

# SISTEMA DE LOCALIZAÇÃO *INDOOR* COM BASE NO PROTOCOLO *ZIGBEE*

Luís Miguel Rodrigues Couto



Mestrado em Engenharia Electrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

2013



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

Candidato: Luís Miguel Rodrigues Couto, N° 1060843, 1060843@isep.ipp.pt

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

Co-orientação científica: Eng. António Meireles, ajmm@isep.ipp.pt



Mestrado em Engenharia Electrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

20 de novembro de 2013



À minha Namorada, aos meus Pais e ao meu Irmão...



## *Agradecimentos*

Foram várias as pessoas, que em maior ou menor medida, contribuíram de alguma maneira para a realização deste projecto, e que eu apenas posso expressar os meus mais sinceros agradecimentos.

Em primeiro lugar gostaria de expressar um agradecimento aos meus orientadores, Lino Figueiredo e António Meireles pela oportunidade de participar neste projecto, pela forma como orientaram o meu trabalho e pela dedicação e contributo para que este seja um êxito.

Deixo também uma palavra de apreço a todos os meus amigos, que passaram muitas horas ao meu lado na “316”, pelas palavras de apoio, sugestões e incentivos.

Não posso também esquecer de deixar uma palavra de agradecimento a D.<sup>a</sup> Lurdes, a Joana e ao Luís pelo apoio e preocupação.

Quero expressar o meu mais sincero e profundo agradecimento aos meus pais, Maria Augusta Couto e Luís Couto por todo o apoio, preocupação e por todo o esforço que tiveram para que eu pudesse terminar a minha formação académica. Obrigado por todos os valores que me passaram e pela forma como me inculcaram alegria de viver.

Não posso deixar de agradecer ao meu irmão, Joel Couto, pelas suas palavras de força e por sempre acreditar em mim.

Para o final, guardei o agradecimento mais importante, para pessoa mais especial da minha vida; a Tânia, por todo o apoio que sempre me deu, por estar sempre a meu lado nos bons e maus momentos, por ter demonstrado uma fé inabalável e por acima de tudo ter sido a minha força.

Muito Obrigado a todos!



## *Resumo*

Actualmente, os sistemas de localização são uma área em forte expansão sendo que para espaços exteriores existe uma grande variedade de sistemas de localização enquanto que para espaços interiores as soluções são mais escassas.

Este trabalho apresenta o estudo e implementação de um sistema de localização *indoor* baseado no protocolo *ZigBee*, utilizando a informação da intensidade de sinal recebido (RSSI - *Received Signal Strength Indication*). Para a realização deste projecto foi necessário iniciar uma pesquisa mais pormenorizada do protocolo *ZigBee*. O dispositivo móvel a ser localizado é o módulo *XBee Serie 2* que se baseia no mesmo protocolo. Posto isto, foi necessário efectuar um estudo sobre sistemas de localização existentes e analisar as técnicas de localização utilizadas para ambientes interiores. Desta forma utiliza-se neste projecto uma técnica que consiste na análise de *fingerprinting*, onde é criado um mapa com os valores RSSI para diferentes coordenadas do espaço físico. As intensidades de sinal recebido são relativas a dispositivos *XBee* instalados em pontos fixos de referência. Para calcular a localização do dispositivo móvel é utilizado o algoritmo K-NN (K- *Nearest Neighbors*) que permite estimar a posição aproximada do dispositivo móvel.

Por último é descrito todo o desenvolvimento do projecto assim como a apresentação e discussão de resultados.

### ***Palavras-Chave***

Sistema de localização, *Indoor*, *ZigBee*, *Fingerprinting*, RSSI.



## *Abstract*

Nowadays, the localization systems are a growing technology in the market. For outdoor environments there is a variety of localizations systems, however for indoor environments the solutions are not so abundant.

This project presents the study and implementation of an indoor localization system based on ZigBee protocol, using information about signal strength. To realize it, it was necessary begin a detailed research about ZigBee protocol. The mobile device to be localized is XBee Serie 2, it is based on ZigBee. After that, was studied the localization systems on the market and analyse the techniques of localization for indoor environments.

This project uses a technique based on fingerprinting analysis, where it is created a map with RSSI values to different space coordinates. The received signal intensity is related to XBee devices installed in fixed reference points. To calculate the device mobile's localization it was used the K-NN algorithm.

To conclude it was described all development steps of the as well as the presentation and discussion of the results.

### ***Keywords***

Localization System, Indoor, ZigBee, Fingerprinting, RSSI.



## *Résumé*

Actuellement, les systèmes de localisation sont un domaine en fort expansion, pour les espaces extérieur il existe une grande variété de système pendant que pour les espaces intérieur les solutions sont plus rare.

Ce travail présente une étude et l'implémentation de un système de localisation *indoor* basée sur le protocole *ZigBee*, utilisant l'information d'intensité du signal reçu (RSSI). Pour la réalisation de ce projet il y a la nécessité de faire une recherche sur le protocole *ZigBee*. Le dispositif mobile utiliser pour faire la localisation est le *XBee Serie 2*, basée sur le même protocole. Après ce là, il faut réaliser une étude sur les systèmes de localisation existant et analyser les technique de localisation utilisé pour les espaces intérieur. Ainsi on utilise en ce projet une technique qui consiste en faire l'analyse de *fingerprinting*, leur on doit créer un mappe avec les valeurs RSSI pour les différent coordonnées de l'espace physique. Les intensités de signal reçus sont relatives au dispositif *XBee* installé dans des points de référence. Pour calculer la localisation du dispositif mobile on utilise l'algorithme K-NN qui permet estimer une position approximé du dispositif.

Pour finir on fait la description du développement du projet ainsi come la présentation e discussions des résultats.

### ***Mots-clés***

Système de localisation, *Indoor*, *ZigBee*, *Fingerprinting*, RSSI.



# Índice

<b>AGRADECIMENTOS</b> .....	<b>I</b>
<b>RESUMO</b> .....	<b>III</b>
<b>ABSTRACT</b> .....	<b>V</b>
<b>RESUME</b> .....	<b>VII</b>
<b>ÍNDICE</b> .....	<b>IX</b>
<b>ÍNDICE DE FIGURAS</b> .....	<b>XI</b>
<b>ÍNDICE DE TABELAS</b> .....	<b>XIII</b>
<b>ACRÓNIMOS</b> .....	<b>XV</b>
<b>1. INTRODUÇÃO</b> .....	<b>1</b>
1.1. CONTEXTUALIZAÇÃO .....	1
1.2. OBJECTIVOS .....	2
1.3. ORGANIZAÇÃO DO RELATÓRIO .....	3
<b>2. ESTADO DA ARTE</b> .....	<b>5</b>
2.1. CARACTERÍSTICAS E FACTORES DETERMINANTES NAS REDES SEM FIOS .....	8
2.2. APLICAÇÕES DAS REDES DE SENSORES SEM FIOS .....	9
2.3. TÉCNICAS DE LOCALIZAÇÃO .....	12
2.4. SISTEMAS DE LOCALIZAÇÃO .....	24
<b>3. ZIGBEE</b> .....	<b>33</b>
3.1. TIPOS DE DISPOSITIVOS .....	34
3.2. TOPOLOGIAS .....	35
3.3. MODOS DE OPERAÇÃO DA REDE .....	36
3.4. TRANSMISSÃO DE DADOS .....	40
3.5. ENDEREÇAMENTO .....	41
3.6. SERVIÇOS DE SEGURANÇA .....	42
3.7. CAMADAS DO PROTOCOLO <i>ZIGBEE</i> .....	43
3.8. DISPOSITIVOS <i>XBEE</i> .....	46
<b>4. DESENVOLVIMENTO DO PROJECTO</b> .....	<b>53</b>
4.1. ARQUITECTURA DO SISTEMA .....	53
4.2. <i>HARDWARE</i> UTILIZADO PARA DESENVOLVIMENTO DO PROJECTO .....	55
4.3. <i>SOFTWARE</i> UTILIZADO PARA DESENVOLVIMENTO DO PROJECTO .....	60
4.4. FUNCIONAMENTO DO SISTEMA DESENVOLVIDO .....	61
4.5. CONFIGURAÇÃO DA REDE .....	63

4.6.	TRAMAS PRINCIPAIS DA REDE.....	65
4.7.	SOFTWARE DESENVOLVIDO PARA O SISTEMA DE LOCALIZAÇÃO. ....	68
4.8.	FLUXOGRAMAS DO SISTEMA DESENVOLVIDO.....	69
<b>5.</b>	<b>TESTES E RESULTADOS.....</b>	<b>73</b>
5.1.	VARIAÇÃO RSSI COM A DISTÂNCIA NUM ESPAÇO EXTERIOR.....	74
5.2.	ANÁLISE RSSI PARA DIFERENTES ORIENTAÇÕES E DISTÂNCIAS EM ESPAÇO INTERIOR.....	75
5.3.	ANÁLISE RSSI PARA CENÁRIO DE TESTES.....	80
<b>6.</b>	<b>CONCLUSÕES.....</b>	<b>87</b>
	<b>REFERÊNCIAS DOCUMENTAIS.....</b>	<b>91</b>
	<b>ANEXO A. ESPECIFICAÇÕES MÓDULO XBEE SERIES 2.....</b>	<b>95</b>
	<b>ANEXO B. MAPA VALORES RSSI PARA DIFERENTES DISTÂNCIAS E DIFERENTES</b> <b>ÂNGULOS. ....</b>	<b>97</b>
	<b>ANEXO C. CÓDIGO C# DA APLICAÇÃO DESENVOLVIDA.....</b>	<b>99</b>
	<b>ANEXO D. CÓDIGO C DE FUNCIONAMENTO DO ROUTER MÓVEL.....</b>	<b>109</b>

## Índice de Figuras

Figura 1	Componentes de um nó sensor [1].....	6
Figura 2	Técnica de tempo de chegada [2].....	13
Figura 3	Técnica AoA com orientação conhecida (a) e sem orientação conhecida (b) [2].....	14
Figura 4	Gráfico da relação RSSI com distância entre transmissor e emissor [33].....	18
Figura 5	Exemplo de um mapa de <i>fingerprints</i> [2].....	20
Figura 6	Fases da técnica de localização por <i>fingerprinting</i> [33].....	22
Figura 7	Sensor Active Badge [1].....	25
Figura 8	Sistema <i>Active Bat</i> [10].....	26
Figura 9	Rede de Sensores do sistema <i>Ubisense</i> [10].....	28
Figura 10	Sistema <i>LandMarc</i> [2].....	29
Figura 11	Sistema de localização baseado no <i>Location Engine</i> [2].....	30
Figura 12	Sistema implementado na Master's room, Mid Sweden University [18].....	31
Figura 13	Topologias da rede <i>ZigBee</i> [1].....	35
Figura 14	Estrutura <i>Superframe</i> [11].....	37
Figura 15	Transmissão directa de dados em redes <i>beacon</i> habilitado [11].....	38
Figura 16	Transmissão directa de dados em redes com <i>beacon</i> não habilitado [11].....	38
Figura 17	Transmissão indirecta de dados em redes com <i>beacon</i> habilitado [11].....	39
Figura 18	Transmissão indirecta de dados numa rede com <i>beacon</i> não habilitado [11].....	39
Figura 19	(a) <i>Broadcast</i> , (b) <i>Multicast</i> , (c) <i>Unicast</i> [5].....	41
Figura 20	Pilha protocolar <i>ZigBee</i> [8].....	43
Figura 21	A camada de rede <i>ZigBee</i> e a sua interacção com a MAC e APL [5].....	45
Figura 22	Camada de aplicação [5].....	45
Figura 23	Principais modelos de Módulo <i>XBee</i> da <i>Digi International</i> [46].....	47
Figura 24	Arquitectura dos módulos <i>XBee</i> [16].....	48
Figura 25	Estrutura das mensagens em modo API [47].....	50
Figura 26	Exemplo de uma estrutura de trama API de transmissão [47].....	50
Figura 27	Radiação de Antenas de diferentes módulos <i>XBee</i> . a)Antena Dipolo b) <i>Chip</i> Antena c) <i>Whip</i> Antena [48].....	51
Figura 28	Arquitectura do sistema.....	54
Figura 29	Dispositivo <i>XBee</i> Series 2 com antena <i>Whip</i> .....	55
Figura 30	Diagrama de Pinos do Módulo <i>XBee</i> Series 2 [50].....	57
Figura 31	Diagrama de fluxos de dados na interface UART [50].....	58
Figura 32	<i>Arduino</i> Uno.....	58
Figura 33	Ligações entre <i>Arduino</i> e <i>ZigBee</i> [4].....	59

Figura 34	Esquema de montagem do regulador LM117 .....	60
Figura 35	Software X-CTU para configuração dos dispositivos <i>XBee</i> .....	60
Figura 36	Funcionamento do sistema desenvolvido.....	63
Figura 37	Janela de operações de configuração .....	64
Figura 38	Estrutura das várias tramas API.....	65
Figura 39	Interface Gráfica da fase de criação do mapa .....	68
Figura 40	Interface Gráfica da fase de localização do sistema desenvolvido .....	69
Figura 41	Fluxograma da fase de criação do mapa de <i>fingerprinting (offline)</i> .....	70
Figura 42	Fluxograma da fase <i>online</i> do sistema de localização .....	71
Figura 43	Fluxograma das acções do <i>router</i> de referência.....	72
Figura 44	Gráfico RSSI em relação à distância para ambiente exterior - Teste 1 .....	74
Figura 45	Gráfico RSSI em relação à distância para ambiente exterior - Teste 2 .....	74
Figura 46	Gráfico RSSI em relação à distância para ambiente exterior - Teste 3 .....	75
Figura 47	Exemplo do teste interior para diferentes orientações e distâncias.....	76
Figura 48	Variação dos valores RSSI para a mesma distância (0,1 m) variando o ângulo em relação à referência.....	76
Figura 49	Variação dos valores RSSI para a mesma distância (0,5 m) variando o ângulo em relação à referência.....	77
Figura 50	Variação dos valores RSSI para a mesma distância (1 m) variando o ângulo em relação à referência.....	77
Figura 51	Variação dos valores RSSI para a mesma distância (2 m) variando o ângulo em relação à referência.....	77
Figura 52	Variação dos valores de RSSI para o mesmo ângulo (0°) para diferentes distâncias ...	78
Figura 53	Variação dos valores de RSSI para o mesmo ângulo (40°) para diferentes distâncias .	79
Figura 54	Variação dos valores de RSSI para o mesmo ângulo (60°) para diferentes distâncias .	79
Figura 55	Esquemático da plataforma de teste.....	80
Figura 56	Representação cromatográfica do mapa RSSI em relação ao Nó 1 de referência a) Momento 1 b) Momento 2 c) Momento 3 .....	81
Figura 57	Representação cromatográfica do mapa RSSI em relação ao Nó 2 de referência a) Momento 1 b) Momento 2 c) Momento 3 .....	82
Figura 58	Comparação dos mapas RSSI. a) Valores RSSI obtidos com Nó 1 colocado na posição do Nó 2. b) Valores RSSI obtidos com Nó 2 na posição original .....	82
Figura 59	Comparação dos mapas RSSI. a) Valores RSSI obtidos com Nó 2 colocado na posição do Nó 1. b) Valores RSSI obtidos com Nó 1 na posição original .....	83
Figura 60	Diferentes posições analisadas.....	84

## *Índice de Tabelas*

Tabela 1	Comparação de diferentes tecnologias para redes sem fios [8].....	7
Tabela 2	Especificações do módulo <i>XBee Series 2</i> .....	55
Tabela 3	Funcionalidade dos pinos do módulo <i>XBee Series 2</i> [50] .....	56
Tabela 4	Configurações efectuadas nos três nós da rede .....	64
Tabela 5	Estrutura da mensagem <i>Remote command Request</i> .....	66
Tabela 6	Estrutura da mensagem <i>Remote Command Response</i> .....	66
Tabela 7	Estrutura da mensagem <i>ZigBee Transmit Request</i> .....	67
Tabela 8	Valores de RSSI para diferentes orientações de antena .....	84
Tabela 9	Valores médios RSSI para funcionamento dos nós de referência individualmente e em conjunto.....	85



## *Acrónimos*

AES	-	<i>Advanced Encryption Standard</i>
AoA	-	<i>Angle of Arrival</i>
API	-	<i>Application Programming Interface</i>
APS	-	<i>Application Support Sublayer</i>
BPSK	-	<i>Binary Phase Shift Keying</i>
CSMA- CA	-	<i>Carrier sense multiple access with collision avoidance</i>
ED	-	<i>End Device</i>
GECAD		<i>Grupo de Investigação em Engenharia do Conhecimento e Apoio à Decisão</i>
GPS	-	<i>Global Positioning System</i>
GSM	-	<i>Global System for Mobile Communications</i>
ITALH	-	<i>Information Technology For Assisted Living at Home</i>
LQI	-	<i>Link Quality Indicator</i>
MAC	-	<i>Medium Access Control</i>
MIT	-	<i>Massachusetts Institute of Technology</i>
NLDE	-	<i>Network Layer Data Entity</i>
NLME	-	<i>Network Layer Management Entity</i>
NWK	-	<i>Network</i>

O-QPSK	-	<i>Offset Quadrature Phase-Shift Keying</i>
PHY	-	<i>Physical</i>
RF	-	<i>Rádio Frequência</i>
RFID	-	<i>Radio-Frequency Identification</i>
RSS	-	<i>Received Signal Strenght</i>
RSSF	-	<i>Redes Sensores Sem Fios</i>
RSSI	-	<i>Received Signal Strenght Indicator</i>
SAP	-	<i>Service Acces Point</i>
TDoA	-	<i>Time –difference of arrival</i>
ToA	-	<i>Time of arrival</i>
ToF	-	<i>Time of flight</i>
UART	-	<i>Universal Asynchronous Receiver/Transmitter</i>
USB	-	<i>Universal Serial Bus</i>
WSN	-	<i>Wireless Sensor Network</i>
ZDO	-	<i>ZigBee Device Objects</i>





# 1. INTRODUÇÃO

Os sistemas de localização são definidos como um sistema de cálculo da localização. Isto é, este sistema realiza um processo que possibilita obter informação sobre a localização de um objecto/dispositivo móvel num determinado espaço respeitando um conjunto de referências [32]. São baseados em redes de sensores sem fios, sendo que as tecnologias mais utilizadas são: *Bluetooth*, *WiFi* e *ZigBee*.

Os sistemas de localização surgiram inicialmente para suporte a aplicações militares em ambientes exteriores (*outdoor*). O GPS (*Global Positioning System*) foi uma das principais tecnologias que surgiu, pois, permitia obter informação mais precisa de um determinado terreno em situações de guerra. Ao fim de algum tempo deixou de ser utilizado apenas para fins militares podendo também ser utilizado por cidadãos comuns, em aplicações de navegação, localização de veículos, controlo de *stocks* de mercadorias, activação de eventos através da proximidade de pessoas, entre outras.

## 1.1. CONTEXTUALIZAÇÃO

Os sistemas de localização podem ser utilizados em dois tipos de situações: espaços exteriores e espaços interiores.

Um exemplo de localização aplicada em espaços exteriores, como solicitado anteriormente, é o GPS. No entanto, a tecnologia não é eficaz nesse meio devido a instabilidade e à fraca qualidade do sinal recebido. É de salientar, que no nosso dia-a-dia

passamos grande do tempo em espaços fechados, pelo que nos últimos anos têm surgido vários sistemas que possibilitam posicionamento nos mesmos.

Um sistema de localização permite determinar a posição de um dispositivo móvel possibilitando assim serviços como navegação, verificação ou monitorização. As informações de localização de utilizadores ou dispositivos podem melhorar o desempenho de uma rede sem fios, para planeamento da rede, adaptação da rede, balanceamento de carga, entre outras.

Alguns destes sistemas têm sido utilizados para controlo do interior de unidades hospitalares, possibilita aos utentes uma utilização mais eficiente dos recursos médicos melhorando a sua orientação nos complexos hospitalares.

O posicionamento interior pode ser também utilizado para orientação de turistas num museu permitindo uma visualização do que está exposto, no sítio correcto, e na sequência certa.

Sistemas de localização interior têm também aplicação em escritórios e casas inteligentes, em automação e controlo industrial.

O trabalho desenvolvido toma rumo neste sentido; aplicação de um sistema de localização em espaços interiores, sendo mais um contributo para a relevância deste tipo de sistemas.

## **1.2. OBJECTIVOS**

O objectivo deste projecto consiste no desenvolvimento de um sistema suportado por uma rede de comunicação sem fios baseada no protocolo *ZigBee*, que possibilite a localização num ambiente *indoor* de um objecto ou pessoa. O desenvolvimento do projecto foi subdividido em diversas tarefas mais simples, tais como:

- Estudo do Protocolo *ZigBee*:
  - Estudo teórico referente ao protocolo.
  - Estudo sobre os módulos *ZigBee* utilizados desde ligações físicas, modos de funcionamento e métodos de transmissão de dados.

- Análise tecnológica sobre o protocolo para conhecer as suas valências, configurações e tipos de tramas de dados que suporta.
- Estudo das técnicas de localização existentes.
- Estudo e análise do comportamento do RSSI para diferentes ambientes.
- Desenvolvimento completo de uma rede sem fios utilizando o protocolo *ZigBee*.
- Realização de testes para avaliar o comportamento do sistema de localização.

### **1.3. ORGANIZAÇÃO DO RELATÓRIO**

O presente relatório encontra-se estruturado em seis capítulos. No primeiro capítulo, denominado “Introdução” é realizada uma breve introdução ao tema, ao contexto da tese e quais os objectivos propostos para a realização da mesma.

O segundo capítulo, denominado por “Estado da Arte”, pretende apresentar as redes de sensores sem fios, os seus constituintes e principais áreas de aplicação das mesmas. Dentro das áreas de aplicação será dado particular destaque aos sistemas de localização e as técnicas de localização que utilizam.

No terceiro capítulo, denominado “*ZigBee*” é feita uma descrição do protocolo *ZigBee* e do seu funcionamento. É realizado um enquadramento teórico para melhor compreensão do mesmo, abordando as topologias de rede, que tipos de dispositivos podem fazer parte da mesma e como os dispositivos comunicam entre si. São também apresentados os dispositivos XBee baseados em protocolo *ZigBee*.

O capítulo seguinte, capítulo 4, denominado “Desenvolvimento do projecto”, explica a forma como foi desenvolvido o sistema de localização *indoor* e as opções tomadas para a sua criação. Será descrita a arquitectura do sistema, a técnica de localização implementada, a forma como foi configurada a rede e os principais dados que nela circulam.

Segue-se o capítulo 5, denominado “Testes e Resultados”, onde são apresentados os testes realizados ao longo do desenvolvimento do projecto e os consequentes resultados obtidos.

Por último, o capítulo 6, denominado “Conclusões”, apresenta as conclusões tiradas relativamente ao desenvolvimento do trabalho, propondo sugestões para futuros projectos

na perspectiva do melhoramento e aperfeiçoamento do sistema de localização em espaços interiores.

## 2. ESTADO DA ARTE

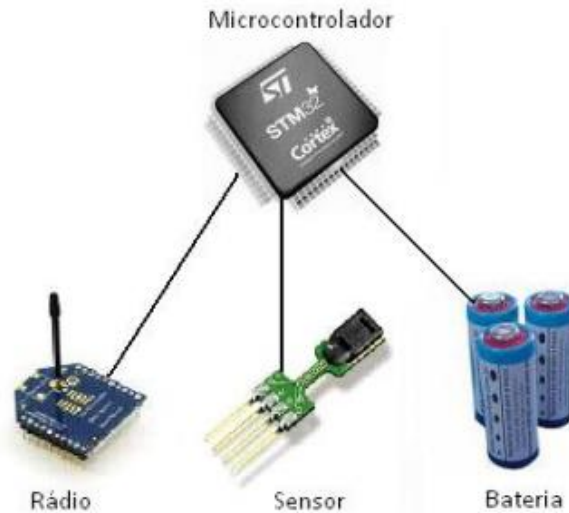
Actualmente existem vários meios de transmissão de informação, que pessoas, empresas ou instituições podem utilizar. Um dos tipos de transmissão mais utilizado é a transmissão sem fios. O telemóvel, que veio eliminar a necessidade de cabos para realizar comunicações telefónicas entre pessoas, é porventura, um dos mais importantes desenvolvimentos nesta área, devido à importância que tem no quotidiano das pessoas; poucas são as que não possuem um equipamento destes!

As potencialidades deste tipo de tecnologia são notórias e por este motivo acabam por se expandir para outras áreas. Desde então foram aparecendo diversos equipamentos, como por exemplo, computadores pessoais, sistemas de posicionamento, *smartphones*, que se inserem num grupo de equipamentos que utilizam tecnologias de comunicação sem fios.

A evolução das tecnologias de comunicações sem fios, aliada à evolução tecnológica dos microcontroladores, dos sensores e dos circuitos integrados, possibilitam a passagem para uma nova área de desenvolvimento, as redes de sensores sem fios (RSSF) ou em inglês *Wireless Sensor Network* (WSN).

As RSSF são uma tecnologia que tem como objectivo recolher informação do meio que as rodeia. Elas são constituídas por um conjunto de nós sensores, que conseguem comunicar entre si. Os nós pertencentes às RSSF são geralmente compostos por um sensor (ou

vários), um microcontrolador, um dispositivo de comunicação e uma bateria. Tal como ilustrado na Figura 1.



**Figura 1** Componentes de um nó sensor [1]

Cada constituinte do nó tem a sua função, o sensor é o elemento que detecta uma grandeza física (temperatura, aceleração, pressão, entre outros) e converte essa grandeza num sinal eléctrico variável (analógico).

O microcontrolador é o dispositivo utilizado em maior número nas WSN. É composto por processador e memória, possuem normalmente memória *flash* e RAM (*Random Access Memory*), conversores analógicos digitais e portas de comunicação, tudo isto combinado com um baixo consumo de potência.

Para haver uma interligação entre os nós, é necessário existir em todos eles um dispositivo de comunicação, geralmente, o tipo de comunicação mais utilizada é a rádio frequência.

As fontes de alimentação são utilizadas para fornecer energia ao nó, sendo as baterias o método mais utilizado. A escolha do tipo de bateria é importante, é necessário ter-se em conta o seu volume, que pode influenciar o *design* do sensor, capacidade e condições de temperatura. As baterias utilizadas com mais frequência são: linear simples, lítio NR e lítio *Coin Cell* [1].

Os nós pertencentes à rede trabalham entre si, de forma a fazer chegar os dados recolhidos ao nó gestor da rede, que habitualmente está conectado a um computador, que processa os dados recolhidos recorrendo a um aplicação desenvolvida para o efeito.

Os três protocolos de comunicação mais utilizados nas RSSF são o *WiFi*, *Bluetooth*, *ZigBee*.




O *WiFi* foi desenvolvido com base nas normas IEEE 802.11 sendo um protocolo destinado a aplicações que precisem de um elevado débito, como é o caso da transferência de ficheiros ou o acesso à Internet e conteúdos multimédia. Funciona na gama dos 2,4 GHz com taxas de transmissão que chegam até 54 Mbps [20].

O *Bluetooth*, baseado nas normas IEEE 802.12.1 e 802.11, é utilizado para comunicação entre um computador e os seus periféricos. Permite ligação entre telemóveis para troca de ficheiros de uma forma simples. Em termos de transmissão, possui um alcance que ronda os 10 metros e um débito de 1 Mbps, funcionando na gama dos 2,4 GHz. [20].

O *ZigBee*, alicerçado na norma IEEE 802.15.4, é um protocolo de comunicação que apresenta uma pilha protocolar pequena e um consumo energético baixo. O alcance da transmissão pode até aos 100 metros. Esta tecnologia trabalha na gama dos 2,4 GHz (globalmente), 915 MHz (América) e 868 MHz (Europa). O seu débito geralmente é de 250 Kbps [20].

A Tabela 1 apresenta uma comparação entre os três *standards* referidos.

**Tabela 1 Comparação de diferentes tecnologias para redes sem fios [8]**

	 ZigBee™	 Bluetooth®	 WiFi™
<b>Gama de frequências</b>	868MHz, 915Mhz, 2,4GHz	2,4GHz	2,4GHz, 5GHz
<b>Débito</b>	20-250Kbts	1-10Mbps	54Mbps
<b>Consumo</b>	≈30mA TX, Standby 0.20µA	>400mA TX, Standby 0.20mA	>400mA TX, Standby 20mA
<b>Alcance</b>	10-100mts	10mts	100mts
<b>Número máximo de dispositivos por célula</b>	255 Dispositivos activos por rede	8 dispositivos activos	2000 dispositivos numa rede estruturada
<b>Célula Básica</b>	Estrela	Piconnet	BSS
<b>Multiplexagem</b>	DSSS	FHSS	DSSS, OFDM
<b>Pontos fortes</b>	Consumo, N.º de Nós, Fiabilidade, custo	Interoperabilidade, Substituição dos cabos	Elevada Taxa de Transferência
<b>Principais Aplicações</b>	Controlo Remoto, Sensores, Dispositivos alimentados por bateria	Periféricos de PC e Telemóveis, PDA's	Internet, Transferências de Ficheiros, Vídeo/Aúdio

Todos os *standards* têm os seus pontos fortes, conseguem ser melhores para um determinado tipo de aplicação e possuem melhor desempenho em certas características. Todos eles conseguem trabalhar na gama dos 2,4 GHz. No que toca ao consumo de energia o *ZigBee* destaca-se por apresentar o consumo mais baixo (30 mA). Em termos de alcance o *ZigBee* e o *WiFi* conseguem atingir os 100 m, uma grande diferença em relação ao *Bluetooth* que apenas consegue atingir os 10 m. Cada *standard* consegue ter apenas um número limitado de dispositivos por nó, sendo o *WiFi* o que pode agregar o maior número de dispositivos por nó.

## 2.1. CARACTERÍSTICAS E FACTORES DETERMINANTES NAS REDES SEM FIOS

As RSSF podem ser vistas como um tipo especial de rede *Ad Hoc*, não utilizam um dispositivo central para encaminhamento dos seus dados. No entanto, cada nó transmite os dados dos nós vizinhos de forma comunitária até à estação de destino.

Tratando-se as RSSF de um tipo especial de rede, elas apresentam uma série de particularidades que as distingue das restantes: a sua escalabilidade, na maioria das vezes, atinge uma ordem de grandeza maior que a das redes convencionais; os nós pertencentes à rede são em grande parte estacionários, o que leva comparativamente a outras redes com nós moveis, que a sua variabilidade de topologia se deva ao recurso limitado de energia e não à mobilidade dos nós; as redes de sensores sem fios utilizam com frequência *broadcast* para comunicação, ao contrário das *Ad Hoc*, que por exemplo, utilizam comunicação ponto-a-ponto.

As redes de sensores sem fios podem ser categorizadas de acordo com a sua composição, organização, mobilidade, densidade, aquisição de dados, capacidade limitada de energia e segurança:

- **Composição** – Quando uma rede de sensores possui nós sensores iguais a nível de *hardware* é classificada como homogénea. Na situação inversa é denominada heterogénea.
- **Organização** Uma rede de sensores é definida como plana ou hierárquica. Se os nós sensores realizarem as mesmas funções e forem pares entre si a rede é

caracterizada como plana. Se existirem grupos de nós a realizarem funções diferentes a rede é definida como hierárquica.

- **Mobilidade** – Uma RSSF é considerada estacionária quando a posição dos nós não se altera quando estão na zona de monitorização.
- **Densidade** – Quando os nós sensores se encontram distribuídos na rede de uma forma regular na área que se pretende analisar, essa mesma rede é considerada balanceada, se isso não acontecer e os nós sensores estiverem distribuídos de forma irregular a rede é classificada como irregular. Quando o número de nós por unidade de área é grande, a rede é designada de densa. Se essa concentração for baixa é considerada espaçada.
- **Aquisição de dados** – A aquisição de dados pode ser executada de quatro maneiras: contínua, os dados são recepcionados de forma contínua e enviados pela rede; quando existem pedidos; programada, o acontecimento a ser monitorizado é agendado; reactiva, se acontecer uma modificação no ambiente monitorizado os dados são enviados.
- **Capacidade limitada de energia** – A fonte de alimentação dos nós sensores são as baterias, quanto maior for a sua duração melhor. Alguns nós sensores são colocados em sítios de difícil acesso, logo um dos parâmetros a ter em conta tem de ser o consumo de energia, para evitar substituições regulares de baterias.
- **Segurança** – Os nós sensores usam a comunicação por radiofrequência, este aspecto torna as redes mais frágeis a ataques [1].

