# TraceMe

## Sistema de localização de pessoas e objectos

**por**

**Maximino da Silva Paralta**

Dissertação submetida à
UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO
para obtenção do grau de
MESTRE
em Engenharia Electrotécnica e de Computadores, de acordo com o disposto no
DR – 1ª série – A, Decreto-lei n.º 74/2006 de 24 de Março e no
Regulamento de Estudos Pós-Graduados da UTAD
DR, 2ª série – Deliberação n.º 2391/2007

# TraceMe

## Sistema de localização de pessoas e objectos

**Por**

**Maximino da Silva Paralta**

**Orientador: Pedro Miguel Mestre Alves da Silva**

**Co-orientador: Rafael Ferreira da Silva Caldeirinha**

Dissertação submetida à
UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO
para obtenção do grau de
MESTRE
em Engenharia Electrotécnica e de Computadores, de acordo com o disposto no
DR – 1série – A, Decreto-lei n.º 74/2006 de 24 de Março e no
Regulamento de Estudos Pós-Graduados da UTAD
DR, 2ª série – Deliberação n.º 2391/2007

*Orientação Científica:*

Pedro Miguel Mestre Alves da Silva

Professor Auxiliar do

Departamento de Engenharias

Universidade de Trás-os-Montes e Alto Douro

Rafael Ferreira da Silva Caldeirinha

Professor Coordenador do

Departamento de Engenharia Electrotécnica

Escola Superior de Tecnologia e Gestão de Leiria do

Instituto Politécnico de Leiria

*Dedicado aos meus pais, irmãs e amigos.*

UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

Mestrado em Engenharia Electrotécnica e de Computadores

Os membros do Júri recomendam à Universidade de Trás-os-Montes e Alto Douro a aceitação da dissertação intitulada "TraceMe" realizada por Maximino da Silva Paralta para satisfação parcial dos requisitos do grau de Mestre.

Dezembro 2008

**Presidente:** **Carlos Manuel José Alves Serôdio,**
Professor Auxiliar do Departamento de Engenharias
da Universidade de Trás-os-Montes e Alto Douro

**Vogais do Júri:** **José Carlos Meireles Monteiro Metrôlho,**
Professor-adjunto do Departamento Engenharia Informática
da Escola Superior de Tecnologia
do Instituto Politécnico de Castelo Branco

**Pedro Miguel Mestre Alves da Silva,**
Professor Auxiliar do Departamento de Engenharias
da Universidade de Trás-os-Montes e Alto Douro

**Rafael Ferreira da Silva Caldeirinha,**
Professor Coordenador do Departamento de Engenharia Electrotécnica
da Escola Superior de Tecnologia e Gestão de Leiria
do Instituto Politécnico de Leiria

# Resumo

Neste trabalho foi desenvolvido o *software* de um sistema de RFID (*Radio-Frequency Identification*), designado por TraceMe, para determinar a localização em ambientes interiores em tempo real. Além da localização de pessoas e objectos, o sistema permite um mecanismo de controlo de acessos a áreas definidas, alarmes gerados pela falha de comunicação e funcionamento dos equipamentos instalados.

Do sistema proposto são descritos a especificação do sistema, a implementação do projecto, os testes de campo e análise de resultados. A especificação do sistema caracteriza todos os componentes e camadas incluídas na solução, as características de *software* e *hardware* necessários para a implementação do projecto, o algoritmo de localização e os sistemas que interagem com a solução, como a base de dados e o servidor *Web*. A implementação do projecto especifica os equipamentos escolhidos e o desenvolvimento do *software* efectuado. Os testes de campo e análise de resultados descrevem os resultados obtidos e análise do funcionamento do trabalho proposto.

**Palavras-chave:** RFID, sistemas de localização em ambientes interiores, *middleware*, Reverse Ajax, *Direct Web Remoting*, JDBC, Java.

# Abstract

This work consisted on the development a software system for RFID (Radio-Frequency Identification) system named TraceMe and intended to determine the indoor real time location. In addition, the system enables access contro of people and objects in defined areas, alarms generated by communication failure and the operability of the installed equipment.

In the proposed solution it is described the system specification, implementation of the project, field tests and analysis of results. The system's specification characterises all the components and layers comprised by this solution, the software and hardware requirements necessary for the project implementation, the location algorithm and systems that interact with the solution, such as the database and the Web server. In the project implementation section it is specified the selected equipment and the software development carried out. Field tests and analysis of results performed describe the achieved results and an analysis of the functioning of the proposed system.

**Key Words**: RFID, indoor location system, middleware, Reverse Ajax, Direct Web Remoting, JDBC, Java.

# Agradecimentos

Aos orientadores da dissertação, Professor Doutor Pedro Mestre e Professor Doutor Rafael Caldeirinha, pela preciosa orientação, disponibilidade e incentivo.

Ao Professor Doutor Carlos Serôdio e ao Gestor de Projecto da ISA™ Jorge Rodrigues pelos esclarecimentos, orientação e apoio incondicional ao longo da dissertação.

À ISA™ pela oportunidade de realizar esta dissertação no âmbito do trabalho no dia-a-dia.

Ao parceiro integrador da ISA dos equipamentos RFID, principalmente ao João Prudêncio e ao Carlos Dias, pela disponibilidade e ajuda na implementação dos equipamentos.

Ao pessoal do departamento de desenvolvimento de *software* da ISA™ pela ajuda, apoio e incentivo ao longo da dissertação.

A todos os que contribuíram directa ou indirectamente para a realização desta dissertação.

# Índice

# Lista de figuras

# Lista de tabelas

# Acrónimos e definições

| ACRÓNIMO | DESIGNAÇÃO |
|---|---|
| ADC | Analog-to-Digital Converter |
| AIX | Advanced Interactive eXecutive |
| AOA | Angle-Of-Arrival |
| API | Application Programming Interface |
| BM | Business Modules |
| DC | Direct Current |
| DEB | Berkeley Software Distribution |
| DWR | Direct Web Remoting |
| EE | Enterprise Edition |
| EHF | Extremely High Frequency |
| EPC | Electronic Product Code |
| ERP | Enterprise Resource Planning |
| GID-96 | General Identifier |
| GPS | Global Positioning System |
| HF | High Frequency |
| HP-UX | Hewlett Packard UniX |
| HTML | HyperText Markup Language |
| HTTP | Hypertext Transfer Protocol |
| ID | Identification |
| IFF | Identify Friend or Foe |

| | |
|---|---|
| IP | Internet Protocol |
| IPN | Instituto Pedro Nunes |
| ISA | Intelligent Sensing Anywhere, S.A. |
| ISM | Industrial Scientific Medical |
| ISO | International Organization for Standardization |
| JDBC | Java Database Connectivity |
| JES | Java Enterprise System |
| JSP | JavaServer Pages |
| LC | Inductor Capacitor |
| LF | Low Frequency |
| LIS | Location Information System |
| MF | Medium Frequency |
| MRID | Multiple Range Identification |
| PDA | Personal Digital Assistant |
| PHP | Hypertext Preprocessor |
| RF | Radio-Frequência |
| RFID | Radio-Frequency Identification |
| RPC | Remote Procedure Call |
| RSS | Received-Signal-Strength |
| RSSI | Received Signal Strength Indicator |
| RTLS | Real Time Location System |
| RTOF | Roundtrip-Time-Of-Flight |
| SAW | Surface Acoustic Wave |
| SDK | Software Development Kit |
| SHF | Super High Frequency |

| | |
|---|---|
| TCP/IP | Transmission Control Protocol / Internet Protocol |
| TDOA | Time Difference of Arrival |
| TOA | Time-Of-Arrival |
| UHF | Ultra High Frequency |
| UPC | Universal Product Code |
| VHF | Very High Frequency |
| VLF | Very Low Frequency |
| Wi-Fi | Wireless Fidelity |

# 1. Introdução

Este capítulo apresenta o enquadramento, os objectivos e a estrutura da dissertação. O enquadramento pretende apresentar a motivação, o sistema e a tecnologia utilizada no trabalho. Os objectivos apresentam o conceito e as funcionalidades do sistema proposto.

## 1.1. Enquadramento

Os dispositivos portáteis têm tido um elevado desenvolvimento nos últimos anos, com mais recursos e melhor capacidade de comunicação e processamento, permitindo no dia-a-dia cada vez maiores e mais complexos desafios. Este desenvolvimento é proporcionado pela rápida expansão das redes sem fios que permite um aumento de equipamentos móveis, e as aplicações daí obtidas são cada vez mais sofisticadas. As novas tecnologias de localização, com custos cada vez menores, permitem localizar um dispositivo móvel com precisão cada vez maior. Nas grandes empresas e em organizações, procura-se sempre melhorar os meios de comunicação, permitindo elevar o patamar de exigência da localização, pois a localização de cada membro de uma equipa ou a localização de um bem essencial pode ser importante para reduzir drasticamente o tempo de uma acção. Esta obrigação de rapidez de actos que leva, imperativamente, a uma necessidade absoluta de conhecer a localização de todos os meios necessários a uma determinada tarefa ou trabalho, faz com que as grandes empresas de todo o mundo apostem e direccionem os seus recursos cada vez mais para esta área [1].

Os sistemas de localização não só pretendem localizar pessoas, mas também objectos, mercadorias, meios de transporte e até mesmo animais, através de sistemas cada vez mais simples e mais económicos. Desde os mais simples objectos (fixos dentro de um armazém) até aos mais complexos objectos (automóveis), os sistemas de localização são hoje uma tecnologia muito apreciada pela indústria electrónica [2].

A tecnologia de RFID (*Radio-Frequency Identification*) tem sido uma das áreas das telecomunicações na qual têm dado grande empenho no desenvolvimento de sistemas de localização em ambientes interiores. Devido à sua fácil adaptação a todas as áreas de telecomunicações, esta tornou-se, de momento, uma das tecnologias mais procuradas e discutidas pela área de negócio.

## 1.2. Objectivos

Este trabalho propõe uma especificação de *software* para o sistema TraceMe a ser desenvolvido na ISA™ [1] (*Intelligent Sensing Anywhere*, S.A.) em parceria com a UTAD[2] (Universidade de Trás-os-Montes e Alto Douro) e com a ESTG[3] (Escola Superior de Tecnologia e Gestão de Leiria).

O sistema proposto pretende localizar dispositivos RFID, com pouca ou nenhuma intervenção humana no funcionamento do mesmo. Este sistema deverá possuir uma precisão que permita distinguir a localização entre zonas num ambiente interior, através de um algoritmo de localização a ser desenvolvido.

As principais funcionalidades do sistema são: a possibilidade de visualizar a localização de pessoas e objectos em tempo real; localização rápida de uma determinada pessoa ou objecto; identificação de determinadas pessoas ou objectos numa planta; seguimento de pessoas e objectos entre zonas; e permitir a consulta do histórico dos percursos efectuados por uma pessoa ou objecto.

Como o sistema deverá conseguir distinguir zonas físicas, é proposto um sistema de detecção e controlo de acessos às zonas pré-definidas. Assim, deverá ser possível configurar os acessos e detectar acessos não autorizados. Estas detecções deverão ser em tempo real e são consideradas como alarmes do sistema. Outro alarme que deverá ser implementado é a detecção de eventuais falhas no funcionamento dos dispositivos de RFID.

A solução deverá suportar várias integrações de sistemas, permitindo vários tipos de comunicação entre os diversos sistemas. Um dos sistemas é composto por diversos equipamentos RFID, permitindo flexibilidade e independência dos dispositivos. O acesso e a configuração deste sistema deverá ser de uma forma simples e intuitiva. Outro sistema a integrar deverá ser uma base de dados para armazenar toda informação relevante da solução. Um servidor *Web* também deverá ser integrado para alojar uma página *Web*, que permitirá aceder a todas as funcionalidades e configurações da solução. O *site Web* deverá ainda suportar actualizações dinâmicas, nomeadamente na visualização da localização em tempo real.

