na disponibilização de uma maior frequência de *clock* (10MHz) e de uma ligação *full-duplex*, o que o torna particularmente adequado para aplicações que requerem uma comunicação bidirecional simultânea. Por outro lado, requer pelo menos três fios partilhados por todos os dispositivos no barramento, e uma linha de seleção para cada dispositivo conectado.

O protocolo SPI separa os dispositivos em mestres e escravos. Apenas o dispositivo mestre pode enviar pulsos para sincronizar a transferência de dados temporais e todos os outros, que sincronizam os seus relógios com o mestre, são considerados escravos. Neste caso, os dispositivos mestre seriam o *imp* e o *Flyport Wi-Fi*, e os escravos seriam os respetivos acelerómetros.

A implementação do protocolo SPI implica a utilização de quatro conexões, o SCLK (*Clock from Master*), o MOSI (*Master-Output Slave-Input*), o MISO (*Master-Input Slave-Output*) e o SS (*Slave-Select ou Chip-Select*). O MOSI e MISO correspondem às linhas de transmissão de dados, o SCLK corresponde ao *clock* e o SS corresponde ao canal de seleção do escravo. Na (Figura 5.8) é possível verificar o funcionamento do protocolo SPI.

Assumindo que existe apenas um dispositivo escravo conectado ao mestre, a comunicação é inicializada quando o mestre transita o estado da linha SS de 1 para 0. O mestre começa a enviar os pulsos do *clock*, tipicamente a transmissão de dados é efetuada nas

ascensões do *clock*. O mestre inicia a transmissão de dados pela linha MOSI sinalizando o símbolo 1 com o nível alto e o símbolo 0 com o nível baixo. O escravo lê o sinal e retorna informação pela linha MISO. Para finalizar a troca de informação o mestre ascende a linha SS de 0 para 1.

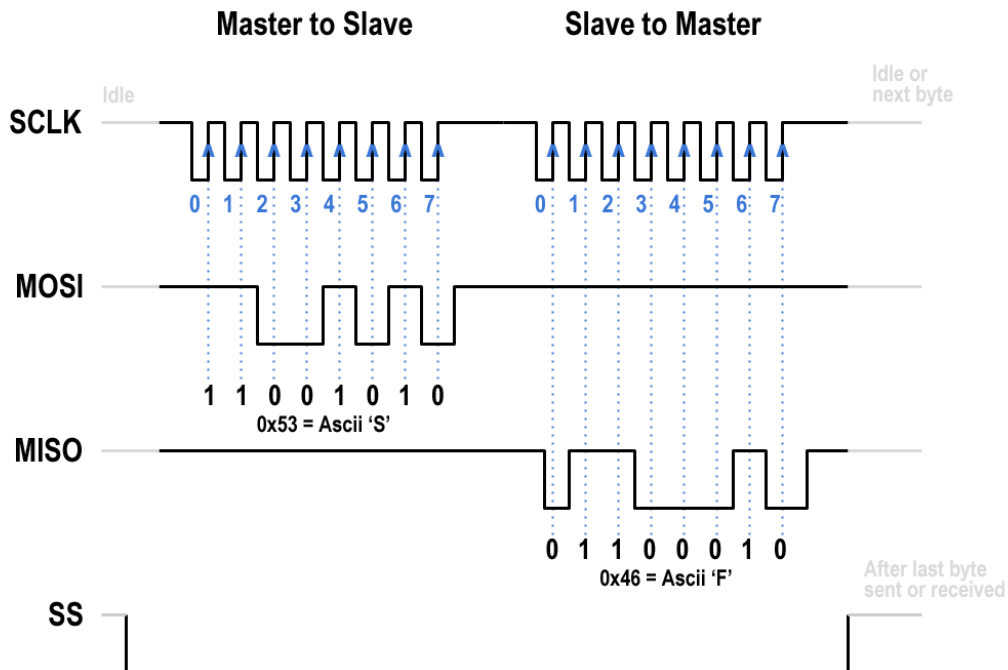


Figura 5.8 - Protocolo SPI (Electric imp, 2014)

O protocolo I²C foi desenvolvido pela Philips com o objetivo de, tal como no caso do SPI, permitir que os seus chips comunicassem uns com os outros. A vantagem do protocolo I²C relativamente ao SPI centra-se na necessidade de utilizar apenas duas conexões, o SDA (*Serial Data Line*) e o SCL (*Serial Clock Line*), para a implementação do protocolo e no facto da implementação ao nível do software ser mais trivial do que a do SPI.

Tal como o protocolo SPI, o I²C separa os dispositivos em mestres e escravos. Apenas o dispositivo mestre pode enviar pulsos na linha SCL para sincronizar a transferência de dados temporais e todos os outros, que sincronizam os seus relógios com o mestre, são considerados escravos. Neste caso, os dispositivos mestre seriam o *imp* e o *Flyport Wi-Fi*, e os escravos seriam os acelerómetros.

Ao contrário do SPI, os escravos são identificados através de um endereço, desta forma, evita-se a utilização de uma linha de seleção para cada escravo.

A frequência de *clock* máxima é de 400kHz, sendo bastante inferior à do SPI.

Na (Figura 5.9) é possível verificar o funcionamento do protocolo I²C. O evento de início ocorre quando o *clock* (SCL) está no nível alto e o canal de dados (SDA) transita do estado 1 para 0. O evento de paragem ocorre quando o *clock* está no estado 1 e o canal de dados transita do estado 0 para o estado 1.

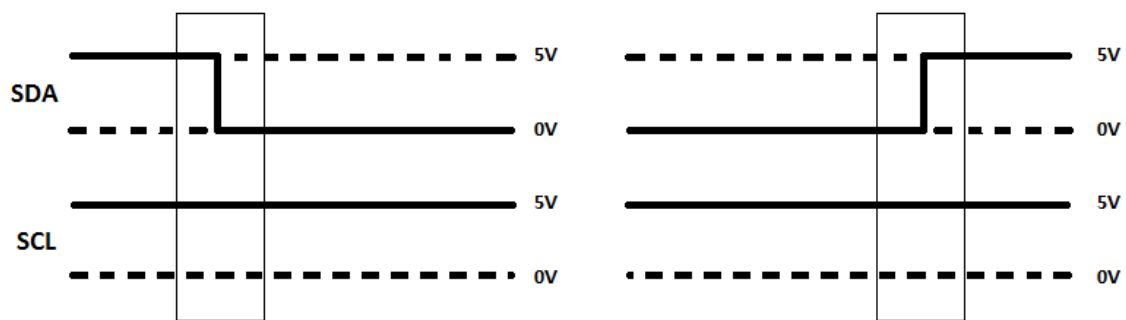


Figura 5.9 - Protocolo I²C (Electric imp, 2014)

Pelo facto da taxa de transmissão não ser um fator crítico e de forma a simplificar a construção da *tag*, o protocolo adotado foi o I²C. A necessidade de utilização de apenas duas conexões foi determinante nesta escolha.

5.3.3. Electric imp

O *Electric Imp* disponibiliza um IDE (*Integrated Development Environment*) online (Figura 5.12) de forma a permitir a programação do dispositivo. Esta característica proporciona bastantes vantagens, pois desta forma, independentemente do local onde a *tag* se encontre é possível programa-la através da Internet sem que seja necessária a utilização de qualquer cabo para transmissão do código desenvolvido no IDE.

Apesar desta grande vantagem, esta *tag* possui uma desvantagem de depender do servidor denominado de *imp cloud*, pois toda a informação que o *imp* envia tem de passar obrigatoriamente por este servidor. Para evitar a interseção do tráfego enviado pelo *imp* com o objetivo de impedir que este enviase as informações para o *imp cloud*, o *Electric imp* cifra a

informação antes de ser enviada e esta é decifrada pelo *imp cloud* como pode ser verificado na Figura 5.10 onde é apresentada a arquitetura do *Electric imp*. A linguagem de programação utilizada pelo *Electric imp* é *Squirrel*. Esta linguagem foi desenvolvida pelo MIT (*Massachusetts Institute of Technology*) e é caracterizada por ser uma linguagem de alto nível, orientada aos objetos. A sua sintaxe é parecida com a linguagem C/C++ e Java mas é uma linguagem muito dinâmica como *Python* ou *Lua* (Squirrel, 2014).

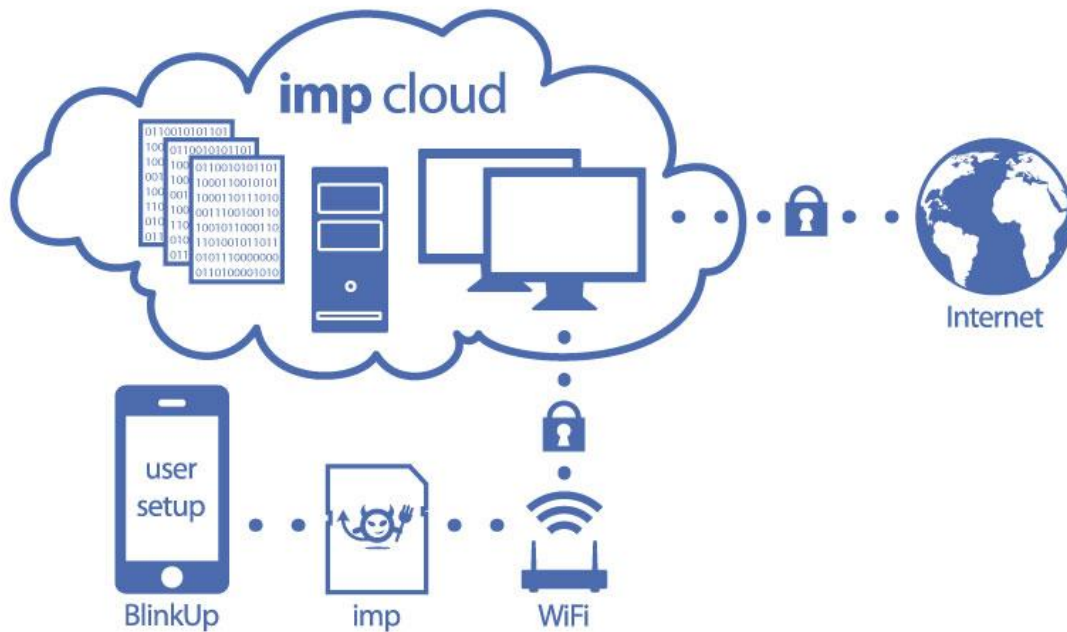


Figura 5.10 - Arquitetura *Electric imp* (*Electric imp*, 2014)

- Análise da plataforma e configuração inicial

Sabendo logo à partida que o *Electric imp* necessita de estar conectado à Internet para funcionar, na primeira fase foi determinado como conecta-lo à Internet e de seguida como o programar. O *Electric imp* é configurado através de uma aplicação para *smartphone*, onde é inserido o SSID e a palavra passe da rede Wi-Fi à qual se pretende conectar. A informação de configuração é transmitida para o *imp* através do ecrã do *smartphone* que ao emitir luz intermitente, envia as definições de rede que foram introduzidas na aplicação. Este processo de configuração é denominado de *BlinkUp*. O *imp* tem a capacidade de receber as configurações desta forma uma vez que é composto por um sensor ótico, como é possível verificar na Figura 5.11. Durante o processo *BlinkUp* é necessário ter em conta que as palavras “*developer edition*”

necessitam de estar tapadas e o sensor ótico necessita de estar o mais próximo possível do ecrã do *smartphone* que está a enviar as configurações.

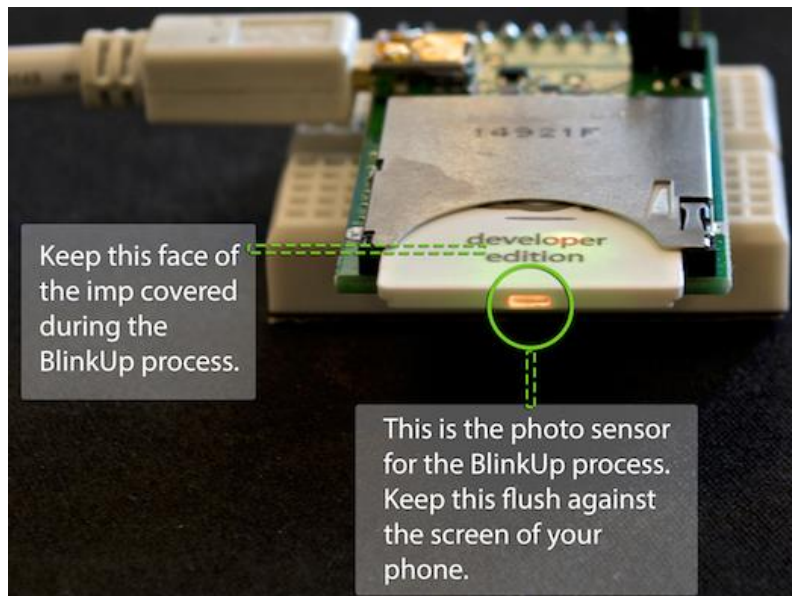


Figura 5.11 - Sensor Ótico do Electric imp (Electric imp, 2014)

O *imp* é composto por leds que têm como objetivo identificar o estado deste. Cada estado está associado a uma determinada cor que tem um significado associado. Tipicamente quando emite luz verde significa que a conexão foi efetuada com sucesso, por outro lado quando emite luz de cor vermelha é porque a conexão não foi efetuada com sucesso. Por fim, quando o *imp* emite a luz laranja é porque não contém nenhuma configuração de rede. Com base no comportamento destes leds, nomeadamente na combinação do número de vezes que um pista emitindo determinada cor é possível realizar um *troubleshooting* dos problemas que podem ocorrer. O site do *Electric imp*¹⁰ está bem documentado acerca deste tema.

Após a configuração inicial do *imp*, seguiu-se a descoberta do IDE onde é possível escrever o código que é transmitido para o *imp* através da Internet e onde também é possível escrever o código que o *imp cloud* vai correr. Neste caso o *imp cloud* é chamado de *agent* e o *imp* de *device*.

A plataforma (IDE) é subdividida em três partes, *agent*, *device* e *logs*. Na Figura 5.12 é possível verificar o aspeto do IDE, onde o *agent* encontra-se na janela da esquerda, o *device* na janela da direita e os *logs* aparecem na janela inferior. Tem ainda a particularidade de dispor de

¹⁰ Documentação sobre *BlinkUp*: <https://electricimp.com/docs/troubleshooting/blinkup/>

três botões, o botão *Check*, o botão *Build and Run* e o botão *Build* que permite o acesso a todas as versões dos códigos desenvolvidos anteriormente. O botão *check* e o *Build and Run*, tal como os próprios nomes sugerem, têm a função de verificar se o código tem erros de sintaxe (botão *Check*) e a função de compilar e enviar o código para o *Electric imp* (botão *Build and Run*).