## 2.2. APLICAÇÕES DAS REDES DE SENSORES SEM FIOS

As redes de sensores sem fios foram desenvolvidas inicialmente para utilização militar. No entanto, a globalização desta tecnologia, possibilitou a sua utilização nas mais diversas aplicações. Entre elas encontra-se, a monitorização de ambientes cuja utilização permite a análise de todas as alterações que se passam nesses locais, a segurança de edifícios e estruturas que permite a protecção e vigilância contra eventuais ameaças, permitem o controlo da qualidade de produtos e máquinas na indústria, assim como no controlo de tráfego, que possibilita a verificação do estado e da fluidez do trânsito em pontos críticos.

A área da saúde recorre também as redes de sensores sem fios para monitorizar o estado de saúde de um paciente. Na domótica são utilizadas para controlar os vários aspectos de uma casa, como o ambiente interior, luminosidade, segurança, entre outros. A localização, pelas redes de sensores sem fios que permite localização de objectos e pessoas numa zona [8][1].

Neste momento existem vários produtos no mercado que utilizam RSSF, pelo que outros ainda se encontrem em desenvolvimento nos centros de investigação. São destacados de seguida alguns trabalhos de investigação assim como seus produtos, para as mais diversas áreas, como por exemplo, área do ambiente, área da saúde, área do desporto e área de segurança, sendo que será dado particular destaque as aplicações que permitem localização.

Na área do ambiente, uma das aplicações em pesquisa chama-se *EcoNet* e foi idealizada na *University of Alberta*, no Canadá. Este projecto é constituído por uma rede de diferentes tipos de sensores (temperatura, humidade e luminosidade) que enviam com periodicidade, para uma estação base, os dados analisados. [17].

Na área da saúde, existem projectos já finalizados e outros que se encontram em desenvolvimento. Os projectos como ITALH (*Information Technology For Assisted Living at Home*) [20] ou o AlarmNet [21] encontram-se a ser desenvolvidos com intuito de fornecer assistência médica à distância.

Na área do desporto, as redes de sensores são utilizados no estudo do controlo e da monitorização do movimento dos desportistas. Esta análise é feita com recurso a módulos de sensores sem fios que conseguem detectar acelerações e forças rotacionais. Esta aplicação foi desenvolvida por um grupo do MIT (Massachusetts Institute of Technology) chamado *sportSebble* [23].

No que diz respeito à segurança, foram criados sistemas que conseguem prever a ocorrência de *tsunamis*. Para tal, são colocados acelerómetros no fundo do oceano, que comunicam com estações transmissoras à superfície da água. O sistema permite avisar as populações antecipadamente, possibilitando a redução do número de vítimas [17].

Na Dinamarca, foi criado um sistema para o controlo de pasto de animais. Este projecto avalia a posição dos animais, e permite o estudo do comportamento do grupo e a sinalização de possíveis animais com problemas de saúde. Enquanto faz esta análise

consegue também analisar a qualidade do pastoreio e a necessidade de abrir novas zonas de pastoreio [24].

As redes de sensores servem também para detecção de objectos. Existe um grupo do MIT, que criou um sistema de sensores, designado de *CargoNet*, que tem como funcionalidade localizar e monitorizar contentores, assim como objectos que se encontrem neles armazenados. São utilizados acelerómetros e microfones que detectam movimentos ou grandes ruídos, respectivamente [25].

Em Espanha, na universidade *Rey Juan Carlos*, foi desenvolvido um sistema de localização de dispositivos móveis para redes *WiFi*. Este sistema utiliza um modelo de propagação teórico das ondas *WiFi* combinado com uma técnica de aproximação probabilística da posição do dispositivo. De entre os modelos de propagação existentes, os autores optaram pelo modelo *breakpoint*, que apenas se preocupa com a distância percorrida pelas ondas *WiFi*, descartando as interferências que obstáculos podem ter na transmissão das ondas. O modelo de propagação das ondas foi utilizado de modo a eliminar a criação de um mapa de intensidades de toda a área abrangida pelo sistema. Neste trabalho compararam a utilização do modelo de propagação face ao mapa de intensidades, chegando a conclusão que o mapa de intensidades obtém resultados com menor erro [26].

Na Universidade de Berlim, Alemanha, foi criado um sistema de localização de dispositivos móveis que é utilizado como base para a disponibilização de informação sobre as obras de arte que se encontram numa localização específica do museu *Pergamon* de Berlim. O sistema consegue estimar onde se encontra o dispositivo móvel através do ponto de acesso ao qual ele se liga à rede [17].

Também na Alemanha, mas desta vez na Universidade de Freiburg, foi desenvolvido um sistema de localização de dispositivos móveis para uma rede *WiFi* a funcionar dentro de um edifício de vários andares. Neste caso o edifício *Computing Center*. Este sistema utiliza como técnica o mapa de intensidades do sinal, técnica essa que será abordada mais à frente neste trabalho, na secção das técnicas de localização em 2.3. Uma característica deste sistema é que o mapa de intensidades está guardado numa base de dados mas também nos dispositivos móveis que utilizam o sistema [27].

O sistema iFind foi implementado no campus do MIT, possibilita que docentes e alunos localizem colegas de trabalho dentro do instituto. Esta aplicação pode utilizar tecnologia *WiFi*, *Bluetooth*, *GPS* e *GSM (Global System for Mobile Communications)* para localizar os dispositivos móveis. Utiliza o *Place Lab*, aplicação java (de código aberto) para estimar a localização. O *Place Lab* contém um conjunto de algoritmos de localização para cada um dos protocolos. Em *WiFi*, o sistema sabe o alcance máximo de cada ponto de acesso e consegue combinar essa informação com as coordenadas espaciais dos pontos de acesso que o dispositivo móvel consegue alcançar, obtendo assim uma área onde o dispositivo se encontra [17].

Para desenvolver uma rede de sensores sem fios que permita efectuar a localização de uma pessoa ou objecto, é necessário perceber quais as técnicas de localização existentes para suportar o sistema. Na secção seguinte, 2.3, são apresentadas as técnicas de localização existentes assim como os seus princípios de funcionamento [17][3].

## **2.3. TÉCNICAS DE LOCALIZAÇÃO**

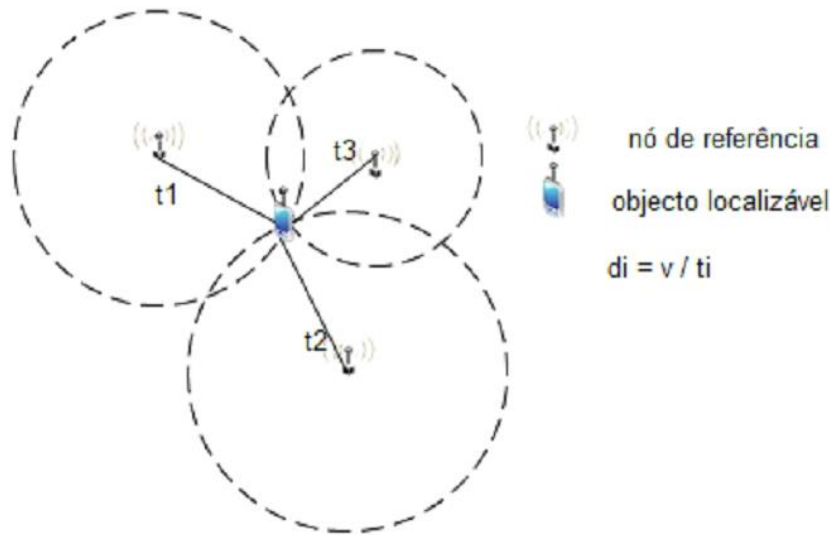
Os sistemas de localização podem ser organizados por categorias segundo diferentes técnicas que são usadas para estimar a localização dos dispositivos móveis. Não se pode dizer que existe um modelo perfeito para definir a localização *indoor* de um objecto ou pessoa. Existem sim, várias técnicas que permitem uma aproximação aos valores reais de posicionamento. Nesta secção são apresentadas as técnicas de localização existentes: tempo de chegada, diferença do tempo de chegada, tempo de voo, angulação, proximidade, análise de padrões, *RSSI* e *fingerprinting*.

### **2.3.1. TEMPO DE CHEGADA**

No método do tempo de chegada (*Time of arrival - ToA*) [28] é utilizado o tempo de chegada de um sinal entre o transmissor e o receptor para determinar uma posição, daí o seu nome. O tempo gasto ao longo do percurso desde o transmissor de sinal até ao receptor é utilizado para calcular a distância entre eles. Esta técnica requer a sincronização da rede, uma vez que uma falha na medição do tempo, 1  $\mu$ s por exemplo, pode levar a um erro de centenas de metros.

Na Figura 2 é ilustrada a técnica de triangulação utilizada no método do tempo de chegada, em que  $t_1$  e  $t_3$  representam o tempo de chegada entre o objecto que se pretende localizar e

os dispositivos que servem como referência. Obtendo o tempo de chegada do sinal e sabendo o tempo de propagação do mesmo é possível calcular a distância entre eles. O raio da esfera centrada no dispositivo de referência é a distância calculada. A intersecção das circunferências, são necessárias no mínimo 3, permite obter a localização aproximada do objecto que se pretende localizar. Esta técnica é a mesma que é utilizada pelos dispositivos GPS.



**Figura 2** Técnica de tempo de chegada [2]

### 2.3.2. DIFERENÇA TEMPO DE CHEGADA

A técnica da diferença do tempo de chegada (*Time –difference of arrival - TDoA*) [28] é baseada na aquisição do tempo de transmissão. Para cálculo do tempo de transmissão são utilizados dois sinais diferentes, com velocidades diferentes de propagação. O sinal que possui maior velocidade de propagação, e que por isso chega primeiro ao receptor é usado para indicar o início da contagem temporal. A recepção do segundo sinal, com menor velocidade de propagação, termina a contagem de tempo, obtendo-se assim a diferença do tempo de chegada. Recorrendo a velocidade de propagação do segundo sinal é possível obter-se uma relação distância-tempo.

### 2.3.3. TEMPO DE VOO

A técnica do tempo de voo (*Time of flight - ToF*) [28] permite uma fácil implementação, sendo apenas necessário saber o tempo de comunicação entre dois pontos. Nesta técnica o emissor efectua uma contagem de tempo a partir do momento em que envia o sinal até receber a resposta. O tempo de transmissão é depois calculado realizando a compensação

do tempo de processamento do receptor e dividindo o tempo pelos dois sentidos de comunicação. O tempo obtido é utilizado para calcular as distâncias e determinar as posições relativas entre dispositivo emissor e dispositivo receptor.

#### 2.3.4. Angulação

A metodologia de angulação [29] utiliza a análise de ângulos para determinar a localização de dispositivos. Tipicamente é usado o Ângulo de Chegada (AoA – *Angle of Arrival*) que consiste na análise do ângulo entre a direcção de propagação do sinal incidente e uma direcção de referência (orientação). Cada nó de referência possui um vector de antenas ou antenas direccionais que irão receber os sinais enviados pelo dispositivo a localizar. São analisados os ângulos de chegada recebidos tendo em conta o eixo principal que funciona como referência para análise dos mesmos.

A posição do objecto é calculada utilizando funções trigonométricas que relacionam os ângulos de chegada recebidos em cada nó de referência e as distâncias entre cada um deles. Na Figura 3 encontram-se ilustrados dois exemplos do método AoA. No exemplo (a), como se conhece a orientação do dispositivo a localizar é possível determinar o ângulo absoluto em relação aos dispositivos de referência  $b_1$  e  $b_2$ . Utilizando estes dois valores angulares consegue-se estimar a localização do objecto. No exemplo (b) como não se sabe a orientação do dispositivo a localizar não se consegue obter o ângulo absoluto. Contudo é possível determinar a localização do dispositivo recorrendo a diferença de AoA dos diferentes nós de referência. Em ambas os casos a informação acerca da posição dos nós de referência é importante para o processo de localização.

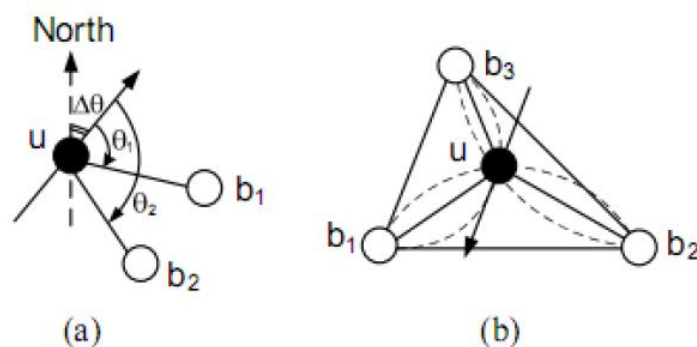


Figura 3 Técnica AoA com orientação conhecida (a) e sem orientação conhecida (b)

[2]

O ponto negativo do método AoA trata-se da *multipath interference*, ou seja, a propagação do sinal por múltiplos caminhos. Pode acontecer que os sinais recebidos não sejam directamente os sinais enviados mas sim uma reflexão dos mesmos, o que desde logo altera o seu ângulo de chegada e leva a uma conclusão errada da posição do dispositivo localizável.

### **2.3.5. Proximidade**

A técnica de proximidade resulta da detecção de objectos devido a sua proximidade com dispositivos de referência. Neste caso o valor da distância entre objecto e nó de referência não é necessário pois apenas se pretende saber que estão próximos um do outro [14].

Um dos métodos utilizado é contacto físico, pelo qual é detectado o contacto entre objecto a localizar com uma referência. Sensores de pressão ou sensores de toque podem ser essas referências, contudo, como se sabe previamente onde elas se encontram saber-se-á também a posição de um objecto quando em contacto com elas [29].

Outro método, que é baseado no conceito de célula de rádio frequência, é a célula de origem. Existe um conjunto de células que possuem um dispositivo de referência. Assim um objecto é detectado no interior de uma das células sabe-se em qual delas se encontra mas não é possível identificar a localização exacta dentro da mesma [29].

A localização recorrendo aos infravermelhos é outra forma de utilizar este método, um emissor é detectado por um determinado sensor quando se encontram próximos.

Outro método amplamente difundido são as etiquetas de identificação por radio frequência (RFID – *radio-frequency identification*) que podem ser colocadas em CDs, livros e outros produtos sendo possível serem detectados por sensores situados nas saídas das lojas [14].

### **2.3.6. Análise de padrões**

A técnica da análise de padrões tem em linha de conta as características que distinguem cada posição do cenário de localização que se pretende observar. O cenário pode ser analisado através de imagens, ou através de intensidade do sinal recebido, RSSI, que será explicado em 2.3.7.

O método de análise de imagens consiste em analisar e processar imagens recolhidas por câmaras de vídeo de forma a encontrar o objecto a localizar. Encontrar a localização do objecto num cenário baseia-se no cálculo da posição do objecto em relação à câmara. Os algoritmos de processamento de imagem dividem-se em dois tipos, *marker* e *markerless*, e são utilizados em sistemas de realidade virtual e aumentada [30][31].

Os algoritmos do tipo *marker* necessitam que os objectos a localizar possuam algum tipo de marca ou característica própria para que possa ser localizado. Os do tipo *markerless* conseguem detectar o objecto mesmo sem que eles possuem qualquer tipo de marca.

### **2.3.7. MEDIÇÃO DE FORÇA DO SINAL RECEBIDO (RSSI)**

O método da medição da força de sinal recebido recorre à análise da potência do sinal recebido no receptor usando modelos de propagação do sinal que indicam como este varia com a distância.

Os modelos utilizados para calcular as perdas relativas à propagação da intensidade do sinal, podem ser divididos em modelos empíricos ou determinísticos. O primeiro grupo utiliza as medições realizadas no ambiente de localização enquanto o segundo tem os seus princípios assentes nos fundamentos de propagação de ondas [2].

Os modelos empíricos realizam uma análise ao ambiente de localização tendo em conta tudo o que o influencia. Utilizando os dados da análise prévia do ambiente consegue-se estimar os parâmetros que o descrevem e efectuar a sua calibração [2].

Os modelos determinísticos são os mais eficientes, mas possuem um entrave, são de difícil implementação uma vez que necessitam de uma análise detalhada da propagação do sinal e da sua relação com as características do ambiente [2].

O RSSI é uma das variáveis utilizadas no desenvolvimento do sistema de localização abordado neste trabalho, sendo por isso descrito de seguida de uma forma mais detalhada.

As WSN usam muitas vezes sistemas baseados em RSSI. Nestes sistemas são colocados módulos em posições específicas, sendo assim os nós de referência. Ao realizar-se a medida de intensidade entre os nós sensores e nós móveis, o sistema consegue assim calcular a localização dos nós móveis com base no sinal de rádio. Deste modo o RSSI pode ser usado para desenvolver um sistema de localização simples, sem necessidade de

equipamento adicional [34]. Vários protocolos possibilitam a utilização do valor de RSSI, sendo que um dos que utiliza esta variável é o protocolo *ZigBee* e os seus dispositivos. O protocolo *ZigBee* será apresentado de uma forma mais detalhada no capítulo 3.

A força de sinal recebido (RSS - *Received Signal Strength*), a energia do sinal, pode ser medida para cada pacote recebido. A energia do sinal medido é quantificada formando o indicador da força de sinal, conhecido por RSSI [2].

A potência com que um sinal chega a um receptor é uma função, que possui entre outros factores, a distância em relação ao emissor. Este aspecto é usado para determinar a distância entre eles, utilizando como referência o RSSI. Quanto mais longe se encontrar um emissor menor será a energia do sinal quando chega ao receptor. A equação (1) conhecida como Fórmula de *Friss*, é utilizada para relacionar potência de um sinal recebido com a distância a partir da qual foi enviado [35].

$$P_r = P_t \frac{G_t G_r \lambda^2}{(4\pi)^2 d^n} \quad (1)$$

Onde,  $P_r$  é a Potência da antena de recepção em watts;  $P_t$  a Potência enviada pela antena de transmissão em watts;  $G_r$  o Ganho da antena de recepção;  $G_t$  o Ganho da antena de transmissão;  $\lambda^2$  o Comprimento de onda,  $\lambda = c/f$ , onde  $c$  é a velocidade da luz (299792458 m/s) e  $f$  é a frequência do sinal;  $d$  a distância em metros;  $n$  a Constante de propagação do sinal. No caso de não existir obstáculos entre os dispositivos que comunicam entre si o valor de  $n$  é igual a 2. Em espaços fechados onde existem obstáculos e interferências o valor de  $n$  pode ir de 2 até 3,5.

Considerando-se  $P_r$ ,  $P_t$ ,  $G_r$ ,  $G_t$ , e  $\lambda$  constantes, chega-se à equação (2):

$$P_r = \frac{k}{d^n} \quad (2)$$

Onde  $k$  é uma constante e  $P_r$  está em Watts.

A grande maioria dos módulos *Zigbee*, expressa o RSSI em dBm. A conversão da potência de watts para dBm é feita recorrendo a equação (3).

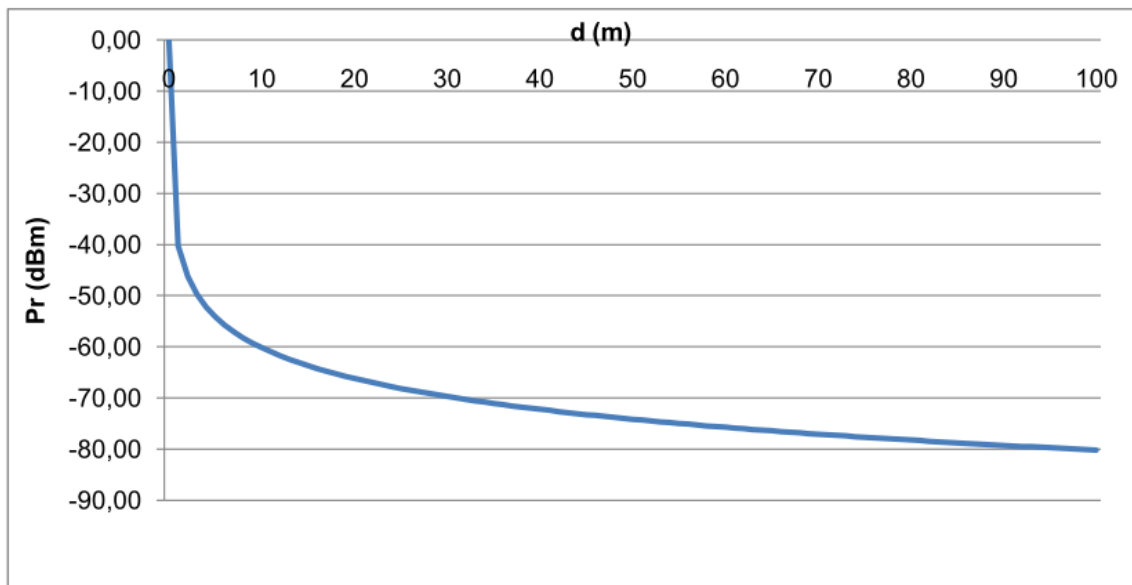
$$P[dBm] = 10 \log(P[W] \cdot 10^3) \quad (3)$$

Substituindo-se a equação (2) na (3) obtém-se a equação (4), onde o RSSI corresponde à potência recebida em dBm.

$$RSSI = -10 n \log(d) + K \quad (4)$$

Onde  $n$  é a constante de propagação do sinal,  $d$  é distância em metros entre transmissor e receptor e  $K$  o valor de sinal recebido a um metro de distância.

A equação (4) [34] mostra que o valor da força do sinal diminui com o aumento da distância. O gráfico da Figura 4 representa o comportamento ideal da intensidade de sinal recebida em relação à distância para um ambiente exterior sem obstáculos ( $n=2$  e  $k= 40$  dBm).



**Figura 4** Gráfico da relação RSSI com distância entre transmissor e emissor [33]

Existem quatro características associadas ao RSSI: gama de valores, precisão, linearidade e média do período. O valor do RSSI é quantificado em dBm, a gama de valores é o intervalo de energia de sinal, desde o mínimo ao máximo que o receptor consegue medir. A precisão do RSSI indica o erro médio relacionado com cada medição da intensidade de sinal recebido. A linearidade do RSSI mostra o desvio máximo de trama de RSSI numa linha recta em relação a potência de sinal recebido real (em escala em logarítmica). A média do período é o valor médio da força de sinal recebido num determinado período de tempo.

### 2.3.8. Factores que influenciam o RSSI

Existem diferentes factores que afectam o valor do RSSI num espaço fechado. Entre eles encontram-se a presença/ausência de utilizadores, altura do dia, tipo de edifício e material,

distância do transmissor e tipo de rádio. Outros factores não menos importantes são a orientação da antena, direccionalidade e tipo [36]:

- *Ausência ou Presença de utilizador:* Em termos da recolha de dados, recolha dos valores da força de sinal, a presença ou ausência do utilizador afecta essa recolha. Com a presença de pessoas no sistema os valores da força de sinal recolhidos são diferentes dos valores obtidos na ausência delas.
- *Orientação e tipo de antena:* Existem diferentes tipos de antena e cada uma delas tem o seu próprio padrão de radiação, sendo elas não uniformes. Para antenas diferentes, os valores de força de sinal analisados serão forçosamente diferentes. A orientação da antena do emissor e do receptor durante a comunicação afecta o valor da força de sinal.
- *Variabilidade do transmissor:* Dois transmissores configurados da mesma maneira podem ter um comportamento diferente. Quando se configura um transmissor para enviar pacotes de dados a um nível de potência, não significa que os dados são enviados exactamente a esse nível de potência e mesmo ao longo do tempo esse valor pode variar. Este aspecto pode alterar a força de sinal recebido, mesmo que o receptor se encontre na mesma posição.
- *Variabilidade do receptor:* A sensibilidade dos receptores é diferente para cada um deles. Dois receptores colocados na mesma posição, com todos os outros factores que afectam a força do sinal constantes, podem emitir diferentes indicadores da força de sinal.
- *Reflecção e multiplicação de sinal:* Num espaço fechado os sinais transmitidos sofrem reflecção, tanto nas paredes como em objectos presentes nesse ambiente. Este fenómeno provoca a duplicação do sinal, enquanto o sinal original e o sinal reflectido seguem diferentes “caminhos” mas chegam ao receptor praticamente ao mesmo tempo. O receptor não tem forma de distinguir os dois sinais recebidos medindo assim a intensidade do sinal recebido de ambos.
- *Momento da medição:* O momento do dia em que é realizada a medição influencia o valor da intensidade de sinal obtido. Realizar uma medição durante a tarde pode indicar valores diferentes de uma medição realizada durante a noite.

### 2.3.9. FINGERPRINTING

O método de localização que utiliza a análise de cena por *fingerprinting* é utilizado para contornar as adversidades encontradas em ambiente *indoor*. Esta metodologia analisa uma determinada cena, examinando as características electromagnéticas do cenário; ou seja utiliza a força do sinal recebido. Esta metodologia é implementada criando um mapa de energias de sinal para uma determinada área, em relação a um ou mais dispositivos de referência; é denominado de mapa *fingerprinting*, exemplificado na Figura 5. Uma das principais dificuldades na utilização deste método passa pela criação coerente e eficaz do mapa, isto porque o valor da força do sinal captado para um determinado ponto tem de ser único. A existência de pontos com valores de força de sinal repetidos tornaria o mapa de *fingerprinting* impreciso. Uma desvantagem desta técnica é que ela exige muito tempo em termos de implementação e sempre que se altera a área de interesse é obrigatório criar outra base de dados.

A propriedade de sinal utilizada para criação do mapa de *fingerprinting* é o RSSI, apresentado em 2.3.7. Este indicador da força do sinal não apresenta porém uma variação linear em relação à distância, nem apresenta um valor fixo ao longo do tempo para um determinado ponto.

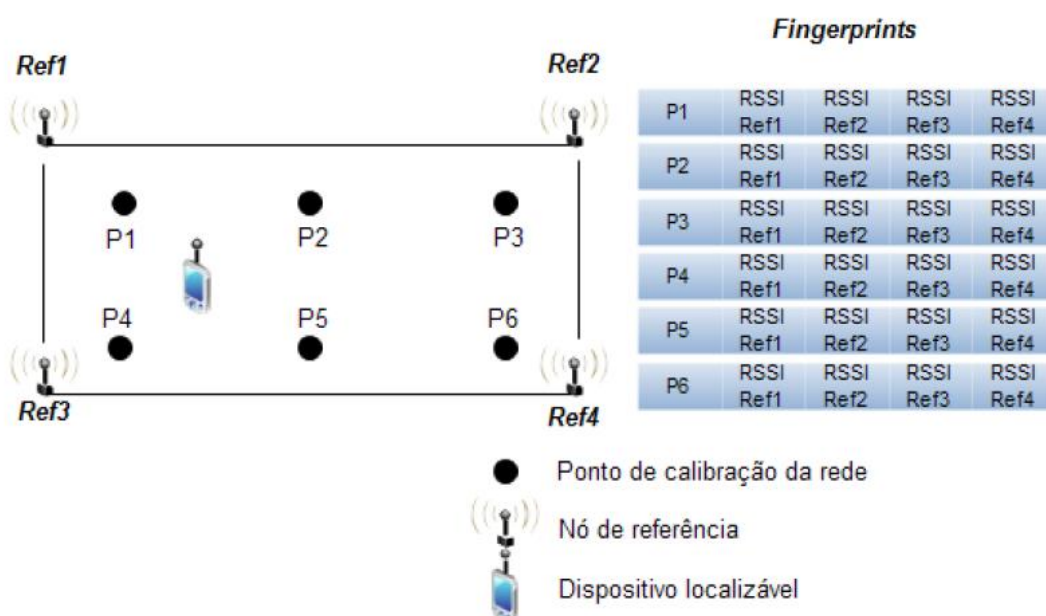


Figura 5 Exemplo de um mapa de *fingerprints* [2]

As amostras de RSSI obtidas para um determinado ponto indicam onde se encontra esse ponto; ou seja, têm-se a indicação das coordenadas espaciais do mesmo. Devido a esta relação dos sinais com a posição são distinguidos dois domínios: o domínio dos sinais e o domínio espacial. Estes dois domínios relacionam-se entre si pois um valor no domínio dos sinais corresponde a um só valor no domínio espacial [14].

O domínio espacial representa o espaço, isto é, os diferentes pontos do mapa *fingerprinting*. Os sistemas de localização podem ter dois tipos de sistemas de indicação da posição:

- *Sistema Simbólico*: Neste tipo de sistema o mapa de *fingerprinting* é construído na forma de etiquetas. Por exemplo, numa habitação é atribuída uma etiqueta a cada divisão. Apenas se consegue saber em que divisão se encontra o dispositivo a localizar.
- *Sistema físico*: Neste sistema é utilizado um sistema de coordenadas, que podem ter duas ou três dimensões. A indicação da orientação é facultativa e as coordenadas podem ser absolutas ou relativas. Um sistema físico permite um maior detalhe e precisão de localização que um sistema simbólico.

Recorrendo ao exemplo da habitação, num sistema simbólico saberíamos apenas que um dispositivo se encontra na divisão cozinha, enquanto num sistema físico saberíamos em que ponto exacto da cozinha (coordenada) ele se encontra. A vantagem do sistema simbólico é ter uma implementação mais fácil [14].

O domínio dos sinais representa os valores de RSSI obtidos para cada referência do mapa. Cada coordenada do mapa em termos espaciais tem um valor de RSSI associado, quanto mais pontos de referência existirem maior será este domínio.

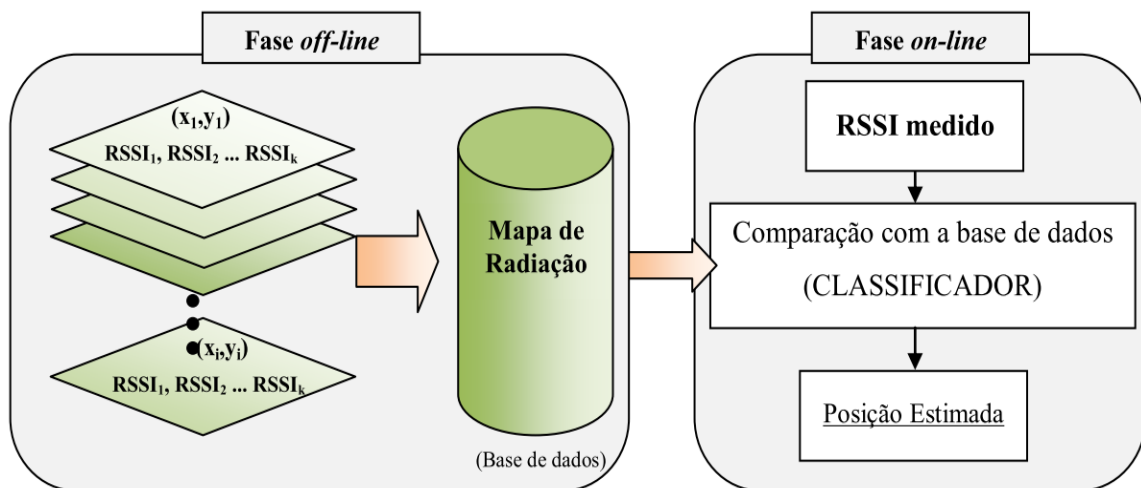
Para implementar o método de *fingerprinting* e efectuar a criação do mapa de valores de energia é necessário dividir o trabalho em duas fases, a fase *Offline* e a fase *Online* [37]:

- Fase de calibração ou fase *Offline*: Numa primeira fase é necessário efectuar a criação do mapa de valores de energias, criar uma ligação entre os valores do domínio dos sinais com os valores de domínio espacial, ou seja, um valor de energia de sinal corresponde a um ponto no mapa.

- Fase Online: Nesta fase o sistema encontra-se em funcionamento, os valores da força de sinal recebidos são comparados com os valores gravados no mapa, tratados por um algoritmo de localização com o intuito de obter a sua localização aproximada no domínio espacial.

A fase *Online* é muito dependente da fase de calibração, pois qualquer erro ou indefinição na fase de calibração pode tornar a fase *Online* inviável.

Na Figura 6 estão representadas as fases da técnica de localização por *fingerprinting*.



**Figura 6** Fases da técnica de localização por *fingerprinting* [33]

Para concretizar as duas fases referidas anteriormente (*Offline* e *Online*), de uma forma correcta, é necessário seguir duas abordagens, abordagem determinística ou abordagem probabilística [38]:

- Abordagem determinística: Cada coordenada definida no mapa é caracterizada pelo indicador da força de sinal, o valor médio e o desvio padrão das várias medições efectuadas na fase de calibração. O número de referências no sistema influencia o número de coordenadas para cada ponto.
- Abordagem Probabilística: Quando se utiliza uma abordagem probabilística, está-se a utilizar um algoritmo probabilístico, como por exemplo o teorema de *Bayes*, teoria da aprendizagem estatística ou redes neuronais artificiais, para obter o melhor valor no domínio espacial de um valor do domínio dos sinais obtido na fase *Online*.