[1] Sítio da ISA: *http://www.isa.pt/* (Ultima vez visitado em Dezembro 2008)

[1] Sítio da UTAD: *http://www.utad.pt/* (Ultima vez visitado em Dezembro 2008)

[1] Sítio da ESTG: *http://www.estg.ipleiria.pt/* (Ultima vez visitado em Dezembro 2008)

## 1.3. Estrutura da Dissertação

O desenvolvimento do sistema proposto nesta dissertação envolve um estudo intenso da tecnologia em análise, a sua história, princípios de funcionamento, tipos de comunicação, características técnicas, protocolos e normas. Sendo assim é apresentado no segundo capítulo um resumo do estado da arte do RFID actual em sistemas de localização de ambientes interiores. Também são apresentados neste capítulo os métodos para sistemas de localização, *middlewares* comerciais relacionados com a localização, tecnologias de integração com um sistema RFID e soluções e aplicações de localização de pessoas ou objectos.

No terceiro capítulo é descrita a especificação do sistema proposto, incluindo uma breve descrição da solução, a arquitectura lógica e física, as características do *software* e *hardware*, o *middleware* que permite a comunicação com todos os componentes do sistema, os algoritmos de localização desenvolvidos, a estrutura da base de dados, o servidor *Web* e a interligação entre camadas.

A implementação da especificação é descrita no quarto capítulo, onde são apresentados os equipamentos RFID, as plataformas que interagem com a solução, o desenvolvimento efectuado no *middleware* e o desenvolvimento realizado nas plataformas.

O quinto capítulo é dedicado aos testes e validações, nomeadamente a definição da plataforma de testes, a descrição dos equipamentos utilizados, a descrição do *software* utilizado e os testes práticos realizados.

A conclusão da presente dissertação é apresentada no sexto e último capítulo. Neste capítulo são apresentadas algumas das conclusões obtidas e são feitas recomendações para trabalho futuro.

# 2. Sistemas de localização de pessoas em ambientes interiores

Neste capítulo são descritas as características técnicas, o fundamento e a uma breve história da tecnologia RFID, os equipamentos, o tipo de comunicação, os princípios de funcionamento, os protocolos e as normas, os *middlewares* comerciais, as soluções e as aplicações no âmbito da localização de pessoas em ambientes interiores.

## 2.1. Introdução

O sistema de localização mais conhecido é o sistema de posicionamento global por satélite (GPS), criado pelo Departamento de Defesa dos Estados Unidos na década de 80 [1]. O GPS é uma rede de satélites que fornece a posição, em coordenadas geográficas, de um ponto qualquer da superfície terrestre. Com a utilização desta tecnologia, surgiram várias aplicações que usam a informação de localização de um indivíduo para lhe fornecer serviços personalizados. Na década de 90 houve um maior impulso nos sistemas de localização. Os operadores de telecomunicações utilizam a usa própria rede para determinar a localização, apesar da precisão ser muito menor do que a oferecida pelo sistema GPS, o seu custo é muito inferior a este [1].

Em ambientes interiores os sistemas anteriormente descritos apresentam desempenhos reduzidos, permitindo a utilização de outras tecnologias, como o RFID, que permitiu criar aplicações que integram informação de localização com serviços. Além da localização, surgiram outros serviços como o controlo de acessos, gestão de *stocks* e controlo de assiduidade.

## 2.2. Breve história do RFID

A origem da tecnologia RFID está associada à 2ª Guerra Mundial pelo uso da técnica de transmissão por rádio frequência com o intuito da identificação automática. Este projecto foi criado sob o comando do escocês *Sir* Robert Watson-Watt [2], no qual os britânicos criaram o primeiro identificador activo de amigo ou inimigo (IFF, *Identify Friend or Foe*). A implementação do sistema na altura foi a colocação de um transmissor em cada avião britânico, quando esses transmissores recebiam sinais das estações de radar no solo, começavam a transmitir um sinal de resposta, que identificava o avião como amigo [2]. Na mesma década, em 1948, Harry Stockman fez uma investigação concreta sobre o conceito do

RFID, no seu trabalho "*Communication by Means of Reflected Power*" [3], a possibilidade do uso da potência reflectida como meio de comunicação.

A partir da década de 50, com os primeiros testes laboratoriais de pequenos dispositivos rádio, o RFID conheceu avanços significativos, sobretudo a partir da década de 60, cientistas e académicos dos Estados Unidos, Europa e Japão realizaram pesquisas e apresentaram estudos a explicar como a energia RF poderia ser utilizada para identificar objectos remotamente. No final dessa década, empresas começaram a comercializar sistemas anti-roubo rádio para determinar se um produto já tinha sido pago ou não. Estes sistemas eram constituídos por etiquetas de vigilância electrónica (EAS - *Eletronic Article Surveillance*) que utilizam um *bit* para distinguir se já foi pago ou não.[2].

Na década de 70 observou-se a explosão do desenvolvimento de sistemas RFID. Várias entidades aperceberam-se do enorme potencial desta tecnologia, começando as primeiras rivalidades, com o surgimento das primeiras patentes. A 23 de Janeiro de 1973, Mario Cardullo faz o registo de patente de uma etiqueta activa de RFID. Nesse mesmo ano, o californiano Charles Walton regista também uma patente de um *transponder* (*tag* [1]) passivo utilizado para destrancar a porta de um automóvel sem necessidade de chave. É ainda nesta década, que o interesse desta tecnologia passa a ser público e surgem os primeiros sistemas RFID para animais [4].

A partir da década de 80, o RFID entra definitivamente nos planos da indústria e do comércio mundial [4], nos Estados Unidos, em áreas de controlo de mercadorias, meios de transporte, acesso de pessoas e identificação animal. Na Europa, os esforços eram concentrados, principalmente, no uso da tecnologia para identificação animal, actividades industriais e controlo de acesso em rodovias [2].

Finalmente, a partir da década de 90, o RFID torna-se presente e largamente comum no dia-a-dia das pessoas, com o surgimento de normas reguladoras (conforme será apresentado na secção 2.7) e aplicações comerciais (secção 2.11) a custos reduzidos. Na Tabela 2-1 encontra-se um resumo do desenvolvimento do RFID ao longo dos tempos.

Tabela 2-1 – Resumo por décadas do desenvolvimento do RFID (retirado de [5]).

| Década | Eventos |
|---|---|
| 1940-1950 | Invenção e rápido desenvolvimento do radar durante a 2ª Guerra Mundial<br>Início de funcionamento do RFID em 1948 |
| 1950-1960 | Primeiras explorações da RFID e experimentações laboratoriais |
| 1960-1970 | Desenvolvimento da teoria da RFID e primeiras aplicações experimentais no terreno |
| 1970-1980 | Explosão no desenvolvimento da RFID e aceleração dos testes<br>Implementações embrionárias de RFID |
| 1980-1990 | Aplicações comerciais de RFID entram no mercado |
| 1990-2000 | Surgimento de normas<br>RFID é largamente utilizado começando a fazer parte da vida de cada um |

[1] *Tag* – Etiqueta electrónica de RFID.

Nos dias de hoje, a tecnologia de RFID está em quase todo o lado. O seu uso é tão usual que já nem se dá conta da sua presença. Assim, e como em muitos outros casos de novas tecnologias, poder-se-á dizer que foi uma ascensão rápida, forte e com enraizamento cada vez mais profundo em todos os sectores da sociedade [4]. Como por exemplo a tecnologia está integrada nos passaportes, portagens (Via Verde® [1]), bibliotecas, lojas e metro (por exemplo a *tag* andante do metro do Porto).

## 2.3. Dispositivos

Existem tipicamente três grandes grupos de dispositivos, as *tags*, os leitores e as antenas, que desempenham diferentes papéis num sistema RFID.

### 2.3.1. *Tags*

As *tags* são umas etiquetas electrónicas, que podem ser encontradas sob diferentes funcionamentos: activos, semi-activos ou passivos. As suas dimensões e aplicações variam conforme o formato, pelo que são descritos em seguida, de forma breve, as características de cada um deles:

- **Activos:** Ao contrário das *tags* passivas, as *tags* activas têm uma fonte de energia interna que alimenta o seu circuito integrado e fornece energia para o envio de sinais de transmissão de dados para o leitor. Por oposição as *tags* passivas, estes não necessitam da energia do sinal recebido para funcionar, podendo assim ter um papel mais independente do leitor. Este tipo de funcionamento permite à *tag* a realização de tarefas mais complexas, deixando este que deixa de ser um simples "espelho de identificação" para o leitor. As *tags* activas são, por isso, geralmente maiores, mais complexos e com um alcance muito superior em relação aos *tags* passivas. Têm também uma maior capacidade de armazenamento de dados, uma memória para escrita superior e suportam componentes exteriores como sensores ou outros dispositivos semelhantes. Na Figura 2-1 encontra-se alguns exemplos de *tags* activas [6].

Figura 2-1 – Exemplo de uma *tag* activa (retirado de [4]).

[1] Sítio da Via Verde®: http://www.viaverde.pt/ViaVerde/vPT/ (Ultima vez visitado em Dezembro 2008)

- **Passivos:** Não necessitem de bateria (ou outra fonte de energia) interna para o seu funcionamento. Em vez disso *a tag* aproveita a energia enviada pelo leitor para alimentar os seus circuitos e transmitir os seus dados armazenados de volta. Uma *tag* passivo tem que ter por isso uma constituição muito simples e com um número de elementos reduzido. Em virtude da ausência de bateria, a *tag* passiva pode ter uma longa vida de funcionamento sem precisar de qualquer manutenção. Pode suportar condições mais extremas sem colocar em causa o seu funcionamento, é geralmente mais pequeno que as *tags* activas e a sua produção em massa originam a ter custos de produção muito baixos. Na Figura 2-2 encontra-se alguns exemplos de *tags* passivas [6].

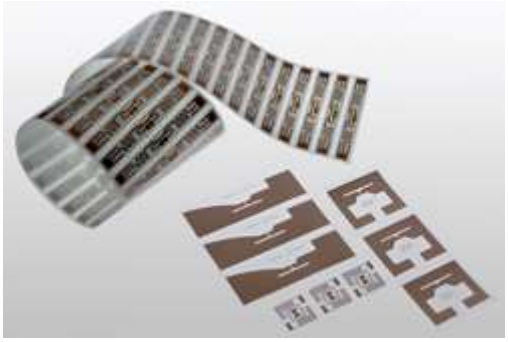


Figura 2-2 – Exemplos de *tags* passivas (retirado de [7]).

- **Semi-Activos:** Estas *tags* são um híbrido de tecnologias dos dois outros funcionamentos, agrupando algumas vantagens e também desvantagens. Usualmente permitem que se consigam alcances iguais as *tags* activos e são igualmente alimentados. A principal diferença para com as *tags* activos está relacionada com o facto de as semi-activas não estarem permanentemente activos, ou seja, necessitam de receber um sinal eléctrico proveniente de uma antena para que estabeleçam uma comunicação. Além disso são igualmente menos dispendiosos que as *tags* activas [6].


Figura 2-3 – Exemplo de uma *tag* semi-activa (retirado de [8]).

Existem inúmeras bandas nas quais os diversos tipos de *tags* operam, as vantagens e as desvantagens de utilizar determinadas frequências encontrar-se na Tabela 2-2. Esta análise centra-se nas soluções actualmente disponíveis no mercado e poderão sofrer alterações de futuro.

Tabela 2-2 – Vantagens e desvantagens de cada Banda de Frequência (retirado de [6]).

| Banda de Frequência | Benefícios | Problemas | Aplicações Típicas |
|---|---|---|---|
| 100-500 kHz | • Baixo custo<br>• Melhor penetração em objectos não metálicos | • Baixo a médio alcance de leitura<br>• Velocidade de leitura baixa | • Controlo de acessos<br>• Controlo de inventário |
| 10-15 MHz | • Baixo a médio alcance de leitura<br>• Velocidade de leitura média | • Apresenta custos superiores às da banda inferior | • Controlo de acessos |
| 850-950 MHz | • Grande alcance de leitura<br>• Velocidade de leitura elevada | • Dispendioso em termos de hardware | • Identificação de veículos e sistemas de controlo de entradas |
| 2,4-5,8 GHz | • Grande alcance de leitura<br>• Velocidade de leitura elevada | • Dispendioso em termos de hardware | • Identificação de veículos e sistemas de controlo de entradas<br>• RTLS (*Real Time Location System*) - Geração 802.11 de WLAN |

As *tags* podem operar em multi-frequências, utilizando diferentes bandas de frequências para suportar várias aplicações. Recorrendo a Tabela 2-2, uma *tag* para funcionar num sistema de controlo de acessos e identificação de veículos precisa de utilizar duas bandas de frequências, para o controlo de acessos podia ser a banda 100-500kHz ou 10-15 MHz, para a identificação podia ser a banda 850-950 MHz ou 2,4-5,8 GHz. Portanto a *tag* teria dois modos de operação com frequências diferentes, F1 (para ser detectada numa curta distância – controlo de acessos) e F2 (para conseguir transmitir em grandes distâncias – identificação). Através do conceito das frequências F1 e F2 a *tag* consegue suportar várias aplicações distintas.