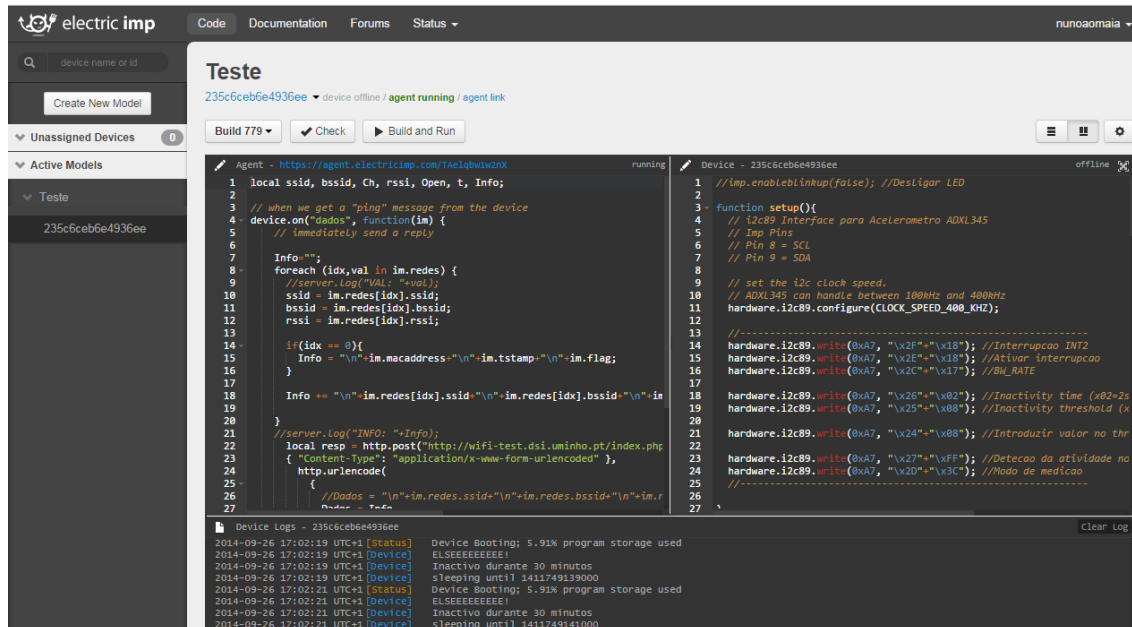


Figura 5.12 - IDE Electric imp

- Conexão à rede Wi-Fi e *scan* dos APs

Na segunda fase foi efetuada a conexão à rede Wi-Fi (através do processo *BlinkUp*) e descoberta a forma de detetar os APs que se encontram na proximidade do *imp*. Como foi referido anteriormente, o *imp* é programado através da linguagem *Squirrel* que é caracterizada por ser uma linguagem de alto nível, tornando o trabalho bastante mais simplificado na medida em que através de relativamente poucas linhas de código se consegue realizar bastantes operações. Desta forma para a deteção dos APs é apenas necessário invocar a função *imp.scanwifinetworks()*.

- Recolha de informações sobre APs e geração do *timestamp*

Após a deteção dos APs, seguiu-se a terceira fase onde foi implementada a forma de realizar o *scan* e de recolher as informações sobre os APs na vizinhança. Na altura em que é feito o *scan* procede-se à geração do *timestamp* UNIX com o objetivo de registar o momento em que foi efetuado o *scan* e foram recolhidas as informações dos APs. Sabendo que o sinal Wi-Fi

varia no tempo, decidiu-se que quando a *tag* é acordada pelo temporizador (*timer*), são feitos três *scans* Wi-Fi com o objetivo de posteriormente no servidor ser possível realizar uma aproximação do valor do RSSI naquele local, pois como os três *scans* foram realizados quando a *tag* estava estática garante-se que ela não saiu do local. Nesta situação é colocada uma *flag* com o valor 0 na mensagem enviada ao servidor e quando a *tag* é acordada pelo acelerómetro é colocada uma *flag* com o valor 1.

- Envio dos dados para o servidor

Como referido anteriormente, as informações enviadas pelo *imp* têm de passar obrigatoriamente pelo *agent* (*imp cloud*). Assim só depois de chegar ao *agent* é que podem ser reenviadas para o “nosso” servidor. Para tal, foi necessário estudar como enviar as informações do *imp* para o *agent* e posteriormente do *agent* para o servidor. A transmissão dos dados do *imp* para o *agent* é feito através da função *agent.send()* disponibilizada pela API do *Electric imp*, onde são enviados como argumento os dados para que, posteriormente no *agent* seja construída a mensagem com estrutura referida na Figura 4.3 da secção 4.2.

A construção da mensagem é feita no *agent* com o objetivo de reduzir tempo de processamento do lado da *tag*. Após a construção da mensagem, o *agent* procede à realização do *HTTP POST*. Os passos referentes ao envio dos dados podem ser visualizados na Figura 5.13.

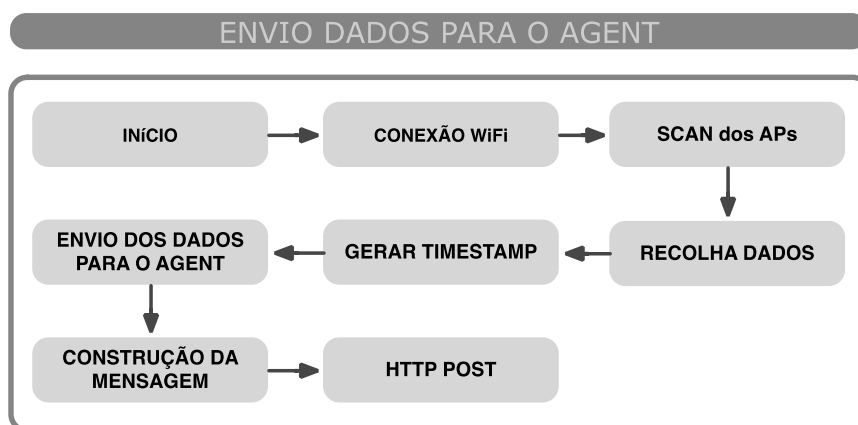


Figura 5.13 - Envio dos dados para o Agent

- Implementação do ciclo para o *scan*

Na quinta fase foi efetuada a implementação do ciclo para a realização do *scan* e da colocação da *tag* a dormir (modo *sleep*) após o *post* dos dados. Com o objetivo de reduzir o

consumo energético, optou-se por adormecer a *tag* nos momentos em que esta não esteja em movimento. Nesta fase foi utilizada uma função que coloca a *tag* no modo *sleep* mas com a particularidade de a manter conectada à rede Wi-Fi. Desta forma, não se poupava o máximo de energia mas para a realização dos primeiros testes era a que garantia um comportamento mais adequado. Com esta função a *tag* ficava num estado *sleep* durante o tempo definido, como seria para testar o funcionamento do *scan* e passagem para o *agent* das informações recolhidas no *scan*. O tempo definido nesta fase era relativamente curto (30 segundos) de forma a reduzir o tempo de espera necessário à *tag* realizar o *scan*.

Na Figura 5.14 pode ser visualizado o algoritmo da amostragem periódica.

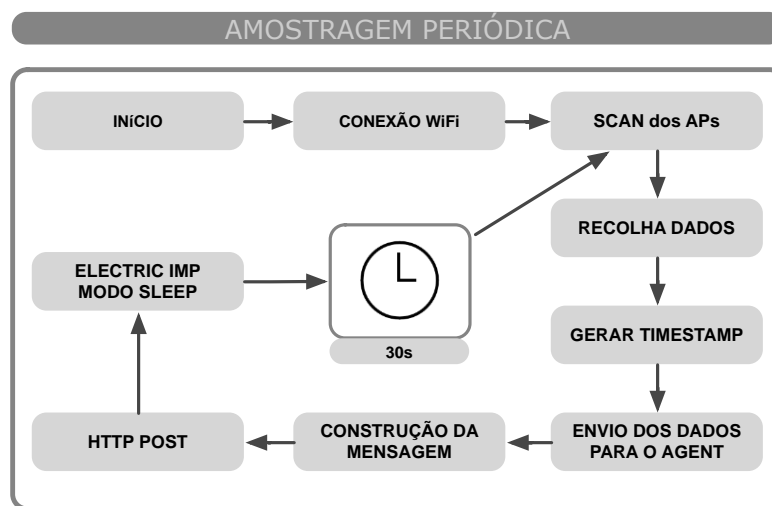


Figura 5.14 - Amostragem periódica Electric imp

- Integração do acelerómetro na *tag* e deteção da movimentação

Na sexta fase foi efetuada a integração do acelerómetro na *tag* e posteriormente a deteção da movimentação da *tag*. Como referido e justificado na subsecção 5.3.2, o protocolo escolhido para a comunicação entre o acelerómetro e o *imp* foi o I²C.

O primeiro passo para a integração do acelerómetro passou por perceber o funcionamento do protocolo I²C. De seguida procedeu-se ao estudo do *datasheet*¹¹ com o objetivo de conseguir a comunicação entre o *imp* e o acelerómetro.

¹¹ *Datasheet* axelerómetro: http://www.analog.com/static/imported-files/data_sheets/ADXL345.pdf

Inicialmente, procedeu-se à configuração do acelerómetro apenas para a realização de leituras sucessivas dos valores de todos os eixos. Após a conclusão destas configurações, procedeu-se ao estudo do *datasheet* de forma a configurar o acelerómetro com o objetivo de detetar a movimentação da *tag*. Esta fase demonstrou-se bastante trabalhosa e demorosa pela exigência de estudo acerca do funcionamento das interrupções. Após este estudo, procedeu-se à configuração de uma interrupção no pino INT 2 (do acelerómetro) e no pino 1 (do *Electric imp*) para quando o acelerómetro detetar atividade e inatividade (Figura 5.15).



Figura 5.15 – Configuração Acelerómetro ADXL345

Quando é gerada uma interrupção é verificado o motivo pelo qual a interrupção ocorreu. Caso seja uma interrupção gerada por inatividade, o acelerómetro e a *tag* são colocados no modo *sleep*. Caso contrário, é realizado um *scan* dos APs e são efetuados todos os procedimentos até ao *HTTP POST* realizado pelo *agent* (Figura 5.16).

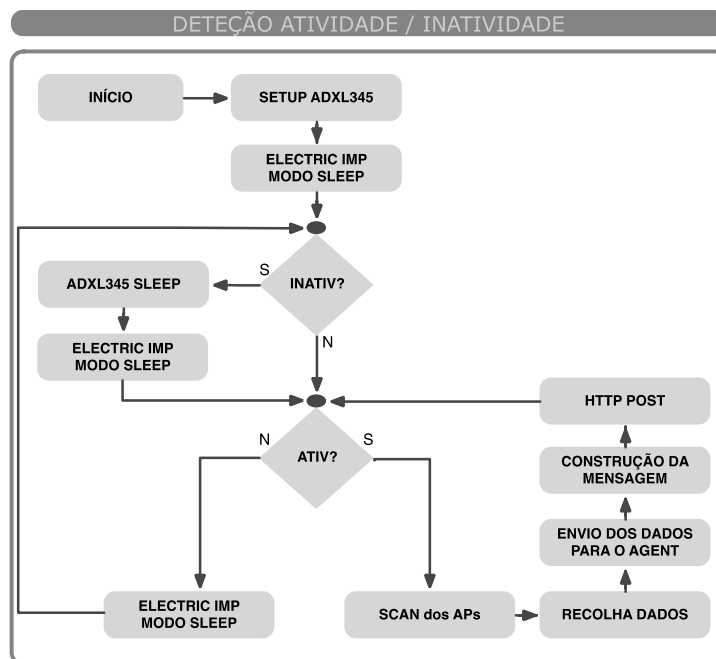


Figura 5.16 - Algoritmo deteção atividade / inatividade Electric imp

- Potenciar a utilização do acelerómetro

Na sétima fase o objetivo passou por potenciar a utilização do acelerómetro e gerir o ciclo de funcionamento da *tag* em função do movimento. Esta foi a fase dos ajustes das configurações, de forma a possibilitar que o comportamento da *tag* correspondesse àquilo que lhe era suposto.

Na quinta fase, foi implementado um algoritmo para colocar a *tag* no modo *sleep*, no entanto, nessa fase a *tag* encontrava-se conectada à rede Wi-Fi mesmo quando se encontrava nesse modo. Como a conexão Wi-Fi é dos fatores que provoca um maior gasto energético, revelou-se necessário configurar *tag*, de forma que quando esta se encontra no modo *sleep* esteja desconectado da rede. Sabendo que a conexão Wi-Fi só é necessária quando a *tag* está em movimento ou quando esta é acordada pelo *timer*, procedeu-se à escolha de uma função que desconecta a *tag* da rede Wi-Fi e a coloca num “sono profundo”, garantindo desta forma um menor gasto energético.

Utilizando a referida função, a única forma de despertar a *tag* é através de interrupções, que tanto pode ser uma interrupção através de um *timer*, como uma interrupção gerada externamente pelo acelerómetro. Esta distinção é feita pelo *imp* através de uma função da API

denominada *hardware.wakereason()* que verifica a fonte da interrupção, retornando um valor inteiro. Caso esta função retorne o valor 3, significa que foi detetada uma interrupção através do pino 1 (da *tag*), correspondendo a uma interrupção gerada pelo acelerómetro (e.g. atividade ou inatividade), caso retorne o valor 1 significa que foi gerada uma interrupção do *timer*.

Para cada uma das interrupções referidas são despoletados procedimentos distintos. No caso da interrupção gerada pelo *timer*, a *tag* realiza os três *scans*, envia-os para o *agent* e de seguida é colocada no modo *sleep*.

Quando é detetada uma interrupção no pino 1, a *tag* é colocada no modo *powersave* e é lido o *buffer* do acelerómetro. Mediante esta leitura é possível determinar se a interrupção foi gerada por inatividade ou atividade da *tag*. Caso tenha sido gerada por atividade, o acelerómetro é colocado no modo de medição (com maior precisão), para que seja possível acompanhar (caso necessário) a movimentação da *tag*. Posteriormente, é efetuado um *scan* e são enviadas para o *agent* as informações obtidas através deste, por sua vez o *agent* realiza o *post* para o servidor. Após as informações terem sido enviadas para o *agent* é realizada mais uma leitura ao *buffer* do acelerómetro. Caso haja dados que indiquem que tenha havido movimentação, é feito outro *scan* e as informações são enviadas para o *agent*, caso contrário, o *buffer* é limpo, o acelerómetro é colocado no modo de medição (com mais poupança) e a *tag* é colocada no modo *sleep*.

As leituras do *buffer* do acelerómetro são efetuadas com um período de 2 segundos, visto que a deteção de toda a movimentação da *tag* afeta a eficiência energética. Deste modo, potencia-se a utilização do *buffer* e entre as leituras do *buffer* a *tag* é colocada no modo *sleep* com conectividade Wi-Fi.

Após ter sido detetada movimentação da *tag*, o acelerómetro entra em contagem decrescente para que seja gerada uma interrupção de inatividade. Caso a *tag* esteja inativa durante 2 segundos, é gerada a interrupção de inatividade e a *tag* é adormecida.

Quanto ao ciclo de funcionamento da *tag*, foi definido que esta ficaria no modo *sleep* nos momentos em que esta se encontra imóvel. Relativamente à amostragem periódica, no caso do *Electric imp*, foi determinado que o ciclo para a realização do *scan* teria um intervalo de 30 minutos, ou seja, a cada 30 minutos é feito um *scan*.

Todos os procedimentos efetuados na construção do software da *tag* relativa ao *Electric imp* são apresentados na Figura 5.17.