A técnica de *fingerprinting* foi utilizada no desenvolvimento deste projecto como será possível verificar mais adiante neste relatório na secção 4.1.

### 2.3.10. ALGORITMOS DE LOCALIZAÇÃO

Os algoritmos de cálculo da localização analisam os valores de RSSI guardados na base de dados e as suas respectivas coordenadas, e compara-os com as amostras de RSSI obtidas já com o sistema em funcionamento de forma a encontrar a sua possível localização. Dois algoritmos muito utilizados são *K-Nearest Neighbor* e as redes neuronais [39]:

*K-Nearest Neighbor* - A base deste algoritmo consiste no cálculo das distâncias entre os valores guardados na base de dados (mapa *fingerprinting*) e os valores obtidos na fase *online*. Quando se fala em distância está-se a falar na diferença entre os valores de RSSI guardados e os valores de RSSI da fase *Online* e não numa distância física propriamente dita. A localização é estimada pelos *fingerprintings* que possuem distância mais pequena. O valor escalar que identifica os valores de RSSI do mapa de *fingerprinting* tem de estar bem definido; normalmente usa-se a média ou mediana [39][33].

Uma analogia que permite perceber a forma como este algoritmo funciona é a do diagnóstico médico. Um médico perante os sintomas de um paciente, pode comparar esses sintomas com os registos de pacientes já diagnosticados, sintomas parecidos, com o propósito de auxiliar no diagnóstico de determinada doença.

Para calcular a distância entre valores da fase *Offline* e valores da fase *Online* são utilizados geralmente dois métodos, distância Manhattan e distância Euclidiana. A equação (5) apresenta o cálculo da distância de Manhattan.

$$d_i = (I'_1 - I_{i1}) + (I'_2 - I_{i2}) + \dots + (I'_k - I_{ik}) \quad (5)$$

Sendo,  $d_i$  a distância de Manhattan ao ponto de colecta “i”;  $I_{ik}$  o valor RSSI armazenado no mapa correspondente ao ponto de colecta “i” em relação a estação base “k”;  $I'_k$  o valor RSSI lido pelo dispositivo móvel correspondente à Estação Base “k”.

O cálculo da Distância Euclidiana é apresentado na equação (6), onde “i” identifica o ponto de colecta e “k” identifica a estação base.

$$d_i = \sqrt{(I'_1 - I_{i1})^2 + (I'_2 - I_{i2})^2 + \dots + (I'_k - I_{ik})^2} \quad (6)$$

Sendo, o  $d_i$ : distância de Euclidiana ao ponto de colecta “i”;  $I_{ik}$  o valor RSSI armazenado no Mapa correspondente ao ponto de colecta “i” e à estação base “k”;  $I'_k$  o valor RSSI lido pelo dispositivo móvel correspondente à Estação Base “k”.

Redes Neurais – As redes neuronais são essencialmente utilizadas quando o comportamento do RSSI é complexo para ser analisado matematicamente. Uma rede neuronal é uma estrutura de processamento que funciona como uma caixa negra e é constituída por um conjunto de neurónios, há uma tentativa de modelar o sistema recorrendo a uma função complexa e não linear [39][14].

Cada neurónio possui várias entradas multiplicadas por pesos sinápticos, um somador, e uma função, geralmente, não linear que limita a amplitude da saída do neurónio.

A associação dos neurónios do sistema pode ser treinada com as diversas amostras de *fingerprinting* da base de dados com o intuito de calcular os pesos sinápticos associados a cada uma das entradas. O processo de treino consiste em forçar cada um dos *fingepoints* captados na fase *Offline* a saída pretendida, isto é, as coordenadas dessa localização. Existem várias técnicas de treino como por exemplo a *one step-secant*, e a de *back propagation error* [14].

## **2.4. SISTEMAS DE LOCALIZAÇÃO**

O facto de se querer perceber como a localização de dispositivos móveis é efectuada em redes de sensores sem fios, levou ao estudo de alguns sistemas já existentes. Nesta secção são apresentados alguns desses sistemas bem como o seu modo de funcionamento. Existem vários sistemas de localização *indoor* desenvolvidos com recurso às mais diversas tecnologias, como GPS, RFID, *WiFi*, *ZigBee* entre outros. As tecnologias *wireless* são as mais utilizadas actualmente. O objectivo desta pesquisa e estudo, é retirar informação que possa ser útil no desenvolvimento do trabalho.

### **2.4.1. ACTIVE BADGE LOCATION SYSTEM**

O *Active Badge Location System* [40] foi desenvolvido no *Olivetti Research Laboratory*, em Cambridge, é um sistema baseado na tecnologia de infravermelhos (IV).

Ele possui um conjunto de crachás electrónicos, *badge*, representado na Figura 7, que não é mais que um transmissor de infravermelhos, que pode ser acoplado ao objecto que se

pretende detectar. O sistema possui também um conjunto de sensores colocados ao longo da área de localização, todos estes dispositivos estão conectados a um servidor de rede.



**Figura 7 Sensor Active Badge [1]**

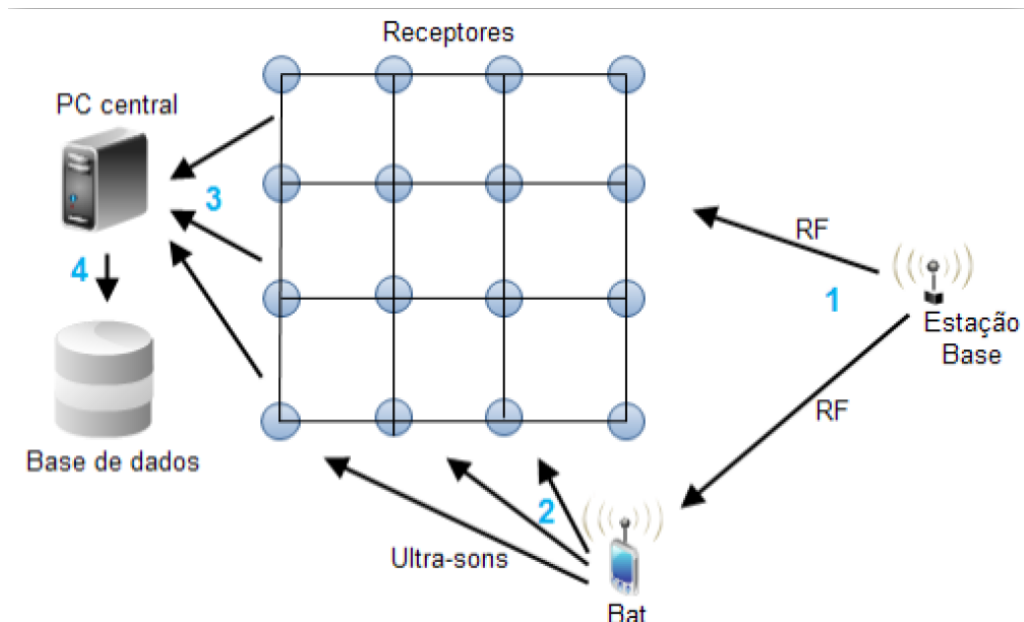
O *badge* envia periodicamente um sinal infravermelho com o seu ID, que vai ser captado pelos receptores dispostos ao longo do ambiente. A informação recolhida pelos receptores é enviada para o servidor de rede, que irá processar esses dados por *software* sendo assim possível estimar a localização do emissor. O sistema permite que se consiga ver a posição gráfica dos *badge*, por exemplo, na planta de um piso de um edifício, ou noutro formato criado pelo utilizador.

O alcance deste sistema é baixo, cerca de 5 metros, encontra-se também limitado pelas posições dos sensores de referência. A luz solar directa altera as condições da transmissão infravermelha alterando as condições de funcionamento do sistema [1].

#### **2.4.2. ACTIVE BAT- ULTRASONIC LOCATION SYSTEM**

O sistema *Active Bat* [41] surgiu como alternativa ao *Active Badge* por parte da mesma empresa, com a diferença deste sistema ser baseado no TDoA entre sinais ultra-sónicos e sinais de rádio frequência.

A composição deste sistema, como podemos visualizar na Figura 8, é feita por uma rede de receptores, organizados em matriz, um computador que funciona como unidade central do sistema e por dispositivos móveis, os *Bats*, que são colocados em objectos ou pessoas a localizar.



**Figura 8 Sistema Active Bat [10]**

A estação base do sistema envia um pedido através de um sinal RF (Rádio Frequência) que é detectado pelos *Bats* e pelos receptores. Quando o *Bat* recebe este pedido, emite um sinal para rede de sensores, neste caso um sinal ultra-sónico. Os sensores de referência vão receber estes dois sinais, medir a diferença temporal entre eles e enviar essa mesma informação para uma estação central. A estação central faz análise da informação recebida e calcula a localização do *Bat* avaliando a distância entre dispositivos [10].

### 2.4.3. RADAR

O sistema de localização RADAR [42] é sustentado em rádio frequência e utiliza dispositivos com o padrão IEEE 802.11. Os métodos utilizados para calcular a localização são a triangulação e análise de cenário. O sistema é constituído por várias estações base para que a cobertura do sinal seja maior.

É preciso analisar os valores de potência de sinal para cada posição da área de localização, ou seja criar um mapa de *fingerprints*. Posteriormente os valores captados serão comparados com este mapa, de forma a identificar a localização do dispositivo.

Para calcular a posição de um dispositivo este sistema utiliza o algoritmo K-NN. Este algoritmo calcula a distância Euclidiana entre os sinais recebidos e o mapa de *fingerprints*. A posição do objecto é definida considerando a menor distância Euclidiana [10].

#### 2.4.4. UBISENSE

O *Ubisense* [43] é um sistema de localização em tempo real, desenvolvido para aguentar diversas aplicações de forma integrada, em ambientes amplos e com bastantes utilizadores. Foi criado pela Universidade de *Cambridge* e possibilita uma maior exactidão em relação a sistemas que usam RFID ou *WiFi*.

O sistema é composto por: *tags* activas que enviam pulsos UWB (*Ultra-wide-band*), sensores fixos de referência que recebem os sinais enviados pelas *tags* e uma interface que permite visualizar a posição.

As *tags* possuem um transmissor UWB e um transceptor de rádio 2,4 GHz que possibilita a troca de mensagens de controlo entre *tag* e os sensores. Elas transmitem pulsos UWB a uma frequência de 6 a 8 GHz com uma duração curta quando um sensor de referência realiza um pedido.

Os sensores são sensíveis aos pulsos UWB de curta duração e baixa potência enviados pelas *tags* pois permite uma filtragem mais fácil e uma melhor distinção entre sinais reflectidos.

Cada sensor de referência detecta o angulo de chegada do sinal enviado pela *tag* recorrendo a um *array* de antenas. Eles comunicam e são programados via *Ethernet* ou via *Wireless*, isto consoante arquitectura do sistema desenvolvido.

Todos os dados da rede de sensores, ilustrada na Figura 9, serão enviados para a plataforma de *software* que irá gerir toda a informação possibilitando assim a visualização da localização num ambiente gráfico, e também outras informações que poderão estar relacionadas com os sensores [10].

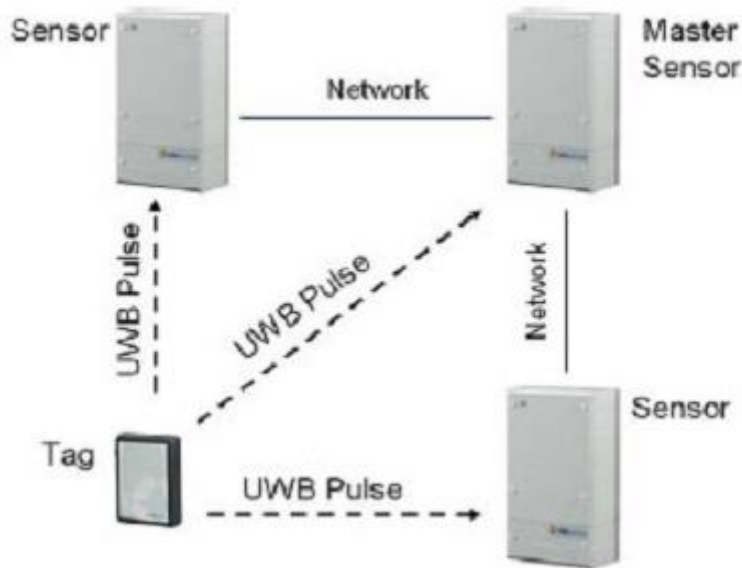


Figura 9 Rede de Sensores do sistema *Ubisense* [10]

#### 2.4.5. LANDMARC

O *LandMarc* (*Location Identification based on Dynamic Active RFID Calibration*) [44] é um sistema que recorre a tecnologia RFID para estimar localização de objectos.

Este sistema é composto por leitores RFID que permitem ler a informação enviada pelas *tags* RFID. Existem as *tags* activas e passivas. As passivas não precisam de baterias, têm alcance limitado e tamanho reduzido, por sua vez, as activas possuem bateria e têm maior alcance.

A parte física do sistema *LandMarc* é formada por leitores RFID, *tags* activas RFID e um servidor que realiza a comunicação com os leitores e a estimativa da localização.

Neste sistema existem *tags* com coordenadas de referência definidas previamente e que são a base para a determinação da localização das *tags*. O sistema é ilustrado na Figura 10.

As *tags* enviam periodicamente um sinal com o ID. O leitor mede a potência do sinal da *tag* a localizar e a intensidade das *tags* de referência que se encontram mais perto. Essa informação é encaminhada para o servidor que, recorrendo a um algoritmo de K-NN, estima a localização das *tags*.

O número de leitores e o número de *tags* de referência que se encontrem próximas da *tag* a localizar influenciam a exactidão do sistema.

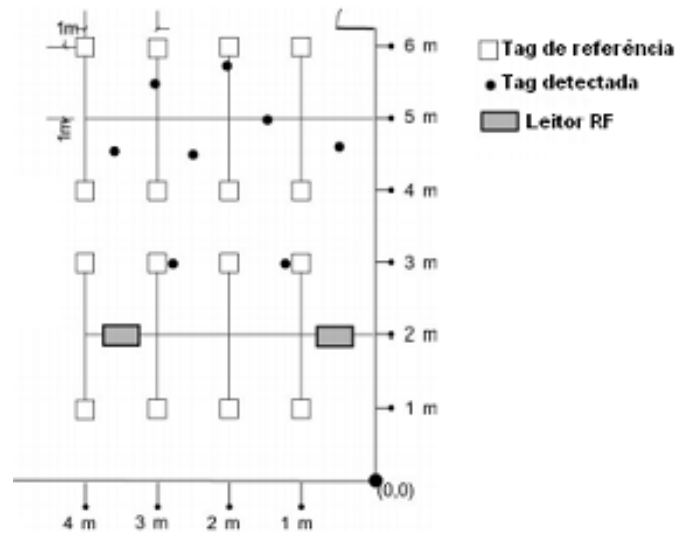


Figura 10 Sistema LandMarc [2]

#### 2.4.6. ZIGBEE LOCATION ENGINE

O chip CC2431 foi desenvolvido pela Texas Instruments, ele possibilita efectuar localização em redes wireless recorrendo ao protocolo ZigBee.

O chip possui um módulo de hardware que é utilizado pelos dispositivos terminais para estimar a posição do location engine, módulo que se pretende localizar.

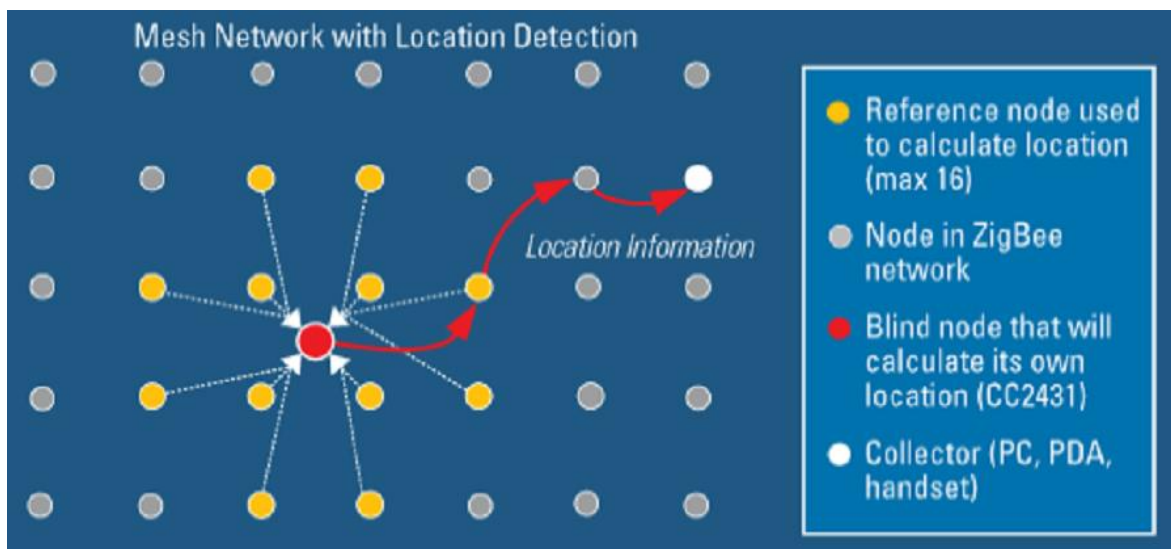
Para a localização são precisos três tipos de nós: Blind Node, Reference Node e Location Dongle [49]:

- *Blind Node:* nó que não se conhece a sua localização. Quando recebe uma solicitação para localização efectua o cálculo da sua posição baseado no location engine. Após efectuar o cálculo da localização envia essa informação para o Location Dongle.
- *Reference Node:* um dispositivo imóvel com posição (X,Y) definida e que indica a sua localização todas as vezes que recebe um pedido para isso de outros dispositivos.
- *Location Dongle:* dispositivo que realiza a ligação entre as aplicações de um computador e a rede que é composta por Blind Nodes e References Nodes. Possibilita deste modo a configuração da rede de sensores e a recepção das mensagens de localização.

O processo de localização é efectuado da seguinte forma:

- O pedido de localização é enviado pelo *blind node* por mensagem *broadcast* para todos os *Reference Node*;
- Os *Reference Nodes* respondem para o *Blind* com informação da sua posição (X,Y) e o valor de RSSI guardados na recepção do pedido de localização emitido pelo *Blind Node*.
- O *Blind Node* recorre ao *location engine* para calcular a posição por triangulação. O *location engine* consegue calcular a distância a um *reference node*, utilizando a posição desse *reference node* e o valor de RSSI registado, recorrendo a um modelo de perdas por propagação do sinal RF, realizando e calculando a distância. Utilizando três distâncias obtidas e com a informação da posição fixa do *reference node* é possível calcular a posição do *blind node*. Após este processo envia a informação da posição calculada para o *location dongle*.
- O *Location Dongle* envia a informação para uma aplicação, que permite uma identificação do *Blind* numa interface com o utilizador.

O sistema *location engine* encontra-se ilustrado na Figura 11.



**Figura 11** Sistema de localização baseado no *Location Engine* [2]

A exactidão deste sistema para aplicações de localização é cerca de 3 a 5 metros. Este módulo é indicado para aplicações onde é necessário um baixo consumo energético.

#### 2.4.7. INDOOR LOCAL POSITIONING SYSTEM FOR ZIGBEE, BASED ON RSSI

O projecto *Indoor Local Positioning System for ZigBee* consiste num sistema de localização e monitorização de movimentos, que é alicerçado sobre uma rede de sensores sem fios e recorre à avaliação dos valores RSSI da tecnologia *ZigBee*.

A rede de sensores é constituída por *End Devices* (ED), um *Coordinator* e *Routers*:

- Vários *End Devices* que são os dispositivos que se pretende localizar.
- O *Coordinator* que realiza a gestão de toda a rede de sensores.
- Quatro *Routers* que permitem a captação dos sinais provenientes dos ED e analisam o respectivo valor de RSSI.

Na Figura 12 está ilustrada a implementação do protótipo.

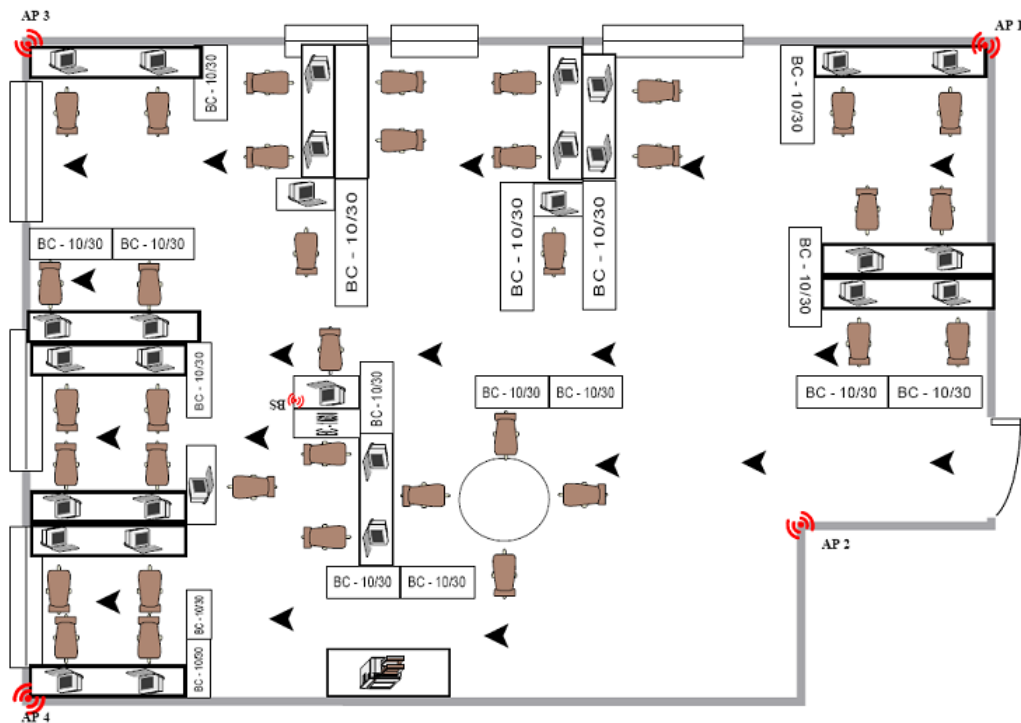


Figura 12 Sistema implementado na Master's room, Mid Sweden University [18]

A técnica utilizada para a localização dos *end devices* é a de *fingerprinting*.

Os dispositivos utilizados, para qualquer uma das funcionalidades acima apresentadas são baseados em plataformas SENTIO [18].

Existe uma fase posterior à implementação do projecto, designada por *offline*, onde são definidos os locais que se quer obter uma *fingerprint*, esse locais são definidos tendo em atenção a exactidão e o intervalo de confiança que se pretende para o sistema. Depois de se colocarem os *routers* em cantos opostos, o valor de RSSI pertencente a cada local é medido através de sinais *ZigBee* enviados por um ED colocado nos locais referidos. Os valores de RSSI analisados são guardados numa base de dados sendo assim criado uma mapa de valores desse local. Nesta fase são também determinadas as distâncias dos pontos seleccionados a um ponto de referência, valores que são igualmente inseridos na base de dados sendo organizados com os valores de RSSI correspondentes. Já na fase *online*, os *routers* analisam os sinais captados e verificam o valor de RSSI e reencaminham esse valor medido ao ED que o transmitiu. Esse ED cria um vector RSSI (vector de 4 coordenadas RSSI correspondentes a cada *router*) que é encaminhado para o sistema de informação. A posição do ED é estimada quando se efectua a comparação entre o vector RSSI e a base de dados, recorrendo às distâncias Euclidianas.

Um factor relevante para um bom funcionamento do sistema é a colocação dos *routers*, sendo aconselhada uma distância média de 10 metros entre eles para um melhor desempenho. As variações de sensibilidade do receptor e do transmissor são outros aspectos a salientar tal como a orientação da antena ou os efeitos de interferência na propagação das ondas, aspectos que afectam o RSSI. Neste projecto chegou-se também a conclusão que o número de amostras de valores RSSI na construção do mapa não é relevante, uma vez que a curva média tende a ser a mesma para vários números de amostras [18].

# 3. ZIGBEE

A *ZigBee Alliance* é uma associação constituída por várias empresas de diferentes segmentos, entre elas a *Motorola, Philips, Samsung, Siemens, Analog Devices, Texas Instruments*. O objectivo desta aliança é desenvolver produtos e padrões *wireless* que sejam de baixo custo, baixo consumo, seguros e confiáveis. Partindo deste pressuposto desenvolveram o protocolo *ZigBee* junto do IEEE.

Os alicerces da tecnologia hoje designada por *ZigBee* foram estabelecidos no protocolo *Home RFLite* criado pela *Philips*. A sua primeira versão foi apresentada ao público em 27 de Julho de 2005, o protocolo foi revisto em 2007 sendo definido com *ZigBee PRO*. O termo zig zag adicionado a palavra abelha em inglês, *Bee*, deu origem ao nome *ZigBee*. Isto resulta da semelhança entre uma topologia de rede em malha *ZigBee* com a forma que as abelhas se comportam numa colmeia. As abelhas quando saem à procura de néctar voam em zig zag comunicando com outras da mesma colmeia, dando informações sobre distância, direcção e localização de alimentos.

O *ZigBee* é um *standard* que define um conjunto de protocolos de comunicação para dispositivos, com baixas taxas de transmissão de dados, sem fios. Os dispositivos *ZigBee* podem trabalhar em três bandas de frequência: 868 MHz , 915 MHz e 2,4 GHz. Na europa é utilizada a frequência de 868 MHz para várias aplicações, na América do Norte e

Austrália é utilizada a frequência de 915 MHz por sua vez a frequência de 2,4 GHz é usada em toda a parte. Na maioria das vezes é escolhida a frequência de 2,4 GHz, pois é a que permite uma taxa de transmissão mais elevada e também fornece um maior número de canais.

Quanto à modulação, é utilizado O-QPSK (*Offset Quadrature Phase-Shift Keying*) para a banda dos 2,4 Ghz e BPSK (*Binary Phase Shift Keying*) para os 915 ou 868 MHz.

A taxa máxima a qual os dispositivos *ZigBee* conseguem transmitir é de 250 kbps. Este género de dispositivos são óptimos para redes em que se precisam grandes taxas de transmissão de dados e também onde se pretende que os custos não sejam altos. O alcance máximo da comunicação num espaço fechado (*indoor*) é de 150 metros e fornece um máximo de 65535 nós por coordenador.

O protocolo *ZigBee* é o futuro das redes de sensores sem fios, pois possui uma valência muito importante nos dias de hoje; baixo consumo energético.

### **3.1. TIPOS DE DISPOSITIVOS**

Uma rede *ZigBee* é composta por diferentes dispositivos, cada um deles com as suas próprias funções na rede. O *ZigBee* distingue três tipos de dispositivos lógicos; o coordenador, *router* e o *end-device*:

- O coordenador é um dispositivo fundamental na rede pois é ele que inicia e faz a gestão da mesma, como por exemplo, admitir outros nós na rede (*routers* e *end-devices*), atribuir endereços de rede e encaminhamento de pacotes de dados. Qualquer tipo de rede que seja criada nunca pode ter mais de que um coordenador. Outro aspecto importante, é que o coordenador necessita de ser alimentado por uma fonte de energia estável e confiável pois este nó processa muitos dados e tem de estar ligado continuamente.
- O *router* é outro dos dispositivos que pode estar presente na rede, ele funciona como um mensageiro entre nós, sem que seja necessário a utilização do coordenador, a sua principal tarefa é o reencaminhamento dos dados. Ao contrário do coordenador, podem existir múltiplos *routers* na rede.

- Por último, temos o *end device*, nó terminal de uma rede. Este nó apenas consegue comunicar com *routers* ou com o coordenador, ou seja, não consegue fazer a retransmissão de mensagens provenientes de outros nós terminais. É possível colocar este dispositivo em modos que permitam poupança de energia, como por exemplo no modo *sleep*.

### 3.2. TOPOLOGIAS

A criação da rede é controlada pela camada de rede do *ZigBee*, tendo por base uma das duas topologias especificadas pelo IEEE 802.15.4, em estrela ou ponto a ponto. A topologia ponto a ponto pode ser dividida em dois formatos, malha e árvore. Na Figura 13 encontram-se ilustradas as topologias referidas.

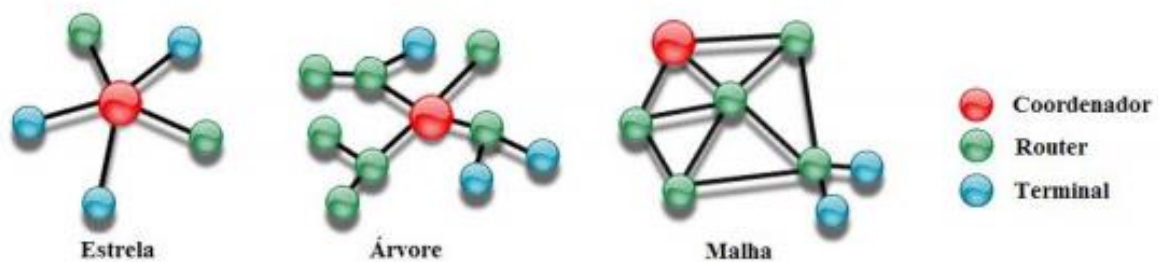


Figura 13 Topologias da rede *ZigBee* [1]

Na topologia em estrela, a mais simples das topologias de rede, cabe ao coordenador *ZigBee* o controlo da rede. Ele assume um papel central e comunica directamente com todos os dispositivos terminais da rede, consegue comunicar com até 65535 terminais. Toda a informação que flui na rede passa pelo nó central, é ele que inicia e mantém os dispositivos na rede. Nesta topologia apenas existe um caminho entre coordenador e *end-device*. Caso exista uma falha na rede deixa de haver comunicação pois não existe caminho alternativo. Outra aspecto negativo é o congestionamento a qual está sujeito o coordenador, devido a constante comunicação com todos os *end devices* da rede.

Na topologia em malha não existe uma centralização como acontece na topologia em estrela, os *routers* e coordenador tem liberdade para enviar dados para qualquer dispositivo na rede. Nesta topologia é fácil aumentar fisicamente a rede, sendo possível criar uma rede que englobe uma área de grandes dimensões. A comunicação entre dispositivos é quase

sempre garantida pois mesmo havendo uma falha em algum, existem sempre caminhos alternativos para fazer chegar a mensagem.

A topologia em árvore tem algumas semelhanças com a topologia em malha, a principal diferença encontra-se no facto de ser estabelecida uma hierarquia através da qual é realizada a distribuição de dados. No topo desta hierarquia encontra-se o coordenador sendo o nó central da rede. A partir do nó central surgem várias ramificações primárias de onde saem por sua vez ramificações secundárias. As ramificações primárias conseguem apenas comunicar entre si utilizando o coordenador como intermediário. Por seu lado as ramificações secundárias conseguem comunicar entre si passando a informação pelo *router* até ao destinatário [1] [5].

### **3.3. MODOS DE OPERAÇÃO DA REDE**

A sincronização das transmissões das tramas ocorre no modo *non-beacon*, utilizando para esse fim o método CSMA-CA (*Carrier sense multiple access with collision avoidance*). O nó que deseja transmitir verifica previamente a ausência de tráfego antes de iniciar qualquer transmissão, com o objectivo de evitar transmissões simultâneas. Quando o canal se encontra como livre, o nó inicia a transmissão. Quando algum nó detecta uma colisão, ele termina a sua transmissão e aguarda um tempo aleatório antes de voltar novamente a transmitir.

No modo *beacon* o coordenador pode operar na rede recorrendo a uma estrutura designada *superframe*, ilustrada na Figura 14, utilizada para garantir o acesso ao canal. O *superframe* é iniciado com um *beacon* e é seguido por 16 intervalos de tempos iguais. Os nove intervalos iniciais (designados por *CAP-Contention Access Period*) que podem ser utilizados por algum dispositivo, e os sete intervalos seguintes (chamados de *GTS-Guaranteed Time Slots*) são reservados e podem ser alocados por um dispositivo através de um pedido.