### 2.3.2. Leitores

Um leitor é um aparelho que permite ler, interpretar e escrever (ao contrário do que o nome faria supor) *tags* RFID. Para tal efeito, o leitor liga-se a uma ou mais antenas por intermédio de um qualquer *interface* definido pelo construtor (mais vulgarmente cabos coaxiais ou de outro género) e usa-as para emitir ondas de rádio, com energia normalmente fornecida pelo próprio leitor. Se uma *tag* encontra no campo de acção da antena, este irá usar a energia dela recebida para emitir o seu próprio sinal, tal como atrás referido. Esse sinal é então capturado por uma ou mais antenas que o transmitem ao leitor e este, por sua vez, irá traduzir o sinal recebido que contém a informação da *tag*. O leitor

permite então redireccionar essa informação da maneira que o utilizador quiser. Na Figura 2-4 é ilustrado um exemplo de um leitor fabricado pela Symbol™ [1] [6].

Figura 2-4 – Leitor fabricado pela Symbol ™ (retirado de [6]).

### 2.3.3. Antenas

Outra parte fundamental destes sistemas consiste nas antenas. Tal como já foi referido, são elas que fazem a interligação entre os leitores e as *tags*, possibilitando a comunicação entre ambos. Normalmente são alimentadas pelo próprio leitor, mas podem ter alimentação própria. Na Figura 2-5 exemplifica as antenas do tipo de parede, HF (*High Frequency*) e em portal [6].


Figura 2-5 – Tipos de antenas: A – Antena de parede; B – Antena de HF; C – Antenas em portal (retirado de [6]).

As antenas de parede são as mais comuns, atinge a maioria das necessidades, uma vez que (à semelhança de qualquer outra) também se podem agrupar podendo fazer as vezes de antenas de portal. Mas o que as distingue para além do seu formato e tamanho é o seu diagrama de radiação e polarização que influenciam em muito a sua eficiência. Tal como outro tipo de antenas, as antenas RFID tem uma polarização circular ou linear (Figura 2-6), dependendo da sua aplicação. De salientar que os leitores que têm as antenas incorporadas, têm o seu diagrama de radiação bastante direccional [9].

[1] Sítio da Symbol™: http://www.symbol.com (Ultima vez visitado em Dezembro 2008)

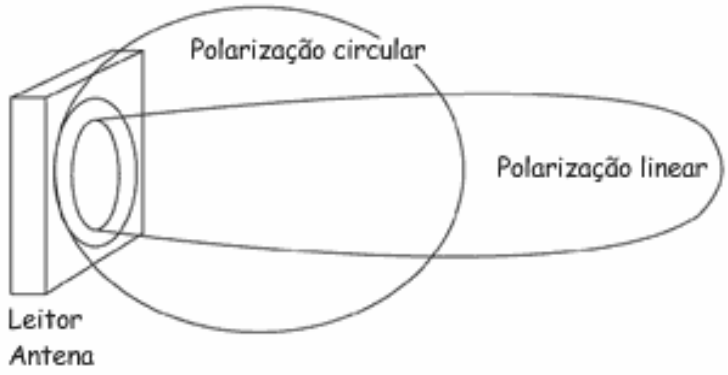



Figura 2-6 – Diagrama de radiação de uma antena RFID (retirado de [10]).

## 2.4. Tipos de comunicação

O processo de comunicação nos sistemas RFID é através de um sinal de rádio-frequência entre a *tag* e o leitor, a sua designação é acoplamento. Os principais tipos de acoplamento são os seguintes:

- **Acoplamento indutivo:** Também designado por aproximação magnética, nestes sistemas a comunicação é feita através da alteração (modulação) dos campos magnéticos em torno das antenas. Este tipo de comunicação é muito simples e baseia-se no princípio de ressonância dos circuitos LC (*inductor capacitor*). O leitor gera um campo magnético alternado com uma determinada gama de frequências. Se a frequência de ressonância do circuito LC estiver dentro dessa gama de frequências, existirá passagem de energia do leitor para o circuito ressonante através da sua indutância. Um exemplo do acoplamento é ilustrado na Figura 2-7.

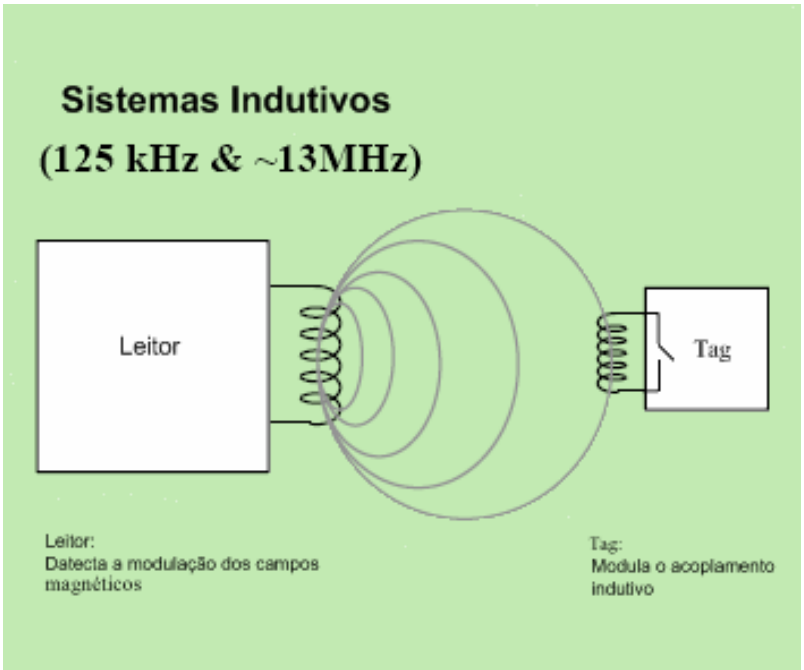



Figura 2-7 – Acoplamento indutivo (adaptado de [11]).

- **Acoplamento de propagação:** Também designado por radiofrequência, nestes sistemas existe a modulação de um sinal RF, que é transmitido entre as antenas dos dois terminais, como num vulgar sistema de rádio. A forma de retorno da informação da *tag* para o leitor é que varia consoante o tipo de *tag* que se está usar, do meio circundante envolvido, entre outros factores. Um exemplo do acoplamento é ilustrado na Figura 2-8.

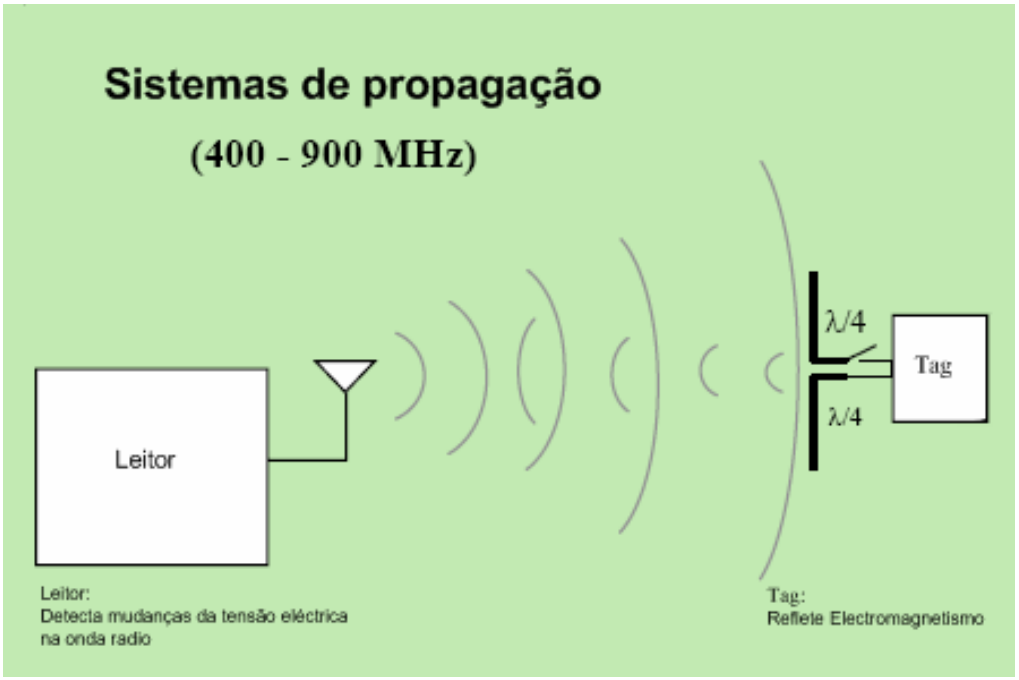



Figura 2-8 – Acoplamento de propagação (adaptado de [11]).

As *tags* com funcionamento por acoplamento indutivo trabalham nas baixas frequências, enquanto as *tags* com acoplamento por radiofrequência trabalham nas gamas de UHF (*Ultra High Frequency*) e microondas.

## 2.5. Princípios de funcionamento

Na Figura 2-9 são ilustrados os vários princípios de funcionamento de sistemas RFID, que descreve a interacção entre os leitores e as *tags*, em particular a fonte de alimentação das *tags* e a transferência de dados entre o leitor e a *tag*.

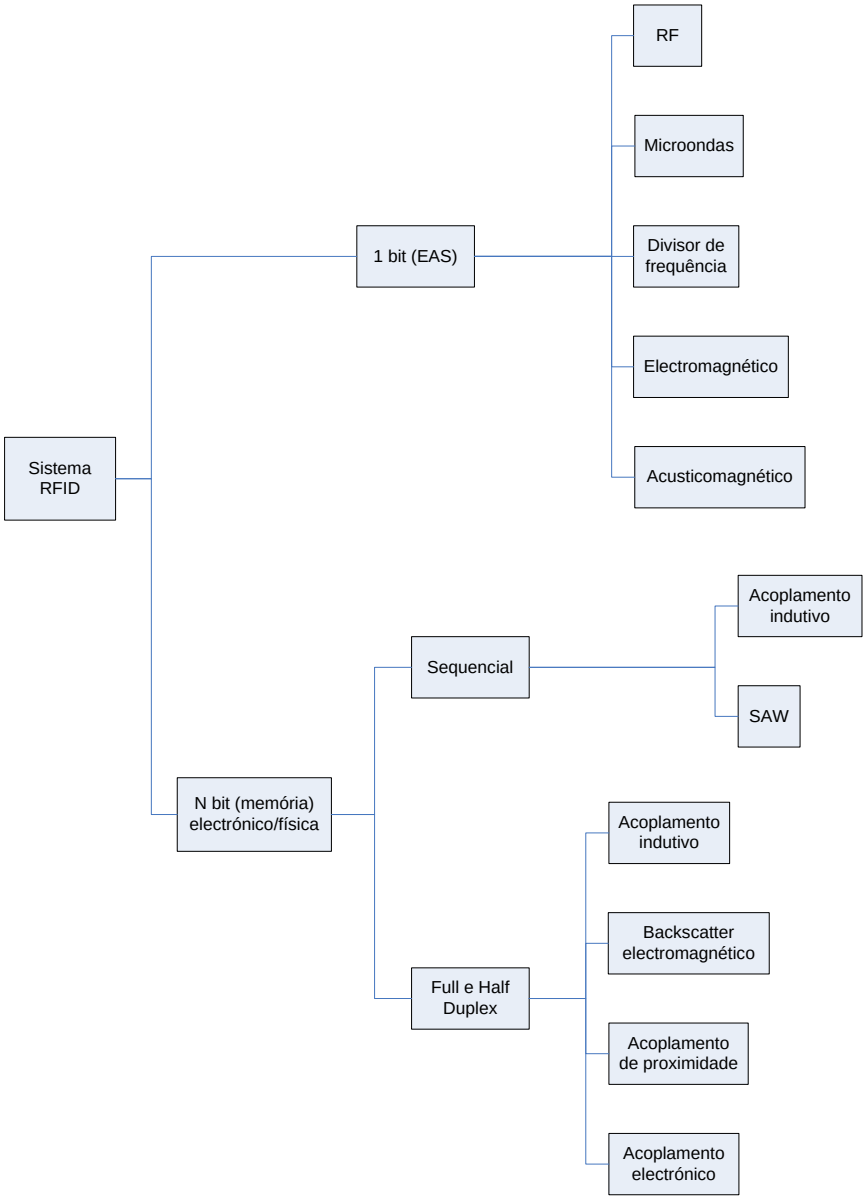



Figura 2-9 – Principais tipos de comunicação de um sistema RFID (adaptado de [12]).