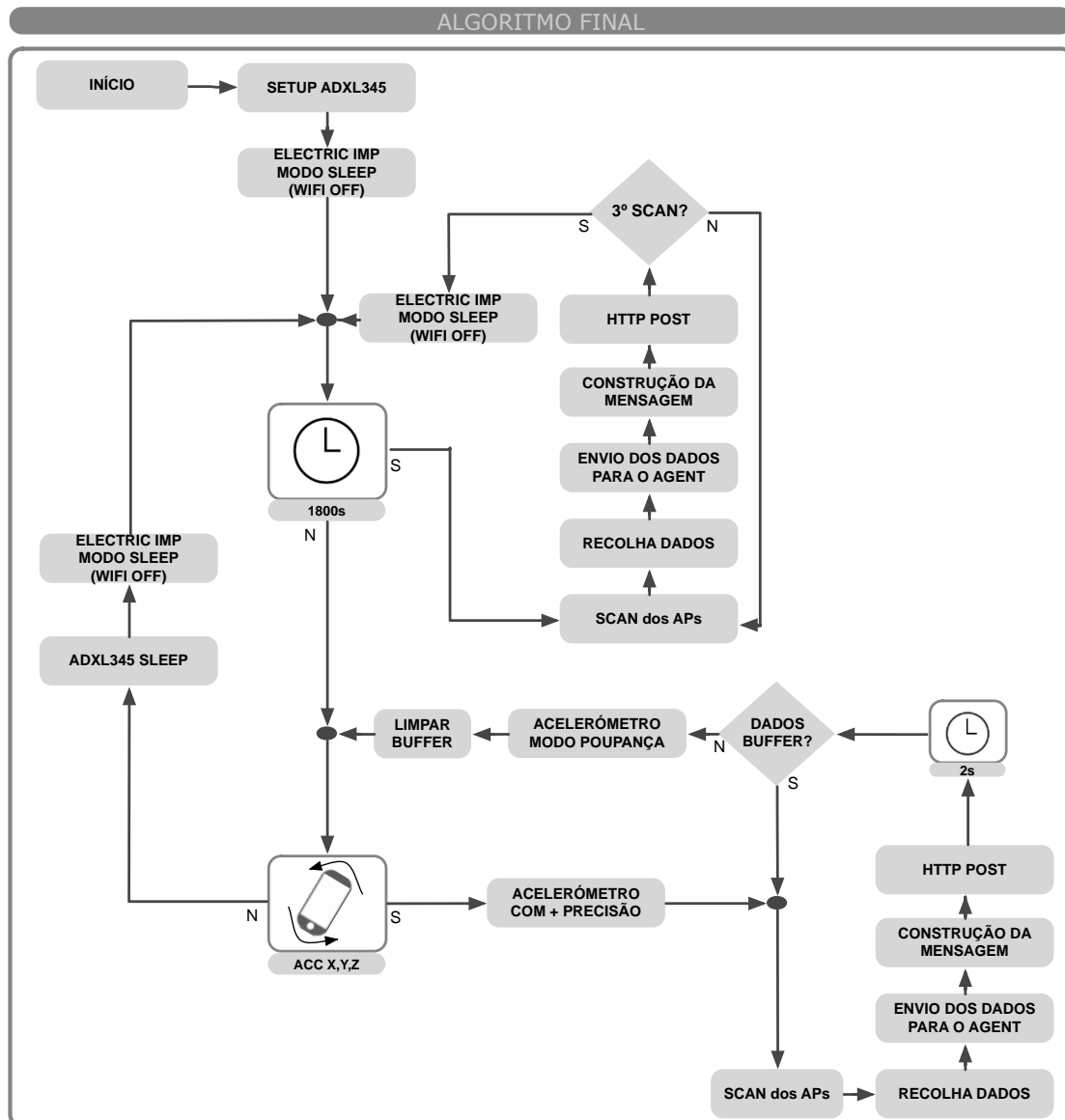


Figura 5.17 – Algoritmo final Electric imp

Após a apresentação do desenvolvimento do *Electric imp*, segue-se a explicação dos passos do desenvolvimento do *Openpicus Flyport Wi-Fi*.

5.3.4. Openpicus Flyport Wi-Fi

O *Openpicus Flyport Wi-Fi* caracteriza-se por ser um dispositivo com funcionamento bastante mais clássico do que o *Electric imp*. À semelhança dos microprocessadores clássicos, a programação do *Flyport Wi-Fi* é desenvolvida utilizando um IDE (Figura 5.18) que necessita de

ser instalado no computador e é transferida para o dispositivo através de uma porta série. A linguagem de programação utilizada é baseada em linguagem C e caracteriza-se como uma linguagem de baixo nível. Como seria de esperar, o processo de compilação e transferência do código para o *Flyport* Wi-Fi é bastante lento.

- Análise da plataforma e configuração inicial.

Inicialmente procedeu-se à análise do IDE de configuração inicial do *Flyport* Wi-Fi. Como referido no parágrafo anterior, a plataforma que permite a programação do *Flyport* Wi-Fi necessita de ser instalada no computador. A primeira tarefa a realizar foi a instalação do software necessário. Para além do IDE, foi necessária a instalação do driver FTDI (*Future Technology Devices International*) que suporta a conversão de RS232 para USB e a instalação do compilador do chip utilizado pelo *Flyport* Wi-Fi (*Microchip C30 Compiler*). Após a instalação do software necessário, procedeu-se à análise do IDE (Figura 5.18). À primeira vista é bastante semelhante com o *Netbeans* ou o *Eclipse*, sendo o acesso às configurações e opções de implementação bastante simples e direto. Ao contrário do *Electric imp*, o *Flyport* Wi-Fi não necessita de nenhuma configuração inicial.

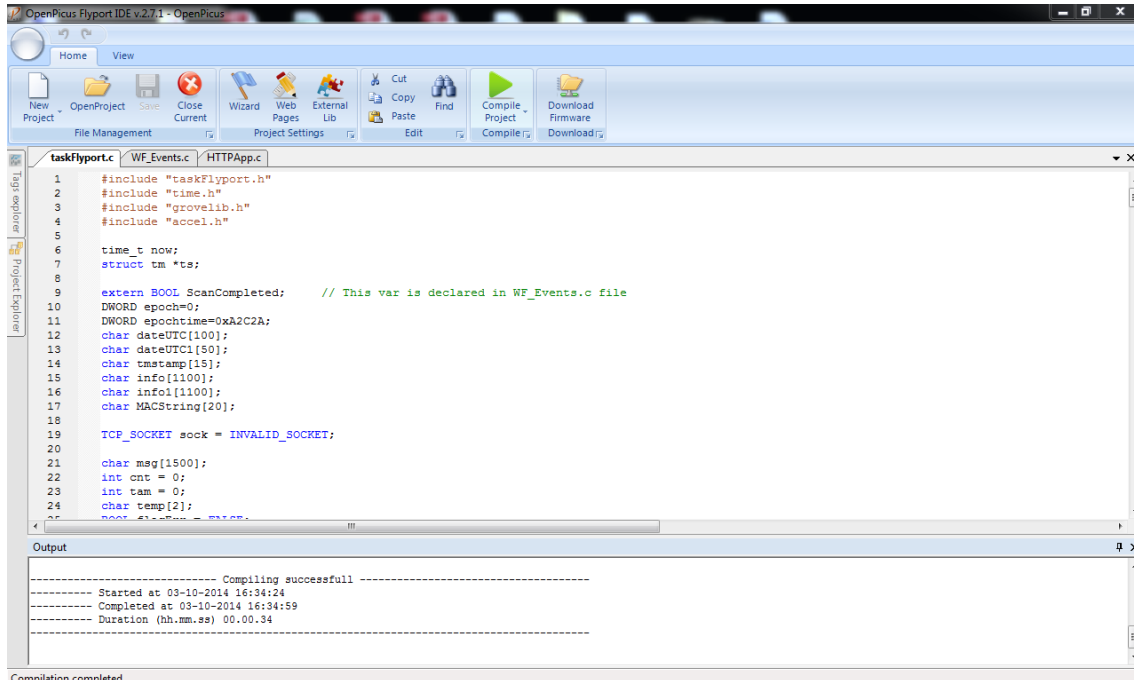


Figura 5.18 - IDE Openpicus Flyport

- Conexão à rede Wi-Fi e scan dos APs

Na segunda fase foi efetuada a conexão à rede Wi-Fi e descoberta a forma de detetar os APs que se encontram na vizinhança do *Flyport* Wi-Fi. O primeiro programa implementado no *Flyport* Wi-Fi foi um programa que estabelecia conexão à rede Wi-Fi. Para realizar esta tarefa existem duas hipóteses, aplicar umas funções que recebem como parâmetros o SSID, palavra passe etc., ou então proceder-se à aplicação das configurações através do *Wizard* disponível no IDE. A solução adotada passou por aplicar a configuração Wi-Fi pelo *Wizard*.

- Recolha de informações sobre APs e geração do *timestamp*

Após a deteção dos APs, seguiu-se a terceira fase onde foi implementada a forma de realização do *scan* e de recolha das informações sobre os APs na vizinhança. No caso do *Flyport* Wi-Fi, existe a particularidade do chip Wi-Fi não permitir que a *tag* esteja conectada à rede Wi-Fi para realizar o *scan* das redes.

Aquando do tratamento do *scan*, como referido na subsecção 4.2, optou-se por colocar uma *flag* a 1 na mensagem quando o *scan* é efetuado com a *tag* em movimento e uma *flag* a 0, caso contrário. Tal como no *Electric imp*, quando a *tag* é acordada pelo *timer*, são realizados três *scans* Wi-Fi com o objetivo de posteriormente no servidor ser possível realizar uma aproximação do valor do RSSI naquele local. As informações sobre os APs que se encontram na vizinhança da *tag* são obtidas através da função que efetua o *scan* Wi-Fi e retorna uma estrutura que contém todos os parâmetros da rede.

Após a realização do *scan*, procede-se à geração do *timestamp* UNIX com o objetivo de registar o momento em este foi efetuado e foram recolhidas as informações dos APs. De realçar que, para gerar o *timestamp* é necessária a conexão à internet¹² devido ao facto de este ser obtido através da função *SNTPGetUTCSeconds()* que envia um pedido SNTP (*Simple Network Time Protocol*) ao servidor NTP (*Network Time Protocol*) e este retorna os segundos que decorreram desde 1 de Janeiro de 1970 até à data em que é feito o pedido.

Finalizadas as tarefas de *scan* e geração do *timestamp* procede-se à construção das mensagens que irão ser enviadas para o servidor.

¹² http://wiki.openpicus.com/index.php/Setting_RTCC_using_SNTP

- Envio dos dados para o servidor

Na quarta fase foi implementada a solução responsável por enviar os dados para o servidor.

Como referido anteriormente, a linguagem de programação utilizada pelo *Flyport* Wi-Fi caracteriza-se como uma linguagem de baixo nível. Este facto torna-se bastante evidente no momento em que se processa o envio dos dados para o servidor, uma vez que requer um conjunto de tarefas, nomeadamente, criação do *socket* TCP (*Transmission Control Protocol*), conexão do *socket* ao servidor e criação da *string* com o *HTTP POST*. Só depois da realização desta última tarefa, é possível proceder à escrita do *socket* enviando o *HTTP POST* (Figura 5.19).

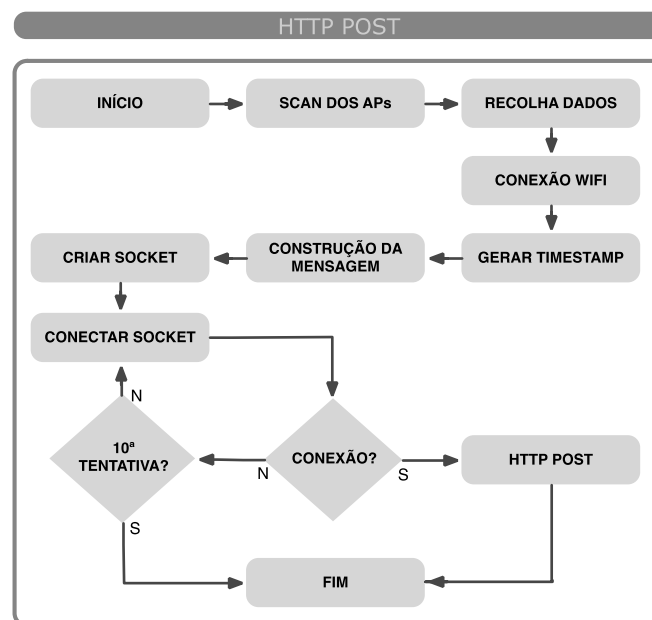


Figura 5.19 - Http Post Flyport Wi-Fi

- Implementação do ciclo para o *scan*

Com o objetivo de reduzir o consumo energético, optou-se por adormecer a *tag* nos momentos em que esta não se encontra em movimento. Nesta fase foi utilizada uma função que coloca a *tag* no modo *sleep* e a desconecta da rede Wi-Fi. Assim, a *tag* só pode ser despertada através da interrupção gerada pelo acelerómetro ou através do disparo de um alarme. O alarme

do *Flyport* Wi-Fi tem a particularidade de apresentar uma lista de opções de tempo limitada, ou seja, não é possível colocar um valor pretendido que não esteja nessa lista descrita na sua API¹³.

Atendendo a este facto, foi decidido utilizar um alarme de 1 minuto (o alarme disponível no nível abaixo deste é de 10 segundos e acima é de 10 minutos) para que fosse possível verificar o comportamento do *Flyport* Wi-Fi. Desta forma, de 1 em 1 minuto a *tag* é despertada, realiza o *scan*, conecta-se à rede Wi-Fi e calcula o *timestamp*. Em seguida, são realizadas um conjunto de tarefas presentes na Figura 5.20, passando pelo *HTTP POST* e finalizando no *timer*.

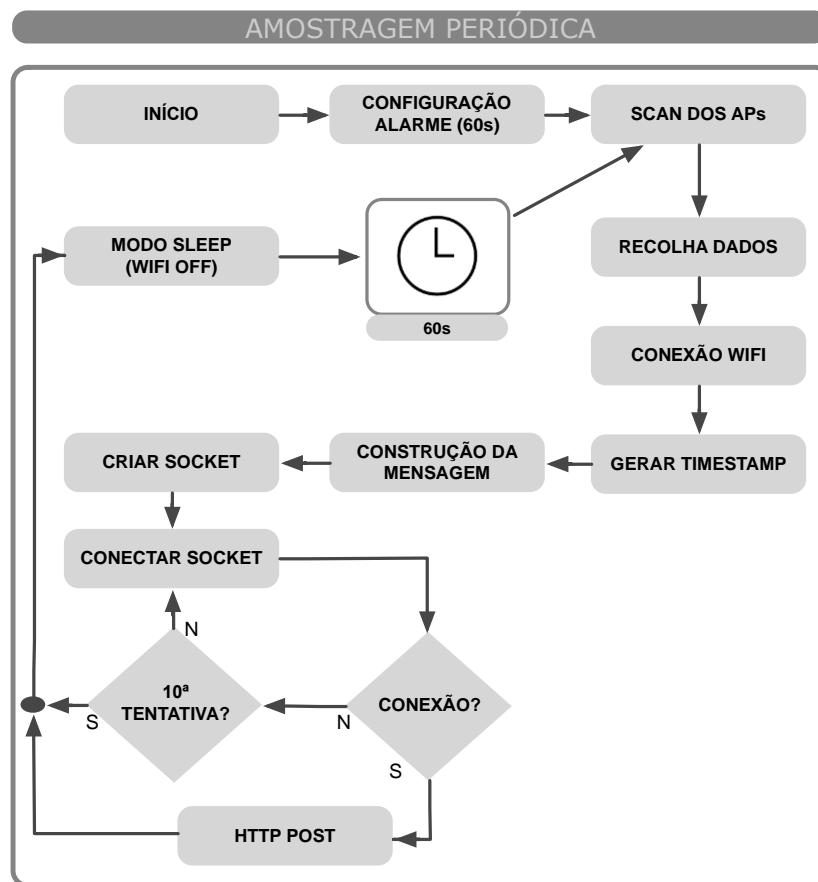


Figura 5.20 - Algoritmo quinta fase

- Integração do acelerómetro na *tag* e deteção da movimentação

Na sexta fase foi efetuada a integração do acelerómetro na *tag* e posteriormente a deteção da movimentação da *tag*. Como referido e justificado na subsecção 5.3.2, o protocolo escolhido para a comunicação entre o acelerómetro e a *tag* foi o I²C.