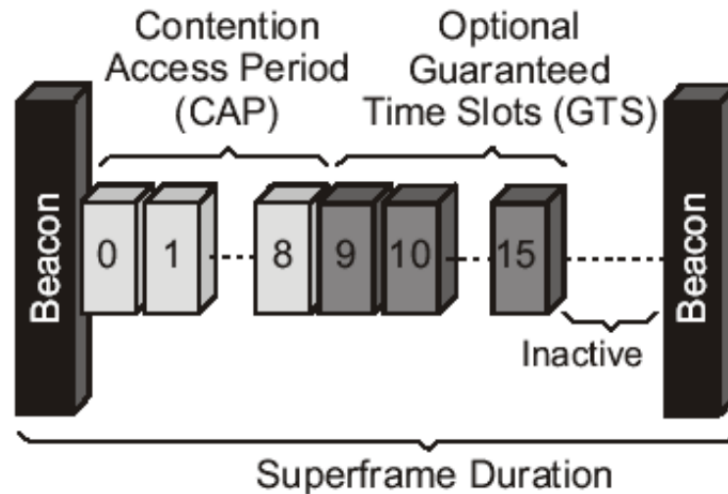


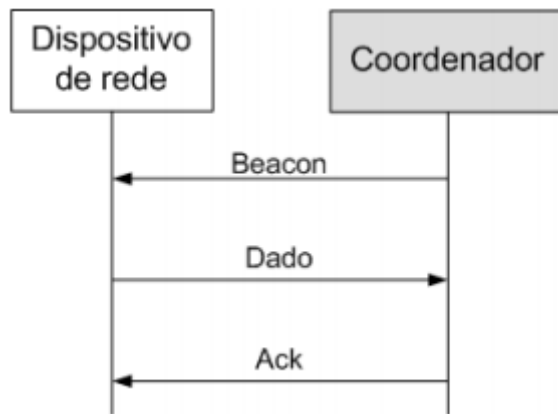
Figura 14 Estrutura *Superframe* [11]

Os *beacons* não são suportados nas redes *ZigBee* com topologia malha, logo não podem ser usados com método de sincronização nessas mesmas redes. Nas redes com topologia árvore os *beacons* podem ser utilizados. Os *beacons* nos nós filhos são atrasados, e a sua retransmissão a partir dos nós filhos é feita com uma compensação no tempo de transmissão.

Uma rede suportar ou não a transmissão de *beacons* afecta directamente o mecanismo utilizado para a transferência de dados. Dispositivos com baixa latência utilizam redes com *beacon* habilitado, se não necessitarem deste tipo de dispositivos opta-se por desabilitar o *beacon*.

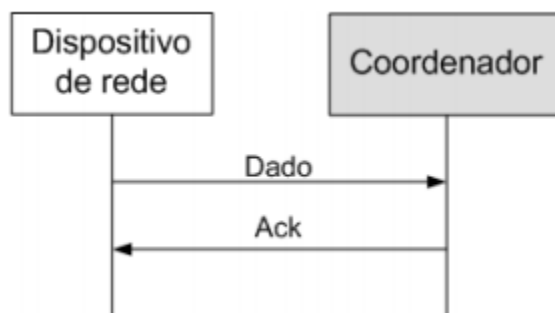
Existem dois modos de transferência de dados sendo, transmissão de dados directa e transmissão de dados indirecta [5] :

- Transmissão de dados directa: Quando um dispositivo pretende transmitir informação ao coordenador, numa rede com *beacon* habilitado, ele primeiro aguarda pela recepção de um *beacon* da rede, como é possível visualizar na Figura 15. O dispositivo efectua a sincronização com a estrutura *superframe* quando recebe o *beacon*. Estando sincronizado o dispositivo consegue transmitir a trama de dados ao coordenador. Por sua vez, o coordenador quando recebe a informação com sucesso envia uma trama de confirmação.



**Figura 15** Transmissão directa de dados em redes *beacon* habilitado [11]

Nas redes com *beacon* não habilitado, quando um dispositivo precisa de transferir informação basta apenas transmitir a informação ao coordenador e esperar pela trama de confirmação do coordenador, como ilustrado na Figura 16.



**Figura 16** Transmissão directa de dados em redes com *beacon* não habilitado [11]

- Transmissão de dados indirecta: Este modo de transferência de dados é utilizado para transferir dados do coordenador para o dispositivo. O dispositivo escuta ciclicamente o *beacon* da rede, se ele receber um *beacon* do coordenador indicando que existe uma mensagem pendente isso significa que o coordenador lhe quer enviar informação. Sendo assim, o dispositivo envia um comando MAC (*Medium Access Control*) pedindo os dados, o coordenador envia uma trama de confirmação para alertar que recebeu o pedido de dados. Posto isto o coordenador envia os dados pendentes, o dispositivo envia uma trama de confirmação dos mesmos e após receber a confirmação a mensagem é removida da lista de mensagens pendentes. Esta sequência é mostrada na Figura 17.

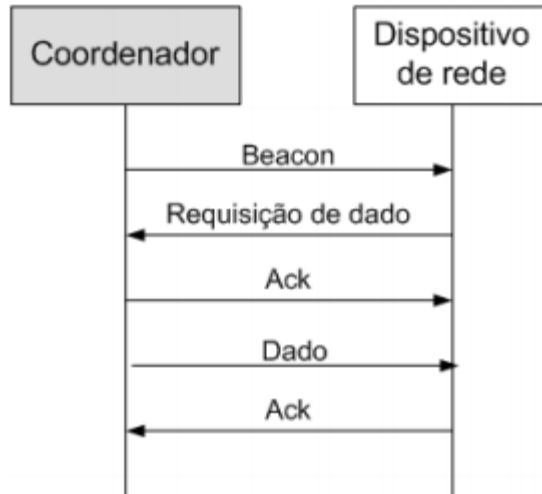


Figura 17 Transmissão indirecta de dados em redes com *beacon* habilitado [11]

Numa rede com *beacon* não habilitado, a informação pendente é armazenada até que o dispositivo apropriado entre em contacto e faça o pedido da informação. O dispositivo envia um comando MAC pedindo informação ao coordenador, a uma taxa de *pooling*. O coordenador confirma a recepção do pedido de dados enviando uma trama de confirmação. Se existem dados pendentes, o coordenador transmite-os para o dispositivo. Se não existem dados pendentes, o coordenador transmite uma trama de dados com cabeçalho de comprimento zero, para indicar que não existem dados pendentes. O dispositivo confirma que recebeu efectivamente os dados enviando uma trama de confirmação. Estas operações são mostradas na Figura 18.

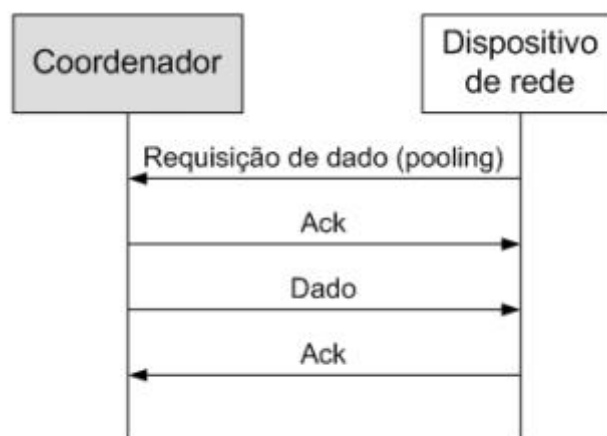


Figura 18 Transmissão indirecta de dados numa rede com *beacon* não habilitado [11]

Como é possível verificar, em ambas as transmissões de dados existe a possibilidade de *beacon* habilitado ou não habilitado. A grande diferença em utilizar um método ou outro reside na coordenação.

Quando o *beacon* está habilitado, estamos perante um modo totalmente coordenado, ou seja todos os dispositivos sabem quando se ligar e comunicar uns com os outros. Neste caso, o coordenador da rede “acorda” e envia sinalização (*beacon*) para os dispositivos existentes na rede. Quando um dispositivo recebe um *beacon* tem de verificar se tem alguma informação para enviar. Se não tiver, o dispositivo volta a “dormir”. O facto da comunicação ocorrer apenas quando pedida, faz com que neste modo o consumo de energia seja o mínimo possível.

No modo em que o *beacon* não está habilitado, as comunicações e transferências de informação entre dispositivos da rede e coordenador são muito menos coordenadas uma vez que os dispositivos podem comunicar sem qualquer restrição com o coordenador. Isto pode causar interferências entre diferentes dispositivos e obriga o coordenador a estar permanentemente acordado para procurar sinais, tornando o consumo de energia maior [5].

### **3.4. TRANSMISSÃO DE DADOS**

No protocolo *ZigBee* podemos classificar a transmissão de dados em três tipos: *broadcast*, *multicast* e *unicast* [5].

A mensagem *broadcast* é destinada a todos os dispositivos que recebam a mensagem, ou seja, a todos os dispositivos que estejam ligados ao mesmo canal de frequência, independentemente do seu endereço ou indentificador na rede. Quando um dispositivo recebe um pacote de dados, analisa o endereço de destino do pacote recebido para apurar se é ele o destinatário da informação.

O método *multicast* envia informação para um determinado grupo de dispositivos na mesma rede, em vez de toda a rede. Cada grupo de dispositivos é identificado por um ID de grupo de 16 bits. Dispositivos pertencentes ao mesmo grupo são designados de *group members*. Um dispositivo não é exclusivo de um grupo, pode pertencer a vários grupos, a lista dos grupos *multicast* aos quais pertence é registado na chamada tabela *multicast*. Um dispositivo não necessita de ser membro de um grupo *multicast* para ser capaz de usar *multicasting* para aceder a um membro. Existem dois modos de operação *multicasting*:

*member* e *nonmember*. No modo *member*, uma transmissão *multicast* é iniciada por um dispositivo pertencente ao grupo e enviada para os restantes membros. No modo *nonmember*, a mensagem pode ser enviada, indirectamente, por um dispositivo que não pertença ao grupo *multicast*. O dispositivo que não pertence ao grupo *multicast* envia a mensagem para um dispositivo que pertença ao grupo a partir do qual irá ser feita a transmissão para todos os dispositivos do grupo.

As mensagens *unicast* são enviadas directamente para um só dispositivo na rede, dado que a mensagem contém o endereço específico do dispositivo.

Na Figura 19 encontram-se ilustrados os três métodos de comunicação mencionados anteriormente.

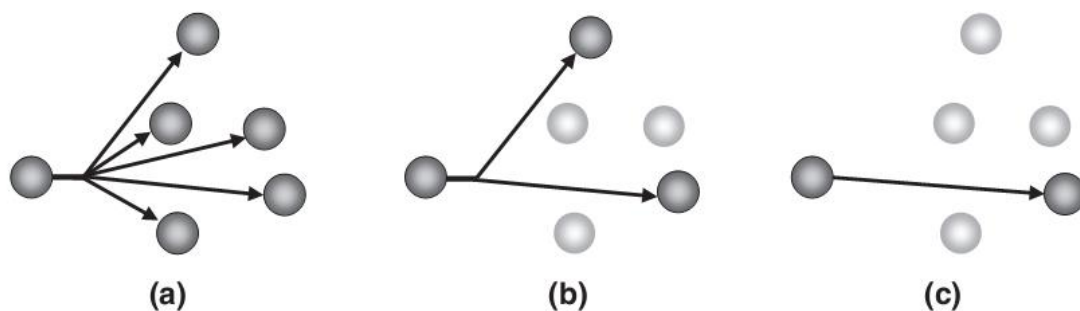


Figura 19 (a) *Broadcast*, (b) *Multicast*, (c) *Unicast* [5]

### 3.5. ENDEREÇAMENTO

Os dispositivos *ZigBee* podem ser identificados na rede graças a dois tipos de endereços, um deles é o endereço IEEE, digamos que é o número de série do dispositivo, um identificador único do dispositivo. O outro é o endereço da rede, que lhe é atribuído quando um dispositivo se liga a uma rede *ZigBee* [15].

O endereço IEEE possui 64 bits, é o endereço que torna o dispositivo único assegurando que não existe mais nenhum com o mesmo endereço a nível físico.

O endereço de rede, que contém 16 bits, identifica o dispositivo na rede. Este endereço é definido para uma rede, mas pode existir também noutra rede, ou seja, em redes diferentes podem existir dispositivos com endereços de rede iguais. O nó pai (*router* ou coordenador) atribui um endereço de rede a um nó filho que é adicionado à rede. O único dispositivo que

tem endereço de rede predefinido é o coordenador (0x0000), os restantes dispositivos podem ter qualquer um dos endereços que se encontrem disponíveis.

A maioria das trocas de informação que se realizam na rede é feita recorrendo ao endereço de rede dos dispositivos transmissores e receptores. Como estes endereços apenas possuem 16 bits, as mensagens que circulam na rede têm um tamanho menor, sendo utilizado menos memória na gravação dos endereços de rede [15].

### **3.6. SERVIÇOS DE SEGURANÇA**

O protocolo *ZigBee* fornece diversos serviços de segurança que se tornam uma mais-valia desta tecnologia. Os mecanismos de segurança utilizados são os do IEEE 802.15.4 que distingue os seguintes, *Advanced Encryption Standard* (AES) e o *Counter mode and Cipher block chaining Message authentication code* (CCM) [45][16].

Os mecanismos de segurança asseguram ao nível da infra-estrutura assim como dos dados da aplicação. Em termos de infra-estrutura é realizado o controlo de acesso à rede, são sinalizados os dispositivos de confiança, é garantida a plenitude dos pacotes enviados e prevenção de transporte não autorizado. Ao nível dos dados de aplicação existem quatro mecanismos de segurança: *freshness*, integridade da mensagem, autenticação e encriptação.

O mecanismo *freshness* não é mais do que rejeitar dados que sejam duplicados. O dispositivo rejeita a trama recebida se o valor de *freshness* não tiver mudado em relação à última comunicação.

A integridade da mensagem garante que uma mensagem não foi modificada durante o transporte.

A autenticação executa a confirmação de que o emissor da mensagem é seguro. Esta autenticação pode ser feita a nível da rede (para ataques externos à rede) ou ao nível do dispositivo (para ataques internos à rede).

A encriptação assegura que qualquer dispositivo na rede a qual a mensagem não é destinada, não consegue percebê-la. Tal como a autenticação, existe encriptação ao nível da rede e do dispositivo.

### 3.7. CAMADAS DO PROTOCOLO ZIGBEE

O protocolo *ZigBee* é implementado com base no padrão IEEE 802.15.4 que está encarregue da Camada Física (PHY - *Physical*) e de acesso ao meio (MAC) em *Personal Area Networks* de baixo custo e baixas taxas de transmissão. O *ZigBee* define a Camada de Rede (NWK - *Network*) e de Aplicação (APL). Na Figura 20 é possível visualizar a pilha protocolar *ZigBee*.

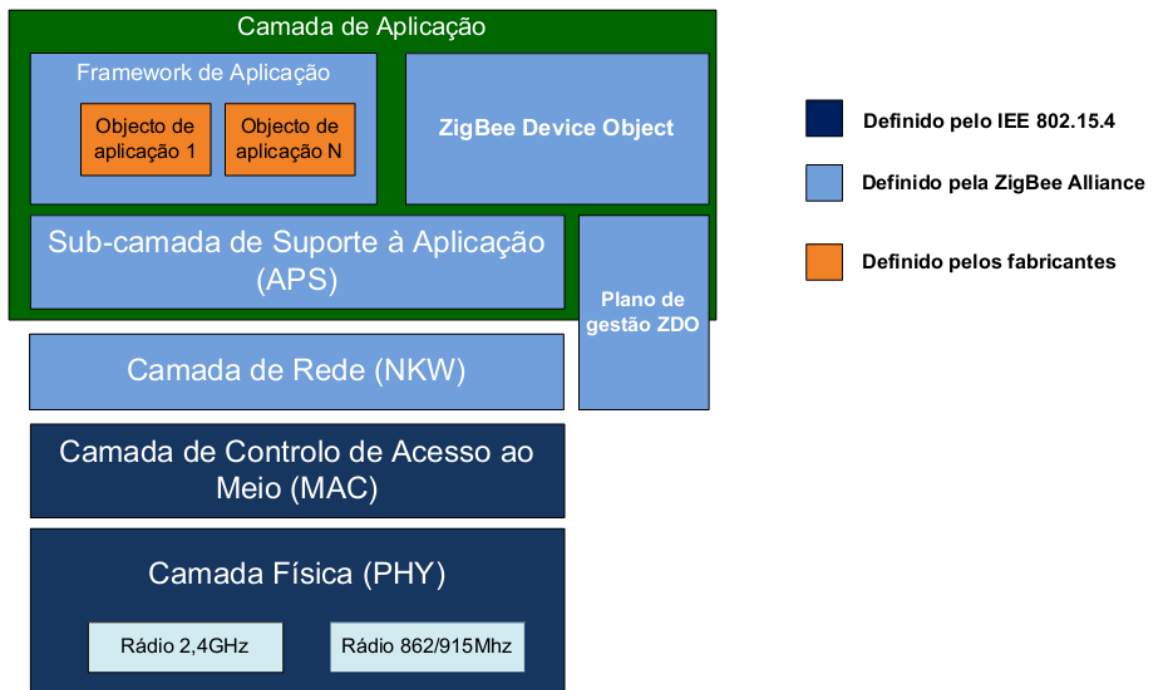


Figura 20 Pilha protocolar *ZigBee* [8]

#### 3.7.1. CAMADA FÍSICA

A camada física é a mais baixa da pilha protocolar. Esta camada codifica os bits que são enviados e descodifica os bits recebidos, escolhe o canal e verifica que não está a ser utilizado por outros dispositivos na rede, pela detecção de energia (ED – *Energy Detection*) e pela indicação da qualidade de ligação (LQI – *Link Quality Indicator*).

Sempre que um dispositivo quer transmitir uma mensagem, primeiro ele entra em modo recepção para detectar e estimar o nível de energia do canal para o qual quer transmitir. Este processo é designado por *energy detection*.

A camada física fornece o indicador de qualidade do *link* que assinala a qualidade do sinal recebido. A intensidade do sinal recebido pode ser utilizada como medida da qualidade do sinal, é a medida da energia total do sinal recebido. Outro método para avaliar a qualidade do sinal é a relação sinal/ruído. A combinação destes dois também permite verificar a qualidade do sinal. O LQI é passado para a camada MAC e disponibilizado para camadas acima na rede. Pode dizer-se que o trajecto que tenha maior LQI tem mais probabilidade de sucesso na entrega da mensagem [5].

### **3.7.2. CAMADA MAC**

A camada MAC proporciona interface entre a camada física e a camada de rede. Esta camada é responsável por gerar mensagens com um formato específico, *beacons*, utilizadas para efectuar sincronismo dos nós da rede. Esta camada consegue também controlar o acesso aos canais de radiofrequência, utilizando o mecanismo de prevenção de colisão CSMA. O tipo de dispositivos e a estrutura de tramas aceites na rede são especificados nesta camada [5].

### **3.7.3. CAMADA DE REDE**

A camada de rede é, hierarquicamente, a primeira camada que é definida pela norma *ZigBee*. Esta camada ocupa-se do nível rede relativo a comunicação. Examina a estrutura de rede e trata do roteamento e das funções de segurança das mensagens enviadas. Esta camada possui mecanismos para incorporar e decompor redes, sendo ela quem controla o início e o fim de ligação de cada dispositivo na rede. Esta camada pode também descobrir novos dispositivos, registando a informação relativa aos mesmos [5].

Como é possível visualizar na Figura 21 a camada de rede tem duas entidades que fornecem serviços à camada de aplicação (APL). O *Network Layer Data Entity* (NLDE) realiza a transmissão de dados, para utilizar este serviço é usado o ponto de acesso NLDE *Service Access Point* (SAP). Os serviços de gestão da camada de rede são tratados pelo *Network Layer Management Entity* (NLME). O NLME-SAP é utilizado para aceder a esta entidade. Esta camada tem as suas próprias constantes e atributos, identificadas por *nwkc* e por *nwk*, respectivamente. Os atributos da camada de rede são guardados na *Network Information Base* (NIB).

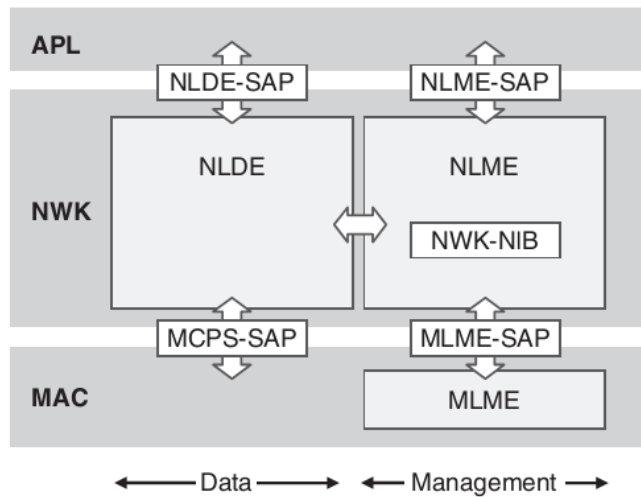


Figura 21 A camada de rede ZigBee e a sua interação com a MAC e APL [5]

### 3.7.4. CAMADA DE APLICAÇÃO

A camada de aplicação, última camada da pilha protocolar ZigBee, está destinada apoiar a aplicação cliente/fabricante dentro do dispositivo ZigBee. Está dividida em três secções, como é possível visualizar na Figura 22: *application support sublayer* (APS), *ZigBee Device Objects* (ZDO) e *application framework*.

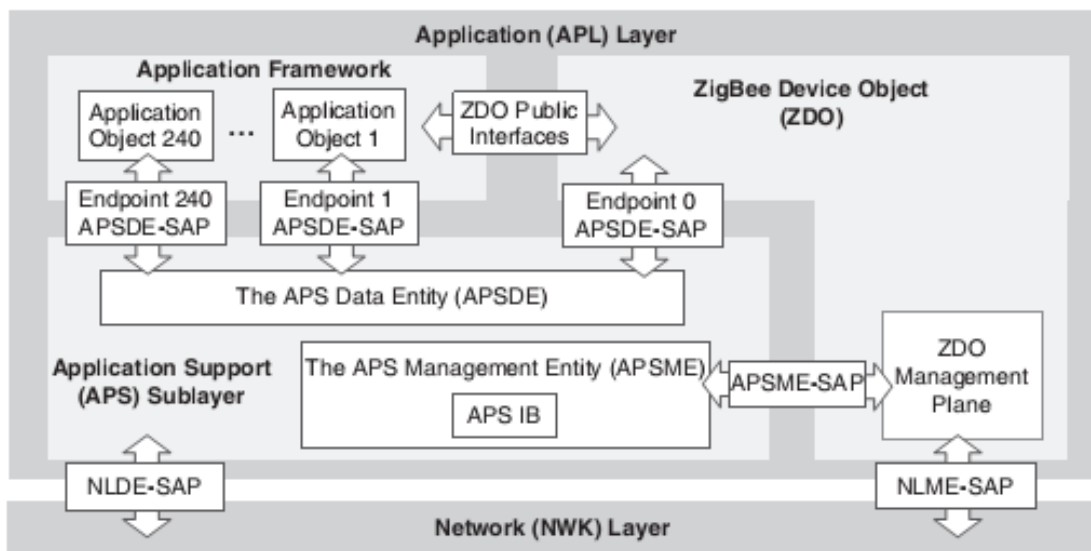


Figura 22 Camada de aplicação [5]

O *application framework* é o onde os objectos de aplicação são guardados para controlar e administrar as camadas do protocolo num dispositivo ZigBee. Os objectos de aplicação são criados pelos fabricantes, sendo personalizados para várias aplicações. Pode existir até 240

objectos de aplicação de um só dispositivo. Os objectos de aplicação recorrem ao ponto de acesso APSDE-SAP para troca de dados entre eles.

Outra das secções da camada de aplicação é a *ZigBee Device Objects*. Funciona como interface entre a subcamada APS e a *application framework* e é responsável pelos recursos de gestão da rede, tais como, gestão de dispositivos, chaves de segurança e estratégias de segurança.

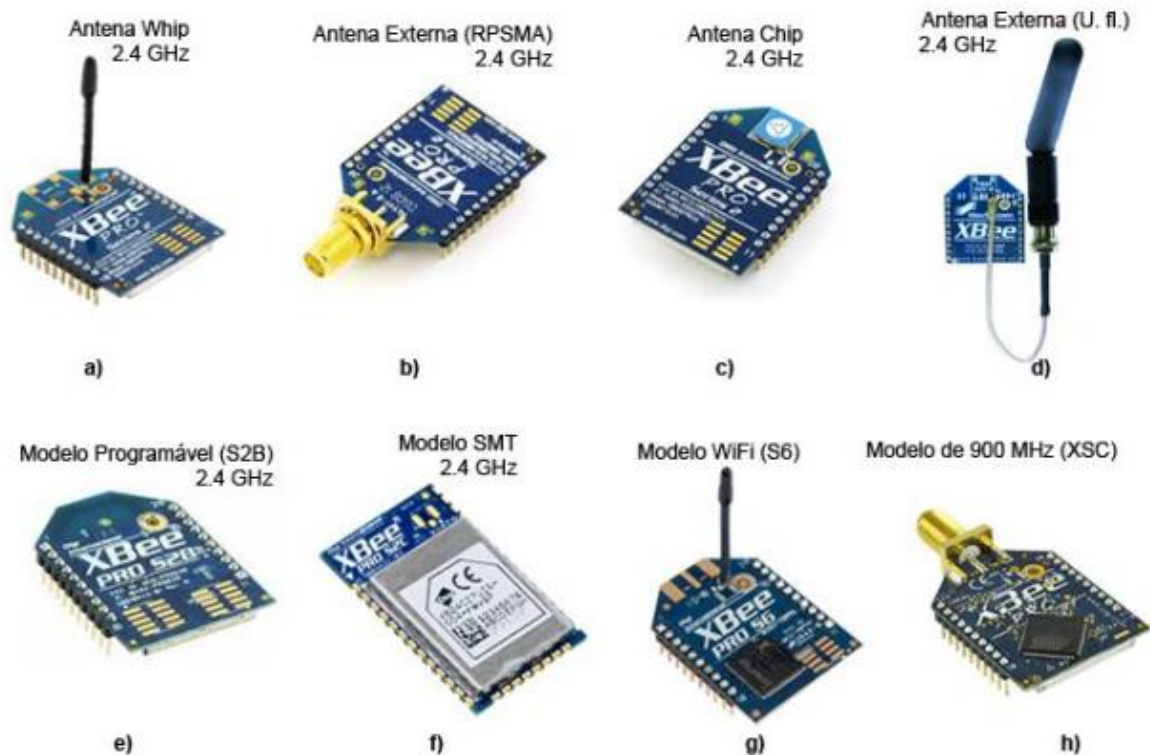
O terceiro componente presente na camada de aplicação é a subcamada de suporte à aplicação APS. Ela trabalha como interface entre os serviços facultados pela camada de rede e a camada de aplicação, que realiza, a manutenção das tabelas de roteamento, encaminhamento de mensagens entre dispositivos, gerenciamento endereços, mapeamento do endereço de 64 bits no endereço de rede (16 bits) e auxílio na realização do transporte de dados [5].

### **3.8. DISPOSITIVOS XBEE**

Actualmente existem vários dispositivos de *hardware* baseados no protocolo *ZigBee*, um modelo muito conhecido e bastante utilizado é o *XBee*, fabricado pela líder de mercado *Digi International*.

Os módulos *XBee* são constituídos por um microcontrolador e um transceptor. O microcontrolador possui o *firmware* com a implementação do protocolo *ZigBee* e enumeração do comportamento do dispositivo (*Coordinator, Router, End Device*). Cada dispositivo tem dois endereços, o MY de 16 bits e o Número de Série de 64 bits. O MY é o endereço de rede, incerto, é atribuído de forma automática pelo coordenador assim que o nó se liga à rede. Por sua vez o número de série é único para cada dispositivo produzido.

Os módulos *XBee* são fabricados em diversos tipos, variando de acordo com o modelo de antena, encapsulamento, frequência de operação e protocolo utilizado. Na Figura 23 são ilustrados os principais modelos.



**Figura 23 Principais modelos de Módulo XBee da Digi International [46]**

Como é possível observar existe um grande leque de soluções de módulos XBee existentes no mercado, logo, antes de escolher um modelo é necessário analisar algumas características essenciais do mesmo como frequência de operação, taxa de transmissão, alcance, tipo de antena, potência de transmissão e consumo energético [46] :

- Frequência de operação e taxa de transmissão – Existem modelos que funcionam na frequência de 2,4 GHz e outros na frequência de 915 e 868 MHz. Quanto mais baixa for a frequência de operação maior será a taxa de transmissão, mas por outro lado, menor será o alcance.
- Alcance e Antena – O alcance do dispositivo e o tipo de antena que utilizam está directamente ligado. Dispositivos com antenas *on chip* (c) têm um alcance menor, porém o espaço físico que ocupam é menor. Dispositivos com antenas externas (d) possibilitam um alcance maior, porém o espaço físico que ocupam é grande. Os dispositivos com antena *Whip* (a) tentam ser o meio-termo dos referidos anteriormente, uma vez que conseguem um alcance intermédio e ocupam um espaço físico intermédio.

- Potência de transmissão e Consumo Energético – Os módulos *XBee* são fabricados em duas versões, a versão *XBee PRO* e a *XBee*. A grande diferença entre estas duas reside na potência de transmissão, no modelo *PRO* essa potência é maior. Consequentemente, o consumo de potência dos modelos *XBee* é menor em relação ao *XBee PRO*.

### 3.8.1. ARQUITECTURA

Os módulos *XBee*/*XBee-PRO* utilizam os dispositivos da *freescale* (transceptor e microcontrolador). A arquitectura dos módulos *XBee* é ilustrada na Figura 24.

Os módulos *XBee* internamente possuem três blocos, o processador, o transceptor e um *switch* RF. O processador (MC9S08GT60) funciona como interface com um *host*, é através dele que se consegue realizar configurações, recepção e envio de dados. O transceptor (MC13193) realiza a interface rádio a 2,4 GHz. Por último, temos o interruptor RF que permite variar entre o modo *receive* e *transmit*. Quando é necessário o módulo transmitir a ligação da antena é direccionada para o *transmit* do transceptor. Depois de efectuar a transmissão o transceptor coloca novamente a antena ao *receive*, ficando aguardar a recepção de pacotes *ZigBee* [16].

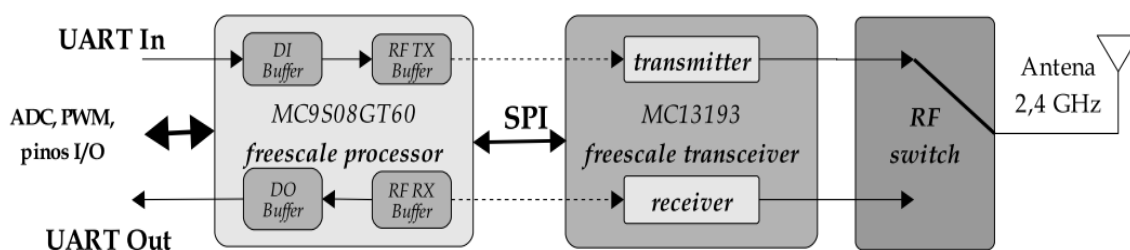


Figura 24 Arquitectura dos módulos *XBee* [16]

### 3.8.2. FIRMWARE

Cada módulo *Xbee* possui um microcontrolador interno, que executa um programa, conhecido como *firmware*. São distribuídos quatro *firmwares* *XBee* 802.15.4, *XBee* ZNet 2.5, *XBee* ZB, DigiMesh 2.4 [4].

- *XBee* 802.15.4 – Primeiro *firmware* utilizado com o módulo *XBee* série 1. Este *firmware* permite apenas passar informação de um só nó para outro, com resposta entre eles. A rede assente neste *firmware* consegue apenas comunicação até ao

alcance máximo de um nó individual, ou seja, não existe saltos ou encaminhamento, as mensagens não conseguem passar por múltiplos nós.

- *XBee ZNet 2.5* e *XBee ZB* – O *XBee ZNet 2.5* é um ajuste do protocolo *ZigBee* efectuado pela DIGI, e *XBee ZB* é a implementação do próprio protocolo podendo ser utilizado por rádios de outros fabricantes que suportem o protocolo *ZigBee*. Um e outro possibilitam o encaminhamento em malha, fazendo com que os pacotes possam passar por diversos nós até chegarem ao seu destino.
- DigiMesh 2.4 – Este *firmware*, implementa também uma topologia em malha, mas permite que todos os nós adormeçam ao mesmo tempo. O processo de sincronismo altera o desempenho da rede, a velocidade de reencaminhamento baixa, mas possibilita que técnicas de poupança de energia sejam utilizadas em todos os nós.