Como se pode observar na Figura 2-9 podemos dividir os tipos de comunicação de um sistema RFID em dois grupos, o 1 *bit* e o N *bit*. O primeiro grupo está dividido por RF, microondas, divisor de frequências, electromagnético e o acústico-magnético. O segundo grupo corresponde a dois subgrupos, o Sequencial e o *Full* e *Half Duplex*. O primeiro subgrupo está dividido pelo acoplamento indutivo e pelo SAW (*Surface Acoustic Wave* - comunicação na dispersão superficial das ondas acústicas a baixa velocidade), o segundo subgrupo está dividido pelo acoplamento indutivo, *backscatter* electromagnético, acoplamento de proximidade e acoplamento electrónico.

O grupo 1 *bit*, o princípio de funcionamento baseia-se na transmissão de um bit que representa apenas dois estados, 1 e 0. Apesar desta limitação, o grupo 1 *bit* é bastante utilizado, a sua principal aplicação são as *tags* electrónicas anti-roubo nas lojas (EAS). Este tipo de funcionamento caracteriza-se por ser geralmente rápido e descontínuo, pois exige apenas uma resposta da *tag* para o leitor. As *tags*

pertencentes a este grupo não têm necessidade de grandes quantidades de informação nem electrónica complexa para o seu normal funcionamento [12].

O grupo N *bit* as *tags* utilizam microprocessador electrónicos como dispositivos de transporte de dados, mas tem uma baixa capacidade de armazenamento de dados, aproximadamente alguns kBs (*quilobyte*). Para ser possível escrever ou ler da *tag* é necessário transferir dados entre a *tag* e o leitor. Esta transferência ocorre através de um de dois procedimentos, *full* e *half duplex* ou sequencial. Este é o tipo de funcionamento mais indicado para *tags* com intuito de localização, pois pretende-se obter uma constante monitorização do deslocamento da *tag*, dentro de uma determinada área de cobertura do leitor [12].

## 2.6. Características técnicas

Devido ao facto dos sistemas RFID produzirem e radiarem ondas electromagnéticas, são classificados como sistemas de rádio. Portanto, é necessária a determinação da frequência de operação para que não existem interferências de outros serviços de rádio, televisão ou rádio móvel (policia, serviço de segurança, industria). Na implementação de um sistema RFID é necessário considerar a frequência de operação dos outros sistemas de rádio, pois estes limitam de forma significativa o funcionamento dos sistemas. Por isso, utilizam-se frequências reservadas especificamente para aplicações industriais, científicas ou médicas. Estas frequências são conhecidas como frequências ISM (*Industrial Scientific Medical*) as quais também podem ser utilizadas para aplicações em RFID. Na Figura 2-10 são apresentadas as frequências de operação utilizadas em sistemas RFID [13].

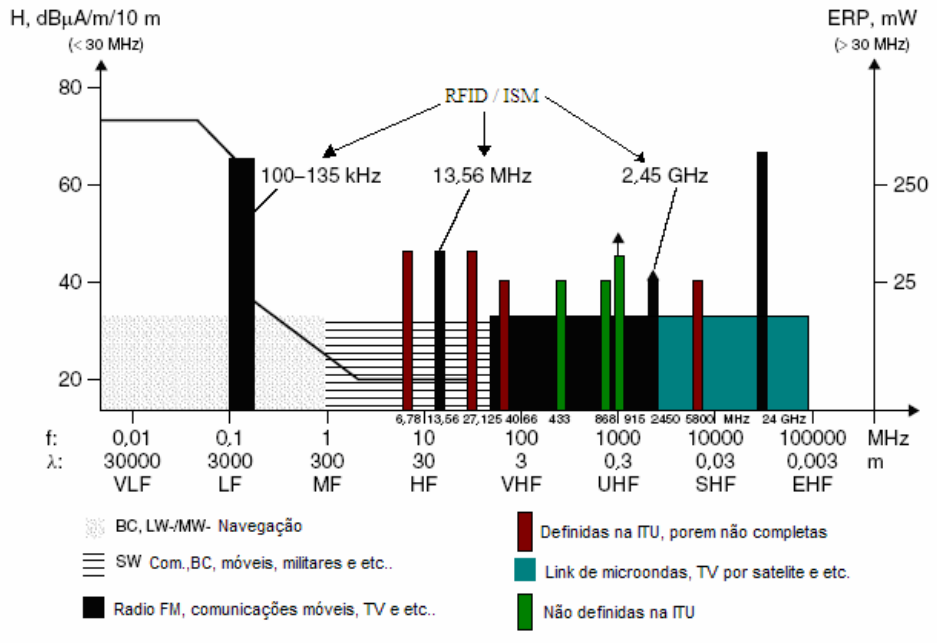



Figura 2-10 – Potências máximas permitidas por frequência em RFID (retirado de [13]).

Existem sistemas de RFID a funcionar desde ao 100kHz até aos 5,8GHz, prevendo-se que no futuro possam mesmo vir a ocupar frequências próximas dos 24GHz. Na Tabela 2-3, estão resumidas as bandas de funcionamento dos diferentes sistemas de RFID, bem como a regulamentação, alcance típico, vantagens e alguns comentários [4].

Tabela 2-3 – Resumo das bandas de funcionamento da RFID (adaptado de [4]).

| Frequência | Regulação | Alcance típico | Vantagens | Comentários |
|---|---|---|---|---|
| <135 kHz | Banda ISM, alta potência | <10cm (passivo) | Boa penetração em líquidos | Controlo acesso |
| 6.78 MHz<br>13.56 MHz<br>27.125 MHz | Banda ISM, regulação praticamente igual em todo o mundo | <1m (passivo) | Penetração média em líquidos | Smart Cards, Controlo de acesso, Imobilização de veículos |
| 433 MHz | Banda ISM para dispositivos de comunicação de curto alcance, Banda não uniforme | <100m (activo) | Funciona bem em ambientes com metais | *Tags* activas |
| 888-956 MHz | Banda não uniforme mundialmente | <10m (passivo US)<br><4m (passivo UE) | O melhor alcance para comunicações passivas | Normas Wal-Mart, DoD |
| 2.45 GHz | Banda ISM | <3m (passivo)<br><50m (SAW) | Alternativa para os 900 MHz | Wi-Fi, Bluetooth |
| 5.4-6.8 GHz<br>24,05-24,5 GHz | Bandas salvaguardadas para uso futuro | | | |

Apesar das diferentes bandas de funcionamento dos sistemas RFID, é possível agrupá-los em três grandes grupos consoante a frequência de operação. Assim, os sistemas de baixa frequência situam-se entre os 100-500 kHz, os sistemas de média frequência entre os 6.78 – 433 MHz e os de alta frequência acima dos 888 MHz [4].

O alcance é outro dos parâmetros importantes na caracterização dos sistemas. Desta forma, existem sistemas de curto alcance tipicamente até 10 cm e que funcionam a gamas de frequências abaixo dos 135kHz, sistemas de médio alcance com frequências a rondar os 15 MHz, e os sistemas de longo alcance para frequências UHF e microondas. O alcance também é fortemente condicionado pela arquitectura da *tag* utilizada, sabendo que o alcance das *tags* passivas é muito inferior ao alcance das *tags* activas [4].

A Figura 2-11 mostra a distribuição do mercado global, de 2000 a 2005, para *tags* nas várias frequências de operação, em milhões de unidades. Observa-se um acentuado crescimento no uso da tecnologia, em especial nas frequências de 13,56 MHz e nas frequências menores que 135 kHz [13].

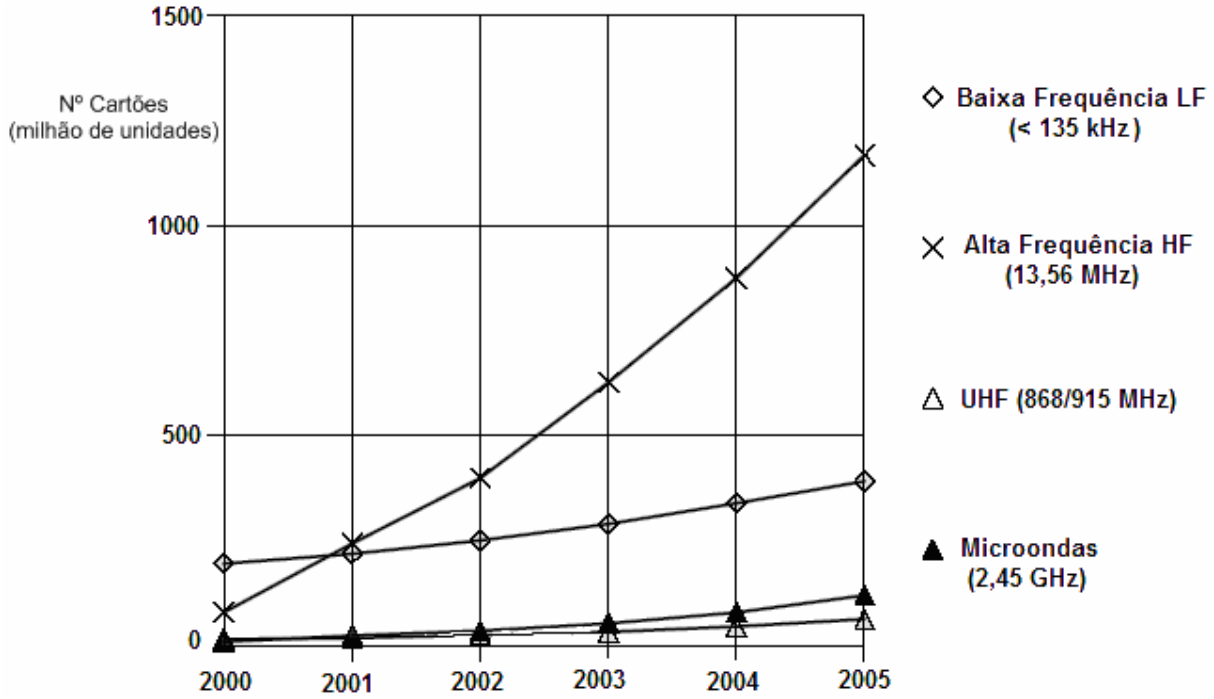



Figura 2-11 – Distribuição do mercado de *tags* por frequências de operação (adaptado de [13]).

## 2.7. Protocolos e normas

Dependendo do tipo de *tag*, a quantidade de informação pode variar desde alguns *bytes* a vários *Megabytes*. A informação dentro das *tags* pode ser organizada em vários formatos, desde que a *tag* e o leitor concordem num único formato de funcionamento. Muitos formatos são proprietários, contudo alguns *standards* estão a emergir. O EPC (*Electronic Product Code*) é considerado o *standard* RFID que irá substituir o UPC (*Universal Product Code)*, usado pelo código de barras. O novo EPC usa o GID-96 (*General Identifier*) da organização EPCglobal [1]. O GID-96 contém 96 *bits* ou seja 12 *bytes* de informação, exemplificado na Tabela 2-4 [6].