¹³A respetiva lista encontra-se no site: <http://wiki.openpicus.com/index.php?title=RTCC>

Após a junção do *Flyport* Wi-Fi com o acelerómetro, o primeiro procedimento para a configuração do acelerómetro foi investigar como transmitir e receber informações deste. Aquando deste procedimento chegou-se à conclusão que afinal a API do *Flyport* Wi-Fi ainda não estava desenvolvida o suficiente, verificando-se que o suporte existente para o acelerómetro era bastante diminuto, tendo apenas uma biblioteca, que se revelou limitada, disponível para download no site do *Openpicus*¹⁴. Perante a impossibilidade do envio das configurações para o acelerómetro através do código principal, a solução passou pela alteração do próprio código da biblioteca, fazendo com que o acelerómetro fosse inicializado com os comandos introduzidos diretamente na biblioteca.

Tal como no *Electric imp*, a configuração inicial implementada passou pela realização de leituras sucessivas dos valores de todos os eixos. Em seguida, passou-se ao estudo do *datasheet*¹⁵, de forma a realizar a configuração do acelerómetro com vista à deteção de atividade e inatividade da *tag*. A configuração do acelerómetro com vista a deteção da atividade e inatividade, pode ser visualizada na Figura 5.21.

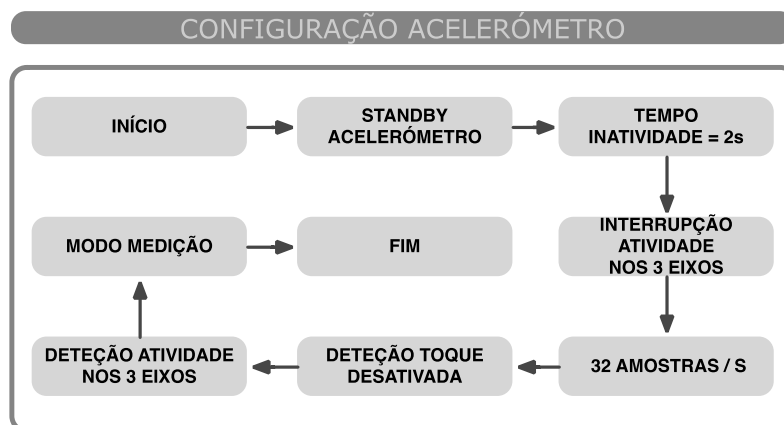


Figura 5.21 - Configuração acelerómetro MM7660FC

À semelhança do acelerómetro do *Electric imp*, este foi configurado de forma a gerar uma interrupção quando deteta movimento e entrada no modo *sleep* quando deteta inatividade. De forma que o *Flyport* Wi-Fi detete interrupções é necessário configurar um pino como sendo uma interrupção. O pino escolhido foi o digital 2 disponível na placa *Grove*, que possui uma ligação ao pino 5 do *Flyport* Wi-Fi. Quando esta interrupção é gerada, são desativados o alarme e

¹⁴ Respetivo site: http://wiki.openpicus.com/index.php/GROVE_-_3_axis_accelerometer

¹⁵ *Datasheet* acelerómetro: http://www.freescale.com/files/sensors/doc/data_sheet/MMA7660FC.pdf

a interrupção e são efetuados todos os processos até ao *HTTP POST* ser concluído. Nessa altura o alarme e a interrupção são ativos.

O algoritmo de deteção da atividade e inatividade pode ser verificado na Figura 5.22, onde se destaca que o *setup* do acelerómetro corresponde à configuração do acelerómetro que foi descrita anteriormente.

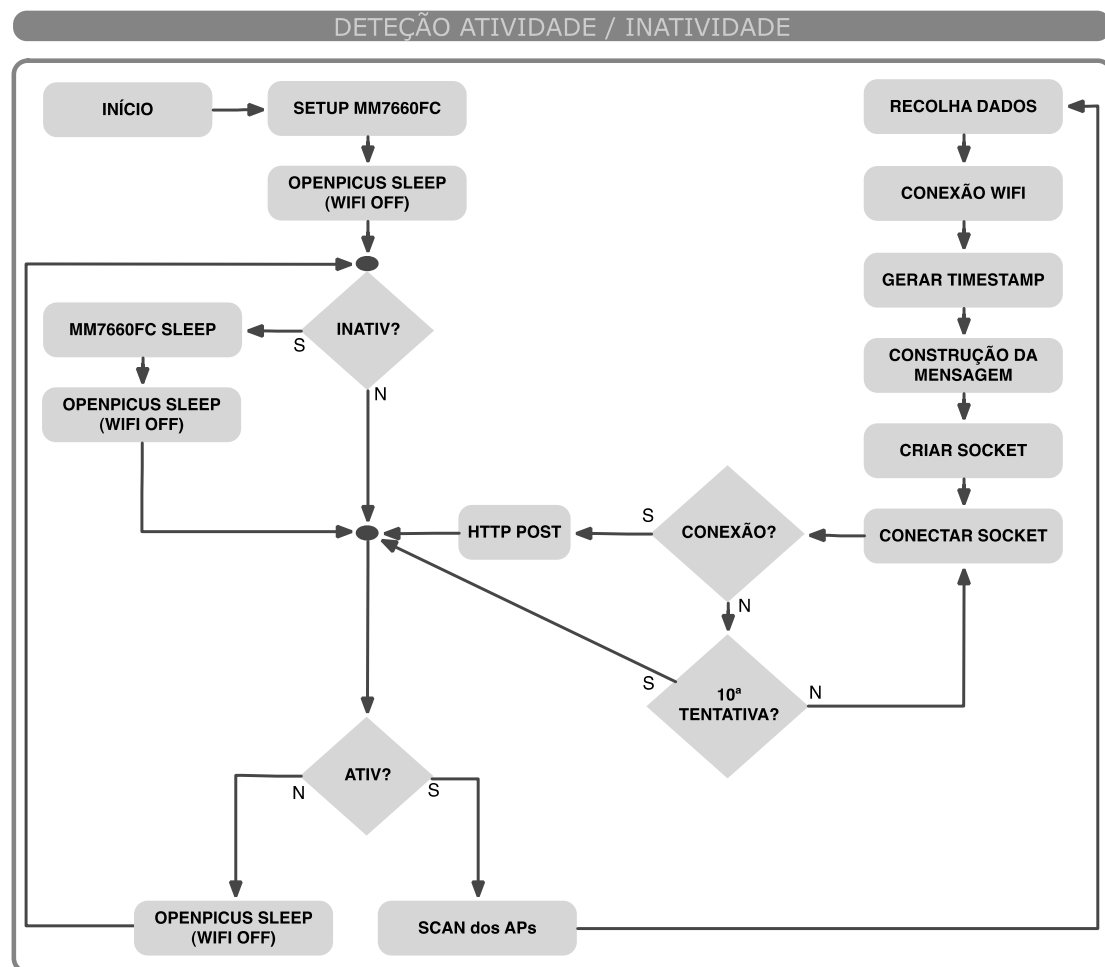


Figura 5.22 – Algoritmo deteção atividade / inatividade Flyport Wi-Fi

- Potenciar a utilização do acelerómetro

Na sétima fase, o objetivo centrou-se em potenciar a utilização do acelerómetro e gerir o ciclo de funcionamento da *tag* em função do movimento. Tal como no caso do *Electric imp*, esta foi a fase dos ajustes das configurações, de forma que o comportamento da *tag* correspondesse ao pretendido.

A *tag* só pode ser acordada através de uma interrupção ou do alarme gerado pelo *timer*. Quando a *tag* é despertada, é verificado o motivo pelo qual isso ocorreu. Caso tenha sido detetada uma interrupção através do pino 5 significa que foi gerada uma interrupção pelo acelerómetro (e.g. atividade ou inatividade), por outro lado, caso tenha sido gerado um alarme pelo *timer* a *tag* desativa a interrupção no pino 5, realiza os três *scans*, cria a mensagem, envia-a para o servidor, ativa a interrupção no pino 5 e entra no modo *sleep*.

Quando é detetada uma interrupção no pino 5, a *tag* desativa a interrupção no pino 5, desativa o *timer* e realiza uma leitura do *buffer* do acelerómetro, para determinar qual a origem da interrupção. Caso seja verificada uma interrupção de atividade, a *tag* efetua um *scan*, cria a mensagem, envia-a para o servidor e verifica se o *buffer* do acelerómetro contém algum dado. Este processo é repetido enquanto o acelerómetro tiver dados no *buffer*. Ao contrário do *Electric imp*, não é colocado nenhum atraso entre as leituras do *buffer*, visto que o processo de conexão à rede Wi-Fi é bastante lento.

Quando o acelerómetro deixa de conter dados no *buffer*, a *tag* ativa a interrupção no pino 5, ativa o *timer* e entra no modo *sleep*. Quanto ao ciclo de funcionamento da *tag*, foi definido que esta ficaria no modo *sleep* enquanto permanecer imóvel.

Na quinta fase, foi implementado um algoritmo onde a *tag* era colocada no modo *sleep*, no entanto, nessa fase o *timer* foi configurado com um ciclo de 1 em 1 minuto. Para a versão final, esse tempo revelou-se demasiado curto, pelo que se alterou para um ciclo de 1 hora.

Todos os procedimentos efetuados na construção do software da *tag* relativa ao *Flyport* Wi-Fi são apresentados na Figura 5.23.

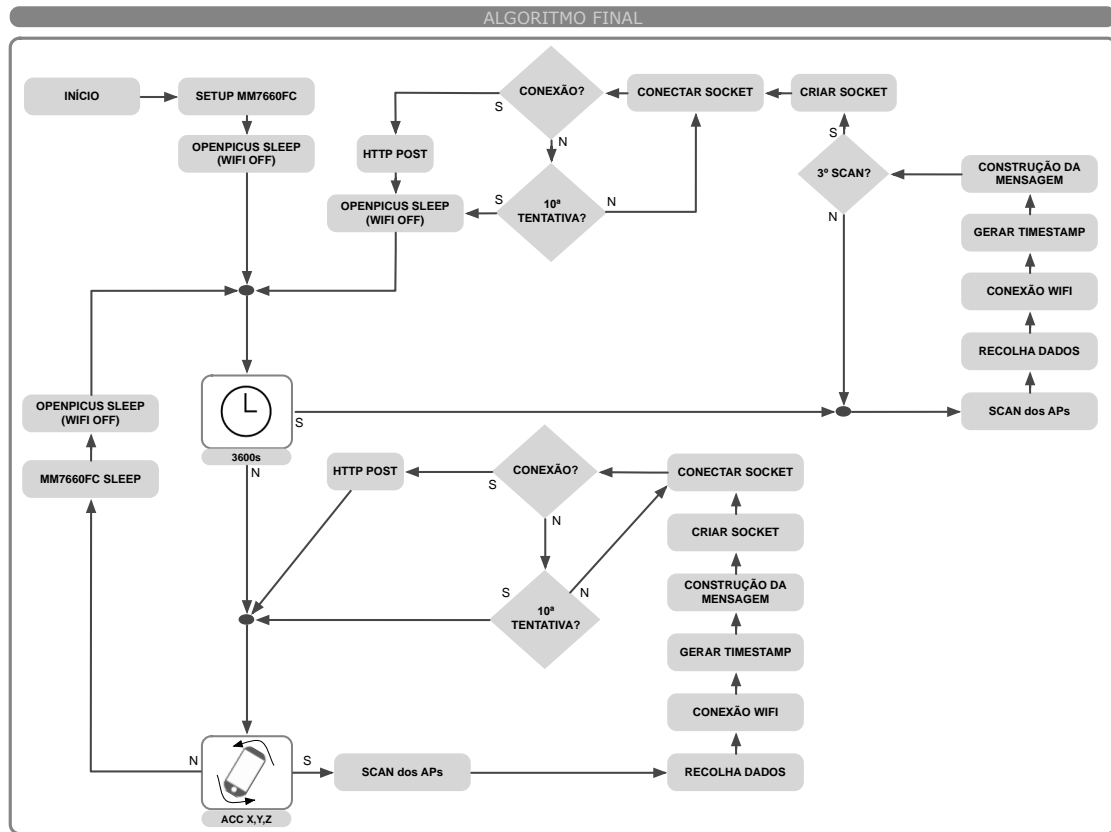


Figura 5.23 - Algoritmo Final FLYport Wi-Fi

6. Medição dos Consumos e Análise da Autonomia

Neste capítulo é feita a análise do sistema, apresentando-se a medição dos consumos e análise da autonomia e fiabilidade de ambas as *tags*.

6.1. Medição dos Consumos

Antes de se iniciar os testes de autonomia e fiabilidade, foi utilizado um osciloscópio para verificar quais os consumos que cada *tag* estava a realizar, tanto no modo *sleep* como no modo ativo. Com estas medições foi possível implementar pequenas alterações que fizeram com que os consumos reduzissem. Nas figuras apresentadas em seguida, a medição da *tag* é representada pela cor amarela e a azul a medição efetuada ao acelerómetro.

6.1.1. Medições Electric imp

Nas primeiras medições do consumo energético efetuadas ao *Electric imp* verificou-se que os LEDs, tinham um grande impacto nos consumos energéticos. Verificou-se ainda que com a ativação do modo *setpowersave* foi possível poupar energia nos momentos em que o *agent* processa os dados e realiza o *http post*. Foi ainda possível determinar o tempo que demora desde que a *tag* acorda até entrar no modo *sleep*. As medições efetuadas ao acelerómetro e às *tags* foram efetuadas com recurso ao osciloscópio Tektronix TDS 2012B¹⁶.

Para as medições foram utilizadas duas resistências, uma de 1 Ω e outra de 750 Ω , e um condensador de 0.047 μ F. Inicialmente experimentou-se utilizar uma resistência de 10 Ω para medir o consumo do *Electric imp*, mas verificou-se que a queda de tensão era demasiado grande fazendo com que o *imp* não se conseguisse conectar à rede Wi-Fi. Desta forma, utilizou-se a resistência de apenas 1 Ω em conjunto com o condensador para manter a tensão mais estável. Para as medições do acelerómetro foi utilizada uma resistência de 750 Ω .

¹⁶ http://www2.tek.com/cmsreplive/psrep/13295/3GM_19558_2_2011.02.10.07.25.34_13295_ES-MX.pdf

Devido à utilização de resistências diferentes, a análise dos gráficos apresentados de seguida requer atenção pois apesar de ambos estarem com a mesma escala, só é possível chegar a alguma conclusão determinando a corrente através da lei de *Ohm*. A utilização direta da lei de *Ohm* deveu-se ao facto de se estar perante corrente continua.

No caso do *Electric imp*, foram realizadas medições quando este efetua um *scan* acordado pelo acelerómetro, realiza dois *scans* consecutivos quando acordado pelo acelerómetro, realiza um *scan* acordado pelo acelerómetro e com os LEDs ligados e *prints* no código, e realiza três *scans* quando acordado pelo *timer*. Desta forma, foi possível comparar várias situações diferentes, nomeadamente quando a *tag* consumia mais e de que forma seria possível reduzir o consumo energético.

A primeira medição efetuada foi com a *tag* com os LEDs ativos e *prints* no código. Na Figura 6.1 (a azul o acelerómetro e a amarelo a *tag*) é possível verificar esta primeira medição, onde se conclui que a *tag* demora cerca de 25 segundos desde que acorda até ao momento em que é colocada no modo *sleep* e que o maior gasto energético está relacionado com o momento em que é feita a conexão à rede Wi-Fi e com o momento em que realiza o *scan*.

Foi possível verificar que os fatores que contribuíam para um maior gasto energético no *Electric imp* era o facto de este depender totalmente da conexão à internet, o que fazia com que os *prints* que estavam no código induzissem a *tag* a realizar mais envios de informação para o *imp cloud* prolongando o tempo de execução.