### **3.8.3. MODOS DE OPERAÇÃO DO XBEE**

Os módulos *XBee*, com qualquer dos *firmwares*, podem ser utilizados em dois modos diferentes: modo Transparente AT e no modo API (*Application Programming Interface*). Os possíveis modos são seleccionados previamente [4].

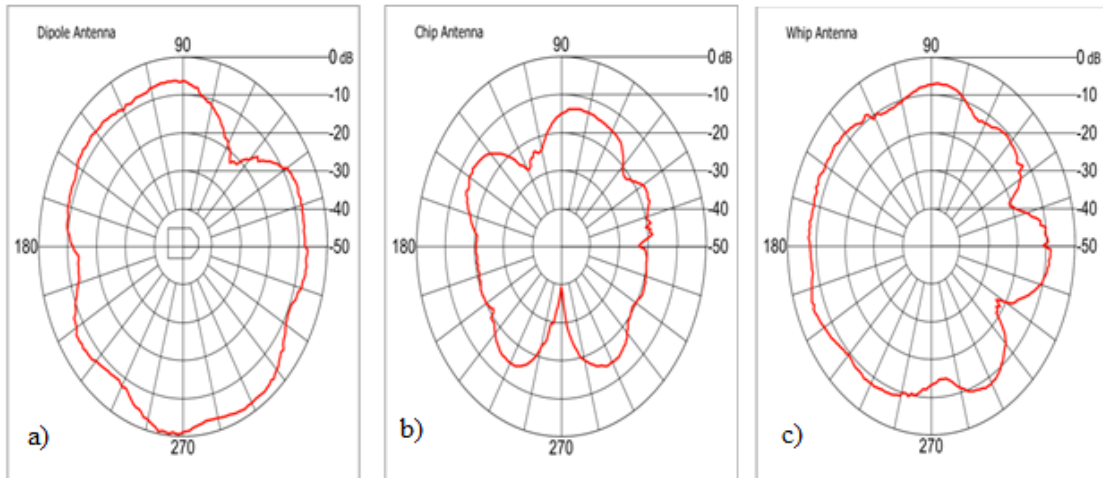
No modo transparente a interacção entre *XBee* e Microcontrolador/Computador é feita de uma forma directa, este modo é o que vem por defeito em todos os módulos. O modo de funcionamento é simples, uma vez que os dispositivos transmitem directamente para a porta série toda a informação recebida. Os dados transmitidos não sofrem alteração em relação à sua forma original, porém, oferecem um baixo nível de robustez, sendo um método ineficaz de comunicar com computadores. Neste modo as configurações são realizadas através de comandos AT.

No modo API a informação que circula pela rede é interpretada e verificada. Existe uma estrutura de trama que permite realizar configurações, enviar e receber dados. Na Figura 25 encontra-se ilustrada a estrutura de uma mensagem API.



### 3.8.4. RADIAÇÃO DAS ANTENAS

Os dispositivos *XBee* podem possuir diferentes tipos de antenas. Consoante o tipo de antena o padrão de radiação pode variar. Na Figura 27, são apresentados os gráficos do padrão de radiação para antenas do tipo dipolo, *Chip* e *Whip*.



**Figura 27 Radiação de Antenas de diferentes módulos *XBee*. a)Antena Dipolo b) *Chip* Antena c) *Whip* Antena [48]**

Das três radiações de antena apresentadas, aquela que é mais omnidireccional é a antena *whip*, isto significa, emite igualmente em todas as direcções. Ou seja, o posicionamento do módulo é indiferente durante a sua utilização.



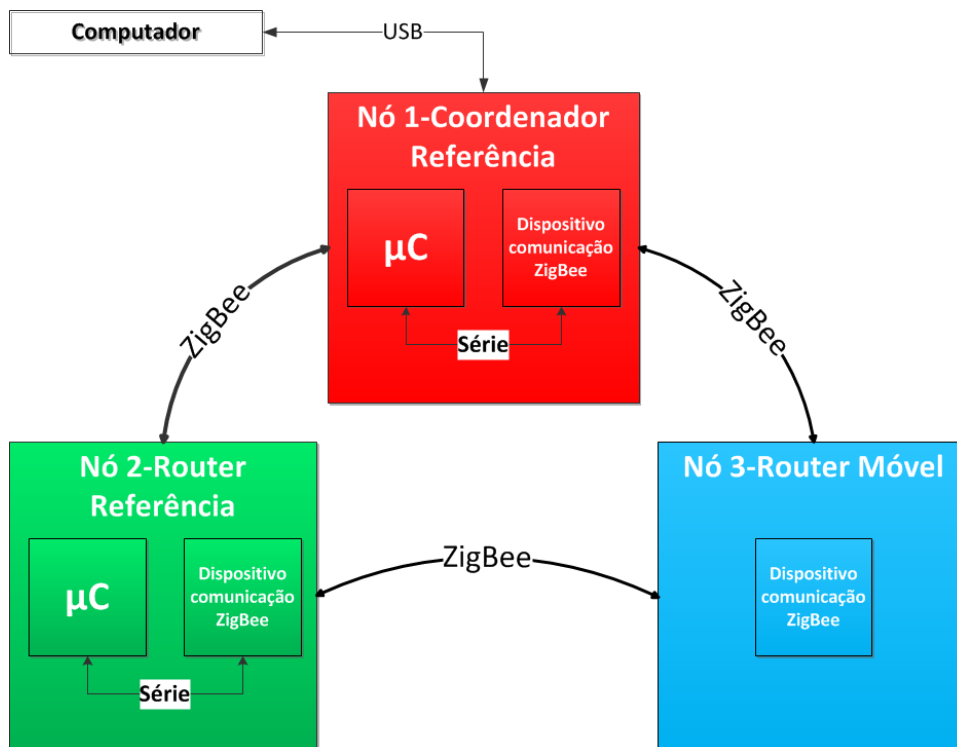
# 4. DESENVOLVIMENTO DO PROJECTO

O principal objectivo deste projecto é o desenvolvimento de um sistema localização *indoor* recorrendo ao protocolo *ZigBee*. Este capítulo pretende demonstrar o desenvolvimento prático de todo o sistema.

## 4.1. ARQUITECTURA DO SISTEMA

O sistema de localização sem fios é desenvolvido tendo em conta a estrutura das RSSF, isto é, os nós constituintes da rede são compostos por um microcontrolador, um dispositivo de comunicação sem fios, uma fonte de alimentação e um sensor. Neste caso não é necessário os nós possuírem sensor, uma vez que a variável a ser avaliada é disponibilizada pelo próprio dispositivo de comunicação. Os nós sensores pertencentes à rede irão comunicar entre si utilizando o protocolo *ZigBee*.

Como é possível observar na Figura 28, o sistema é composto por um coordenador de referência (Nó 1), um *router* que funciona como nó de referência (Nó 2) e um *router* que funciona como nó móvel (Nó 3).



**Figura 28** Arquitectura do sistema

A rede é composta por três tipos de nós:

- Coordenador de referência (Nó 1) – Nó composto por um dispositivo *ZigBee* e ligado ao computador via USB através do módulo *Arduino*. Este é o nó principal da rede, é ele que coordena toda a informação da rede e executa os pedidos de localização. Para além destas tarefas, funciona também como um dos nós de referência do sistema.
- *Router* de referência (Nó 2) – Este nó é composto por um dispositivo *ZigBee* em modo *router*. Funciona também como nó de referência, situando-se por isso numa posição de referência. Este módulo efectua pedidos de RSSI a um nó móvel e reencaminha esse valor para o coordenador.
- *Router* Móvel (Nó 3) – Nó composto por um dispositivo *ZigBee*, apenas responde aos pedidos de RSSI.

#### 4.2. *HARDWARE* UTILIZADO PARA DESENVOLVIMENTO DO PROJECTO

Nesta secção serão apresentadas as escolhas realizadas, em termos de *hardware*, para os constituintes dos nós sensores da rede.

Em termos de dispositivo de comunicação, a escolha recaiu sobre o *XBee Series 2* da *Digi International*, com antena *Whip*, ilustrado na Figura 29.

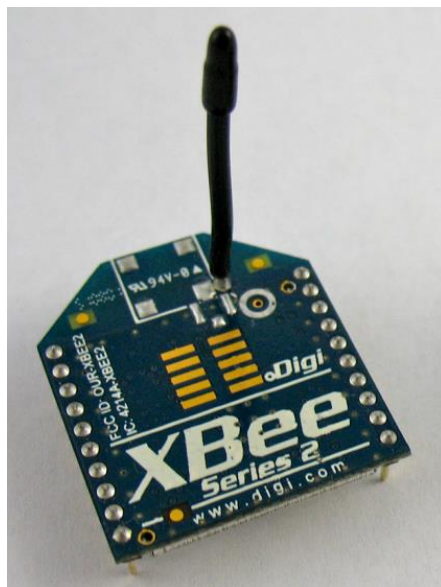


Figura 29 Dispositivo *XBee Series 2* com antena *Whip*

Após análise das principais características do módulo *XBee*, apresentadas na Tabela 2, chegou-se à conclusão que este módulo seria o indicado para a realização deste sistema.

Tabela 2 Especificações do módulo *XBee Series 2*

<b>Alcance Indoor</b>	40 m
<b>Alcance Outdoor</b>	120 m
<b>Potência de transmissão</b>	2 mW
<b>Taxa de transmissão</b>	250000 bps
<b>Tensão de alimentação</b>	2,8 – 3,4 V
<b>Corrente de transmissão</b>	40 mA (alimentado por 3,3 V)
<b>Corrente de recepção</b>	40 mA (alimentado por 3,3 V)
<b>Frequência</b>	2,4 GHz
<b>Topologias de rede suportadas</b>	Ponto a ponto, Estrela, Malha

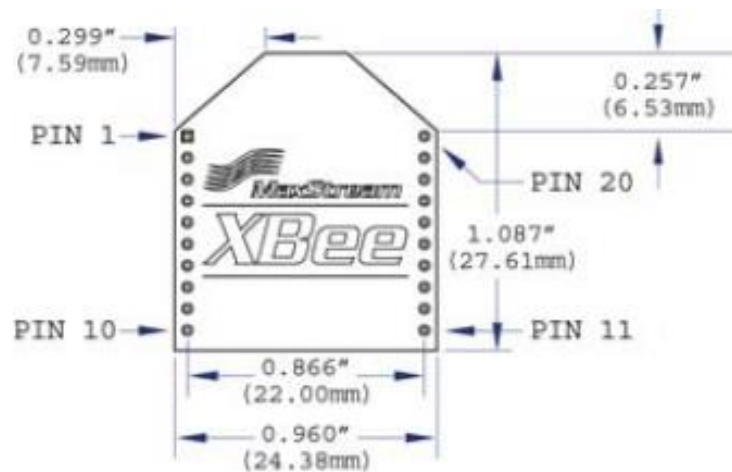
Em termos de alcance *indoor*, tem um alcance máximo de 40 m, valor mais do que suficiente uma vez que os dispositivos nunca serão colocados a distâncias tão grandes. O dispositivo possui valores de consumo energéticos baixos, aspecto cada vez mais importante nos sistemas de hoje em dia. Este módulo requer uma energia mínima para funcionamento, uma tensão de alimentação de 3,3 V e uma corrente de 40 mA. O protocolo de comunicação sem fios que utiliza é obviamente o *ZigBee*. O facto de o módulo suportar todas as topologias de rede *ZigBee* é também importante. Outro ponto importante é a antena do tipo *whip* emitir de forma mais omnidireccional que outros tipos, como analisado em 3.8.4. A sua disponibilidade no laboratório GECAD (Grupo de Investigação em Engenharia do Conhecimento e Apoio à Decisão) e o facto de não existirem muitos estudos acerca deste dispositivo, foram outros dos motivos que levaram à sua escolha. Todas as especificações referentes ao *XBee Series 2* encontram-se no Anexo A.

O módulo *XBee* possui 20 pinos, cada um deles com a sua funcionalidade, apresentadas na Tabela 3.

**Tabela 3 Funcionalidade dos pinos do módulo *XBee Series 2* [50]**

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	DIO12	Either	Digital I/O 12
5	RESET	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI / DIO10	Either	PWM Output 0 / RX Signal Strength Indicator / Digital IO
7	PWM / DIO11	Either	Digital I/O 11
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DIO8	Either	Pin Sleep Control Line or Digital IO 8
10	GND	-	Ground
11	DIO4	Either	Digital I/O 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP / DIO9	Output	Module Status Indicator or Digital I/O 9
14	[reserved]	-	Do not connect
15	Associate / DIO5	Either	Associated Indicator, Digital I/O 5
16	RTS / DIO6	Either	Request-to-Send Flow Control, Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0 / Commissioning Button	Either	Analog Input 0, Digital IO 0, or Commissioning Button

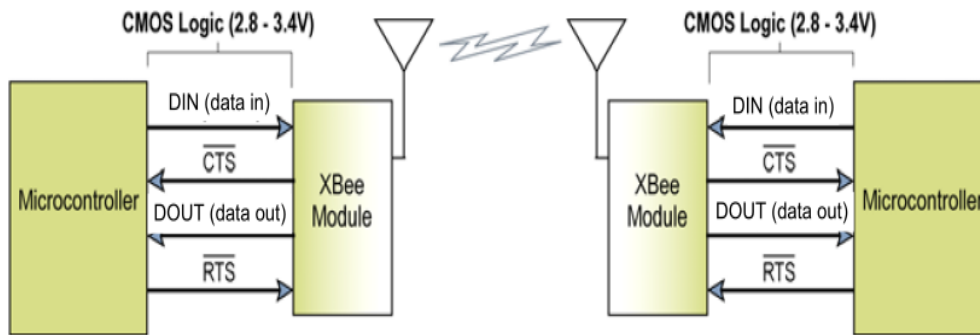
Na Figura 30 encontra-se ilustrado o diagrama de pinos do módulo *XBee Series 2*.



**Figura 30** Diagrama de Pinos do Módulo *XBee Series 2* [50]

Para colocar o módulo *XBee* em funcionamento é necessária a ligação de quatro pinos: VCC, GND, DOUT e DIN. Os pinos VCC e GND são utilizados para alimentação do módulo. Os pinos DOUT e DIN são responsáveis pela saída e entrada de dados através da UART, respectivamente. Os restantes pinos possuem mais de uma função que é determinada através de *software*.

Todos os microcontroladores que possuam interface UART (*Universal Asynchronous Receiver/Transmitter*) podem conectar-se ao controlador conforme demonstra a Figura 31. O módulo UART permite a troca de dados entre módulo *XBee* e microcontrolador. As comunicações dependem de ambas as UART, logo elas têm de ser configuradas com os mesmos parâmetros (taxa de transmissão, paridade, *start* bits, *stop* bits, bits de dados) de maneira a garantir uma correcta troca de dados. Os módulos possuem pequenos *buffers* para recolher os dados série e os dados recebidos por rádio frequência. Quando chegam dados série ao módulo, são armazenados no *buffer* até poderem ser processados. Quando são recebidos dados por rádio frequência estes são transferidos para o *buffer* e enviados através da porta série para o microcontrolador.



**Figura 31** Diagrama de fluxos de dados na interface UART [50]

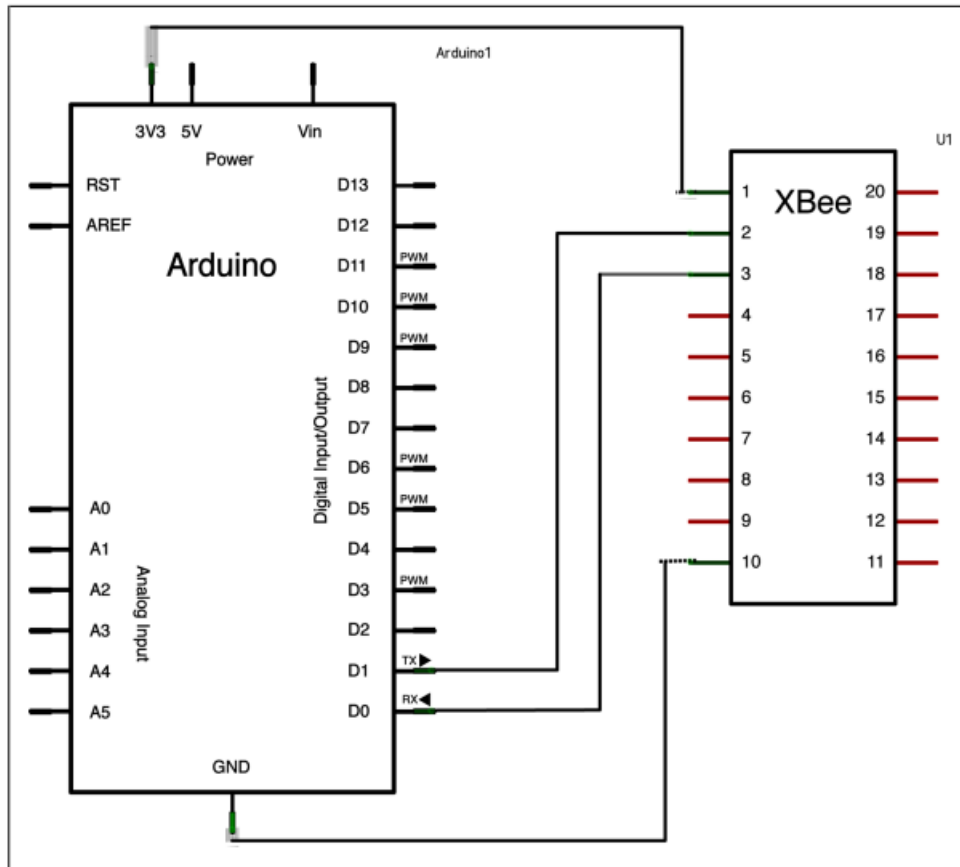
O *Arduino Uno*, ilustrado na Figura 32, foi a plataforma de desenvolvimento utilizada para interação como o *XBee*. A placa consegue importar e exportar conteúdo, a linguagem de programação do *Arduino* é baseada em C/C++. Existem várias versões de modelos *Arduino*, mas todas elas têm como base de funcionamento o microcontrolador ATMEL AVR. No caso do *Arduino uno* o microcontrolador incorporado é o ATmega 328, que possui um cristal com uma frequência de *clock* de 16 MHz.

A plataforma utilizada permite fornecer energia a partir dos seus pinos com uma tensão máxima de 5 V ou 3,3 V e com correntes máximas de 40 mA e 50 mA respectivamente. A tensão de 3,3 V é usada para alimentação do dispositivo *XBee*. Quanto à alimentação da própria placa pode ser feita de duas formas, através de ligação USB ou por fonte de alimentação periférica. Outro aspecto importante é o facto de suportar várias formas de comunicação que permite a troca de dados com o dispositivo *XBee*. Esta plataforma permite o desenvolvimento de várias aplicações sendo um dispositivo bastante flexível.



**Figura 32** *Arduino Uno*

Na Figura 33 encontra-se o esquema de ligações entre *XBee* e *Arduino*, de forma a poderem comunicar através da UART. A alimentação dos nós será realizada através do *Arduino*, isto para os nós de referência (Nó 1 e Nó 2).



**Figura 33** Ligações entre *Arduino* e *ZigBee* [4]

Para o nó móvel (Nó 3) a alimentação terá de ser realizada de outra forma, uma vez que este nó estará em movimento, não é viável estar ligado ao *Arduino*. Este nó será alimentado por uma pilha de 9 V.

A pilha de 9 V não fornece directamente a tensão ao *XBee*, encontra-se ligada a um regulador de tensão que garante os 3,3 V de tensão de alimentação. O regulador de tensão utilizado é o LM117 da *Texas Instruments*, trata-se de um regulador que conseguimos ajustar com a combinação de resistências e condensadores para uma tensão de 3,3 V. Essa montagem encontra-se ilustrada na Figura 34.

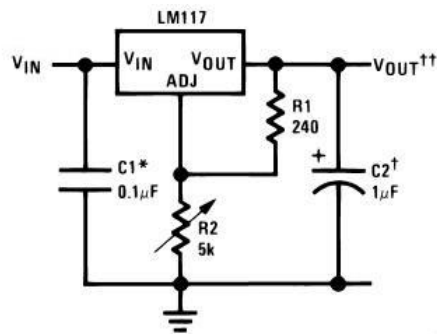


Figura 34 Esquema de montagem do regulador LM117

#### 4.3. SOFTWARE UTILIZADO PARA DESENVOLVIMENTO DO PROJECTO

Para configurar os módulos *XBee* é necessário recorrer ao *software* X-CTU. O X-CTU é o programa oficial, fornecido pela própria *Digi International* para a configuração dos dispositivos. Ele permite configurar os dispositivos a nível de *firmware*, ID da rede, modo de funcionamento, entre outros. Para realizar as configurações, é necessário ter o *Arduino* e o *XBee* ligados através da UART e o computador ligado por USB (*Universal Serial Bus*) ao *Arduino*. Na Figura 35 é possível visualizar o aspecto do *software* utilizado.

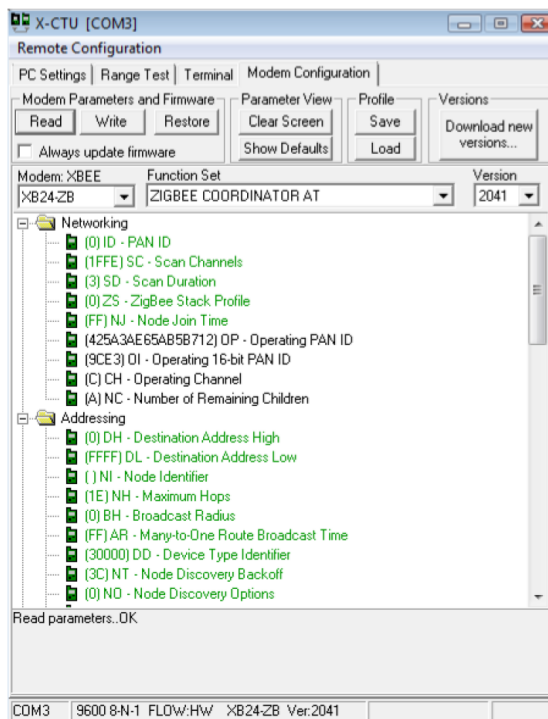


Figura 35 Software X-CTU para configuração dos dispositivos *XBee*

A plataforma *Arduino Uno* é programada com recurso a uma aplicação *open source* designada por *Arduino IDE*, que é fornecida pelo fabricante do *Arduino*.

Para desenvolvimento de interfaces gráficas foi utilizado o software *SharpDevelop*. Este *software* permite o desenvolvimento de programas utilizando a linguagem C#. A aplicação desenvolvida que permite visualizar os valores de RSSI e a posição do *router* móvel foi desenvolvida recorrendo a este *software*. Esta aplicação será apresentada mais a frente na secção 4.4.

#### 4.4. FUNCIONAMENTO DO SISTEMA DESENVOLVIDO

Antes de apresentar qualquer explicação sobre o desenvolvimento do sistema de localização, é necessário explicar como se realiza o processo de localização. Após estudo dos módulos *XBee* foi possível perceber que eles conseguem fornecer o valor RSSI dos sinais que recebem. Sabe-se que é possível com esses valores construir um mapa de RSSI, podendo assim ser utilizada a técnica de localização *fingerprinting*

Esta técnica de localização possui duas fases: A fase *offline* e a fase *online*.

Na fase *offline* é criado o mapa de RSSI. No nosso trabalho cada coordenada do mapa corresponde a dois valores de RSSI, uma vez que temos dois nós referência. Devido a este aspecto o sistema desenvolvido realiza a localização de forma bidimensional.

Na fase *online*, os valores RSSI recebidos de ambos os nós de referência são comparados com os valores gravados no mapa e tratados por um algoritmo de localização para estimar a localização.

Neste trabalho é utilizado o algoritmo K-NN, a base deste algoritmo consiste no cálculo das distâncias entre os valores guardados na base de dados e os valores obtidos na fase *online*. O método para cálculo da distância utilizado neste trabalho foi o das distâncias Euclidianas para estimar a posição do nó móvel.

Por exemplo, se tivermos dois pontos de referência P1 (x,y) e P2 (x,y) guardados com os valores P1 (20,40) e P2 (80,90) e recebermos na fase *online* os valores K (30,20) de um dispositivo móvel. Calculando as distâncias Euclidianas como apresentado na equação 7 e equação 8, obtemos em relação a P1 o valor 22 e em relação a P2 o valor 86, ou seja o dispositivo móvel está mais perto do ponto de referência P1.

$$d_{P1k} = \sqrt{(P_{1x} - K_x)^2 + (P_{1y} - K_y)^2} \quad (7)$$

Onde,  $d_{P1K}$  é a distância Euclidiana K em relação a P1;  $P_{1x}$  é a coordenada x do ponto 1;  $P_{2x}$  é a coordenada y do ponto 1;  $K_x$  é a coordenada x do ponto K ;  $K_y$  é a coordenada y do ponto K.

$$d_{P2K} = \sqrt{(P_{2x} - K_x)^2 + (P_{2y} - K_y)^2} \quad (8)$$

Onde,  $d_{P2K}$  é a distância Euclidiana K em relação a P2;  $P_{2x}$  é a coordenada x do ponto 2;  $P_{2y}$  é a coordenada y do ponto 2;  $K_x$  é a coordenada x do ponto K ;  $K_y$  é a coordenada y do ponto K.

$$d_{P1} = \sqrt{(30 - 20)^2 + (20 - 40)^2} = 22 \quad (9)$$

$$d_{P2} = \sqrt{(30 - 80)^2 + (20 - 90)^2} = 86 \quad (10)$$

O Nó 1 é o dispositivo que controla toda a rede e todo o sistema. Este efectua duas acções: envia o pedido de RSSI ao Nó móvel e envia a ordem ao Nó 2 para este efectuar o pedido de RSSI ao Nó Móvel, tal como ilustrado na Figura 36.

O Nó 2 efectua o pedido de RSSI ao Nó móvel, após receber a ordem enviada pelo Nó 1. De seguida, o Nó 2 após receber resposta do nó móvel reencaminha essa informação para o Nó 1.

A ordem enviada pelo Nó 1 é sinalizada com o caracter “A”, pelo que este não obedece a uma regra podendo ser alterado.

Toda a comunicação entre nós é feita utilizando o protocolo *ZigBee*, sendo as tramas envolvidas nas comunicações apresentadas de uma forma detalhada na secção 4.6.

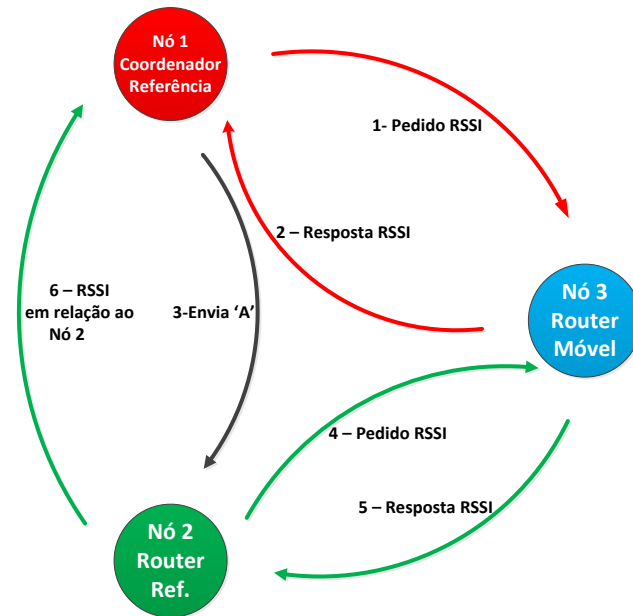


Figura 36 Funcionamento do sistema desenvolvido

#### 4.5. CONFIGURAÇÃO DA REDE

Para que os módulos *XBee* que constituem a rede comuniquem correctamente, têm de ser configurados adequadamente, com recurso ao *software* X-CTU. Desta forma, será explicada a configuração do módulo *XBee* que funciona como coordenador.

Os dispositivos *XBee* vêm configurados de fábrica para funcionarem no modo AT. Para este projecto será utilizado o modo API, onde a informação que circula pela rede é interpretada e verificada. Nesta existe uma estrutura de trama que permita realizar configurações, enviar e receber dados.

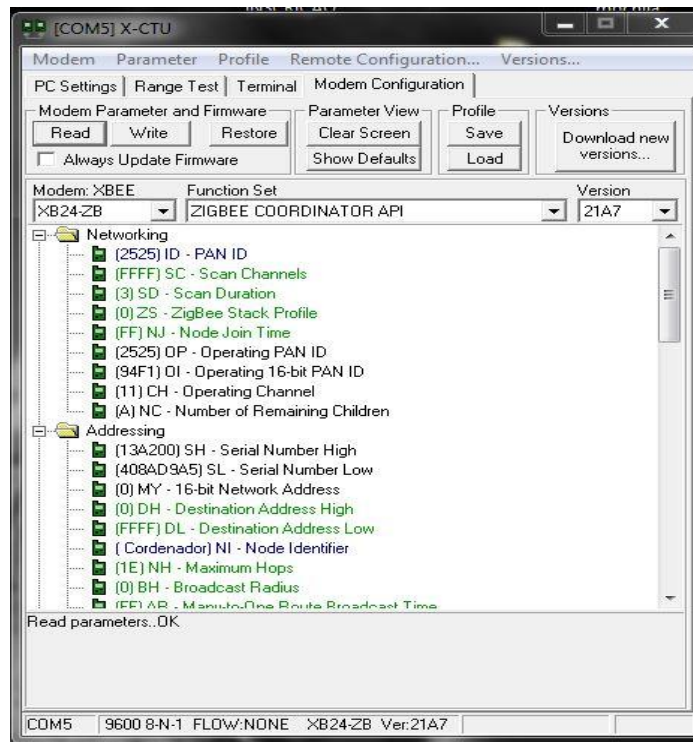
Uma vez que se pretende criar as tramas de informação que circulam na rede, o modo API é o que melhor se adequa. Assim, as primeiras configurações passam pelo modo de operação, *firmware* e tipo de dispositivo. Para este caso concreto, é configurado como coordenador API com a última versão do *firmware XBee ZB*.

Na pasta *networking* configurou-se a rede com a PAN ID igual a 2525, este valor é atribuído a gosto de quem está a desenvolver rede, apenas deve-se ter em atenção que os restantes dispositivos têm de possuir o mesmo PAN ID para pertencerem à mesma rede.

Na pasta *Addressing* existe um campo denominado por *node identifier* que é configurado com um nome definido pelo projectista, facilitando o manuseamento dos nós. Aqui

também é possível saber qual o endereço do dispositivo nos campos *Serial Number High* (SH) e *Serial Number Low* (SL). Refira-se que este é o endereço de série, configurado de fábrica, que torna o dispositivo único, sendo impossível alterá-lo.

As alterações efectuadas são visíveis na Figura 37.



**Figura 37** Janela de operações de configuração

Na Tabela 4 encontram-se listados os parâmetros de configuração de cada nó pertencente à rede.

**Tabela 4** Configurações efectuadas nos três nós da rede

	PAN ID	Node Identifier	Modo Operação	Firmware	SH	SL
<b>Coordenador</b>	2525	Coordenador	Coordenador API	XB24-ZB	13A200	408AD9A5
<b>Router Referência</b>	2525	Router	Router API	XB24-ZB	13A200	40795B4F
<b>Router Móvel</b>	2525	Router 1	Router API	XB24-ZB	13A200	40795AFF

## 4.6. TRAMAS PRINCIPAIS DA REDE

Uma das componentes mais importantes para o desenvolvimento deste protótipo é o valor de RSSI. Este valor é obtido a partir do Nó 1 que envia uma linha de comandos em estrutura API. Antes de se proceder ao envio da mensagem é necessário determinar alguns parâmetros para que a mensagem seja enviada correctamente. Na Figura 38 encontram-se ilustrados os vários tipos de trama possíveis do modo API. Neste trabalho as principais tramas da rede são *Remote Command Request*, *Remote Command Response*, *ZigBee Transmit Request*.