Tabela 2-4 – *Standard* EPC do tipo GID-96 (adaptado de [6]).

| | Cabeçalho | Empresa | Classe do Objecto | Série |
|---|---|---|---|---|
| **Número de bits:** | 8 | 28 | 24 | 36 |
| **Números possíveis:** | | 268,435,455 | 16,777,215 | 68,719,476,735 |

Com a informação organizada desta forma permite que 30,939,155,745,879,204,468,201,375 números únicos sejam criados. O sistema RFID funciona quando uma antena transmite sinais de rádio, esses sinais são detectados pela *tag* emite um sinal de rádio como resposta. Esse sinal é posteriormente interpretado pelo receptor. Caso a *tag* tenha alguma potência de computação, pode efectuar funções de encriptação e desencriptação. Algumas *tags* só permitem o funcionamento de leitura aos seus dados,

[1] Sítio da EPCglobal: *http://www.epcglobalinc.org/home* (Ultima vez visitado em Dezembro 2008)

enquanto outras permitem leitura e escrita. Dependente do funcionamento das *tags*, o leitor pode efectuar a escrita na *tag*. A EPCglobal e a ISO[1] (*International Organization for Standardization*) já definiram vários protocolos, alguns estão descritos na Tabela 2-5 [6].

Tabela 2-5 – Protocolos desenvolvidos pela EPCglobal e ISO (adaptado de [6]).

| Protocolo | Capacidades |
|---|---|
| EPC Generation 1 – Class 0 | "*Read-Only*", pré-programado |
| EPC Generation 1 – Class 1 | "*Write-Once*", "*Read-Many*" |
| ECP Generation 2.0 Class 1 | *"Write-Once", "Read-Many",* versão aceite globalmente. |
| ISO 18000 Standard | "*Read-Only*", pode conter memória para que dados do utilizador possam ser escritos. Este protocolo é composto por diferentes secções dependendo da frequência usada e da intenção de uso. |
| ISO 15963 | *Unique Tag* ID |
| ISO 15961 | Protocolos de dados: Regras de codificação dos dados e funções lógicas de memória. |
| ISO 15962 | Protocolos de dados: interface da aplicação. |

Em Anexo A.1 encontra-se uma lista dos principais fabricantes de sistemas RFID, bem como os produtos comercializados.

## 2.8. Métodos para sistemas de localização

Numa primeira análise, existem duas formas de cálculo da posição de uma *tag*, o *self-positioning*, em que a *tag* calcula a sua posição através dos sinais recebidos de vários leitores com localizações fixas e o *remote-positioning*, em que a localização é feita pela agregação dos sinais recebidos pelos componentes da rede. Normalmente para sistemas de baixo custo o método *remote-positioning* é uma hipótese mais aceitável [14].

Os principais tipos de efectuar o cálculo da distância são as seguintes [15]:

- **Ângulo de Chegada (AOA, *angle-of-arrival*)** – depende bastante do ambiente em redor. Em distâncias maiores torna-se pouco fiável e precisa de agregados complexos de antenas para o cálculo do ângulo de chegada.
- **Potência do sinal (RSS, *received-signal-strength*)** – também é bastante dependente do ambiente em redor. É fortemente prejudicada pela atenuação multi-percurso e *shadowing* e exige exactidão de medição de potência extremamente acentuada para variações de 1 metro de distância.

[1] Sítio da ISO: *http://www.iso.org/iso/home.htm* (Ultima vez visitado em Dezembro 2008)

- **Diferença temporal** – é um método mais simples, onde a exactidão da medida não se degrada substancialmente com a distância. Dentro deste método existem três formatos diferentes, o TOA (*time-of-arrival*), o RTOF (*roundtrip-time-of-flight*) e o TDOA (*time-difference-of-arrival*). No primeiro método, a existência de sincronismo entre o leitor e a *tag* é essencial para uma estimação correcta do tempo de chegada. Já no segundo caso, embora o sincronismo não seja tão influente, existe a necessidade de uma transmissão em *full-duplex* com divisão espectral (uma frequência para *uplink* e uma divisão para *downlink*). No terceiro caso, estas duas imposições não são necessárias, deixando mesmo de haver necessidade da existência de um relógio ou um oscilador local na *tag*. Este passa a funcionar como "espelho". Exige-se, contudo, sincronismo entre os vários leitores.

## 2.9. *Middleware* RFID comerciais

Além das *tags* e dos leitores, um sistema RFID necessita de um *middleware* para transformar dados em informação. O *middleware* recebe dados provenientes das *tags*, detectados pelos leitores, e reenvia para aplicações ou para uma base de dados. A sua principal função é controlar o fluxo de informação entre estes componentes com funções básicas como filtragem de dados inválidos ou redundantes. O *middleware* pode ser considerado como o responsável pela qualidade e usabilidade da informação [16].

Esta secção apresenta uma vista geral de alguns *middlewares* RFID disponíveis no mercado. A inclusão destes é uma amostra representativa e uma breve descrição dos produtos disponíveis. Através destes exemplos é proposto um *middleware* para o projecto TraceMe. As empresas dos *middlewares* estudadas são as seguintes:

- **Sun Microsystems**™ – Esta empresa foi uma das primeiras a entrar no mercado de RFID. A Sun™ fornece uma plataforma baseada em Java designada por "Sun Java System RFID Software". O *software* de RFID da Sun™ é concebido especificamente para suportar níveis elevados de confiabilidade e de escabilidade para as redes de EPC, permite ainda integrar outros sistemas no sistema. Este *software* faz parte da JES (*Java Enterprise System*) e suporta integrações de servidores *standards*, incluindo o "Sun Java Enterprise Integration Server". Os componentes deste projecto são o "RFID Event Manager", o "RFID Management Console", o "RFID Information Server" e o SDK (*Software Development Kit*) para criar adaptadores e aplicações [17]. A Figura 2-12 corresponde aos 3 primeiros componentes.

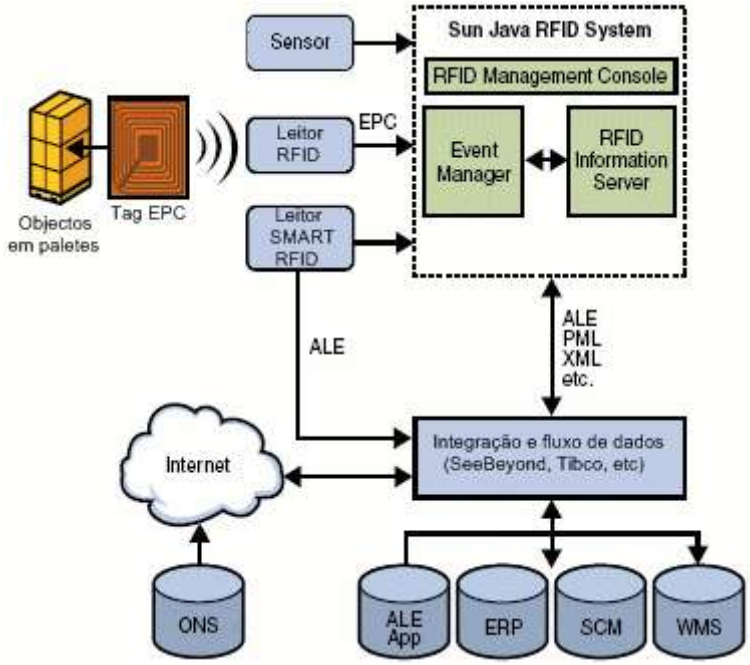



Figura 2-12 – Arquitectura do "Sun Java System RFID Software" (retirado de [18]).

O "RFID Event Manager" processa dados das *tags* que são transmitidos pela rede de leitores. Esse processamento consiste na filtragem, agregação, contagem dos ID's das *tags* e comunica ainda com sistemas integrados. O "RFID Information Server" é uma aplicação Java EE (*Enterprise Edition*) que permite uma *interface* para detecção e validação da informação relativa à *tag*. A informação armazenada inclui o ID da *tag* e os atributos necessários para os eventos da implementação. O "RFID Management Console" é uma aplicação *Web* que permite controlar o funcionamento dos leitores e a gestão dos eventos. Os administradores podem modificar os parâmetros enquanto o sistema está em funcionamento [18].

- **Sybase**™ – Esta empresa apresenta a solução RFID Anywhere® e RFID Anywhere LIS® (*Location Information System*) para sistemas de localização e de rastreamento de objectos. O RFID Anywhere® é um *software* com uma infra-estrutura flexível que integra a lógica do negócio e os processos com uma variedade de recolha automática de dados e tecnologias de sensores, incluído o RFID, código de barras, dispositivos móveis, sistemas de localização e rastreamento e sensores ambientais [19]. O diagrama da solução é apresentado na Figura 2-13.

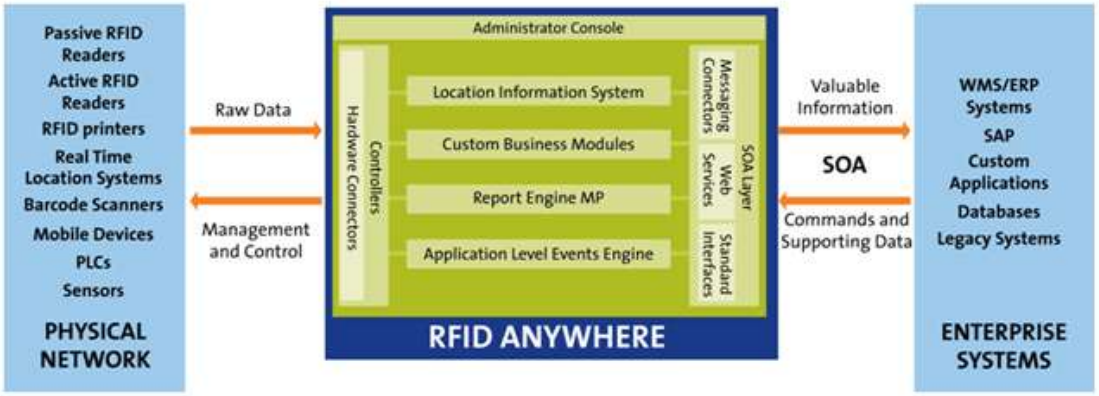



Figura 2-13 – Diagrama do RFID Anywhere® (retirado de [20]).

As principais características desta plataforma são: a arquitectura distribuída; abstracção de *hardware*; as opções de desenvolvimento serem flexíveis; a gestão de rede; o grande nível de segurança; e a facilidade de uso. Os principais elementos do RFID Anywhere são conectores de *hardware* que permitem uma abstracção das API's (*Application Programming Interface*) em relação aos elementos físicos, conectores de transporte que permitem a transferência de informação, permite que possam ser instalados e executados módulos de negócios – BM's (*Business Modules*) – criados em .NET, permite a criação de relatórios de ciclos de eventos e possibilita a geração de relatórios multi-protocolo sobre a actividade das *tags*. Um dos componentes fundamentais é o módulo de negócio. Estes possibilitam que qualquer programador crie a lógica de funcionamento da aplicação utilizando somente uma extensão do Visual Studio®. Os eventos gerados pelo sistema podem ser depois tratados utilizando as funcionalidades da arquitectura .NET. Além disso a interacção com outras aplicações e fontes de dados é facilmente alcançável – fruto dos BM's tratarem-se de uma comum aplicação .NET [6].

O RFID Anywhere LIS foi desenhado para aumentar os níveis de suporte do RFID Anywhere. Esta nova ferramenta agrega um conjunto de tecnologias para permitir a localização e seguimento de objectos. Esta agregação passa por *tags* RFID do tipo activo e passivo, RTLS, *positioning engines*, códigos de barras, sensores e outros. Antes de iniciar o rastreamento dos elementos é necessário que inicialmente seja feita a calibração das localizações e associa-las a coordenadas XY no mapa. O LIS possibilita também a visualização do movimento dos objectos através de um mapa que pode ser importado para o LIS, ou através de elementos importados de um *position engine* [6].