No caso do acelerómetro, é possível verificar que este é responsável por acordar a *tag*. Antes de este ser colocado no modo de medição apresenta uma quebra no consumo pelo facto de ser colocado em *standby* e só é colocado no modo *sleep* quando deteta inatividade.

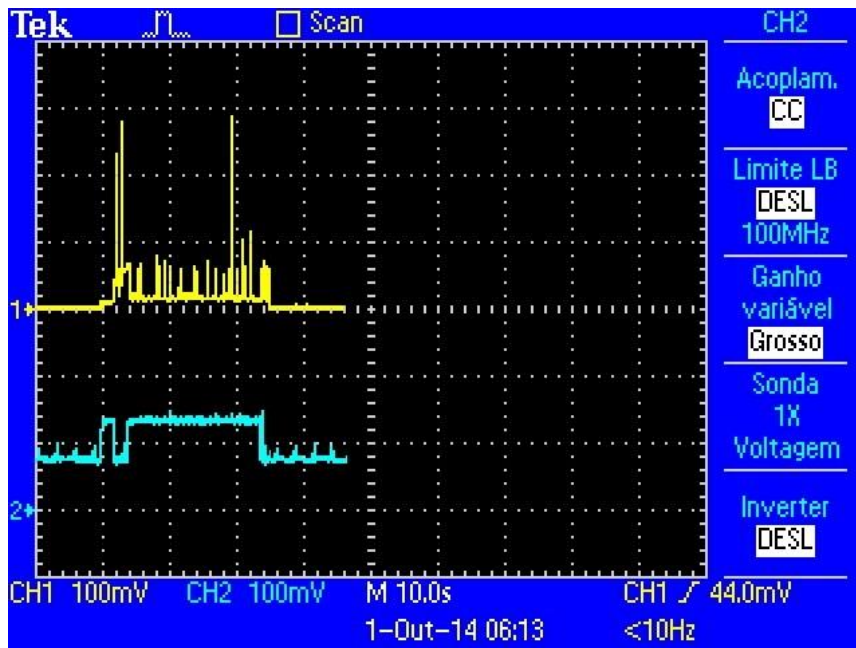


Figura 6.1 - Medição 1 scan com LEDs e Prints

Após esta medição, procedeu-se à alteração das configurações da *tag* desligando os LEDs e eliminando os *prints*. Foi realizada uma nova medição que é apresentada na Figura 6.2.

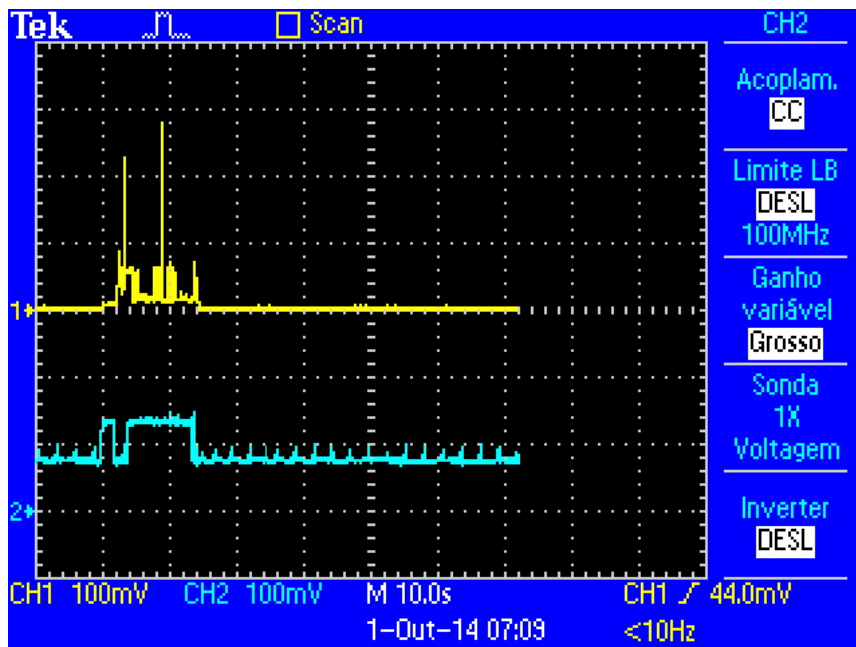


Figura 6.2 - Electric imp 1 scan

É possível verificar que houve uma redução do tempo de execução da *tag*, demorando desta forma cerca de 14 segundos desde que é acordada até entrar no modo *sleep*. Para além disso, é possível verificar que o momento em que a *tag* se conecta à rede Wi-Fi corresponde ao pico verificado aos 5 segundos e o momento em que esta efetua o *scan* corresponde ao pico verificado aos 9 segundos.

O acelerómetro mantém o mesmo comportamento verificado na medição anterior, reduzindo apenas o seu tempo de execução uma vez que este está diretamente relacionado com a execução da *tag*.

A medição apresentada na Figura 6.3 corresponde à realização de dois *scans* consecutivos quando a *tag* é acordada pelo acelerómetro.

Ao analisar a Figura 6.3 verifica-se que, desde quando a *tag* acordou e foi colocada no modo *sleep*, passaram cerca de 18 segundos e que os *scans* são efetuados com espaçamento de 2 segundos. O acelerómetro manteve o comportamento registado nas duas medições anteriores.

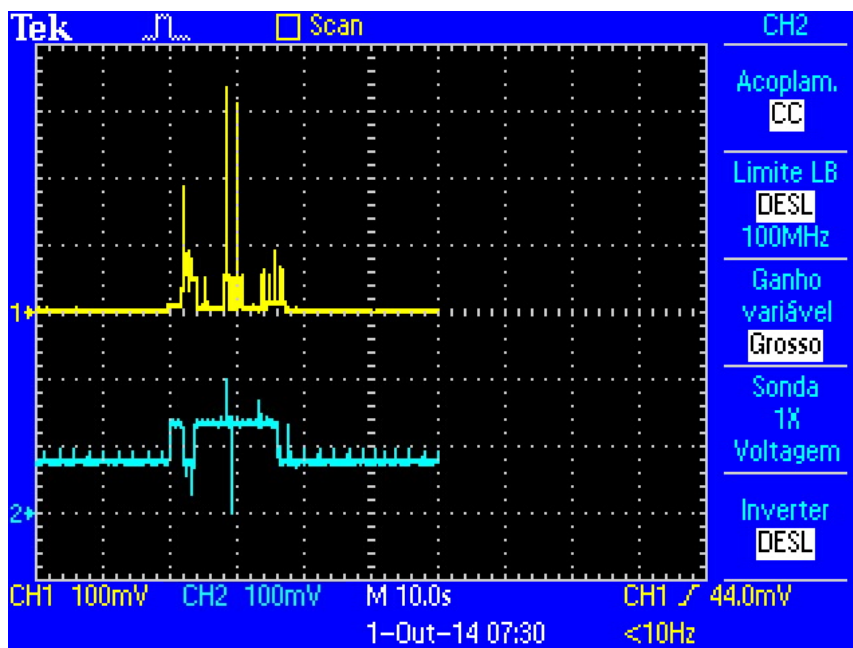


Figura 6.3 - Medição dois scans *Electric imp* ativados pelo acelerómetro

A última medição efetuada ao *Electric imp* foi no caso de a *tag* ser acordada pelo timer. Como explicado na secção 5.3.3, quando a *tag* é acordada pelo timer, esta realiza três scans. Devido a este facto, tornou-se relevante verificar os consumos quando esta situação ocorre.

Na Figura 6.4 é possível verificar a medição da *tag* quando esta é acordada pelo *timer*. A partir desta figura conclui-se que desde que a *tag* acorda até ao momento em que é colocada no modo *sleep*, decorreram cerca de 15 segundos. Apesar de realizar três *scans*, o tempo total despendido é inferior ao tempo da realização de *scans* quando a *tag* é acordada pelo acelerómetro (exceto quando realiza apenas um scan). Este facto deve-se a vários fatores, tais como os *scans* realizados pela *tag* em movimento serem espaçados por 2 segundos e o facto de a *tag* ter de receber a interrupção de inatividade (também 2 segundos) gerada pelo acelerómetro o que faz com que haja um maior tempo de espera para que esta seja colocada no modo *sleep*. Verifica-se ainda que o acelerómetro se manteve no modo *sleep* durante o tempo de execução da *tag*.

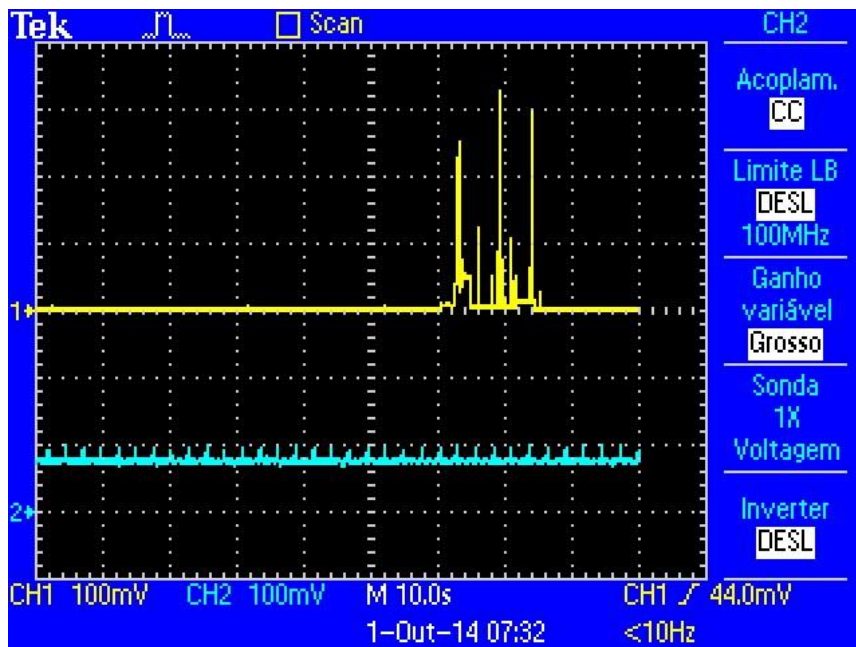


Figura 6.4 - Medição três scans ativados pelo timer

Com base nas figuras apresentadas foi possível produzir um gráfico do tipo onda quadrada onde é descrito o funcionamento geral da *tag*, tendo em conta os tempos de execução e as *flags* que possibilitam a alteração de estados do acelerómetro.

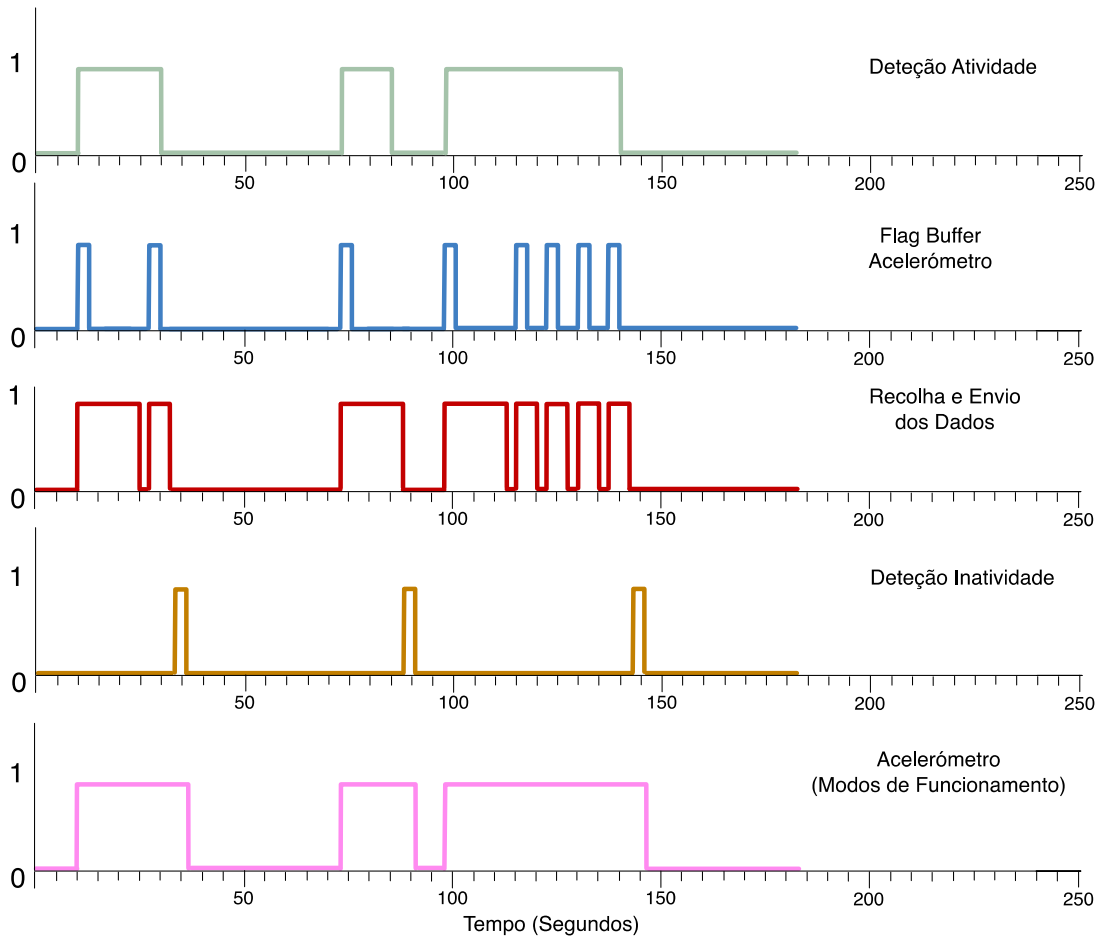


Figura 6.5 - Funcionamento geral *Electric imp*

Com base nas figuras apresentadas anteriormente, foi possível determinar os consumos da *tag* baseada no *Electric imp*.

Em todos os casos apresentados, verifica-se que quando a *tag* está no modo *sleep* consome pouca energia, sendo difícil determinar o consumo nesse modo através do osciloscópio. Assim, recorreu-se a um multímetro para realizar essa medição. Nas medições efetuadas através do multímetro, concluiu-se que quando a *tag* se encontra no modo *sleep* consome $94\mu\text{A}$.

Para determinar os consumos no modo ativo, recorreu-se às imagens obtidas através do osciloscópio. Nas figuras apresentadas é possível determinar que quando a *tag* se encontra no modo ativo a queda de tensão numa resistência de 1Ω é de 60mV , com períodos entre 10 e 20mV , e quando ocorrem os *scans* e envio das informações para o *imp cloud* o máximo registado foi de 340mV .

Aplicando a lei de *Ohm*:

$$U = R \times I \Leftrightarrow I = \frac{U}{R} \Leftrightarrow I = \frac{0,06}{1} \Leftrightarrow I = 60mA$$

Para os períodos em que se verifica 10mV:

$$U = R \times I \Leftrightarrow I = \frac{U}{R} \Leftrightarrow I = \frac{0,01}{1} \Leftrightarrow I = 10mA$$

Quando se verifica 20mV:

$$U = R \times I \Leftrightarrow I = \frac{U}{R} \Leftrightarrow I = \frac{0,02}{1} \Leftrightarrow I = 20mA$$

Nos picos de consumo, o máximo registado foi de:

$$U = R \times I \Leftrightarrow I = \frac{U}{R} \Leftrightarrow I = \frac{0,34}{1} \Leftrightarrow I = 340mA$$

No caso do acelerómetro, quando este se encontra no modo *sleep* foram registadas quedas de tensão entre 70mV e 100mV. Quando se encontra no modo ativo, foram registadas quedas de tensão de 140mV. Neste caso a resistência utilizada foi de 750Ω, pelo que aplicando a lei de Ohm se obtém:

$$U = R \times I \Leftrightarrow I = \frac{U}{R} \Leftrightarrow I = \frac{0,07}{750} \Leftrightarrow I = 93\mu A$$

$$U = R \times I \Leftrightarrow I = \frac{U}{R} \Leftrightarrow I = \frac{0,1}{750} \Leftrightarrow I = 133\mu A$$

No modo *sleep*, o acelerómetro consome entre 93 e 133μA.