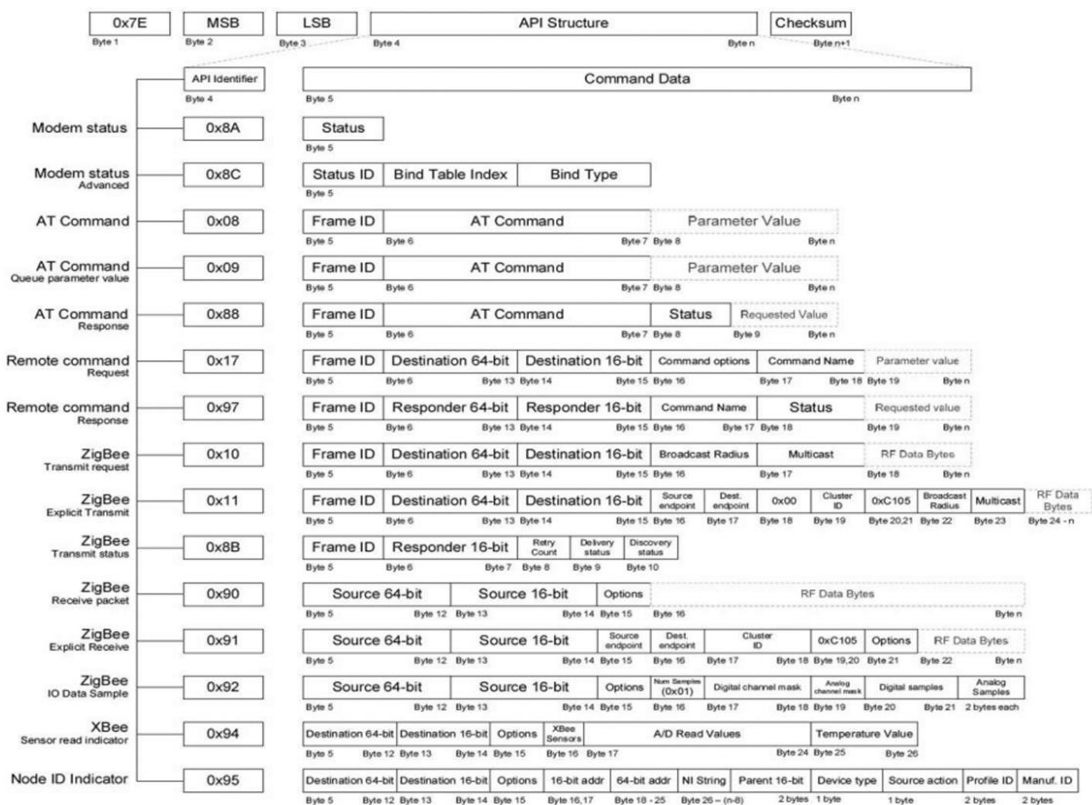


Figura 38 Estrutura das várias tramas API

O início de trama é sempre identificado pelo valor 0x7E. O campo seguinte é o *Length* que indica o tamanho da trama. De seguida temos o campo que sinaliza o tipo de trama. O tipo de trama API criada para efectuar este pedido corresponde a *Remote Command Request*, que permite enviar comandos AT. Ou seja, o campo de identificação do tipo de trama tem de ser igual a 0x17. Para efectuarmos o pedido RSSI directamente para o *Router Móvel* tem de se preencher o campo 64 *Destination Address* com o valor do endereço de série do mesmo. O campo 16 *Destination Network* é colocado a 0xFFFF, que implica fazer um *broadcast* pela rede. No campo *Command Name* é enviado o comando, neste caso o

comando DB, comando que permite efectuar o pedido RSSI. Por último têm-se o *Cheksum*. Todos os valores pertencentes à trama têm de estar em formato hexadecimal. A Tabela 5 mostra a estrutura da mensagem enviada.

**Tabela 5** Estrutura da mensagem *Remote Command Request*

<b>Start Delim</b>	7E
<b>Length</b>	00 0F
<b>API Ident.</b>	17
<b>Frame ID</b>	02
<b>64 Dest. Address</b>	0013A200 40795AFF
<b>16 Dest. Network</b>	FFFF
<b>Options Command</b>	01
<b>Command name</b>	44 42
<b>Cheksum</b>	9B

A mensagem de resposta tem formato semelhante à *Remote Command Request*, no entanto está-se perante uma mensagem do tipo *Remote Command Response*, daí o campo *API Identifier* ter o valor 0x97. A Tabela 6 mostra a estrutura da mensagem de resposta.

**Tabela 6** Estrutura da mensagem *Remote Command Response*

<b>Start Delim.</b>	7E
<b>Length</b>	00 10
<b>API Ident</b>	97
<b>Frame ID</b>	02
<b>64 Source Address</b>	0013A200 40795AFF
<b>16 Source Network</b>	FFFF
<b>Command Name</b>	44 42
<b>Status</b>	0
<b>RequestValue</b>	1D
<b>Cheksum</b>	FE

Nesta trama o campo *64 Source Address* indica o endereço do dispositivo que respondeu ao pedido. O campo *Command Name* indica a que comando se refere a resposta. O campo *status* fornece a informação se o pedido foi recebido correctamente, sendo que o valor 0 indica que a mensagem foi recebida correctamente.

O campo mais importante desta trama é o *request value*, pois indica o valor pedido através do comando DB; neste caso o valor de RSSI. Como o valor está em hexadecimal é necessário convertê-lo para decimal. Neste caso o valor de RSSI é de 0x1D convertendo para decimal obtém-se o valor de -29 dBm.

O Nó 1 envia a ordem (valor “A”) para o Nó 2 sendo que para realização deste processo é necessário enviar uma trama do tipo *ZigBee Transmit Request* representada na Tabela 7. Nesta trama temos também o início de trama, o tamanho, o identificador de trama API que neste caso é o valor 10. O campo de *64 Destination Address* é preenchido com o endereço que permite enviar a mensagem em *broadcast*; poder-se-ia ter utilizado o endereço de série do Nó 2 contudo, optou-se por esta opção pois caso seja necessário adicionar futuramente mais Nós de referência, o sistema já se encontra preparado para tal. O campo *RF data* é utilizado para enviar os dados que se pretende, neste caso o valor “A”, que em hexadecimal é representado por 0x41.

**Tabela 7** Estrutura da mensagem *ZigBee Transmit Request*

<b>Start Delim.</b>	7E
<b>Length</b>	00 0F
<b>API Ident</b>	10
<b>Frame ID</b>	01
<b>64 Dest. Address</b>	00000000 0000FFFF
<b>16 Dest. Network</b>	FFFF
<b>Broadcast radius</b>	0
<b>Options</b>	0
<b>RF Data</b>	41
<b>Cheksum</b>	B1

#### 4.7. SOFTWARE DESENVOLVIDO PARA O SISTEMA DE LOCALIZAÇÃO.

Nesta secção é apresentada a interface gráfica desenvolvida para o sistema de localização. Esta aplicação compreende as duas fases (*offline* e *online*) da técnica de *fingerprinting*.

Num primeiro momento a aplicação permite criar o mapa *fingerprinting* com os valores médios de RSSI para cada ponto físico do ambiente em relação aos nós de referência.

São recebidos 20 valores de RSSI por cada nó de referência, a partir desses valores é calculada a média. A média é guardada num ficheiro de texto.

A aplicação que permite realizar esta acção encontra-se ilustrada na Figura 39. Na interface da aplicação é possível escolher o ponto que se pretende mapear, ou seja guardar a média dos valores RSSI. Para realizar a funcionalidade descrita é necessário pressionar o botão “mapear ponto”. Conforme os dados são recolhidos aparecem no ecrã.

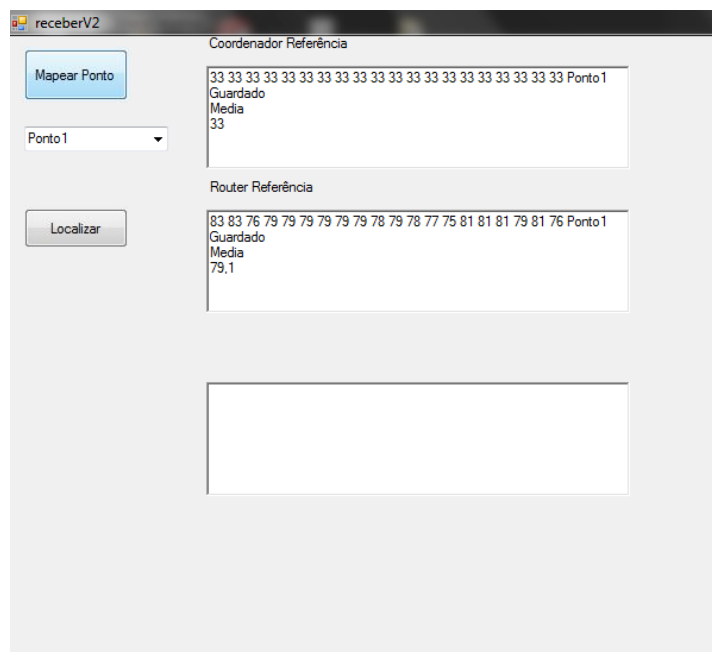
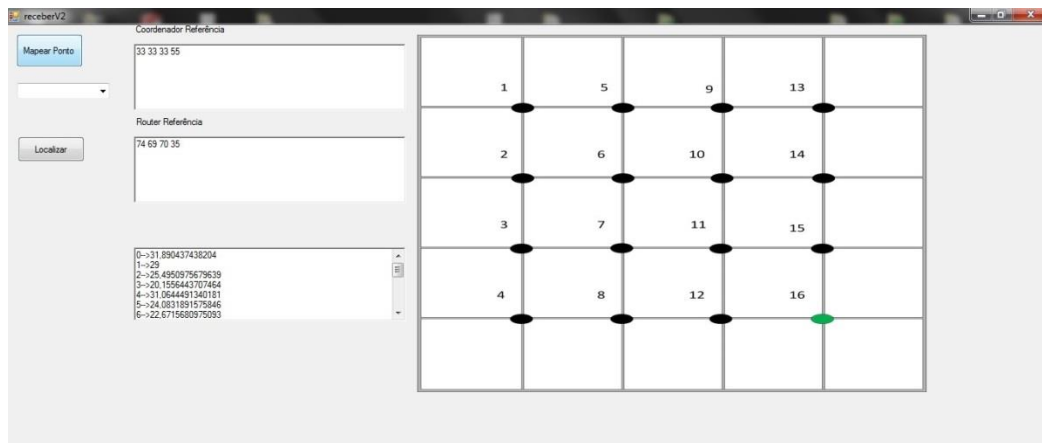


Figura 39 Interface Gráfica da fase de criação do mapa

Após mapear todos os pontos do mapa, é possível realizar a localização do Nó móvel. Para realizar esta acção é necessário premir o botão “localizar”, o sistema fica a ler ciclicamente os valores recebidos de RSSI e calcula as distâncias Euclidianas em relação aos pontos guardados no mapa. Deste modo, consegue indicar a posição mais aproximada do Nó Móvel, assinalando-a no mapa por um ponto de cor verde. A interface que permite obter a

localização do Nó Móvel encontra-se ilustrada na Figura 40. Na secção 5.3 será explicado mais detalhadamente porque o mapa apresenta esta configuração.



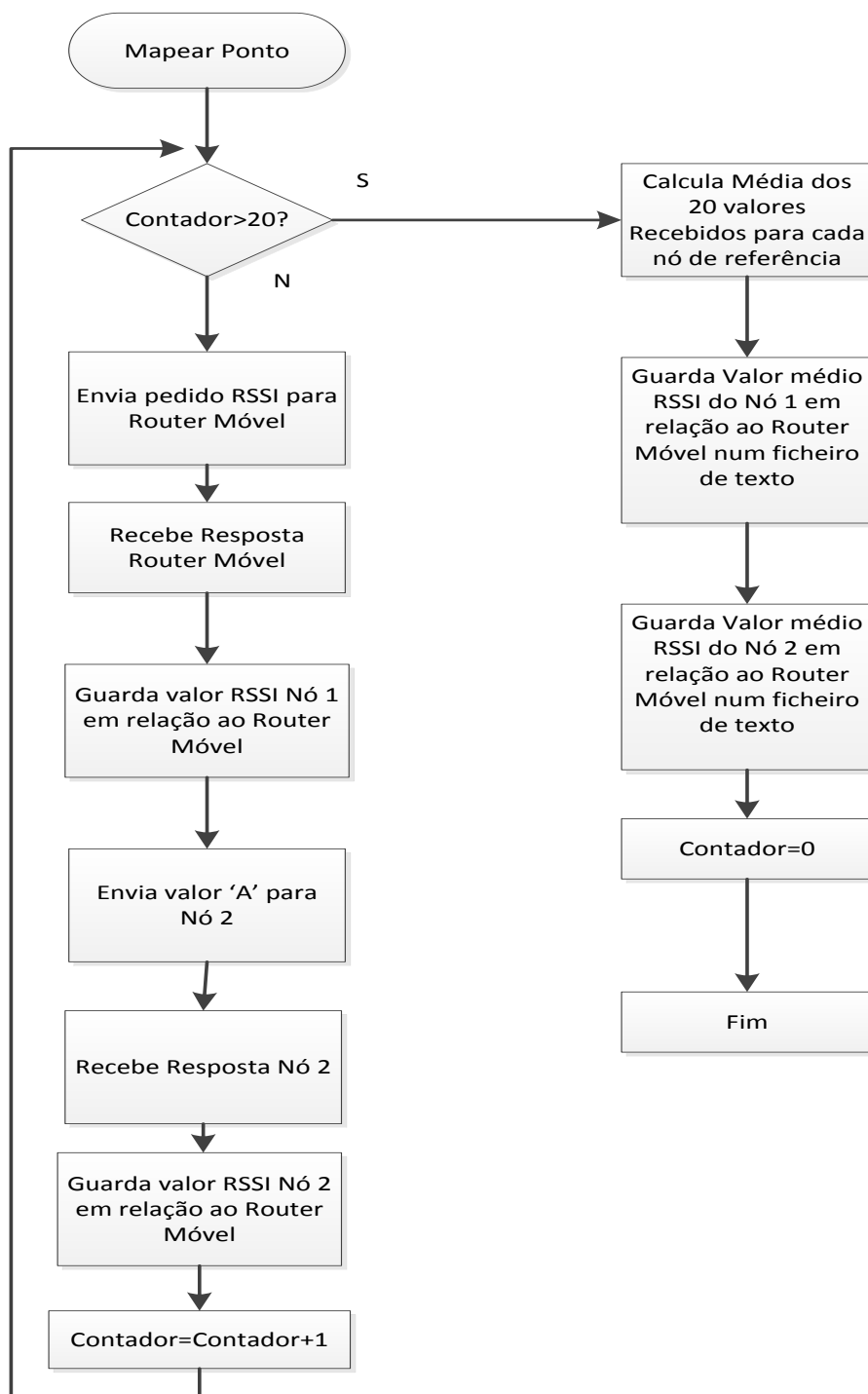
**Figura 40** Interface Gráfica da fase de localização do sistema desenvolvido

#### **4.8. FLUXOGRAMAS DO SISTEMA DESENVOLVIDO**

Para uma melhor compreensão de como funciona o sistema desenvolvido são apresentados os fluxogramas que sustentam a aplicação.

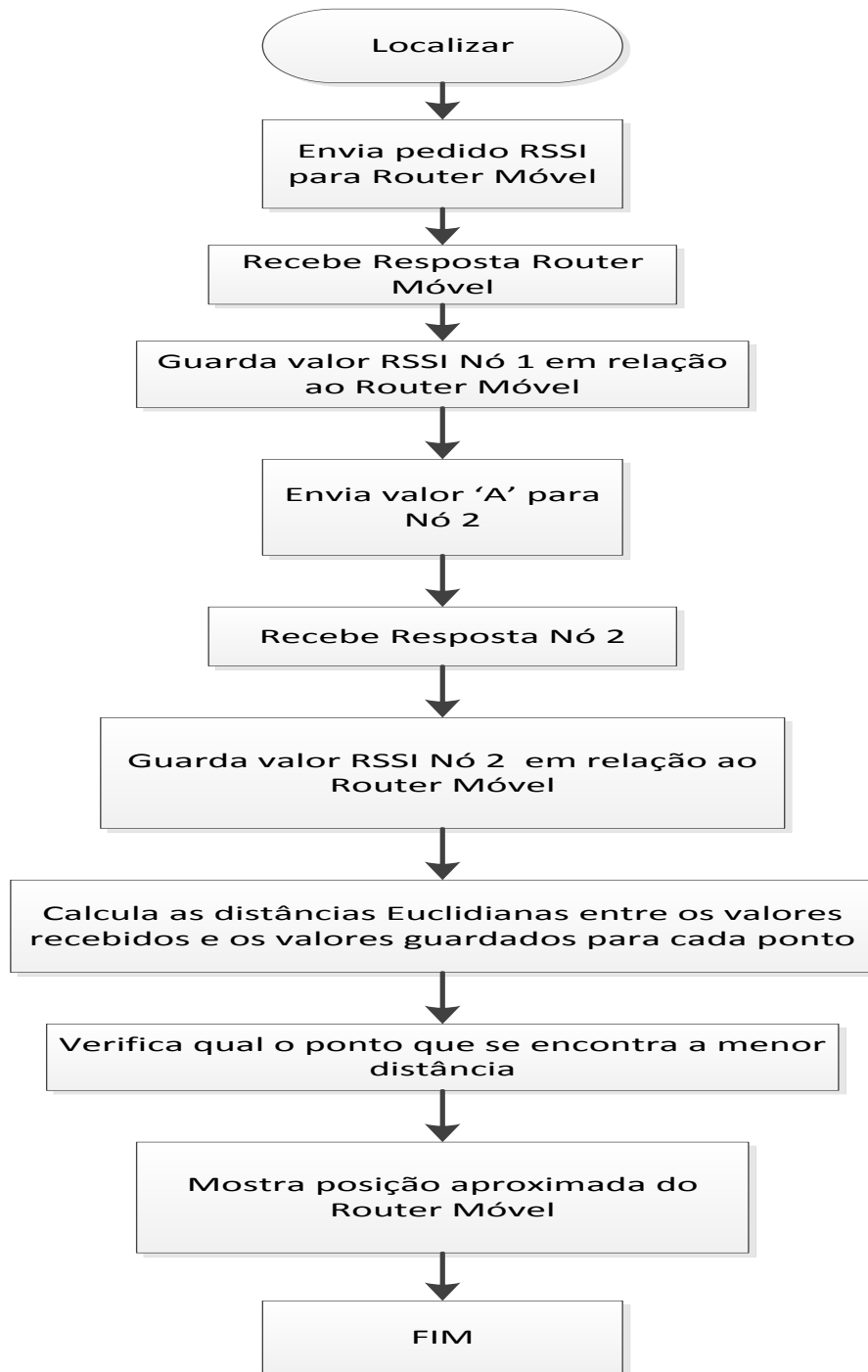
O primeiro fluxograma apresentado, Figura 41, explica o funcionamento da primeira parte da aplicação, onde é criado o mapa *fingerprinting*. O Nó 1 faz o pedido RSSI ao Nó móvel e recebe resposta. De seguida envia ordem para o Nó 2, recebendo o valor de RSSI do Nó 2 em relação ao Nó Móvel.

Estes pedidos são efectuados 20 vezes para cada ponto. Após a obtenção dos 20 valores, calcula-se a média RSSI para ambos os nós de referência e guardam-se estes valores num ficheiro de texto, para depois serem utilizados na fase em que a aplicação realiza as tarefas de localização.



**Figura 41** Fluxograma da fase de criação do mapa de *fingerprinting* (*offline*)

Na Figura 42 está ilustrado o fluxograma que permite obter a localização do Nó móvel. Nesta fase o sistema volta a efectuar o procedimento para obter os valores de RSSI em relação aos nós de referência. Após receber os valores RSSI em relação a cada nó de referência são calculadas as distâncias Euclidianas para cada ponto do mapa. A posição apresentada no mapa é aquela que apresenta menor distância.

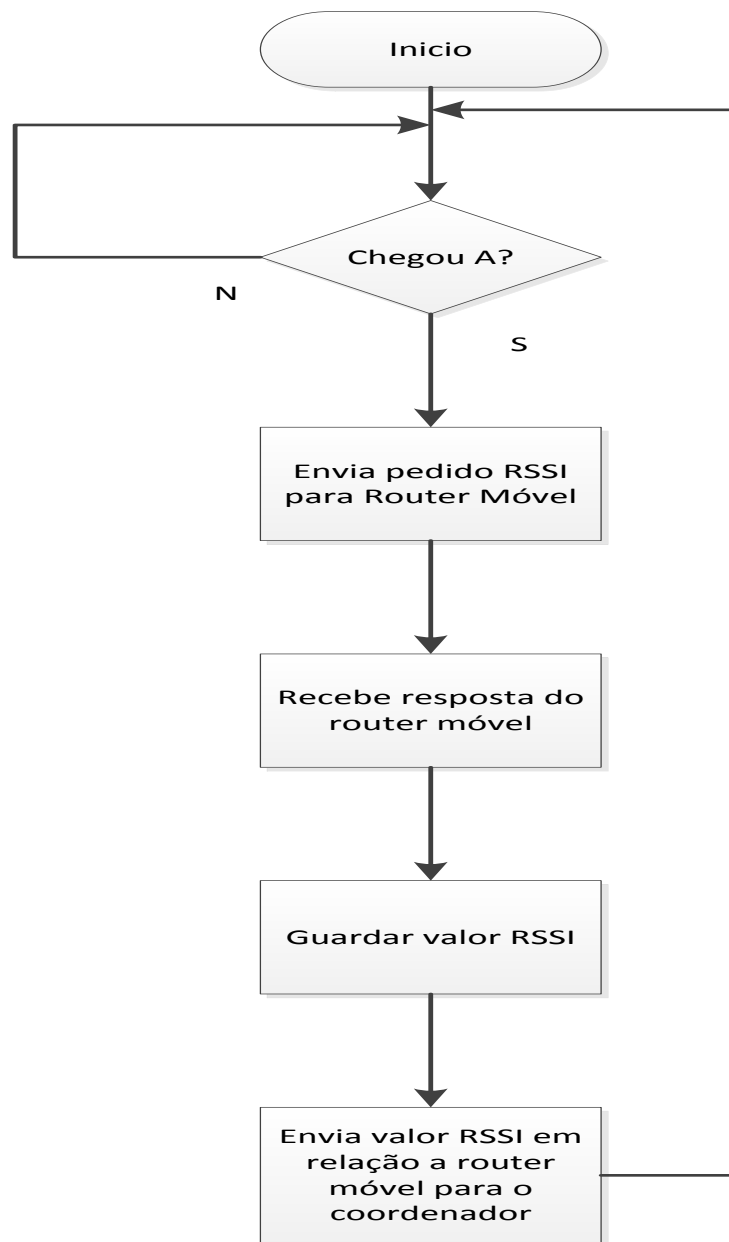


**Figura 42 Fluxograma da fase *online* do sistema de localização**

O código em linguagem C# baseado neste algoritmo para desenvolvimento da aplicação encontra-se no Anexo C.

Outro fluxograma importante no trabalho desenvolvido é o que trata das acções do *router* de referência (Nó 2), ilustrado na Figura 43. O Nó 2 apenas efectua o pedido de RSSI ao *router* móvel após receber a ordem vinda do coordenador de referência. Depois de efectuar

o pedido RSSI envia o valor obtido para o coordenador. O código desenvolvido em linguagem C baseado no fluxograma das acções do *router* encontra-se no Anexo D.



**Figura 43** Fluxograma das acções do *router* de referência

## 5. TESTES E RESULTADOS

Para estudar o comportamento do RSSI e para verificar se a técnica *fingerprinting* é viável para o projecto foi necessário realizar vários testes.

A primeira análise realizada tem como intuito perceber o comportamento do sinal num espaço exterior sem obstáculos, para diferentes distâncias entre dispositivos *XBee*. De seguida, foi realizado um estudo do RSSI com o mesmo objectivo do anterior, mas desta vez num espaço interior, para diferentes distâncias e diferentes ângulos de orientação.

Após a realização destas duas experiências foi criada uma plataforma de testes RSSI, que representa o mapa de 16 pontos definido para o sistema de localização desenvolvido. Nesta plataforma são realizados testes ao RSSI com o intuito de perceber se o seu valor varia ao longo de diferentes momentos temporais. Ainda nesta plataforma, foram realizados testes para perceber como a orientação de antena influencia o valor do RSSI, para um dispositivo colocado num ponto fixo. O último teste realizado nesta plataforma teve como objectivo compreender se os vários dispositivos em funcionamento simultâneo interferem no valor de RSSI.

## 5.1. VARIAÇÃO RSSI COM A DISTÂNCIA NUM ESPAÇO EXTERIOR

O primeiro teste realizado e o mais simples, foi efectuado em ambiente exterior. Analisou-se a variação do RSSI em relação ao aumento da distância. O coordenador de referência e o *router* móvel foram colocados em linha recta, os valores de RSSI foram sendo analisados à medida que se move o dispositivo de metro em metro, até uma distância de 12 metros. Foram realizados 3 testes, que se encontram representados nos gráficos das figuras 44, 45 e 46.

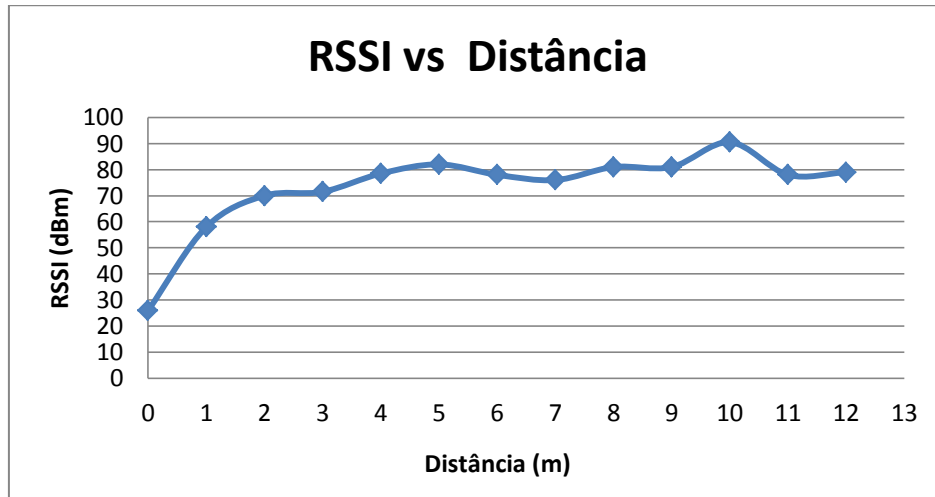


Figura 44 Gráfico RSSI em relação à distância para ambiente exterior - Teste 1

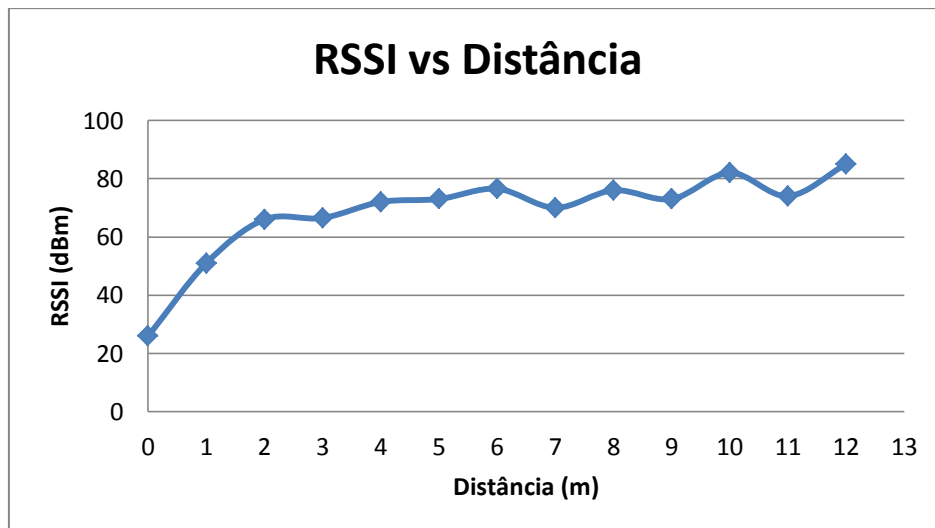
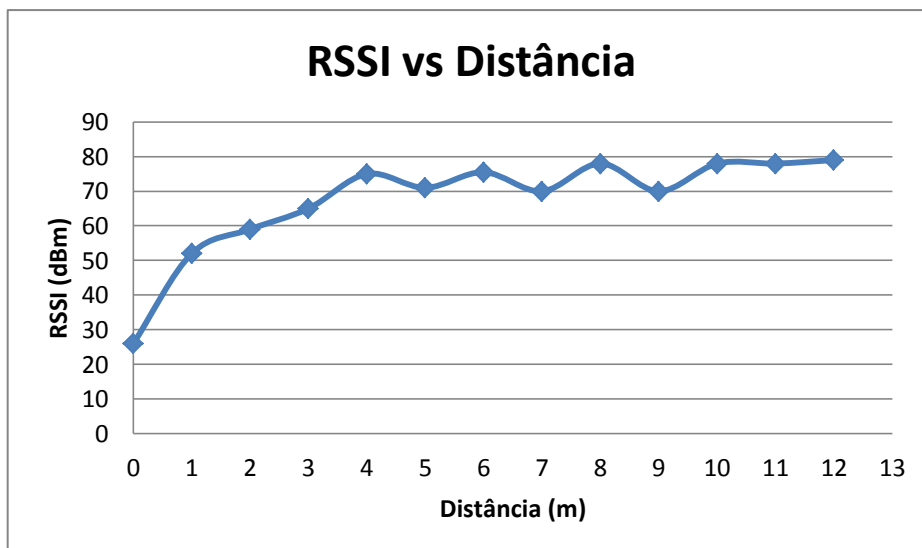


Figura 45 Gráfico RSSI em relação à distância para ambiente exterior - Teste 2



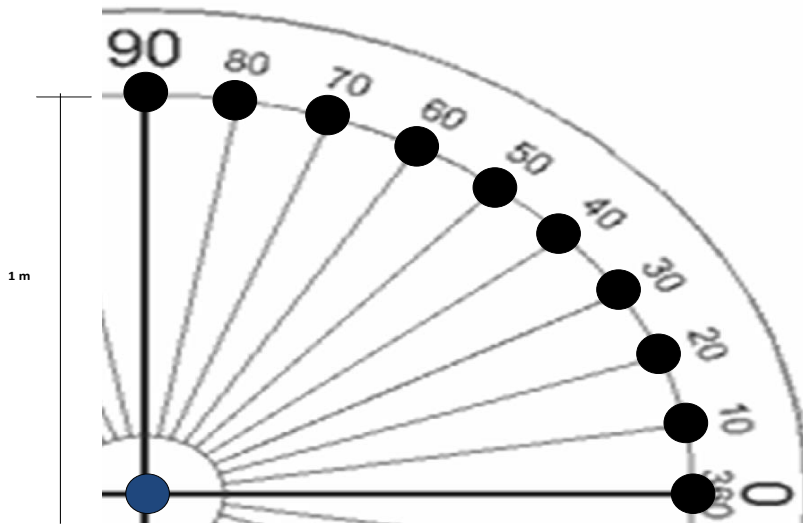
**Figura 46** Gráfico RSSI em relação à distância para ambiente exterior - Teste 3

Nas três análises efectuadas foi possível verificar que o aumento da distância leva a que o RSSI aumente. Este aumento acontece até aos 2, 3 metros. A partir desse ponto e em todos os testes realizados o valor RSSI torna-se muito incoerente. Existem vários pontos, para os três testes, que possuem valores muito próximos para distâncias diferentes.

## **5.2. ANÁLISE RSSI PARA DIFERENTES ORIENTAÇÕES E DISTÂNCIAS EM ESPAÇO INTERIOR**

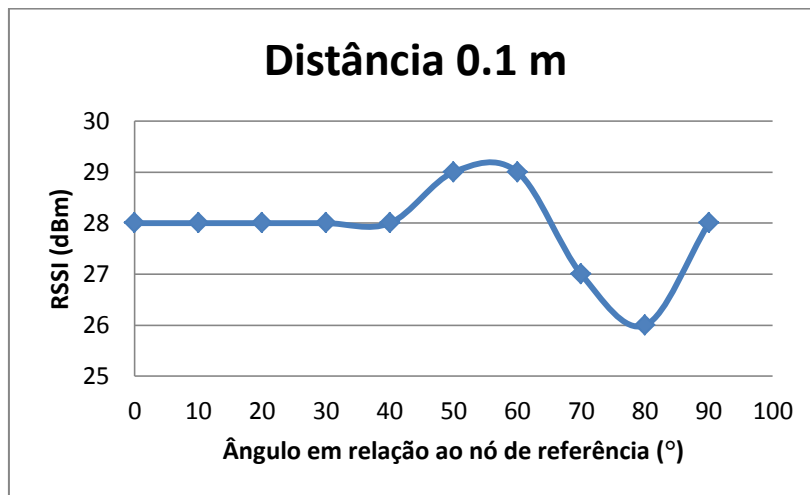
Após realizar a análise para um ambiente exterior, passou-se para um ambiente interior. Nesta fase a análise do valor RSSI foi efectuada com o coordenador numa posição fixa e *router* móvel colocado em diferentes distâncias. Para além de se efectuar a análise em linha recta, ou seja, o *router* móvel é colocado num ângulo de 0° em relação ao coordenador. Foram também realizadas medições para diferentes orientações entre eles. Tentou perceber-se o comportamento do sinal RSSI em relação à distância mas desta vez num espaço interior.