- **logicAlloy**™ – Esta empresa apresenta uma plataforma *middleware* ALE Server® que é uma plataforma baseada nos *standards* EPCglobal (exemplificado na Figura 2-14), desenvolvida em Java, tem uma boa performance e é de fácil utilização, facilitando assim a integração do *hardware* RFID com modelos de negócios existentes. Esta plataforma é código aberto e pode

ser utilizada gratuitamente para fins de desenvolvimento. Para fins comerciais é necessário pagar para obter licença de utilização [21]. Este *middleware* apenas possibilita a identificação e captura das *tags* e não o modo como troca informação com o modelo de negócios, visto que apenas dispõe de um servidor ALE Server® [9].

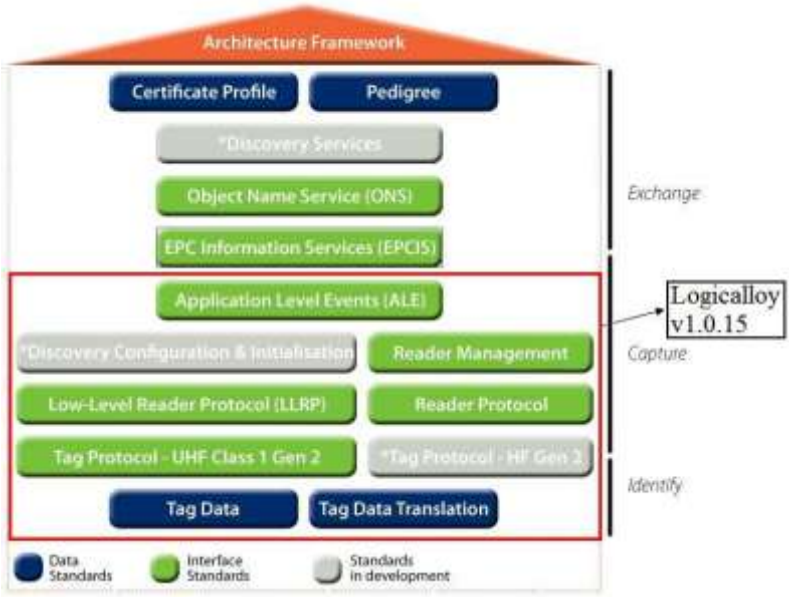



Figura 2-14 – Esquema dos EPCglobal *Standards* (retirado de [9]).

Na Figura 2-15 é ilustrado um exemplo de adição de um leitor RFID no ALE Server®, as configurações são feitas através do acesso *Web*.

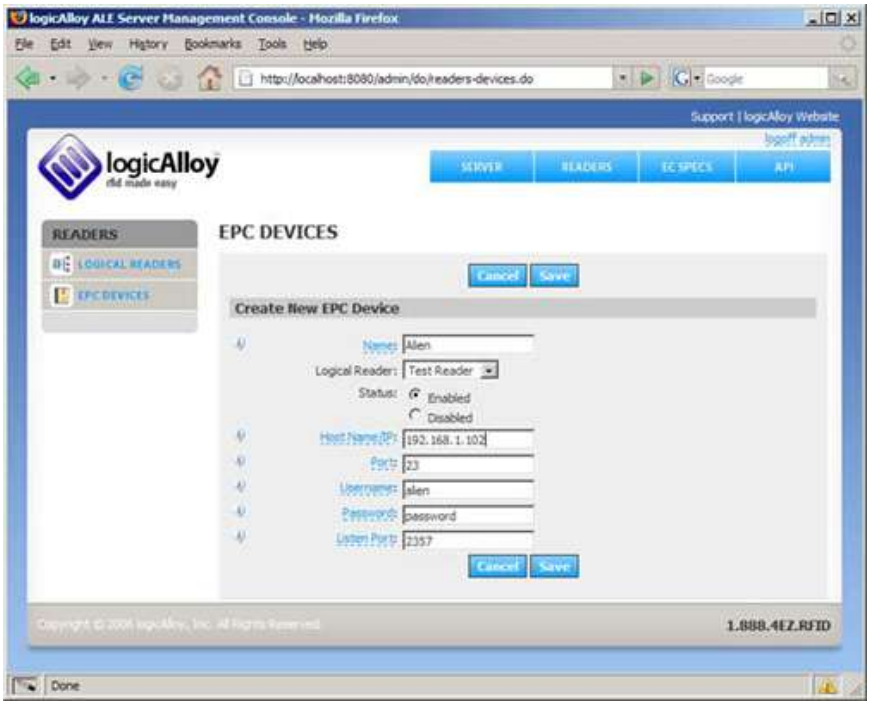



Figura 2-15 – Adicionar um leitor RFID no ALE Server® (retirado de [22]).

## 2.10. Tecnologias de integração

### 2.10.1. JDBC

O JDBC (*Java Database Connectivity*) é um *standard* industrial para conectividade com a base de dados entre a linguagem de programação Java e várias bases de dados SQL (*Structured Query Language*). A tecnologia JDBC permite utilizar a linguagem de programação Java para explorar as potencialidades do "*Write Once, Run Anywhere*" para as aplicações que precisam o acesso aos dados da empresa. Com a activação do *driver* da tecnologia JDBC, é possível interligar todos os dados incorporados mesmo em ambientes heterogéneos [23].

Através do JDBC, é fácil de enviar instruções SQL para qualquer base de dados relacional. Com o JDBC API, não é necessário escrever um programa para aceder a base de dados da Sybase™, nem outro programa para aceder a base de dados Oracle® [1], nem outro para aceder a base de dados PostgreSQL® [2], e assim por diante. Utilizando a API do JDBC é possível criar um programa capaz de enviar comandos SQL para diferentes tipos de base de dados, como exemplifica a Figura 2-16. A combinação entre o Java e o JDBC permite ao programador escrever o seu código e executá-lo em diferentes plataformas [24].

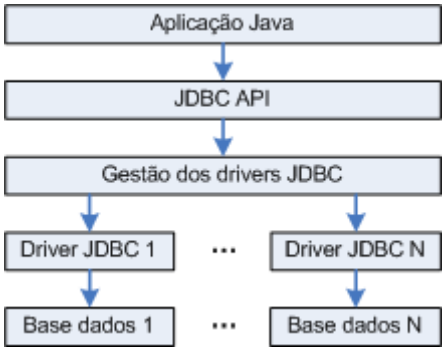



Figura 2-16 – Arquitectura do funcionamento da API do JDBC (adaptado de [24]).

### 2.10.2. JSP

A tecnologia JSP (*JavaServer Pages*) é baseada na linguagem Java que possibilita a criação de páginas *Web* dinâmicas. Esta tecnologia foi desenvolvida pela Sun Microsystems™, que permite criar aplicações do lado do servidor e aceder a base de dados, oferece uma plataforma robusta para o desenvolvimento *Web*.

[1] Sítio do Oracle: http://www.oracle.com/global/pt/index.html (Ultima vez visitado em Dezembro 2008)

[2] Sítio do Postgresql®: http://www.postgresql.org/ (Ultima vez visitado em Dezembro 2008)

Na máquina cliente reside a *interface* utilizada pelo utilizador da aplicação e a parte de acesso a dados e de mecanismo de decisão e processamento no servidor *Web*. É uma arquitectura diferenciada em dois níveis distintos que utiliza a infra-estrutura de redes existentes para facultar informações e aplicações para o utilizador. Tanto para o cliente como para o servidor *Web* a topologia da rede é irrelevante, uma vez que é tratado pelo protocolo TCP/IP (*Transmission Control Protocol* / *Internet Protocol*). Esta abordagem consiste em que cada aplicação *Web*, que corre em máquinas cliente crie uma ligação a uma aplicação servidor para executar os seus processos. Este modelo permite ao JSP aceder directamente a um recurso como uma base de dados, para servir um pedido do cliente, como exemplifica a Figura 2-17 [25].

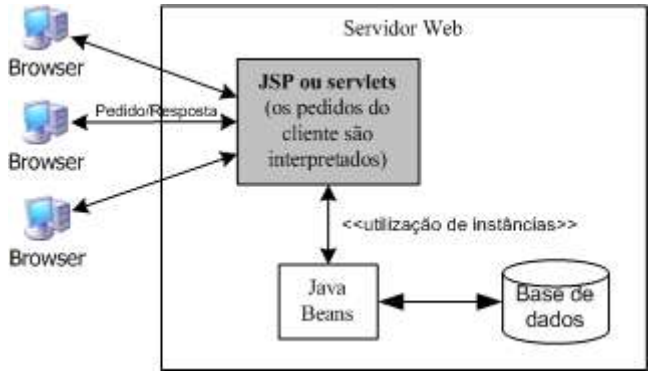



Figura 2-17 – Arquitectura global do funcionamento do JSP (adaptado de [25]).

As páginas JSP têm a capacidade de interceptar e processar um pedido de entrada, obter ou executar as informações recebidas e responder ao cliente, fornecendo uma resposta de uma forma clara e organizada colocando os dados nos *beans* (conjunto de classes e interfaces na forma de pacotes Java). Esta arquitectura baseia-se em pedidos directos à aplicação servidor embutida em código Java, que gera automaticamente uma resposta para ser inserida dentro do HTML (*HyperText Markup Language*). Todo o código Java é colocado pelo servidor dentro do HTML, o que restringe qualquer alteração a uma área limitada, reduzindo a complexidade [25].

O motor JSP analisa os ficheiros .jsp e cria um ficheiro Java *Servlet*. Este ficheiro é compilado apenas num primeiro acesso. É por este motivo que o JSP é provavelmente mais lento a primeira vez que é acedido. Em qualquer outro acesso posterior, a *servlet* especial compilada é executada e a informação é retornada mais depressa [26].

Tal como podemos ver na Figura 2-18, um pedido JSP exige os seguintes passos:

1. O utilizador da aplicação consulta um *Web site* e abre uma página JSP. O *browser* efectua o pedido pela *Internet*;
2. O pedido JSP é enviado para o servidor *Web*;

3. O servidor *Web* reconhece que o pedido vem de uma página de extensão JSP e passa-a para o motor JSP *Servlet (JSP Servlet Engine)*;
4. Se o ficheiro JSP foi chamado pela primeira vez, vai agora ser processado. Caso o acesso ao servidor *Web* não seja o primeiro, o passo seguinte é o passo nº7;
5. De seguida é gerada uma *servlet* especial com base no ficheiro JSP. Todo o HTML exigido é convertido em instruções de impressão de dados (*println statements*);
6. O código fonte da *servlet* é compilado para uma classe;
7. A *servlet* é instanciada assim que são chamados os métodos de início e de serviço;
8. O HTML da resposta da *servlet* é enviado via *Internet*;
9. O resultado HTML é mostrado no *browser* do utilizador.

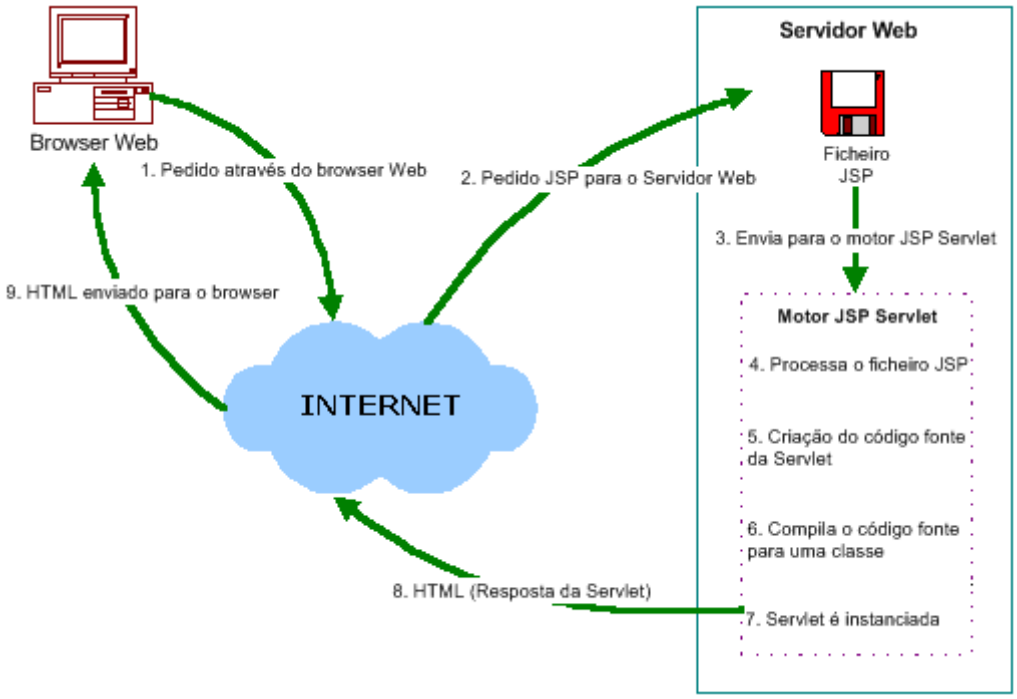



Figura 2-18 – Passos executados por um pedido de JSP (adaptado de [26]).