Para o modo ativo, obtém-se:

$$U = R \times I \Leftrightarrow I = \frac{U}{R} \Leftrightarrow I = \frac{0,14}{750} \Leftrightarrow I = 187\mu A$$

Após o cálculo dos consumos é possível realizar uma comparação dos valores obtidos com os valores anunciados pelos fabricantes, tanto do *Electric imp* como do acelerómetro.

		Anunciado	Medido
Electric imp	Operação normal WiFi modo poupança energia	Entre 5mA e 400mA nas transmissões WiFi	Entre 10mA e 340mA nas transmissões WiFi
	Modo Sleep WiFi desligado	6uA	94uA
Acelerómetro	Modo Sleep	34uA	Entre 93uA e 133uA
	Modo Ativo	50uA	187uA

Tabela 6.1 - Comparação dos consumos anunciados¹⁷ e medidos *Electric imp*

Na Tabela 6.1 verifica-se um maior desfasamento no modo *sleep* do *Electric imp* e nos consumos do acelerómetro. Como no modo *sleep* existe uma grande diferença entre o consumo anunciado e consumo medido, desconectou-se o acelerómetro e verificou-se que o consumo desceu para 23 μ A. Existe também o facto dos consumos anunciados terem sido medidos com uma tensão de alimentação de 2,5V e as medições efetuadas terem sido realizadas com uma alimentação superior a 3,5V, o que provoca um aumento do consumo.

6.1.2. Medições Openpicus Flyport Wi-Fi

No caso do *Flyport Wi-Fi* os LEDs também foram desligados. Foi ainda possível determinar o tempo que demora desde o momento que a *tag* acorda e entra no modo *sleep*. As medições efetuadas aos dois dispositivos (acelerómetro e *tag*) foram realizadas separadamente.

Tal como no *Electric imp*, inicialmente para a realização das medições também foram utilizadas as duas resistências (de 1 Ω e 750 Ω) e um condensador de 0,047 μ F, onde a resistência de 1 Ω em conjunto com o condensador foram utilizados para a medição do consumo energético do *Flyport Wi-Fi* e a resistência de 750 Ω foi utilizada para a medição de consumo do acelerómetro. No entanto verificou-se que a queda de tensão proporcionada pela resistência de 750 Ω era demasiado grande, fazendo com que o acelerómetro não recebesse tensão suficiente para o seu funcionamento. Deste modo, também se utilizou uma resistência de 1 Ω para as

¹⁷ Consumos verificados nos respetivos *datasheets*.

medições do acelerómetro. Como se continua perante corrente contínua, voltou-se a utilizar a lei de *Ohm* para determinar a corrente gasta pela *tag*.

Para a *tag* baseada no *Openpicus Flyport* Wi-Fi, foram realizadas três tipos de medições. A primeira consistiu na *tag* ser acordada pelo acelerómetro e realizar um *scan* tendo a particularidade de realizar a pesquisa para a conexão na rede Wi-Fi em todos os canais. A segunda baseou-se em efetuar medições quando a *tag* era acordada pelo acelerómetro, realizava um *scan* mas ao contrário da medição anterior, neste caso a *tag* só estava configurada para pesquisar a rede Wi-Fi num único canal. Por fim, a última medição consistiu no caso de a *tag* ser acordada pelo timer e realizar os três *scans*.

Ao analisar a Figura 6.6 o que desperta logo à atenção é o tempo que a *tag* demora para realizar um *scan* e a tensão ser praticamente constante durante o tempo de execução.

Para o *scan* ser efetuado a *tag* não pode estar conectada à rede Wi-Fi. Após a realização do *scan*, a *tag* é conectada à rede Wi-Fi onde se verifica o maior gasto de tempo, pois é feita uma pesquisa em todos os canais para onde o AP está a emitir o sinal. De seguida procede-se à criação do *socket* e respetiva conexão ao servidor que por vezes demora algum tempo.

Na Figura 6.6 é possível verificar que o tempo despendido pela *tag* entre o momento em que acorda e entra no modo *sleep* é de cerca de 48 segundos, o que representa desde logo ser bastante tempo. Verifica-se ainda que o maior gasto energético ocorreu aos 42 segundos e corresponde ao momento em que é efetuado o *HTTP POST*. O acelerómetro não apresenta alterações uma vez que os seus consumos são na ordem dos μA , o que torna difícil serem detetados pelo osciloscópio.

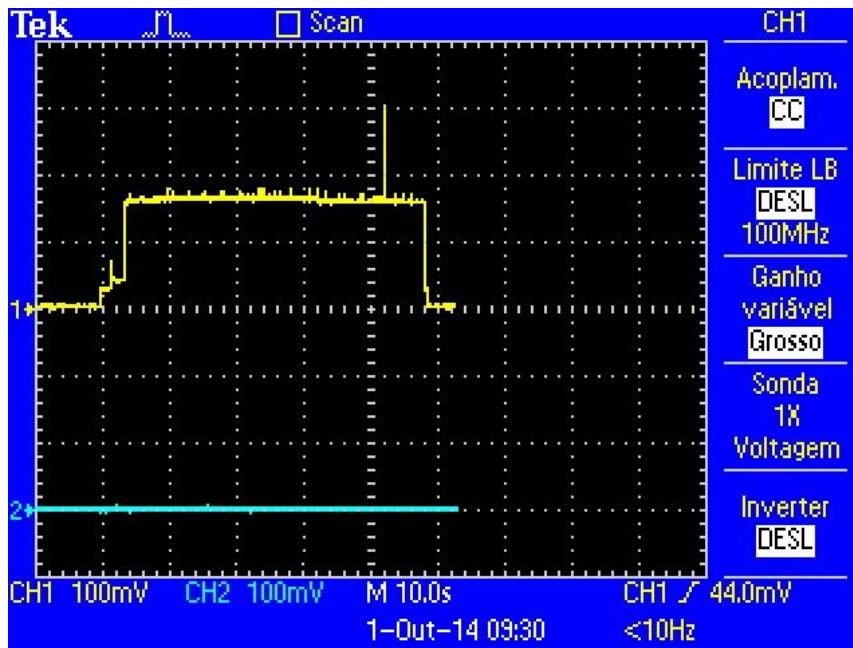


Figura 6.6 - Medição um scan Flyport Wi-Fi ativado pelo acelerómetro com todos os canais

Após a realização da primeira medição, procedeu-se a alterações na configuração da *tag*, nomeadamente na redução da pesquisa de canais pela rede Wi-Fi para apenas 1 canal em vez dos 12 configurados inicialmente.

Desta forma, assistiu-se a uma redução do tempo de execução da *tag* que passou de 48 segundos para cerca de 40 segundos. Na Figura 6.7 é possível verificar o gráfico correspondente à medição da *tag* quando esta é acordada pelo acelerómetro e realiza apenas um *scan*.

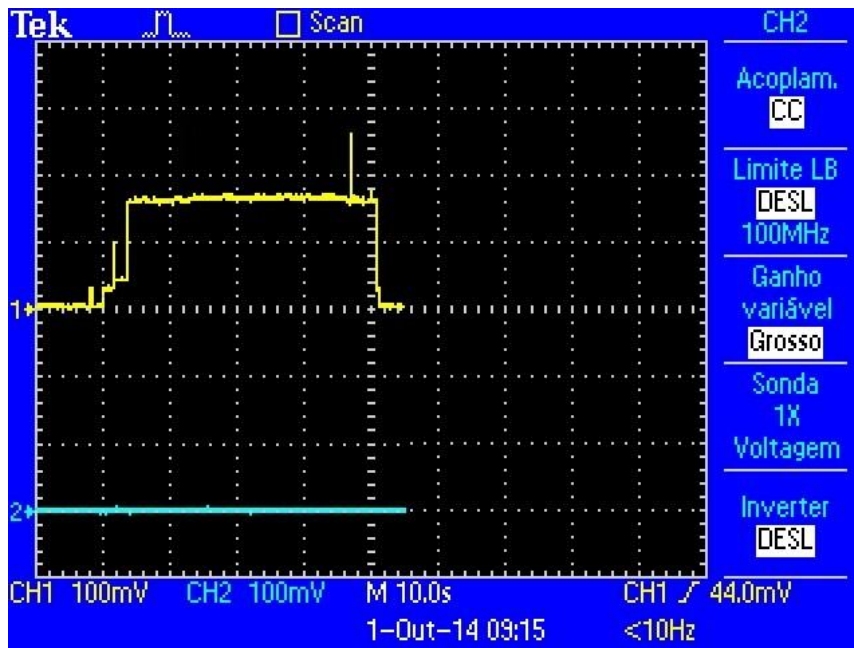


Figura 6.7 - Medição 1 scan acelerômetro 1 canal

Na Figura 6.8 é possível verificar a medição correspondente ao momento em que a *tag* é acordada pelo *timer* e realiza os três *scans*. É possível concluir que o tempo de execução da *tag* desde que acorda até ao momento em que entra no modo *sleep* é de cerca de 52 segundos.

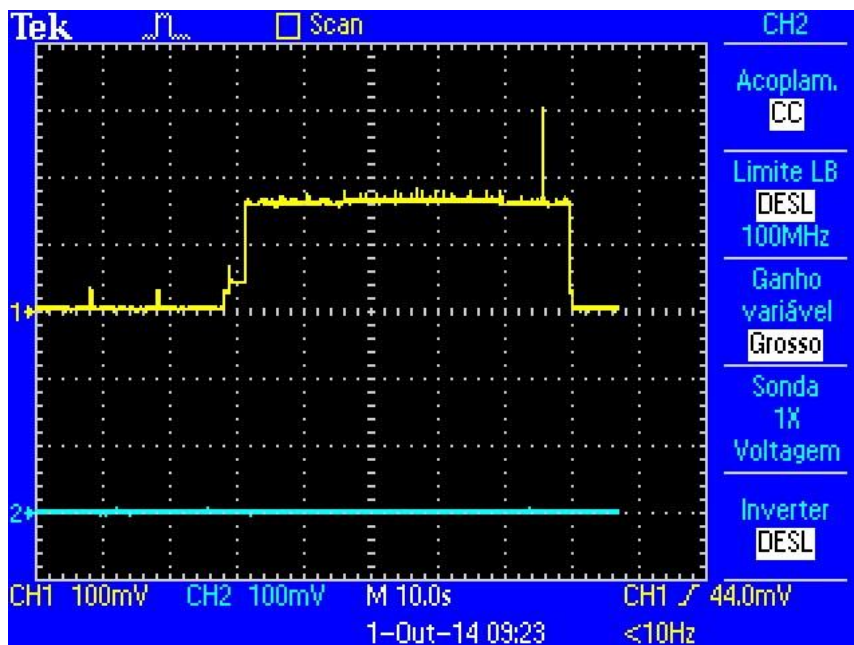


Figura 6.8 - Medição 3 scans Timer Openpicus

Com base nas figuras apresentadas foi possível produzir um gráfico do tipo onda quadrada onde é descrito o funcionamento geral da *tag* baseada no *Flyport* Wi-Fi, tendo em conta os tempos de execução e as *flags* que possibilitam a alteração de estados do acelerómetro.

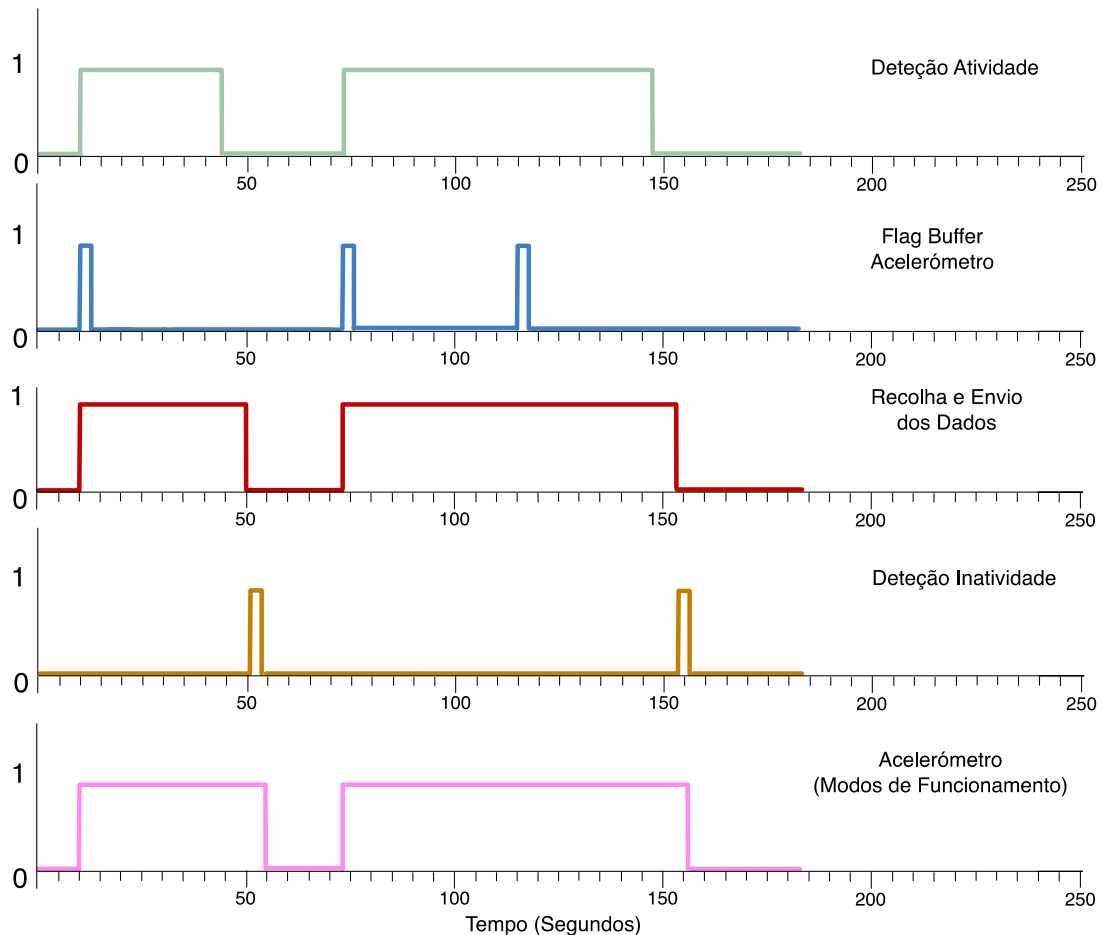


Figura 6.9 - Funcionamento geral *Flyport* Wi-Fi

Após todas as medições realizadas pode-se concluir que a *tag* apresenta um tempo de execução demasiado grande, despendido maioritariamente no momento da conexão à rede Wi-Fi. É possível ainda concluir que a *tag* é bastante vulnerável no facto da conexão do *socket* ao servidor poder demorar algum tempo a ser realizada.

Com base nas figuras apresentadas anteriormente, foi possível determinar os consumos da *tag* baseada no *Flyport* Wi-Fi.