A análise foi feita para diferentes ângulos dos 0 aos 90 graus, com intervalos de 10 graus. Quanto a distância, foram sendo registados os valores de 5 em 5 cm até uma distância de 2 m. Na Figura 47 é apresentado um exemplo do teste realizado, nesta caso 1 m. O ponto azul é o nó fixo, os pontos pretos representam as posições por onde passa o nó móvel e onde são efectuadas as medições de RSSI.

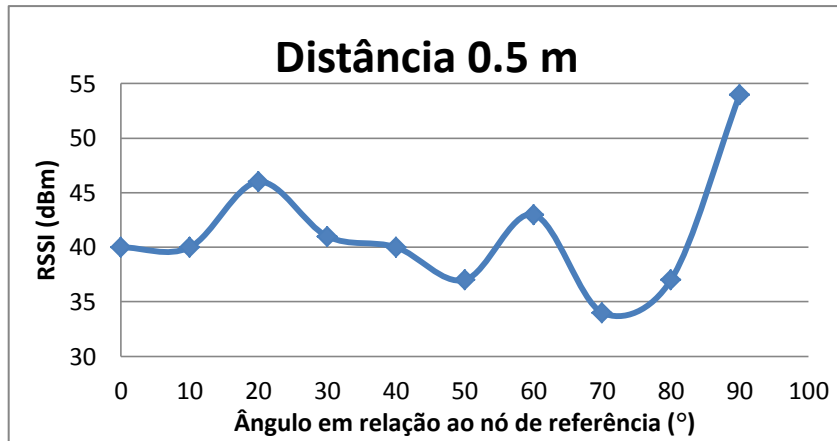


**Figura 47 Exemplo do teste interior para diferentes orientações e distâncias**

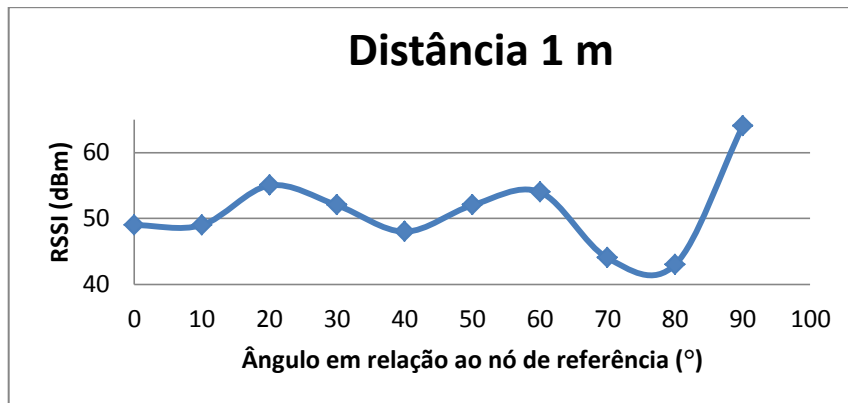
Nesta secção apenas será apresentada a análise para algumas distâncias, 0,1 m, 0,5 m, 1 m e 2 m variando o ângulo em relação ao coordenador de referência correspondentes aos gráficos das figuras 48, 49, 50 e 51 respectivamente.



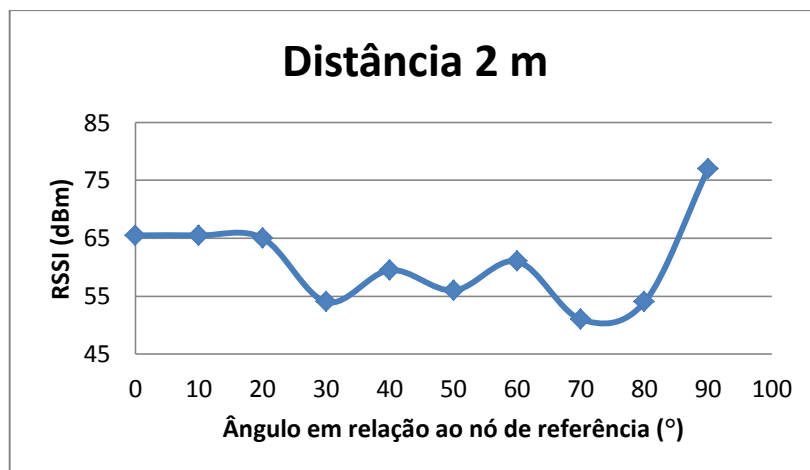
**Figura 48 Variação dos valores RSSI para a mesma distância (0,1 m) variando o ângulo em relação à referência**



**Figura 49** Variação dos valores RSSI para a mesma distância (0,5 m) variando o ângulo em relação à referência



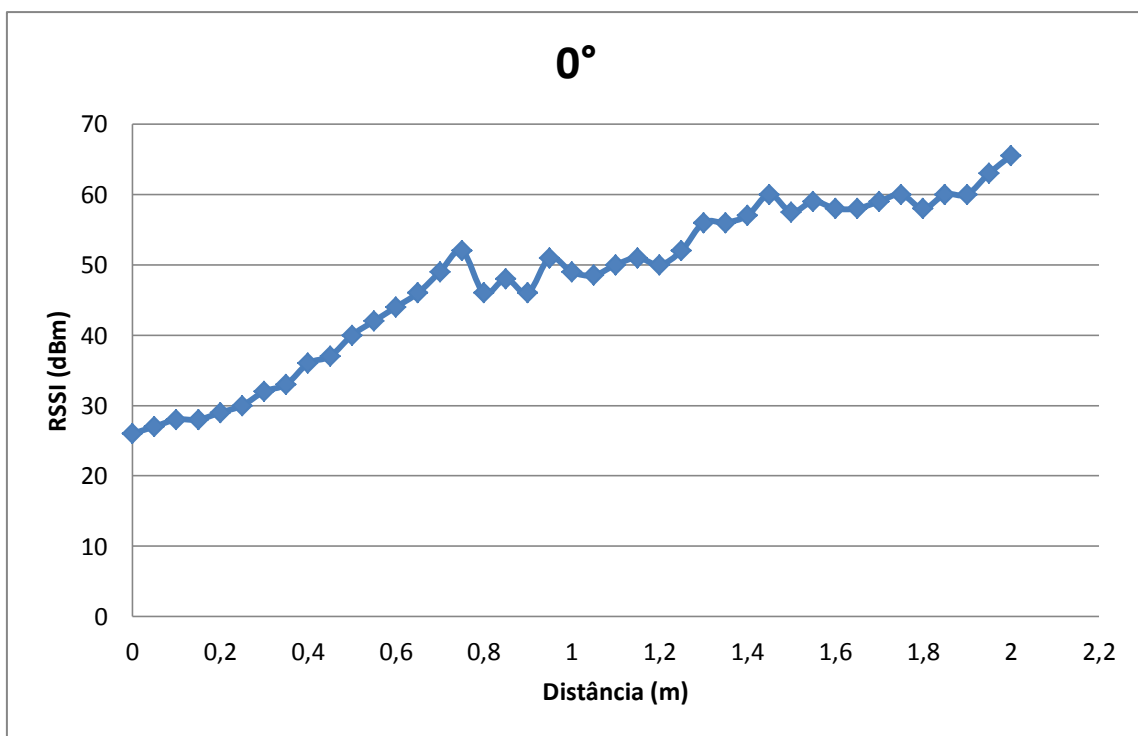
**Figura 50** Variação dos valores RSSI para a mesma distância (1 m) variando o ângulo em relação à referência



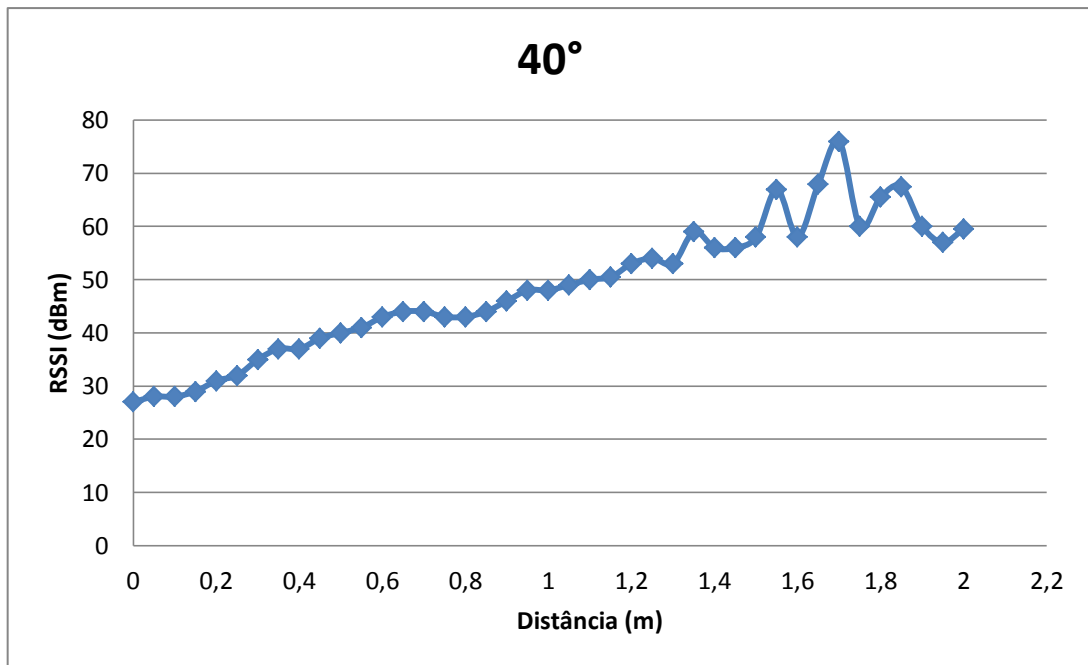
**Figura 51** Variação dos valores RSSI para a mesma distância (2 m) variando o ângulo em relação à referência

Esta análise permitiu verificar que para a mesma distância e variando o ângulo de orientação do nó móvel em relação ao nó de referência, os valores de RSSI sofrem variação.

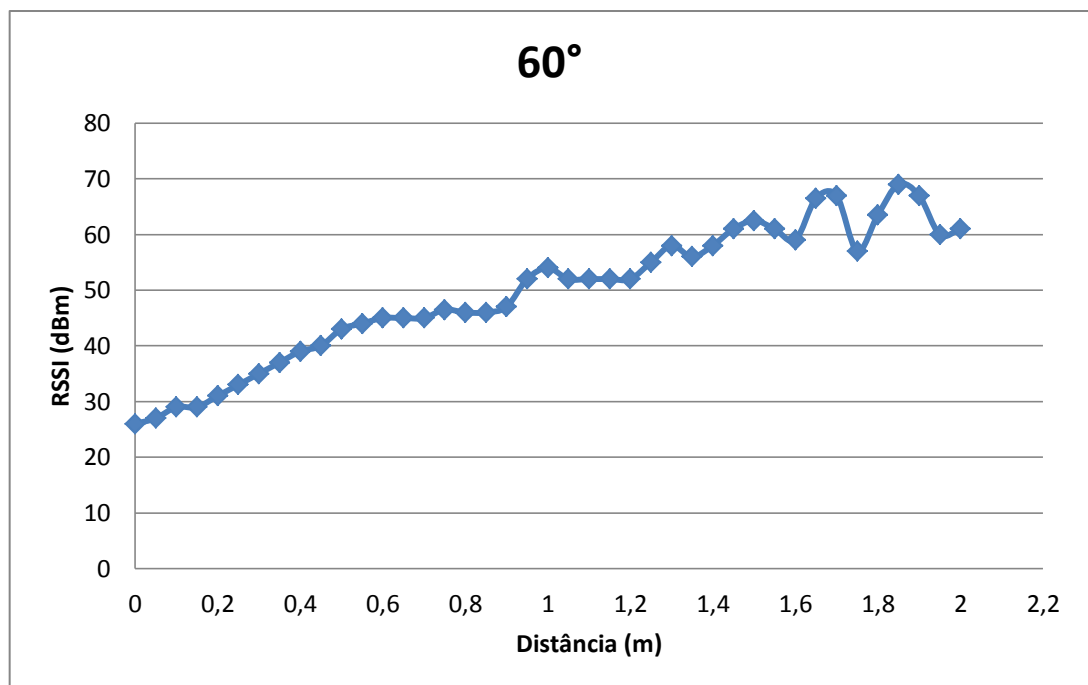
Após analisar a variação do RSSI para a mesma distância mas com ângulo de orientação do nó móvel em relação ao nó de referência diferentes, efectua-se a análise do sinal RSSI para o mesmo ângulo de orientação mas com diferentes distâncias. De seguida são apresentados os gráficos da variação do sinal para o mesmo ângulo de orientação entre dispositivos mas para distâncias diferentes. Essa variação encontra-se representada nos gráficos das figuras 52, 53 e 54 para os ângulos  $0^\circ$ ,  $40^\circ$ ,  $60^\circ$ , respectivamente.



**Figura 52** Variação dos valores de RSSI para o mesmo ângulo ( $0^\circ$ ) para diferentes distâncias



**Figura 53** Variação dos valores de RSSI para o mesmo ângulo (40°) para diferentes distâncias



**Figura 54** Variação dos valores de RSSI para o mesmo ângulo (60°) para diferentes distâncias

Os valores de RSSI obtidos voltaram a ter grande incoerência e não apresentam um aumento progressivo, como já tinha acontecido para a análise num espaço exterior na secção 5.1. Nesta análise é possível observar que os valores de RSSI aumentam consoante

o aumento da distância até cerca de 1 m, após essa distância os valores tornam-se muito incoerentes e obtêm-se mais uma vez valores muito próximos para distâncias diferentes.

Todos os valores recolhidos neste teste encontram-se no Anexo A.

### 5.3. ANÁLISE RSSI PARA CENÁRIO DE TESTES

Após as análises efectuadas anteriormente nas secções 5.1 e 5.2 chegou-se a conclusão que os valores de RSSI apresentam grande incoerência a partir dos dois metros para um espaço exterior e a partir de um metro para um espaço interior. Decidiu-se então criar uma plataforma de teste de 1 m<sup>2</sup> para tentar implementar o sistema de localização e estudar o comportamento do RSSI. A plataforma é constituída por 16 pontos espaçados entre eles por 20 cm. Na Figura 55 encontra-se ilustrada a plataforma de testes.

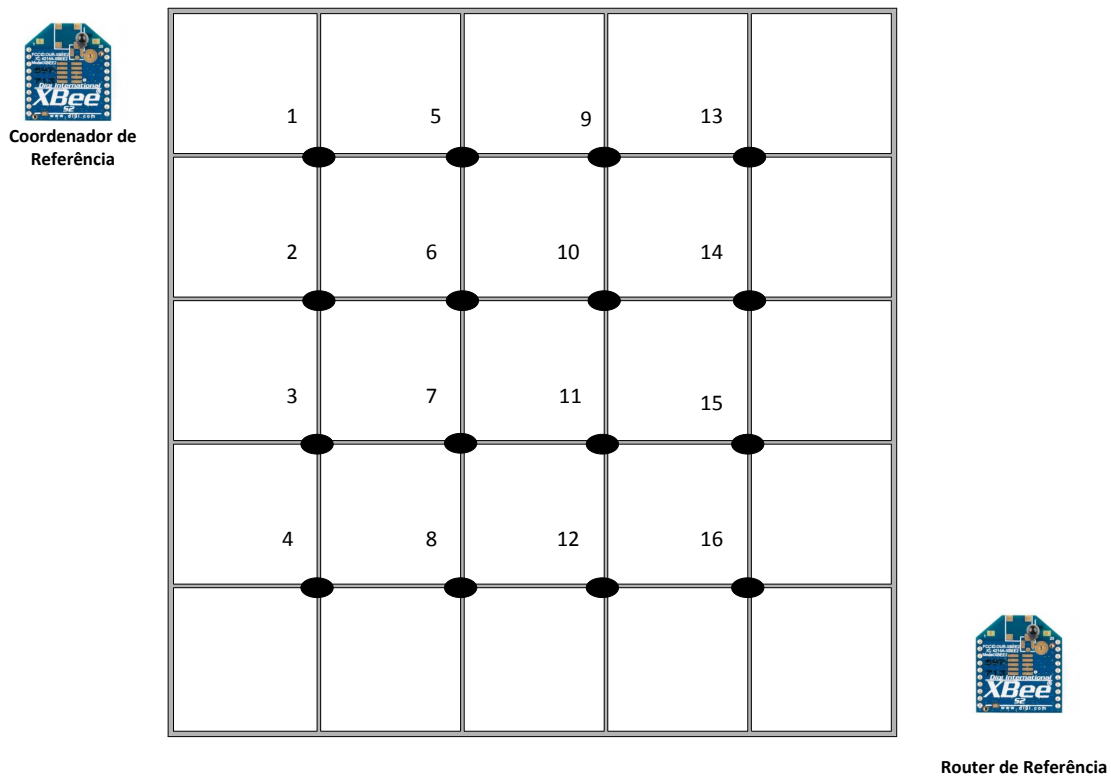


Figura 55 Esquemático da plataforma de teste

Como é possível visualizar os nós de referência encontram-se em cantos opostos, o *router* móvel, ao qual será efectuado o pedido de RSSI, irá ser colocado nos pontos assinalados de 1 à 16. Estes pontos são os pontos utilizados para a criação do mapa de *fingerprinting*.

### 5.3.1. ANÁLISE RSSI EM RELAÇÃO AO COORDENADOR DE REFERÊNCIA

A avaliação dos valores de RSSI em relação ao coordenador de referência (Nó1) e das diferentes posições do *router* móvel, foi realizada para diferentes momentos do dia (Momento 1: 10h; Momento 2: 14h; Momento 3: 20h). A Figura 56 apresenta uma representação cromatográfica, dos diferentes valores médios de RSSI obtidos para cada ponto e em diferentes momentos do dia. Pontos com menor intensidade de sinal, e logo mais próximos do coordenador são representados por cores mais esverdeadas, pontos intermédios são representados por cores amareladas e pontos mais distantes representados por cores mais avermelhadas.

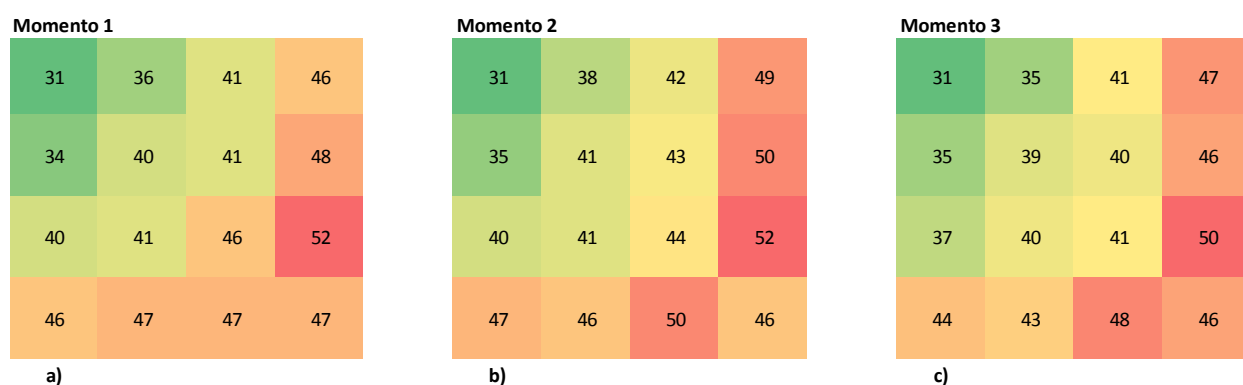
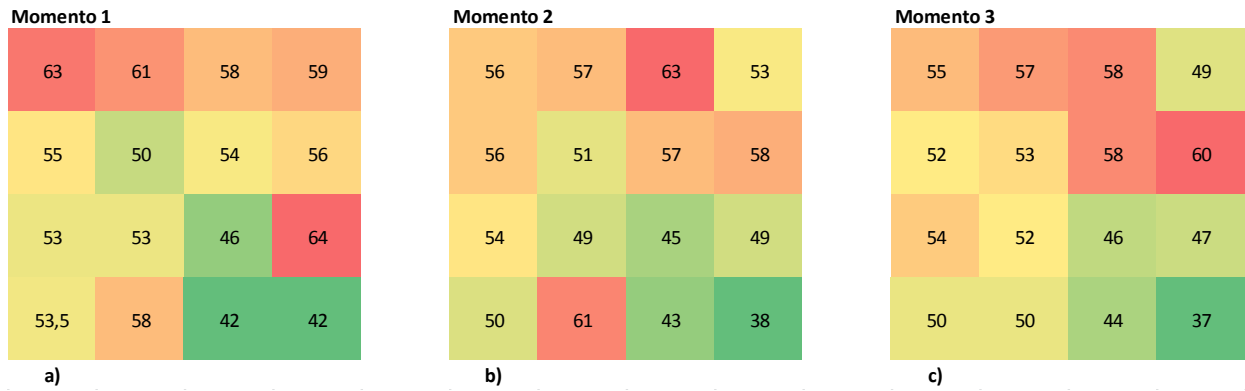


Figura 56 Representação cromatográfica do mapa RSSI em relação ao Nó 1 de referência a) Momento 1 b) Momento 2 c) Momento 3

Como é possível analisar, os valores apresentam variação em relação aos diferentes momentos do dia, são poucos os pontos de análise em que os valores se mantêm constantes. Outro problema encontrado é que existe pontos ambíguos, isto é, no mesmo mapa para distâncias diferentes o valor de RSSI obtido é o mesmo.

### 5.3.2. ANÁLISE RSSI EM RELAÇÃO AO ROUTER DE REFERÊNCIA.

A metodologia de análise foi a mesma que foi utilizada na subsecção anterior 5.3.1. A análise dos valores de RSSI desta vez é do *router* de referência (Nó 2) em relação ao *router* móvel. Na Figura 57 está ilustrada a representação cromatográfica dos valores de RSSI.



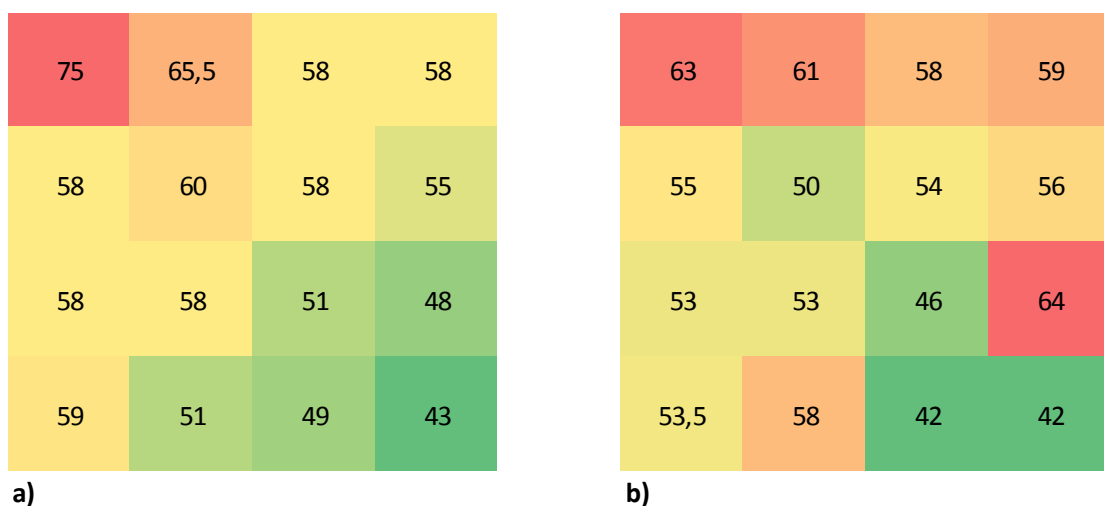
**Figura 57** Representação cromatográfica do mapa RSSI em relação ao Nó 2 de referência a) Momento 1 b) Momento 2 c) Momento 3

Nesta análise, a variação dos valores obtidos em relação a diferentes momentos é também visível. Aqui surge novamente o problema de valores iguais para posições diferentes.

### 5.3.3. ANÁLISE RSSI COM NÓS DE REFERÊNCIA EM POSIÇÕES TROCADAS

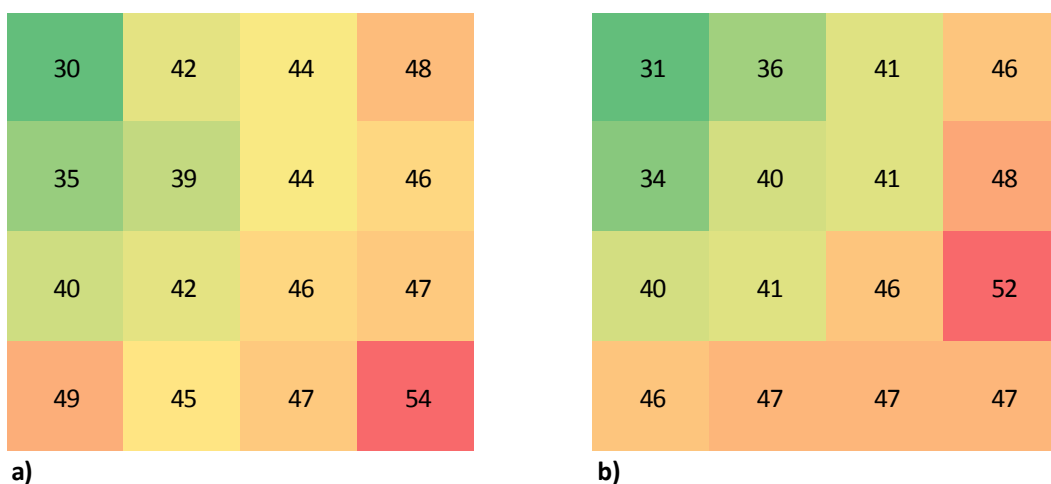
Nesta fase tentou perceber-se até que ponto a sensibilidade de cada dispositivo influencia as medições de RSSI. Para isso alterou-se a posição do Nó 1 para o sítio onde estava o Nó 2 e vice-versa.

Na Figura 58 é apresentado o mapa de valores RSSI para o Nó 1 na posição do Nó 2 comparado com os valores obtidos para Nó 2 na sua posição original.



**Figura 58** Comparação dos mapas RSSI. a) Valores RSSI obtidos com Nó 1 colocado na posição do Nó 2. b) Valores RSSI obtidos com Nó 2 na posição original

Na Figura 59 é apresentado o mapa valores RSSI para o Nó 2 na posição do Nó 1 comparado com os valores obtidos para o Nó 1 na sua posição original.

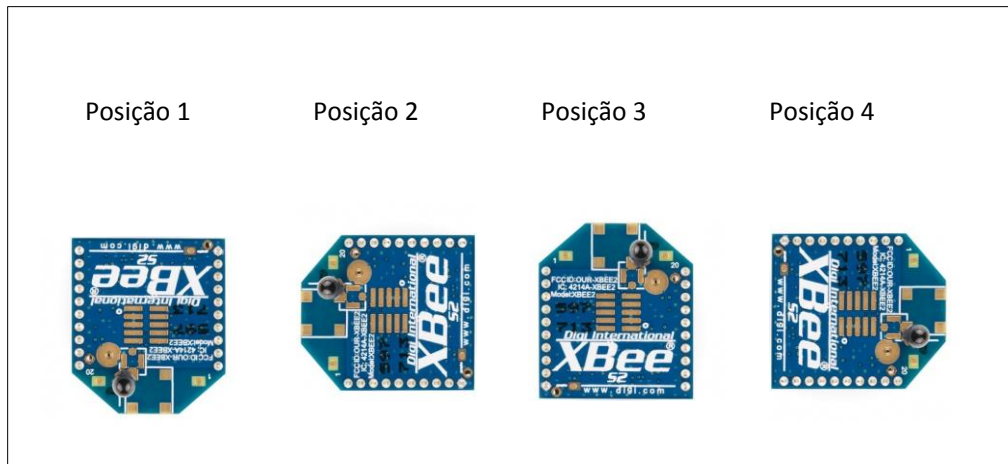


**Figura 59** Comparação dos mapas RSSI. a) Valores RSSI obtidos com Nó 2 colocado na posição do Nó 1. b) Valores RSSI obtidos com Nó 1 na posição original

Como é possível observar nos mapas de *fingerprinting*, dispositivos com o mesmo *hardware* têm comportamento diferentes. Ou seja, dois dispositivos iguais colocados na mesma posição de referência deveriam obter valores RSSI próximos, verifica-se que isso não acontece. Esta situação pode estar relacionada com um dos factores que influenciam o RSSI, estudados na subsecção 2.3.8, neste caso a variabilidade do transmissor.

#### 5.3.4. ANÁLISE RSSI PARA DIFERENTES ORIENTAÇÕES DE ANTENA

Outro dos factores estudados foi a orientação da antena dos dispositivos e até que ponto se traduzia essa diferença de orientação em termos de valores de RSSI. A análise foi realizada entre Nó 1 e *router* móvel. O nó móvel é aquele que irá mudar de orientação, foram analisadas quatro posições do dispositivo como podemos ver na Figura 60. Para estudar este factor não houve necessidade de analisar todos os pontos do mapa, a análise foi feita apenas para dois pontos, o ponto 1 e ponto 16, o ponto mais próximo e mais distante, respectivamente.



**Figura 60** Diferentes posições analisadas

Na Tabela 8 é possível visualizar os valores médios de RSSI obtidos para cada posição de antena em ambos os pontos de análise.

**Tabela 8** Valores de RSSI para diferentes orientações de antena

	Ponto 1				Ponto 16			
	Pos.1	Pos.2	Pos.3	Pos.4	Pos.1	Pos.2	Pos.3	Pos.4
<b>RSSI (dBm)</b>	31	40	39	32	50	53	58	52

Tendo em conta os dados recolhidos é notório que a orientação da antena influencia o valor de RSSI. Por exemplo para o ponto 1, com *router* móvel na posição 1 obtêm-se o valor 31 e na posição 2 o valor 40, trata-se de uma diferença significativa. A orientação do dispositivo móvel influencia os valores de RSSI, logo irá influenciar o sistema de localização.

### 5.3.5. ANÁLISE DE RSSI QUANTO À INFLUÊNCIA ENTRE DISPOSITIVOS

Os valores dos mapas RSSI foram obtidos para cada nó de referência de forma individual, ou seja, apenas um dos nós se encontrava ligado a rede. Nesta fase, ligaram-se os dois nós de referência a rede e ambos efectuaram o pedido de RSSI ao nó móvel. Esta tentativa serve para perceber se há influências entre os dispositivos que possam alterar significativamente os valores de RSSI.

Os valores de RSSI obtidos com os dois nós de referência em conjunto, foram comparados com os valores obtidos com os nós a funcionarem separadamente. Esses valores são apresentados Tabela 9. Esta análise foi também efectuada para os pontos 1 e 16.

**Tabela 9 Valores médios RSSI para funcionamento dos nós de referência individualmente e em conjunto**

	<b>Ponto1</b>		<b>Ponto 16</b>	
<b>Coordenador Ref.</b>	<b>Individual</b>	<b>Conjunto</b>	<b>Individual</b>	<b>Conjunto</b>
<b>RSSI (dbm)</b>	31	31	47	48
<b>Router Ref.</b>	<b>Individual</b>	<b>Conjunto</b>	<b>Individual</b>	<b>Conjunto</b>
<b>RSSI (dbm)</b>	63	59	42	43

Os valores obtidos para a situação individual são idênticos aos valores obtidos para situação em que os dispositivos se encontram a funcionar em simultâneo. Este factor é relevante pois permite dizer que os dispositivos de referência não influenciam os valores RSSI obtidos de um e outro.



## 6. CONCLUSÕES

Este trabalho teve como principal objectivo o desenvolvimento de um sistema de localização para ambientes interiores baseado no protocolo de comunicação sem fios *ZigBee*, objectivo esse que foi cumprido. O sistema desenvolvido é constituído por módulos *XBee Series 2*, onde os dispositivos comunicam entre si através do protocolo *ZigBee*. O sistema é baseado na informação de intensidade de sinal, RSSI, fornecida pelos dispositivos. Para utilizar o RSSI como variável que permita efectuar a localização de dispositivos, foi utilizada a técnica de *fingerprinting*.