### 2.10.3. Reverse Ajax

O Reverse Ajax não é uma tecnologia em si, mas sim um termo que se refere à utilização de um grupo de tecnologias em conjunto, consiste na utilização da longa duração das ligações HTTP de baixa latência para permitir a comunicação entre um servidor *Web* e um *browser*, portanto é uma forma de transmitir dados da página para o servidor e um mecanismo para recolher os dados do servidor para a página, permitindo assim a página receber dados do servidor sem fazer nenhum pedido [27].

A implementação do Reverse Ajax é realizada através de uma biblioteca RPC (*Remote procedure call*) designada por DWR (*Direct Web Remoting*), que consiste em duas partes principais, um servidor com a tecnologia Java Servlet (consiste no desenvolvimento de aplicações *Web*) para processar pedidos e enviar respostas de volta para o *browser* (Figura 2-19).

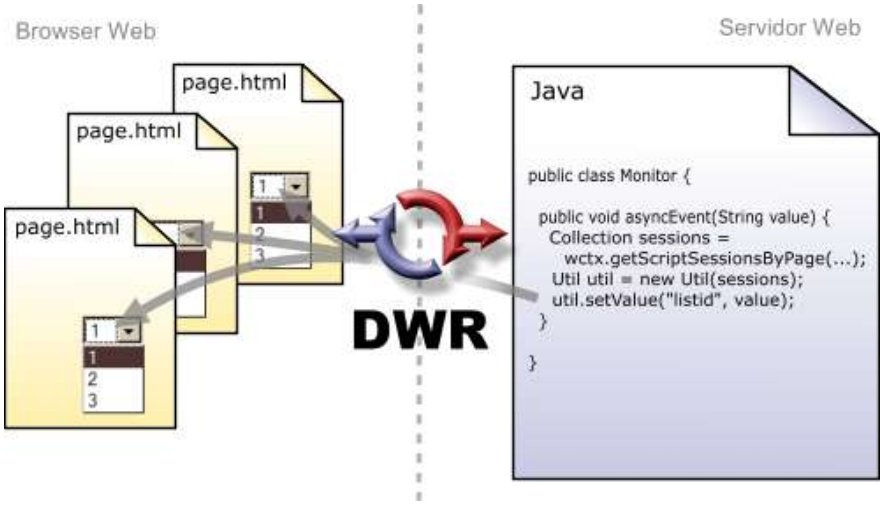



Figura 2-19 – Actualização dinâmica da página (adaptado de [28]).

O *browser* precisa de suportar JavaScript para enviar pedidos e poder actualizar dinamicamente a página (Figura 2-20). Este método de funções remotas desde do Java até o JavaScript permite a execução em páginas *Web* sem precisar de instalar extras [28].

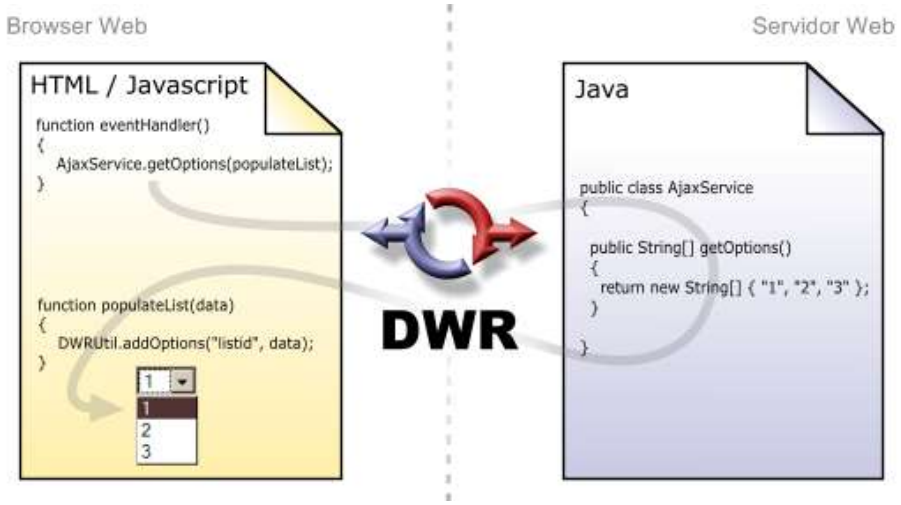



Figura 2-20 – Pedido e envio de resposta (adaptado de [28]).

## 2.11. Soluções e aplicações

O objectivo desta secção é apresentar algumas soluções e aplicações associadas aos sistemas de localização de pessoas ou objectos.

### 2.11.1. AeroScout™

A empresa AeroScout™ [1] utiliza o *standard* Wi-Fi (*Wireless Fidelity*) como base na implementação dos seus produtos e soluções. Através da integração com a norma IEEE 802.11 consegue oferecer soluções que permitem localizar e manter pessoas ou bens em múltiplos ambientes. O sistema permite as funcionalidades de localização em tempo real (RTLS) de pessoas ou bens, utilização de RFID activa de longo alcance, serviços de telemetria e controlo de acessos [6].

Os produtos que são fornecidos pela AeroScout™ englobam desde o sistema de *software* até aos elementos físicos necessários. Em termos de aplicação tem o AeroScout MobileView®, que serve de integração entre um destes elementos: o AeroScout Engine, Cisco Location Appliance ou outras fontes. Para *middleware* existe o AeroScout Engine que trata a informação proveniente dos pontos de acesso Wi-Fi. Este motor permite múltiplas formas de localização com apenas uma infra-estrutura: localização em tempo real, detecção de presença e telemetria. Permite também um funcionamento em diferentes tipos de ambientes, como ambientes interiores e exteriores. A localização feita por esta aplicação é conseguida recorrendo-se ao uso de algoritmos TDOA (*Time Difference of Arrival*) e RSSI (*Received Signal Strength Indicator)* [6]. O diagrama da arquitectura da aplicação AeroScout MobileView® encontra-se na Figura 2-21.

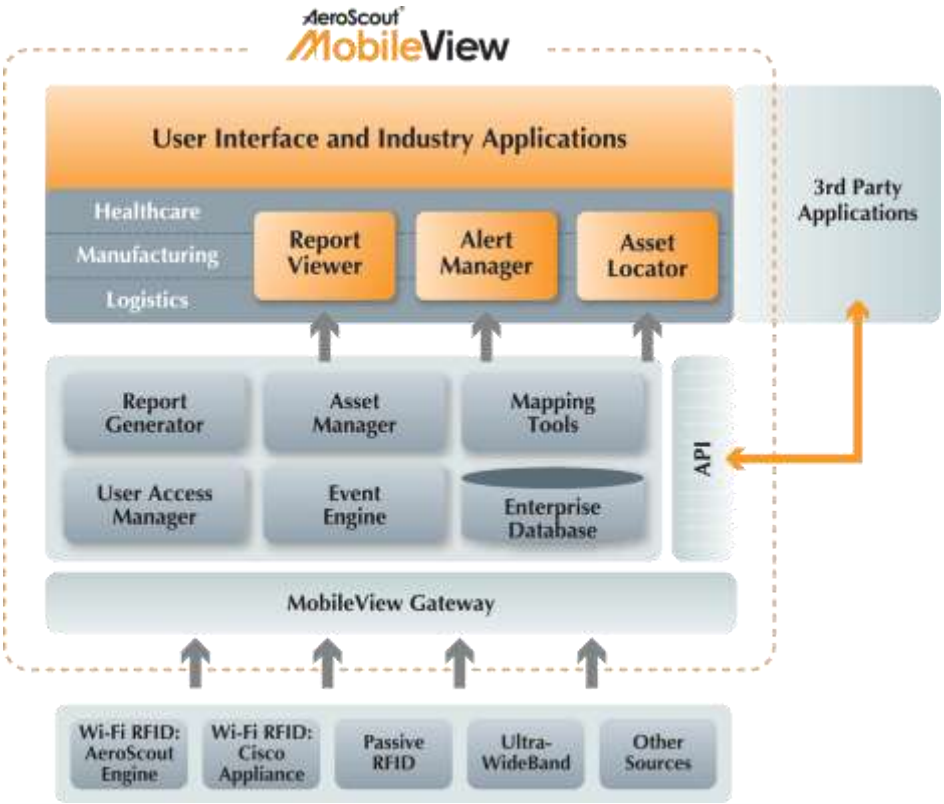



Figura 2-21 – Arquitectura da aplicação AeroScout MobileView® (retirado de [29]).

[1] Sítio da AeroScout™: http://www.aeroscout.com/ (Ultima vez visitado em Dezembro 2008)

Através da Figura 2-21 verifica-se que a infra-estrutura é compatível com os diversos dispositivos. Permitindo assim à aplicação uma independência de um tipo de específico de tecnologia e dispositivo [30].

### 2.11.2. Ekahau RTLS®

A Solução Ekahau RTLS® pertence a empresa Ekahau™ [1], baseada em localização de pessoas e bens em tempo real, utilizando uma infra-estrutura Wi-Fi existente. O mecanismo de localização da Ekahau™ permite as *tags* funcionarem como se fosse um equipamento Wi-Fi *standard*, e portanto torna possíveis comunicações únicas de duas vias, entre as *tags* e as aplicações, entregando dados no formato visual e alarmes audíveis e mensagens de texto [31]. A solução consiste nos seguintes componentes integrados [32]:

- **Ekahau Tracker** – aplicação comercial para localização em tempo real, visualização de alarmes e análise das localizações das pessoas e bens;
- **Ekahau Finder** – consiste num serviço que implementa as funcionalidades do motor de localização, criação dos alarmes, gestão dos equipamentos e serviços para funcionamento correcto do sistema e armazenamento dos *logs* durante a fase de desenvolvimento;
- **802.11 Wi-Fi *access points*** – Serviço que permite a comunicação entre as *tags* e o Ekahau Engine.
- ***Tags*** – RFID activo, devolve a força do sinal entre os pontos de acesso e as *tags* através de uma rede 802.11 para o Ekahau Engine em tempo real;
- **Ekahau Client** – Consiste num agente de *software* que permite localizar/activar equipamentos sem fios. O funcionamento é similar as *tags* mas funciona em segundo plano nos PDA's (*Personal Digital Assistant*), *Tablet* PC's (*Personal Computer*) ou portáteis.

A visão geral da solução Ekahau RTLS® é ilustrada na Figura 2-22.