Em todos os casos apresentados, verifica-se que quando a *tag* está no modo *sleep* consome pouca energia. Para determinar o consumo neste modo, recorreu-se à utilização de um multímetro. Nas medições efetuadas através do multímetro, concluiu-se que quando a *tag* se encontra no modo *sleep* apresenta uma queda de tensão na resistência (com valor 1Ω) de $2,1\text{mV}$. Tendo em conta que a resistência utilizada foi de 1Ω e aplicando a lei de *Ohm* para determinar a corrente, tem-se que:

$$U = R \times I \Leftrightarrow I = \frac{U}{R} \Leftrightarrow I = \frac{0,0021}{1} \Leftrightarrow I = 2,1\text{mA}$$

Para determinar os consumos no modo ativo, recorreu-se às imagens obtidas através do osciloscópio. Nas figuras apresentadas é possível determinar que quando a *tag* se encontra no modo ativo, apresenta uma queda de tensão na resistência de cerca de 160mV e quando realiza o envio das informações para o servidor a tensão máxima registada foi de 300mV .

Aplicando a lei de *Ohm* para se obter os consumos:

$$U = R \times I \Leftrightarrow I = \frac{U}{R} \Leftrightarrow I = \frac{0,16}{1} \Leftrightarrow I = 160\text{mA}$$

Nos picos de consumo, o máximo registado foi:

$$U = R \times I \Leftrightarrow I = \frac{U}{R} \Leftrightarrow I = \frac{0,3}{1} \Leftrightarrow I = 300\text{mA}$$

Em todos os casos apresentados, verifica-se que o acelerómetro apresenta um sinal demasiado fraco para que se consiga determinar a sua tensão através do osciloscópio. Desta forma, foi necessário recorrer a um multímetro para realizar essa medição. Nas medições efetuadas através do multímetro, concluiu-se que quando o acelerómetro se encontra no modo *sleep*, existe uma queda de tensão de $64\mu\text{V}$ e quando se encontra no modo ativo $206\mu\text{V}$. Tendo em conta que a resistência utilizada foi de 1Ω e aplicando a lei de *Ohm* para determinar a corrente, tem-se que:

$$U = R \times I \Leftrightarrow I = \frac{U}{R} \Leftrightarrow I = \frac{0,000064}{1} \Leftrightarrow I = 64\mu\text{A}$$

No modo *sleep*, o acelerómetro consome entre $67\mu\text{A}$.

Para o modo ativo, obtém-se:

$$U = R \times I \Leftrightarrow I = \frac{U}{R} \Leftrightarrow I = \frac{0,000206}{1} \Leftrightarrow I = 206\mu A$$

Após o cálculo dos consumos é possível realizar uma comparação dos valores obtidos com os valores anunciados pelos fabricantes, tanto do *Flyport* Wi-Fi como do acelerómetro.

		Anunciado	Medido
Openpicus	Operação normal WiFi ligado	Entre 162,70mA e 282,50mA nas transmissões WiFi	Entre 160mA e 300mA nas transmissões WiFi
	Operação normal WiFi desligado	39,74mA	40mA
	Modo Sleep WiFi desligado	1,44mA	2,1mA
Acelerómetro	Modo Sleep	42uA	64uA
	Modo Ativo	133uA	206uA

Tabela 6.2 - Comparação dos consumos anunciados¹⁸ e medidos *Flyport* Wi-Fi

Na Tabela 6.2 verifica-se que os consumos medidos estão de acordo com os anunciados, existindo uma pequena variação no acelerómetro devido à sua alimentação ser efetuada com uma tensão mais elevada do que a apresentada no *datasheet*.

Com base nestes cálculos, foi possível realizar pequenas comparações entre as duas *tags*, desde logo o porquê da *tag* baseada no *Electric imp* apresentar uma autonomia superior à baseada no *Flyport* Wi-Fi.

6.2. Autonomia e Fiabilidade

Ambas as *tags* apresentam a impossibilidade de conexão à rede Wi-Fi *eduroam*, devido ao facto desta rede necessitar de autenticação. Perante este facto, os testes de autonomia e fiabilidade tiveram de ser realizados num local em que houvesse uma rede Wi-Fi disponível e onde alguém passasse bastante tempo, com o objetivo de simular o dia-a-dia de uma pessoa no seu local de trabalho ou em sua própria casa. Desta forma, o cenário de testes foi a minha

¹⁸ Consumos verificados nos respetivos *datasheets*.

própria casa, onde para além do meu *router* foram acrescentados outros dois com o objetivo único de ter vários APs disponíveis para serem descobertos pelas *tags*.

Perante este cenário, os testes foram realizados mediante a utilização normal de uma *tag*, tendo em conta o acompanhamento de uma pessoa. De realçar que, durante os dias úteis o grau de utilização das *tags* foi relativamente reduzido dado o facto da disponibilidade de estar em casa grande parte do dia ser reduzida. Inicialmente, verificou-se alguns problemas na monitorização da movimentação por parte da *tag*, baseada no *Electric imp*. Após um estudo sobre a situação verificou-se que o problema se encontrava na configuração do *threshold* da atividade, uma vez que este se encontrava configurado com 2g enquanto a configuração do *threshold* inatividade era de 0,5g. Este facto resultou num fosso entre as duas configurações, o que fazia com que, caso o acelerómetro detetasse uma aceleração superior a 0,5g mas inferior a 2g, este não conseguisse detetar movimento nem inatividade, ficando no modo ativo até que detetasse uma aceleração inferior a 0,5g durante 2 segundos consecutivos.

Este problema ficou resolvido com a configuração do *threshold* atividade para 0,5g. Este foi o único problema detetado na *tag* baseada no *Electric imp*, revelando-se uma *tag* com excelente fiabilidade e bons consumos.

Quanto à *tag* baseada no *Openpicus Flyport* Wi-Fi, foram detetados mais problemas em relação à *tag* baseada no *Electric imp*. Desde logo, verificou-se que o acelerómetro não é tão eficaz como o do *Electric imp*, mesmo em termos de configurações, uma vez que é mais limitado não deixando o programador tomar decisões que seriam importantes, tais como definir o *threshold* de atividade e de inatividade.

Inicialmente não se revelou muito fiável, pois verificaram-se vários problemas com os *sockets*. O problema mais relevante, ocorria quando chegava ao momento do *socket* estabelecer a conexão com o servidor, este não conseguia e retornava *timeout*. A verificação deste problema revelou-se complexa, pois era um problema intermitente, tanto era retornado *timeout* como de seguida conseguia estabelecer conexão. A resolução passou pela reestruturação do código e respetiva limpeza onde foi tudo visto ao pormenor e devido às interrupções é necessário garantir que a *flag* dos *sockets* se esteja inicializada a *false* em todas as instruções percorridas pelo código.

Quanto à autonomia, verificou-se que a limitação de escolha de uma das opções de configuração do alarme fornecidas pela API afetava a autonomia. Desde o início, o tempo definido para o *timer* era de 30 minutos, mas como não existia esta opção nos alarmes do *Flyport* Wi-Fi, foi implementado o alarme de 10 minutos. Este alarme de 10 minutos, em conjunto com um contador, proporcionava a possibilidade da *tag* ficar no modo *sleep* durante os 30 minutos desejados, no entanto esta teria de acordar 3 vezes ao invés de uma, o que não aconteceria no caso de o alarme disponibilizar a opção de 30 minutos. Este facto proporcionava um gasto desnecessário de energia, uma vez que a *tag* acordava duas vezes e apenas incrementava o contador. Desta forma optou-se por alterar o alarme de modo a este acordar a *tag* de 1 em 1 hora.

Quanto à localização, não foram realizados testes para determinar a localização com base nos dados recolhidos pelas *tags*. No entanto, verificou-se que quando as *tags* se encontravam num determinado local paradas, os valores correspondentes ao RSSI recolhidos pelas *tags* não apresentam uma grande variação no tempo, podendo deste modo afirmar que as *tags* apresentam uma boa fiabilidade no que corresponde ao valor do RSSI recolhido.

6.2.1. Determinação da Autonomia

Para a determinação da autonomia foram implementadas duas *scripts*, uma em linguagem C que converte os ficheiros txt, compostos pelos dados enviados pelas *tags* que se encontram guardados no servidor, para o formato CSV (*Comma-Separated Values*) e outra em linguagem VBA (*Visual Basic for Applications*) para realizar os cálculos necessários para determinar a autonomia.

Após a conversão dos ficheiros txt para CSV e a cópia de todos eles para apenas um ficheiro, este foi importado para uma folha de Excel. Nesta, implementou-se uma macro com o código VBA para determinar o tempo em que a *tag* esteve no modo ativo e no modo *sleep*, utilizando-se como base, os tempos extraídos das imagens apresentadas na secção 6.1.

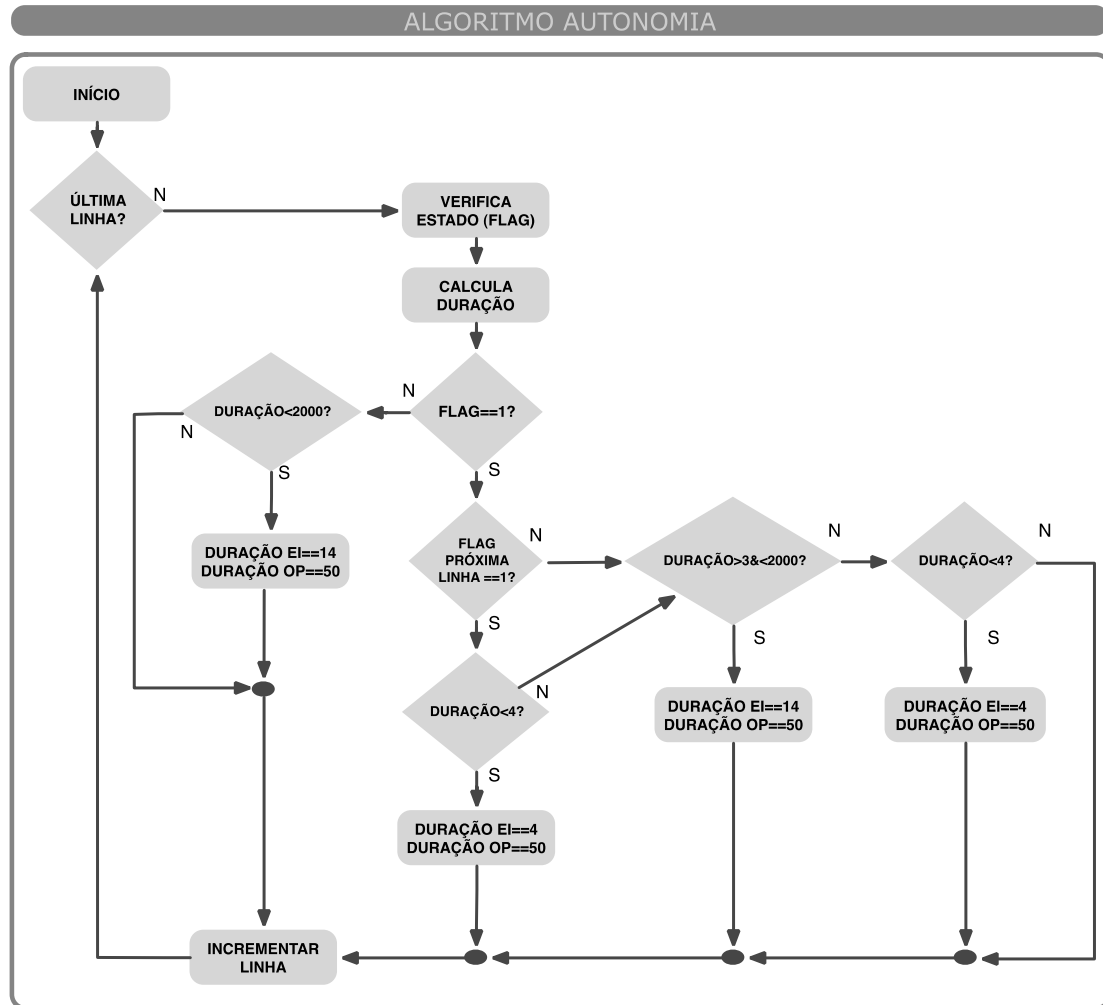


Figura 6.10 - Algoritmo autonomia

- Electric imp

Como referido anteriormente, a *tag* baseada no *Electric imp* apresenta um comportamento e consumos bastante satisfatórios. Após finalizar o teste à *tag* baseada no *Electric imp*, foi determinada a autonomia em número de horas e em percentagem de utilização.

Os resultados obtidos nos dois testes efetuados em relação à autonomia estão representados na Figura 6.11 e em relação à percentagem de funcionamento na Figura 6.12. Tendo em conta que a bateria utilizada pela *tag* é de 1500mAh, como mostra a Figura 6.11 esta esteve em funcionamento durante 550,57 horas, das quais 19,72 foram no modo ativo e 530,85 no modo *sleep*.

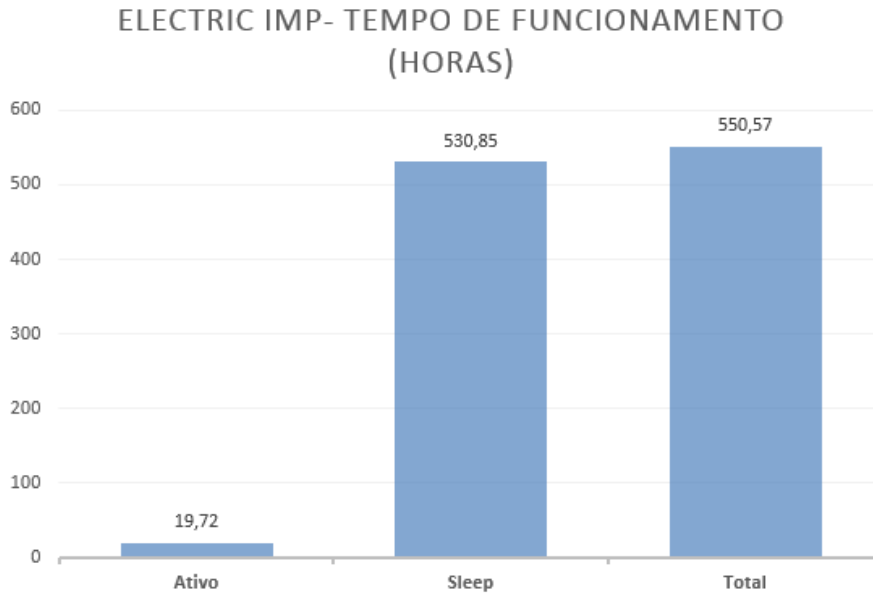


Figura 6.11 - Autonomia Electric imp

Baseando-se na autonomia, é possível determinar a percentagem de funcionamento da *tag* em ambos os modos. Como se pode verificar na Figura 6.12, determinou-se que a *tag* esteve no modo ativo 4% e no modo *sleep* 96% da atividade total.

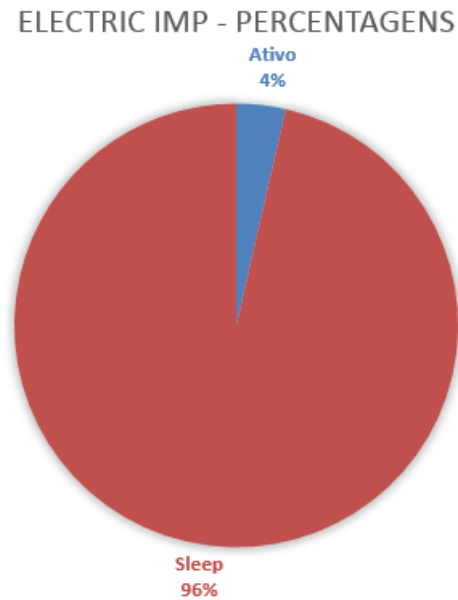


Figura 6.12 - Percentagem Electric imp

- Openpicus Flyport Wi-Fi

Na secção 6.1 foram apresentadas algumas figuras que demonstravam os consumos de ambas as *tags*. No caso do *Flyport* Wi-Fi, foi possível verificar que o seu tempo de execução é demasiado, afetando a sua autonomia. Desta forma, já era esperado que a sua autonomia fosse inferior à da *tag* baseada no *Electric imp*.