A técnica de localização *fingerprinting* possui duas fases: *offline* e *online*. Na fase *offline* é necessário construir um mapa de intensidades de sinal para o espaço onde se pretende implementar o sistema de localização. Esta fase é muito demorada; é necessário um estudo exaustivo ao comportamento do sinal. Depois do mapa de *fingerprints* criado, passa-se a fase *online* onde se implementa o algoritmo K-NN com recurso às distâncias Euclidianas.

Num primeiro momento foram analisadas as técnicas de localização existentes e os seus princípios de funcionamento. Foram também analisados alguns sistemas de localização já existentes, de forma a perceber como as técnicas de localização são utilizadas e implementadas. Para além das técnicas de localização foi também realizado um estudo acerca do protocolo *ZigBee*, de forma a analisar as suas características, especificações,

princípio de funcionamento e de que forma se poderia implementar a rede *ZigBee* para auxílio do sistema de localização. Efectuou-se também uma análise ao dispositivo *XBee* utilizado, para compreender os seus princípios de funcionamento e de que forma poderíamos obter o valor de RSSI necessário para implementação do sistema.

A grande dificuldade encontrada neste trabalho prende-se com o comportamento do RSSI. Foram realizados diversos testes, tanto em espaços exteriores, como em espaços interiores, do comportamento do RSSI.

Nos testes realizados no exterior sem obstáculos, os valores de RSSI foram aumentando à medida que afastávamos os dispositivos *XBee*; situação que apenas se manteve até cerca de dois, três metros. A partir destas distâncias os valores tornam-se muito incoerentes, tendo variações quando comparado com o modelo de propagação teórico estudado. Em certas situações, para distâncias diferentes obteve-se valores RSSI muito próximos.

Realizou-se também um teste num ambiente *indoor*, neste caso, para além do estudo da distância, analisou-se também a influência do ângulo de orientação. Nesta análise o sistema apresenta incoerência a partir de um metro de distância. Um dos factores que poderá ter tido influência no valor RSSI num ambiente *indoor* é a reflexão e a multiplicação de sinal, devido às interferências no meio, como paredes e objectos. Este factor possivelmente tornou o sistema mais instável. Com a obtenção e posterior análise destes resultados ficou também provado que a orientação entre dispositivos influencia o sinal.

Analisando o teste anterior, foi possível verificar que até um metro, as influências que afectam o RSSI, não alteravam muito o comportamento do sinal. Por causa disso o sistema de localização desenvolvido foi implementado dentro desse limite ( $1\text{m}^2$ ) e constituído por 16 pontos de análise.

O primeiro teste realizado na plataforma foi a criação de mapas *fingerprints* para diferentes momentos do dia. Fazendo a análise dos mapas obtidos foi possível verificar que para o mesmo ponto, ou seja, a mesma distância entre nó de referência e nó móvel, os valores de RSSI variavam ao longo do dia. Este factor é um entrave do sistema de localização, pois se o mapa de *fingerprints* for implementado num determinado momento e o sistema for utilizado noutro pode haver uma grande discrepância entre valores RSSI guardados e valores RSSI obtidos aquando da utilização do sistema de localização. Outro aspecto

verificado nas análises dos mapas desenvolvidos foi a existência de posições ambíguas, isto é, dois pontos distintos tinham o mesmo valor RSSI.

Um dos aspectos que voltou a ser analisado e que novamente provocou alterações no sistema foi a orientação da antena dos dispositivos *XBee*.

O sistema de localização desenvolvido foi implementado na plataforma criada. Funciona e consegue localizar o nó móvel. O grande entrave deste está relacionado com a instabilidade do RSSI, que torna o sistema impreciso. O algoritmo de localização utilizado reduz alguma da imprecisão do sistema mas não consegue contornar a instabilidade dos valores RSSI. Por vezes, devido à instabilidade dos valores o sistema de localização indica uma posição do nó móvel, diferente daquela que ele realmente se encontra.

Por fim, pode dizer-se que este trabalho cumpriu os objectivos propostos. O sistema de localização *indoor* funciona, é certo com limitação de distância e nem sempre indica a localização exacta do dispositivo que se pretende localizar, mas devido a grande instabilidade dos valores RSSI e a todos os factores que o influenciam, foi o sistema possível de desenvolver.

Espera-se que este trabalho possa ser uma contribuição importante para projectos na mesma área, uma vez que este serve como "alavanca" e ponto de partida para iniciar o desenvolvimento de um sistema de localização *indoor*. Em trabalhos futuros sugiro que se utilize outro dispositivo *ZigBee*, de forma a estudar se este fornece valores de RSSI mais estáveis e com uma melhor relação com a distância. Ao longo das pesquisas realizadas surgiu como opção para algoritmo de localização as redes neuronais; neste trabalho foi utilizado o K-NN. O desenvolvimento de um sistema de localização baseado nas redes neuronais pode ser também um caso a ser estudado futuramente.



## Referências Documentais

- [1] BRAGA, Tiago – *Monitorização Ambiental em Espaços Florestais com Rede de Sensores Sem Fios*. Dissertação de Mestrado em Engenharia de Telecomunicações e Redes orientada pelo Eng.º Joaquim Azevedo e apresentada na Universidade da Madeira, em 2010.
- [2] BRÁS, Luís – *Desenvolvimento de Sistema de Localização Indoor de Baixo Consumo*. Dissertação de Mestrado em Engenharia Electrónica e Telecomunicações orientada pelo professor José Fonseca e apresentada na Universidade de Aveiro, em 2010.
- [3] CARVALHO, Joel - *Localização de Dispositivos Móveis em Redes Wi-Fi*. Dissertação de Mestrado em Engenharia Informática orientada pelo Dr. Paulo Martins e apresentada na Universidade de Trás os Montes e Alto Douro, em 2007.
- [4] FALUDI, Robert – *Building Wireless Sensor Networks*. O’Reilly, ISBN:978-0-596-80773-3, 2011, United States of America.
- [5] FARAHANI, Shahin *Zigbee Wireless Networks and transceivers*. Newnes, 2008.
- [6] FERNANDES, Maria – *Redes de Sensores para uso Humano*. Dissertação de Mestrado em Engenharia Electrónica e Telecomunicações orientada pelo Dr.Nuno Borges de Carvalho e apresentada na Universidade de Aveiro, em 2010.
- [7] G. N. Sol Teixeira, L. M. Rodríguez Peralta, “Comparing ZigBee, Bluetooth, UWB, and Wi-Fi”, in *Encyclopedia of Networked and Virtual Organizations*, Idea Groups, Vol. I, pp. 288 - 296, ISBN: 978-1-59904-885-7, 2008.
- [8] GOUVEIA, Bruno. - *Dispositivos de Monitorização e Controlo automático de factores climáticos em Museus*. Dissertação de Mestrado em Engenharia Telecomunicações e Redes orientada pelo Dr. Carlos Seródio e apresentada na Universidade de Trás os Montes e Alto Douro.
- [9] L. M. Pestana Leão de Brito, L. M. Rodriguez Peralta, "Basic Concepts of Wireless Sensor Networks", *Encyclopedia of Networked and Virtual Organizations*, Idea Groups, , ISBN: 978-1-59904-885-7, 2008.
- [10] MATOS, Rui – *Sistema de Localização com Base em Tecnologias Sem Fios*. Dissertação de Mestrado em Engenharia Electrónica e Telecomunicações orientada pelo Dr.Nuno Borges de Carvalho e apresentada na Universidade de Aveiro, em 2009.
- [11] MONSIGNORE, Ferdinando - *Módulos de Comunicação Wireless para Sensores ZigBee*. Dissertação de Mestrado em Engenharia Eléctrica orientada pelo Dra. Maria Paiva e apresentada na Universidade de São Paulo, em 2007.

- [12] OLIVEIRA, José – *Correcção de interferência electromagnética em sistemas de localização*. Dissertação de Mestrado em Engenharia Física orientada pelo Dr. Luís Cadillon Costa e apresentada na Universidade de Aveiro, em 2008.
- [13] BARONTI, Paolo; PILLAI, Prashant; CHOOK, Vince; CHESSA, Stefano; GOTTA, Alberto; HU, Y. Fun - “Wireless Sensor Networks: a Survey on the State of the Art and the 802.15.4 and ZigBee Standards”, in Computer communications ISSN 0140-3664.
- [14] PINTO, Hugo – *Desenvolvimento de Modelos de Localização para tecnologias de Redes sem Fios*. Dissertação de Mestrado em Engenharia Electrotécnica e de Computadores orientada pela Dra. Laura Peralta e apresentada na Universidade da Madeira, em 2009.
- [15] SANTOS, João - *WISE-MANager: Criação e Gestão de Sessões Colaborativas em Redes de Sensores sem Fio*. Dissertação de Mestrado em Engenharia Informática orientada pelo Dra. Laura Peralta e apresentada na Universidade da Madeira, em 2010.
- [16] SIMÃO, Pedro – *Controlo de Assiduidade com Multiposto e Comunicações Wireless*. Dissertação de Mestrado em Engenharia Electrónica e Telecomunicações orientada pelo Dr. José Fonseca e apresentada na Universidade de Aveiro, em 2008.
- [17] SOUSA, Tiago – *Rede de Sensores Móveis*. Dissertação de Mestrado em Engenharia Electrotécnica e de Computadores orientada pelo Dr. Armindo Lage e apresentada na Faculdade de Engenharia da Universidade do Porto, em 2008.
- [18] TADAKAMADLA, Shashank – *Indoor Local Positioning System For ZigBee, Based On RSSI*. Dissertação de Mestrado em Ciência da Engenharia Eléctrica orientada pelo Prof. Bengt Oelmann e apresentada na Mid Sweden University, em 2006.
- [19] University Bremen, Germany, <http://www.faculty.iu-bremen.de/schoenw>
- [20] LEE, Jin; SU, Wei-Su; SHEN, Chung-Chou - *A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi*. Information & Communication Research Labs Industrial Technology Institute. The 33<sup>rd</sup> Annual Conference of the IEEE Industrial Electronics Society, Taiwan, 2007.
- [21] EKLUND, Mikael; HANSEN, Thomas; SPRINKLE, Jonathan; SASTRY Shankar – *Information Technology for Assisted Living at Home: building a wireless infrastructure for assisted living*. Engineering in Medicine and Biology 27th Annual Conference, Shaangai, 2005.
- [22] WOOD, A.; VIRONE, G.; DOAN, T; CAO, Q.; SELAVO, L.; WU, Y.; FANG, L.; HE, Z.; LIN, S.; STANKOVIC; J. - *ALARM-NET: Wireless Sensor Networks for Assisted-Living and Residential Monitoring*. Department of Computer Science, University of Virginia, 2006.
- [23] LAPINSKI, Michael - *A Wearable, Wireless Sensor System for Sports Medicine*. Massachusetts Institute of Technology, 2008.
- [24] NADIMI, E.; SOGAARD, H.; BAK, T.; OUDSHOORN, F. - *ZigBee-based wireless sensor networks for monitoring animal presence and pasture time in a strip of new*

- grass. Research*. Center Bygholm, Department of Agricultural Engineering, Horsens, 2007.
- [25] MALINOVSKI, Mateus; MOSKWA, Matthew; FELDMEIERS, Mark; LAIBOWITZ, Mathew; PARADISO, Joseph - *CargoNet: A Low-Cost MicroPower Sensor Node Exploiting Quasi-Passive Wakeup for Adaptive Asynchronous Monitoring of Exceptional Events*. MIT Media Lab, Cambridge.
- [26] SERRANO, Oscar; CAÑAS, José; RODERO, Luis; MATTELLÁN, Vicente - *Robot localization using WiFi signal without intensity map*. Universidad Rey Juan Carlos, 2004.
- [27] ZHOU, Rui - *Enhanced Wireless Indoor Tracking System in Multi-Floor Buildings with Location Prediction*. Computing Center, University of Freiburg, Freiburg.
- [28] LIU, Hui; DARABI, Housang; BANERJEE, Pat; LIU, Jing - *Survey of Wireless Indoor Positioning Techniques and Systems*. IEEE transactions on systems, man, and cybernetics—part c: applications and reviews, 2007.
- [29] HIGHTOWER, Jeffrey; BORRIELLO, Gaetano - *Location Sensing Techniques*. Computer Science and Engineering, University of Washington, Seattle, 2001.
- [30] KATO, H.; BILLINGHURST, M.; POUPYREV, I.; IMAMOTO, K.; TACHIBANA, K. - *Virtual Object Manipulation on a Table-Top AR Environment*. Faculty of Information Sciences, Hiroshima City University; HIT Laboratory, University of Washington; ATR MIC Laboratories, ATR International.
- [31] COMPORT, Andrew; MARCHAND, Éric; CHAUMETTE, François - *A real-time tracker for markerless augmented reality*. ACM/IEEE Int. Symp. on Mixed and Augmented Reality; Tokyo; 2003.
- [32] KAEMARUNGS, Kamol; KRISHNAMURTHY, Prashant - *Modeling of Indoor Positioning Systems Based on Location Fingerprinting*. School of Information Science, University of Pittsburgh, 2004.
- [33] JUNIOR, Valmir - *sistema de localização para ambientes fechados baseado na potência do sinal recebido em rede ZigBee*, Instituto Federal do Espírito Santo; Vitória, 2011.
- [34] PRIWGHARM, Ratana; SRIVILAS, Kiri; CHERNTANOMWONG, Panarat - *Indoor Localization System using RSSI Measurement in Wireless Sensor Network Based on ZigBee Standard*. Faculty of Engineering King Mongkut's Institute of Technology Ladkrabang, 2010.
- [35] XU, Jiuqiang; LIU, Wei; LANG, Fenggao; ZHANG, Yuanyuan; WANG, Chenglong - *Distance Measurement Model Based on RSSI in WSN*. School of Information Science & Engineering Northeastern University, Shenyang; 2010.
- [36] KAEMARUNGS, Kamol - *Design of indoor positioning systems based on location fingerprinting technique*. University of Pittsburgh, 2005.
- [37] RAJ, Mithun; VARGHESE, Koshy - *Positioning of human resources in a construction environment using Zigbee*. IIT Madras, India.

- [38] ROOS T.; MYLLYMÄKI, P.; TIRRI, H.; MISIKANGAS, P.; SIEVÄNEN, J. -*A probabilistic approach to WLAN user location estimation*. International Journal of Wireless Information Networks, July 2002.
- [39] LIN, Tsung-Nan, LIN, Po-Chiang - *Performance Comparison of Indoor Positioning Techniques based on Location Fingerprinting in Wireless Networks*. Institute of Communication Engineering National Taiwan University, International Conference on Wireless Networks, Communications and Mobile Computing, 2005.
- [40] WANT, Roy; HOPPER, Andy; FALCÃO, Veronica; GIBBONS, Jonathan - *The Active Badge Location System*. Olivetti Research Ltd. (ORL)Cambridge, England
- [41] WARD, Andy; JONES, Alan; HOPPER, Andy - *A New Location Technique for the Active Office*.
- [42] BAHL, Paramvir; PADMANABAN, Venkata – *RADAR: An in- building RF-based user location and tracking system*. Microsoft Research, 2000.
- [43] STEGGLES, Pete; GSCHWIND, Stephan – *The Ubisense Smart Space Platform*. A ubisense white paper, 2005.
- [44] NI, Lionel; LIU, Yunhao; LAU, Yiu; PATIL, Abhishek - *LANDMARC: Indoor Location Sensing Using Active RFID*. Kluwer Academic Publishers, 2004.
- [45] SUN, Maoheng; QIAN, Yicheng - *Study and Application of Security Based on ZigBee Standard*. Third International Conference on Multimedia Information Networking and Security, 2011.
- [46] SOARES, Sérgio - *Rede de Sensores Sem Fio Para Localização e Monitoramento de Pequenos Ruminantes*. Trabalho de Final de Curso em Engenharia de computação orientada pelo Prof. Brauliro Leal e apresentado na Universidade Federal do Vale do São Francisco, em 2012.
- [47] SANTOS, Filipe - *Monitorização de Provas de Educação Física*. Dissertação de Mestrado em Engenharia Telecomunicações e Redes orientado pelo Dr. Joaquim Azevedo e apresentada na Universidade da Madeira, em 2009.
- [48] *XBee & XBee-PRO OEM RF Module Antenna Considerations*. Digi Application Note, 2012.
- [49] Tennina, S.; Di Renzo, M.; Graziosi, F.; Santucci, F. - *Locating ZigBee Nodes Using the TI's CC2431 Location Engine: A Testbed Platform and New Solutions for Positioning Estimation of WSNs in Dynamic Indoor Environments*.
- [50] *XBee Series 2 OEM RF Modules*. MaxStream, 2007.

## Anexo A. Especificações módulo XBee Series 2

Specification	XBee Series 2
<b>Performance</b>	
Indoor/Urban Range	up to 133 ft. (40 m)
Outdoor RF line-of-sight Range	up to 400 ft. (120 m)
Transmit Power Output (software selectable)	2mW (+3dBm)
RF Data Rate	250,000 bps
Serial Interface Data Rate (software selectable)	1200 - 230400 bps (non-standard baud rates also supported)
Receiver Sensitivity	-95 dBm (1% packet error rate)
<b>Power Requirements</b>	
Supply Voltage	2.8 – 3.4 V
Operating Current (Transmit)	40mA (@ 3.3 V)
Operating Current (Receive)	40mA (@ 3.3 V)
Power-down Current	< 1 uA @ 25°C
<b>General</b>	
Operating Frequency Band	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)
Operating Temperature	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip, RPSMA, or U.FL Connector
<b>Networking &amp; Security</b>	
Supported Network Topologies	Point-to-point, Point-to-multipoint, Peer-to-peer & Mesh
Number of Channels (software selectable)	16 Direct Sequence Channels
Addressing Options	PAN ID and Addresses, Cluster IDs and Endpoints (optional)
<b>Agency Approvals</b>	
United States (FCC Part 15.247)	Pending
Industry Canada (IC)	Pending
Europe (CE)	Pending



## Anexo B. Mapa Valores RSSI para diferentes distâncias e diferentes ângulos.

Ângulo Orientação (°)	0	10	20	30	40	50	60	70	80	90	
Distância (m)											
<b>0</b>	26	26	26	27	27	26	26	26	26	26	
<b>0,05</b>	27	27	27	28	28	28	27	26	26	27	
<b>0,1</b>	28	28	28	28	28	29	29	27	26	28	
<b>0,15</b>	28	28	30	29	29	29	29	28	27	30	
<b>0,2</b>	29	29	32	31	31	31	31	28	28	32	
<b>0,25</b>	30	30	35	33	32	31	33	29	29	36	
<b>0,3</b>	32	32	38	35	35	32	35	29	30	37	
<b>0,35</b>	33	33	43	38	37	33	37	30	31	40	
<b>0,4</b>	36	36	44	40	37	35	39	31	33,5	44	
<b>0,45</b>	37	37	44	41	39	35	40	32	36	47	
<b>0,5</b>	40	40	46	41	40	37	43	34	37	54	
<b>0,55</b>	42	42	53	42	41	39	44	34	38	58	
<b>0,6</b>	44	44	55	46	43	41	45	35	38	63	
<b>0,65</b>	46	46	63	49	44	42	45	38	38	75	
<b>0,7</b>	49	49	59	52	44	43	45	40	38	60	
<b>0,75</b>	52	52	55	55	43	44	46,5	41	38	53	
<b>0,8</b>	46	46	58	52	43	44	46	41	40	54	
<b>0,85</b>	48	48	56	52	44	48	46	40	42	57	
<b>0,9</b>	46	46	63	49	46	51	47	42	44	63	
<b>0,95</b>	51	51	61	51	48	50	52	43	43	70	
<b>1</b>	49	49	55	52	48	52	54	44	43	64	
<b>1,05</b>	48,5	48,5	58	51	49	50	52	44	44	55	
<b>1,1</b>	50	50	57	51	50	46,5	52	46	45	56	
<b>1,15</b>	51	51	57	52	51	48	52	48	45	58	
<b>1,2</b>	50	50	56	53	53	48	52	48	47	56	
<b>1,25</b>	52	52	55	51	54	49	55	46,5	46	58	

<b>1,3</b>	56	56	56	53	53	50	58	47	47	59
<b>1,35</b>	56	56	54	52	59	53	56	49	47	58
<b>1,4</b>	57	57	56	52	56	52	58	48	48	61
<b>1,45</b>	60	60	58	50	56	52	61	46	48	65
<b>1,5</b>	57,5	57,5	58	51	58	52	62,5	49	48	71
<b>1,55</b>	59	59	62	51	67	52	61	48	49	63
<b>1,6</b>	58	58	60	53	58	51	59	48	49	75
<b>1,65</b>	58	58	60	54	68	53	66,5	53	50	76
<b>1,7</b>	59	59	60	56	76	51	67	50	50	68
<b>1,75</b>	60	60	61	53	60	51	57	50	52	74
<b>1,8</b>	58	58	58	54	66	54	63,5	51	54	66
<b>1,85</b>	60	60	67	55	68	53	69	52	53	76
<b>1,9</b>	60	60	59	55	60	56	67	51	53	65
<b>1,95</b>	63	63	65	58	57	53,5	60	50	55	67
<b>2</b>	65,5	65,5	65	54	60	56	61	51	54	77

## Anexo C. Código C# da aplicação desenvolvida

```
*
* Created by SharpDevelop.
* User: LuísCouto
* Date: 21-10-2013
* Time: 15:53
*
* To change this template use Tools | Options |
Coding | Edit Standard Headers.
*/
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;
using System.Threading;
namespace receberV2
{
    /// <summary>
    /// Description of MainForm.
    /// </summary>
    public partial class MainForm : Form
    {
        public MainForm()
        {
            //
            // The InitializeComponent() call is
required for Windows Forms designer support.
            //
            InitializeComponent();

            //
            // TODO: Add constructor code after
the InitializeComponent() call.
            //

            try{
                if(!portaserie.IsOpen) //abrir
                porta série
                {
                    portaserie.PortName=
"COM5";
                    portaserie.Open();

                }
            }catch
            {
                richTextBox1.Text= "Erro";
            }
        }
    }
}
```

```

    }

    byte[] buf;
    byte[] bu;
    double [] val;
    double [] dist;
    double media=0, media20=0, med=0,
med20=0;
    int flag =0;
    int x = 1;
    int y=0;

    void Button1Click(object sender,
EventArgs e)// botão para mapear pontos
    {
        richTextBox1.Clear();
        richTextBox2.Clear();

        if(portaserie.IsOpen)
        {
            if (flag==0)
            {
                timer1.Start();
                double [] pontos;
                pontos = new double[2];
                flag=1;
            }
        }

    }

    void Timer1Tick(object sender, EventArgs
e)
    {
        double [] pontos;
        pontos = new double[2];
        x++;
        byte[] bytesToSend3 = new byte[19]
{0x7E, 0x00, 0x0F, 0x17, 0x02, 0x00, 0x13, 0xA2,
0x00, 0x40, 0x79, 0x5A, 0xFF, 0xFF, 0xFF, 0x00,
0x44, 0x42, 0x9B};
        portaserie.Write(bytesToSend3, 0,
19);//pedido RSSI

        //Thread.Sleep(100);
        bu = new
byte[portaserie.BytesToRead];// byte array com
tamanho dos dados que estão no buff

```

```

        byte[] bytesToSend = new byte[20] {
0x7E, 0x00, 0x10, 0x10, 0x01, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0x00,
0x00, 0x00, 0x41, 0xB1};

        portaserie.Write(bytesToSend, 0,
20);//envia A router referência

        Thread.Sleep(2000);

        bu = new
byte[portaserie.BytesToRead];// byte array com
tamanho dos dados que estão no buff
        int read1 = portaserie.Read(bu, 0,
portaserie.BytesToRead);

        switch (bu[11])
        {
            case 90 :

                richTextBox1.Text +=
bu[18]+ " ";

                media=media+bu[18];
                break;

            case 91 :

                richTextBox1.Text +=
bu[18]+ " ";

                break;
        }

        switch (bu[41])
        {
            case 90 :

                richTextBox2.Text +=
bu[46]+ " ";

                break;

            case 91 :

                richTextBox2.Text +=
bu[46]+ " ";

                med=med+bu[46];

                break;
        }

        if(x==21)// fez 20 leituras
        {

```

```

        media20= media/20;
        med20=med/20;

        richTextBox1.Text
+=comboBox1.SelectedItem.ToString() +" " +
"Guardado"+"\\r";
        richTextBox1.Text += "Media";
        richTextBox1.Text += "\\r" +
media20;

        richTextBox2.Text
+=comboBox1.SelectedItem.ToString() +" " +
"Guardado"+"\\r";
        richTextBox2.Text += "Media";
        richTextBox2.Text += "\\r" +
med20;

        if (y<2)
        {
            String nomeArquivo =
"C:/testeV2.txt";

                                if
(!System.IO.File.Exists(nomeArquivo))
System.IO.File.Create(nomeArquivo).Close();

            System.IO.TextWriter arqTXT =
System.IO.File.AppendText(nomeArquivo);

            arqTXT.WriteLine(media20); // guarda valor
médio RSSI no ficheiro txt Nó 1

            arqTXT.Close();

                                String
nomeArquivo1 = "C:/testeV21.txt";

                                if
(!System.IO.File.Exists(nomeArquivo1))
System.IO.File.Create(nomeArquivo1).Close();

            System.IO.TextWriter arqTXT1 =
System.IO.File.AppendText(nomeArquivo1);

            arqTXT1.WriteLine(med20); // guarda valor médio
RSSI no ficheiro txt Nó 2

```

```

    argTXT1.Close();

        y++;
    }
    else {

        y=0;

    }

    x=1;
    timer1.Stop();
    media=0;
    media20=0;
    med=0;
    med20=0;
    flag=0;

    }

}

void Button2Click(object sender,
EventArgs e) //botão para realizar localização
{

    richTextBox3.Clear();

    double menor =100000;
    int posicao =0;

    double [] val;
    double [] dist;
    val = new double[16];
    dist = new double[16];

    double []ponto ;
    double [] pont;
    ponto = new double[16];
    pont = new double[16];

    int valor=0;
    int valor1=0;

```

```

        byte[] bytesToSend3 = new byte[19] {0x7E,
0x00, 0x0F, 0x17, 0x02, 0x00, 0x13, 0xA2, 0x00,
0x40, 0x79, 0x5A, 0xFF, 0xFF, 0xFF, 0x00, 0x44,
0x42, 0x9B};
        portaserie.Write(bytesToSend3, 0,
19); //pedido RSSI

        //Thread.Sleep(100);
        bu = new
byte[portaserie.BytesToRead]; // byte array com
tamanho dos dados que estão no buff

        byte[] bytesToSend = new byte[20] {
0x7E, 0x00, 0x10, 0x10, 0x01, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0x00,
0x00, 0x00, 0x41, 0xB1};

        portaserie.Write(bytesToSend, 0,
20); //envia A router referência

        Thread.Sleep(5000);

        bu = new
byte[portaserie.BytesToRead]; // byte array com
tamanho dos dados que estão no buff
        int read1 = portaserie.Read(bu, 0,
portaserie.BytesToRead);

        switch (bu[11])
        {
            case 90 :

                richTextBox1.Text +=
bu[18]+ " ";

                valor=bu[18];
                break;

            case 91 :

                richTextBox1.Text +=
bu[18]+ " ";

                break;
        }

        switch (bu[41])
        {
            case 90 :

                richTextBox2.Text +=
bu[46]+ " ";

                break;
        }

```

```

        case 91 :

                                richTextBox2.Text +=
bu[46]+ " ";
                                valor1=bu[46];

break;
        }

string[] lines =
System.IO.File.ReadAllLines("C:/testeV2.txt"); //
lê valores txt nó1

string[] lines1 =
System.IO.File.ReadAllLines("C:/testeV21.txt");//
lê valores txt nó 2

        for(int i= 0;i<lines.Length ;i++)
        {
                ponto[i]= Convert.ToDouble(lines[i]) ;
                pont[i]= Convert.ToDouble(lines1[i]) ;

        }
        for(int i = 0; i<ponto.Length;i++) //calculo
distancia para cada ponto
        {

                val[i]= Math.Sqrt(Math.Pow(ponto[i] - valor, 2)
+ Math.Pow(pont[i] - valor1, 2));
        }

                for (int i = 0; i < val.Length; i++)
                {
                        richTextBox3.Text += i + "--
>";
                        richTextBox3.Text += val[i]
+" \r";
                }

                for (int i = 0; i < val.Length; i++)
                {

                        if (val[i] < menor)
                        {
                                menor =val[i];
                                posicao = i+1;
                        }

                }
                richTextBox3.Text += "\r";
                richTextBox3.Text += menor + " ";

```

```

        richTextBox3.Text += "Posição"+
posicao +" \r ";

        switch(posicao)
        {
                case 1 :
                        pictureBox1.Image =
Image.FromFile("C:/pontos/ponto1.jpg");

                        break;

                case 2 :
                        pictureBox1.Image =
Image.FromFile("C:/pontos/ponto2.jpg");

                        break;

                case 3 :
                        pictureBox1.Image =
Image.FromFile("C:/pontos/ponto3.jpg");

                        break;
                case 4 :
                        pictureBox1.Image =
Image.FromFile("C:/pontos/ponto4.jpg");

                        break;

                case 5 :
                        pictureBox1.Image =
Image.FromFile("C:/pontos/ponto5.jpg");

                        break;

                case 6 :
                        pictureBox1.Image =
Image.FromFile("C:/pontos/ponto6.jpg");

                        break;
                case 7 :
                        pictureBox1.Image =
Image.FromFile("C:/pontos/ponto7.jpg");

                        break;

                case 8 :
                        pictureBox1.Image =
Image.FromFile("C:/pontos/ponto8.jpg");

                        break;

                case 9 :
                        pictureBox1.Image =
Image.FromFile("C:/pontos/ponto9.jpg");

```

```
        break;

        case 10 :
            pictureBox1.Image =
Image.FromFile("C:/pontos/ponto10.jpg");

            break;

        case 11 :
            pictureBox1.Image =
Image.FromFile("C:/pontos/ponto11.jpg");

            break;

        case 12 :
            pictureBox1.Image =
Image.FromFile("C:/pontos/ponto12.jpg");

            break;

        case 13 :
            pictureBox1.Image =
Image.FromFile("C:/pontos/ponto13.jpg");

            break;

        case 14 :
            pictureBox1.Image =
Image.FromFile("C:/pontos/ponto14.jpg");

            break;

        case 15 :
            pictureBox1.Image =
Image.FromFile("C:/pontos/ponto15.jpg");

            break;

        case 16 :
            pictureBox1.Image =
Image.FromFile("C:/pontos/ponto16.jpg");

            break;

    }

}
```



## Anexo D. Código C de funcionamento do Router Móvel

```
int envio = 0;
int incomingByte = 0;
int recebido = 0;
int buf[20];
int valor;
int x=0;
int flag=0;
String inputString = "";           // a string to
hold incoming data
boolean stringComplete = false;
  int tamanho=0;
void setup() {

  Serial.begin(9600);

}
void loop() {

  if (Serial.available() > 0)
  {

    if (flag==0)  {

      incomingByte = Serial.read();

    }
    if(flag==1)
    {
      x++;
      incomingByte = Serial.read();
      if(x==1){
        if( incomingByte==0)
        {
          Serial.print("Sucesso");

        }
      }
    }
    if(x==2)
    {

      envio=incomingByte;

      Serial.println(envio);

    }

  }

}
```



```

void resposta(int value)
{
    buf[0]= 0x7E; //byte inicial
    buf[1]= 0x00; //tamanho MSB
    buf[2]= 0x0F;//tamnho LSB
    buf[3]= 0x10;//tipo de trama
    buf[4]= 0x01;// id trama
    buf[5]= 0x00;// MSB endereço 64 bits
    buf[6]= 0x00;
    buf[7]= 0x00;
    buf[8]= 0x00;
    buf[9]= 0x00;//LSB endereço 64 64 bits
    buf[10]= 0x00;
    buf[11]= 0x00;
    buf[12]= 0x00;
    buf[13]= 0xFF;//MSB endereço 16 bits
    buf[14]= 0xFE;//LSB endereço 16 64 bits
    buf[15]= 0x00;
    buf[16]= 0x00;
    buf[17]= value;// dados a enviar
//cheksum
    //buf[19]= 0xB1;// cheksum

    long sum = (buf[3] + buf[4] + buf[5] + buf[6]
+ buf[7] + buf[8] + buf[9] + buf[10] + buf[11] +
buf[12] + buf[13] + buf[14] + buf[15] + buf[16] +
value);

    // calculate the proper checksum

    buf[18]=( 0xFF - ( sum & 0xFF));

    for(int i=0;i<19;i++){
        Serial.write(buf[i]);
        delay(50);
    }
}

```