Para o cálculo da sua autonomia, foram realizados três testes. Os resultados apresentados nas Figura 6.13 e Figura 6.14 representam a média dos testes. Tendo em conta que a bateria utilizada pela *tag* é de 1060mAh, como é possível verificar na Figura 6.13 verifica-se que a *tag* esteve em funcionamento durante 123,08 horas, das quais 3,57 foram no modo ativo e 119,52 no modo *sleep*. Com uma bateria de 1500mAh estima-se que a autonomia do *Flyport* Wi-Fi aumentasse para cerca de 174,17 horas.

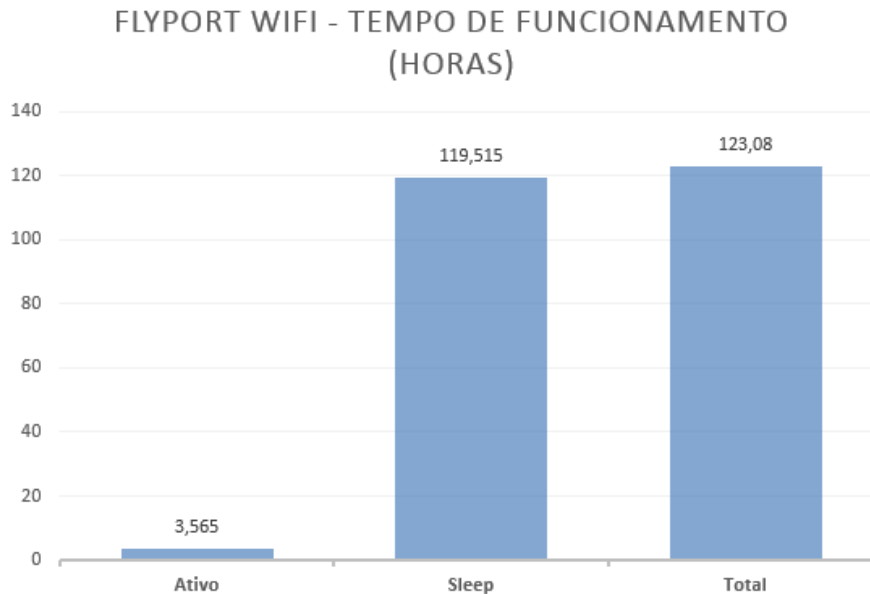


Figura 6.13 - Autonomia Flyport Wi-Fi

Com base nos resultados obtidos, é possível determinar a percentagem de funcionamento da *tag* em ambos os modos. Como se pode verificar na Figura 6.14, determinou-se que a *tag* esteve no modo ativo 3% e no modo *sleep* 97% da atividade total.

FLYPORT WIFI- PERCENTAGENS

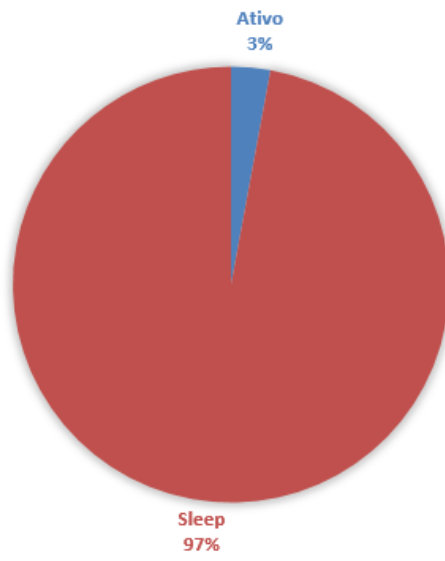


Figura 6.14 - Percentagem Flyport Wi-Fi

7. Conclusões e Trabalho Futuro

Neste capítulo serão apresentadas as conclusões deste trabalho, onde são especificadas as conclusões no processo de recolha de dados e na poupança de energia. Na segunda secção será realizada uma análise dos resultados e na última secção serão apresentados quais os passos futuros para a evolução deste trabalho.

7.1. Conclusões

Fazendo uma comparação direta entre as duas *tags* desenvolvidas, é possível concluir alguns factos. Após todos os testes e medições, é possível concluir que a *tag* baseada no *Electric imp* é a que se adequa melhor ao objetivo proposto nesta dissertação. Fatores como a fiabilidade e autonomia fazem com que esta *tag* seja a melhor escolha.

Quanto ao funcionamento de ambas as *tags*, o *Flyport* Wi-Fi necessita de cabos para a passagem do código e o processo de implementação é mais trabalhoso. Apesar desta pequena desvantagem, o *Flyport* Wi-Fi apresenta a grande vantagem de não depender de servidores externos tal como o *Electric imp* depende do *imp Cloud*.

7.1.1. Recolha de Dados e Poupança de Energia

Como discutido ao longo deste documento, os dispositivos que têm como objetivo a recolha de dados apresentam um forte impacto na sua própria autonomia, sendo por isso fundamental desenvolver uma solução que minimize os seus consumos energéticos.

Sabendo que a autonomia está diretamente relacionada com a capacidade da bateria, consumo da interface Wi-Fi e nível de utilização do processador, chegou-se à conclusão que uma estratégia de utilização de um acelerómetro e realização de amostragens periódicas através de *timers*, seria benéfica para reduzir o tempo de atividade do processador e interface Wi-Fi. Nas

medições efetuadas e apresentadas na secção 6.1, confirmou-se que realizar amostragens contínuas seria impensável, visto que o gasto energético quando as *tags* se encontram ativas é significativo.

A utilização do acelerómetro, possibilitou otimizar a recolha de dados tendo em conta o comportamento da *tag*. O algoritmo desenvolvido permite reduzir os consumos nos períodos em que a *tag* se encontra estática, visto que este deteta quando a *tag* se encontra imóvel e coloca-a num estado de baixo consumo (modo *sleep*). Desta forma, pode-se concluir que a autonomia da *tag* depende do nível de movimentação a que esta está sujeita. O algoritmo implementado, tem em consideração que, caso se detetasse toda a movimentação da *tag*, a eficiência energética é afetada. Devido a este facto, potenciou-se a utilização do *buffer* do acelerómetro, efetuando-se leituras de 2 em 2 segundos e colocando a *tag* num modo *sleep* com conectividade Wi-Fi entre as mesmas.

7.2. Análise dos Resultados

O número de testes realizados poderia ser maior mas devido à limitação temporal imposta pela data de entrega da dissertação não foi possível realizar mais testes. No entanto, os resultados apresentados na subsecção 6.2.1, permitem que se chegue a uma conclusão quanto à autonomia das *tags*.

Comparando diretamente as duas *tags*, não há dúvidas que o *Electric imp* apresenta-se como uma *tag* mais fiável e com uma autonomia muito superior à do *Flyport* Wi-Fi. Este facto foi descrito na secção 6.2, onde foi realizada uma análise à autonomia e fiabilidade das *tags*. Na secção 6.1, foram apresentadas duas tabelas onde se pode comparar os consumos anunciados pelos fabricantes com os consumos medidos nas *tags*. A conclusão a que se chega é que, o *Flyport* Wi-Fi revela consumos em linha dos apresentados pela *Openpicus* e o *Electric imp* apresenta um desfasamento no momento em que se encontra no modo *sleep*. Este desfasamento deve-se ao facto do *Electric imp* ter de alimentar o acelerómetro.

As razões do *Flyport* Wi-Fi apresentar uma autonomia bastante inferior ao *Electric imp*, estão diretamente relacionadas com o elevado consumo no modo *sleep*, mesmo com o Wi-Fi desligado, e com o tempo de execução demasiado grande que esta apresenta.

7.3. Trabalho Futuro

O desenvolvimento das duas *tags* apresentadas neste documento teve em conta apenas as funcionalidades mais importantes. Neste caso, o objetivo foi desenvolver duas *tags* que possibilitassem a sua localização com uma boa autonomia, deste modo, características que não tivessem relação direta com este objetivo não foram equacionadas.

Desta forma, existe a hipótese de evoluir a composição das *tags* com um circuito para verificação do estado da bateria e forma de transmissão dessa informação ao utilizador alertando-o por exemplo, de bateria fraca. Para a implementação desta ideia é necessário ter em consideração de que, esta inovação fará com que o consumo aumente, e num dispositivo deste género, qualquer que seja o hardware adicionado, poderá ter um grande impacto na sua autonomia. Para transmitir a informação do estado da bateria ao utilizador, pode ser adicionado e.g. um LED, vibrador de um *smartphone* ou até um *speaker* de um computador portátil. Com uma estratégia bem definida, esta implementação poder-se-á tornar numa grande mais valia para quem utiliza a *tag*, pois neste momento existe a necessidade de verificar no servidor se a *tag* continua a enviar dados, para confirmar o estado da bateria. Em alternativa, o nível de bateria poderá ser transmitido para o servidor, como um campo adicional das mensagens de posicionamento, permitindo controlar num ponto central todas as *tags*. Desta forma pode efetuar-se manutenção preventiva e substituir ou recarregar a bateria antes desta terminar e pode controlar facilmente as *tags* que não são usadas por pessoas e que poderiam ficar inoperacionais por falta de bateria.

Para além desta evolução, poderá também ser pensada a aplicação de algum acessório, e.g. um botão com o objetivo de enviar pedidos ao servidor e com o objetivo de que a *tag* fique melhor equipada, com outras características que permitam uma comunicação personalizada com o servidor, que não seja apenas a localização. Para melhorar a autonomia, pode ser implementada uma forma de *energy harvesting*. Para a implementação de um sistema destes existem várias possibilidades, a energia pode ser obtida através do efeito piezoelétrico ou até mesmo através do aproveitamento da energia cinética, e.g. os relógios kinetic¹⁹ da Seiko.

¹⁹ <http://www.seikowatches.com/world/technology/kinetic/index.html>

8. Bibliografia

- Bahl, P., & Padmanabhan, V. N. (2000). RADAR: an in-building RF-based user location and tracking system. *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, 2. doi:10.1109/INFCOM.2000.832252
- Chon, J. (2011). LifeMap: A Smartphone-Based Context Provider for Location-Based Services. *IEEE Pervasive Computing*, 10(2), 58–67. doi:10.1109/MPRV.2011.13
- Chon, Y., Talipov, E., Shin, H., & Cha, H. (2011). Mobility prediction-based smartphone energy optimization for everyday location monitoring. *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems - SenSys '11*, 82. doi:10.1145/2070942.2070952
- Eisa, S., & Peixoto, J. (2013). Removing useless APs and fingerprints from WiFi indoor positioning radio maps. *2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, (28-31 October), 1–7. doi:10.1109/IPIN.2013.6817919
- Liu, H., & Darabi, H. (2007). Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(6), 1067–1080. doi:10.1109/TSMCC.2007.905750
- Mok, E., & Retscher, G. (2007). Location determination using WiFi fingerprinting versus WiFi trilateration. *Journal of Location Based Services*, 1(2), 145–159. doi:10.1080/17489720701781905
- Retscher, G., Moser, E., Vredevelde, D., & Heberling, D. (2006). Performance and Accuracy Test of the WLAN Indoor Positioning System “ ipos ” Principle of WLAN Positioning. *Presented at the 3rd Workshop on Positioning, Navigation and Communication WPNC 2006, University of Hannover, Germany*, 7–16.
- Russell, S. (2003). Detecting and Locating Rogue Access Points. Retrieved from <http://www.ee.iastate.edu/~russell/cpre537.s04/Report-Example.pdf>
- Yamasaki, R., Ogino, A., & Tamaki, T. (2005). TDOA location system for IEEE 802.11 b WLAN. *Wireless Communications and Networking Conference , 2005 IEEE*, 4, 2338–2343. doi:10.1109/WCNC.2005.1424880
- Zegelin, C. (2003). Real time location system. *US Patent App. 10/648,712*, 1–8.
- Aeroscout*. (Outubro de 2013). Obtido de <http://aeroscout.com/>
- Centrak*. (Outubro de 2013). Obtido de <http://www.centrak.com/>

Ekahau. (Outubro de 2013). Obtido de <http://www.ekahau.com>

Electric imp. (Setembro de 2014). Obtido de <http://www.electricimp.com>

Navizon. (Outubro de 2013). Obtido de <http://www.navizon.com/>

RedPineSignals. (Outubro de 2013). Obtido de <http://www.redpinesignals.com/>

Squirrel. (Setembro de 2014). Obtido de <http://www.squirrel-lang.org>

Vestac. (Outubro de 2013). Obtido de <http://www.vestac.com/>

VizziaTech. (Outubro de 2013). Obtido de <http://vizziatech.com/>

A. Anexo: Script para o Cálculo da Autonomia

Sub autonomia()

Dim secondDate As Date

Dim startDate As Date

Dim dtmStart As Date

Dim dtmEnd As Date

Dim dblDuration As Double

Dim resultado As Integer

Dim X As Long

Dim data_aux As Integer

Dim aux_dia As Integer

Dim data_inicio As Date

Dim data_fim As Date

Dim data_inicio_aux As Date

Dim contador As Integer

Dim teste As Integer

Dim min As Long

Dim max As Long

Dim duracao As Long

Dim aux As Integer

Dim y As Integer

y = 3

Dim flag As Integer

Dim flag1 As Integer

Dim aux1 As Integer

Dim val As Integer

For X = 1 To 279

```
flag = Range("D" & X).Value
```

```
If (flag = 1) Then
```

```
val = val + 1
```

```
    If (val = 1) Then
```

```
        aux1 = X
```

```
    End If
```

```
flag1 = Range("D" & y).Value
```

```
dtmStart = Format(CDate(Range("A" & X).Value), "yyyy-mm-dd hh:mm:ss")
```

```
dtmEnd = Format(CDate(Range("A" & aux1).Value), "yyyy-mm-dd hh:mm:ss")
```

```
Range("AY" & X).Value = dtmStart
```

```
Range("AZ" & X).Value = dtmEnd
```

```
Range("Bc" & X).Value = flag1
```

```
Range("Bd" & X).Value = X
```

```
Range("Be" & X).Value = y
```

```
duracao = DateDiff("s", dtmEnd, dtmStart)
```

```
Range("BB" & X).Value = duracao
```

```
y = y + 1
```

```
If (flag1 = 1) Then
```

```
    If (duracao < 4) Then
```

```
        Range("AV" & X).Value = 4
```

```
    Elseif (duracao > 3 And duracao < 2000) Then
```

```
        Range("AW" & X).Value = duracao
```

```
        Range("AV" & X).Value = 14
```

```
    End If
```



```
Elseif (flag1 = 0) Then
    If (duracao > 3 And duracao < 2000) Then
        Range("AW" & X).Value = duracao
        Range("AV" & X).Value = 14
    Elseif (duracao < 4) Then
        Range("AV" & X).Value = 4
    End If
End If

Elseif (flag = 0) Then
    val = val + 1

    If (val = 1) Then
        aux1 = X
    End If

    flag1 = Range("D" & y).Value

    dtmStart = Format(CDate(Range("A" & X).Value), "yyyy-mm-dd hh:mm:ss")
    dtmEnd = Format(CDate(Range("A" & aux1).Value), "yyyy-mm-dd hh:mm:ss")

    Range("AY" & X).Value = dtmStart
    Range("AZ" & X).Value = dtmEnd

    duracao = DateDiff("s", dtmEnd, dtmStart)
    y = y + 1

    If (duracao < 2100) Then
        Range("AW" & X).Value = duracao
        Range("AV" & X).Value = 14
    End If
End If

aux1 = X

Next
End Sub
```