[1] Sítio da Ekahau™: http://www.ekahau.com (Ultima vez visitado em Dezembro 2008)

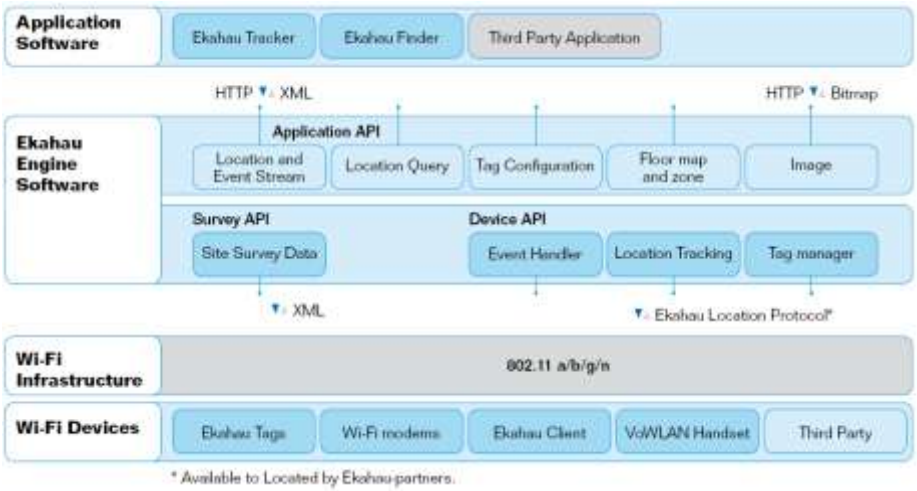



Figura 2-22 – Arquitectura da solução Ekahau RTLS® (retirado de [32]).

### 2.11.3. HealthTrax®

HealthTrax® é uma solução completa da empresa InfoLogix™ [1] para a localização de objectos e pessoas num hospital, inclui *tags* RFID e *software* empresarial, que permite utilizar com todos *standards* de redes sem fios. Como o sistema é baseado pelo Wi-Fi, utiliza a infra-estrutura da rede sem fios do próprio edifício. Um exemplo da aplicação encontra-se na Figura 2-23.


Figura 2-23 – Exemplo da aplicação HealthTrax® (retirado de [33]).

[1] Sítio da InfoLogix™: http://www.infologixsys.com/ (Ultima vez visitado em Dezembro 2008)

### 2.11.4. MRID

A junção entre a QuadTech International™ [1] e a Mtech™ [2] levou ao aparecimento de uma nova tecnologia de RFID, o MRID (*Multiple Range Identification*). Esta tecnologia permite fazer o rastreio de bens, animais, materiais ou de qualquer género, através dum sistema de códigos em multifrequência. Um sistema MRID é bastante similar aos sistemas RFID já atrás referidos substituindo, porém, as *tags* e os leitores RFID, por *tags* e leitores MRID. Isto no entanto não traz qualquer alteração à arquitectura conceptual dos sistemas, que por si consiste numa enorme vantagem. A diferença entre as *tags* consiste que a *tag* MRID transmite múltiplos códigos que são univocamente associados a uma dada unidade. As diferenças entre os leitores MRID e RFID colocam-se apenas na possibilidade de interpretar esses códigos de modo a poder calcular a distância da *tag* ao leitor [6].

O funcionamento desta tecnologia consiste nos múltiplos ID's (*Identification*) de uma única *tag*, emitidos a diferentes níveis de potência. Cada um desses ID's contém um código associado que representa uma certa distância, por exemplo a *tag* de ID 100, possui 3 códigos: 100-010; 100-005; 100-001 (sendo que os dígitos a seguir ao hífen representam uma distância em metros, por exemplo), isto quererá dizer que a *tag* de ID 100 irá emitir 3 sinais de diferentes potências, em que cada sinal leva o código a si associado. Imaginando que o leitor capta apenas o sinal mais forte (associado ao código 100-010), então isso quererá dizer que a *tag* está num raio de 10 a 5 metros do leitor, pois não captou o sinal representativo dos 5 metros [5]. Um outro exemplo pode ser visualizado na Figura 2-24, onde ilustra 6 códigos: 101-100; 101-050; 101-025; 101-010; 101-005; 101-001, onde o código 101-100 corresponde uma distância maior e o código 101-001 indica uma distância menor.

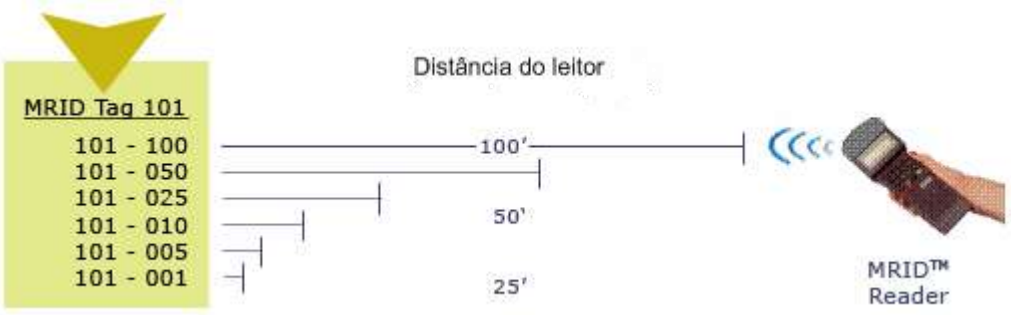



Figura 2-24 – Exemplo ilustrativo do funcionamento do sistema MRID (adaptado de [6]).

Para determinar a distância da *tag*, o leitor calcula qual é o sinal mais forte recebido da *tag*, através de um circuito de discriminação. Isto é especialmente eficaz num leitor manual, onde este pode ser apontado em várias direcções para que o circuito consiga calcular mais eficazmente as potências,

[1] Sítio da QuadTech International™: *http://www.quadtechint.com/* (Ultima vez visitado em Dezembro 2008)

[2] Sítio da Mtech™: http://www.m-techindia.com/ (Ultima vez visitado em Dezembro 2008)

obtendo consequentemente valores de posição mais concisos. Esta tecnologia faz com que estes sistemas se tornem preciosos em situações de emergência ou para organizações com elevado número de bens a serem rastreados e protegidos [6].

### 2.11.5. WiseTrack™

A empresa WiseTrack™ [1] utilizando RFID activo consegue localizar objectos em movimento ou parados. As principais funcionalidades desta solução são a localização em tempo real de uma *tag* em qualquer sítio dentro do edifício, visibilidade dos recursos, controlo automático do inventário, controlo de entrada e saída sem a interacção das pessoas e alarmes de segurança. Na Figura 2-25 é exemplificado a localização do servidor Wisetrack e das antenas (junto as portas das salas e do corredor).


Figura 2-25 – Localização das antenas e do servidor (retirado de [34]).

Um exemplo de controlo de acessos é apresentado na Figura 2-26, onde a *tag* é detectada pela antena e o acesso a sala é activado.

Figura 2-26 – Detecção de uma *tag* (retirado de [34]).

[1] Sítio da WiseTrack™: http://www.wisetrack.com/ (Ultima vez visitado em Dezembro 2008)

Quando uma determinada *tag* é detectada num zona de acesso não autorizado, é criado um alarme como exemplifica na Figura 2-27.


Figura 2-27 – Detecção de um alarme (retirado de [34]).

### 2.11.6. Comparação entre as soluções

Após a descrição de algumas soluções e aplicações relacionadas com sistema de localização de pessoas ou objectos foi realizado uma comparação entre as funcionalidades (Tabela 2-6) de cada sistema e os equipamentos utilizados (Tabela 2-7).

Tabela 2-6 – Comparação entre as soluções apresentadas.

| Solução | Localização tempo real de pessoas | Localização tempo real de bens | Controlo de acesso | Controlo de inventário | Troca de informação detalhada |
|---|---|---|---|---|---|
| AeroScout™ | ✓ | ✓ | ✓ | | |
| Ekahau RTLS® | ✓ | ✓ | ✓ | ✓ | ✓ |
| HealthTrax® | ✓ | ✓ | ✓ | ✓ | ✓ |
| MRID | ✓ | ✓ | | | |
| WiseTrack™ | ✓ | ✓ | ✓ | ✓ | |

Através da Tabela 2-6 verifica-se que existe no mercado soluções para as funcionalidades seleccionadas, existe ainda soluções que conseguem abranger todas as funcionalidades.

Tabela 2-7 – Comparação entre os equipamentos das soluções.

| Solução | *Tags* Activas | *Tags* passivas | Antenas | Leitores | Sensores incorporados | Tecnologia Wi-Fi |
|---|---|---|---|---|---|---|
| AeroScout™ | ✓ | ✓ | | | | ✓ |
| Ekahau RTLS® | ✓ | ✓ | | | | ✓ |
| HealthTrax® | ✓ | ✓ | | | | ✓ |
| MRID | ✓ | ✓ | ✓ | | | |
| WiseTrack™ | ✓ | | ✓ | ✓ | | |

Através da Tabela 2-7 verifica-se que todos os equipamentos seleccionados não são abrangidos, mas verificamos a interligação entre a tecnologia RFID com Wi-Fi em algumas soluções.

## 2.12. Conclusões

O RFID é um mecanismo que possibilita imensas oportunidades. Por ser uma tecnologia bastante apreciada na área das telecomunicações actuais, o mercado disponibiliza sempre novos desafios de implementação.

Como foi possível analisar ao longo deste capítulo, a tecnologia RFID baseia-se em vários princípios de funcionamento, com soluções robustas de extrema utilidade e funcionalidade. Os mais variados locais públicos ou privados são hoje cobertos por redes de pequenas *tags*, que recolhem e processam informação detalhada.

A tecnologia RFID permite desempenhar várias funções diferentes, o funcionamento das *tags* pretendido na implementação, indica o tipo de *tags* e de leitores que são necessários para a sua utilização. Existe 3 tipos diferentes de *tags* em circulação no mercado, activo, passivo e semi-passivo. As *tags* activas têm mais capacidades, permitindo maior alcance do que as *tags* passivos, no entanto estas não precisam de bateria, sendo mais baratas do as activas. As *tags* RFID comunicam com um leitor, que irá transmitir as informações recebidas para um servidor.

Apesar da aparente difusão da tecnologia, as normas de regulação tardaram a surgir e só na década passada começaram a surgir as primeiras normas.

Uma análise de algumas soluções e aplicações comerciais associadas a sistemas de localização de pessoas ou objectos em ambientes interiores foi realizada, em que se verifica soluções interagir com outras tecnologias, mais concretamente o Wi-Fi, e que existe no mercado soluções para as funcionalidades seleccionadas.

Após o estudo das tecnologias a ser utilizadas no sistema são necessários especificar as arquitecturas físicas e lógicas do sistema proposto, as características do *software* e do *hardware*, o *middleware* proposto e os sistemas de integração com a solução. Portanto o próximo capítulo é detalhado a especificação do sistema.

# 3. Especificação do Sistema

O sistema proposto é descrito em detalhe, sendo apresentado a arquitectura física e lógica do sistema, os requisitos do *software* e *hardware*, o mecanismo que interliga os equipamentos e as aplicações, os algoritmos de localização propostos, o diagrama da base dados, o servidor *Web* e a interligação entre camadas.

## 3.1. Descrição do Sistema

O sistema, designado TraceMe, deverá controlar de forma automática e não obstrutiva acessos a diferentes sectores de um edifício, tendo aplicações em hospitais, edifícios públicos, sedes de grandes empresas ou instalações industriais. O controlo automático da detecção do acesso deverá ser efectuado remotamente por comunicação de radiofrequência entre uma *tag* de acesso na posse do utilizador ou de um objecto e um conjunto de antenas localizadas em pontos estratégicos do edifício. Além de poder detectar o acesso a zonas restritas sem necessidade de contacto com fechaduras, teclados ou sistemas biométricos, o sistema deverá ainda localizar a posição de um utilizador específico para um contacto mais expedito.

Deve ainda suportar a localização de equipamentos móveis sensíveis e accionar alarmes de detecção de ausência no edifício e falha de comunicação com os equipamentos do sistema. Na Figura 3-1 é apresentado o logótipo do TraceMe.


Figura 3-1 – Logótipo do TraceMe.

Na Figura 3-2 é apresentado um exemplo de aplicação do TraceMe, onde existem leitores distribuídos por cada divisão e que permitem saber se um dado objecto ou pessoa passou nesse ponto. Tipicamente estas são instaladas em pontos de passagem (detecção na frequência F1, ilustrado na Figura 3-2) e possuem a capacidade de identificar o sentido da passagem da *tag*. Estes leitores comunicam

directamente com um servidor (onde estará toda a inteligência do sistema) através da *Ethernet* [1]. As *tags* devem utilizar o conceito de funcionamento em multi-frequência (secção 2.3.1), a frequência F1 para controlo de acesso e detecção e a frequência F2 para transmitir a identificação.

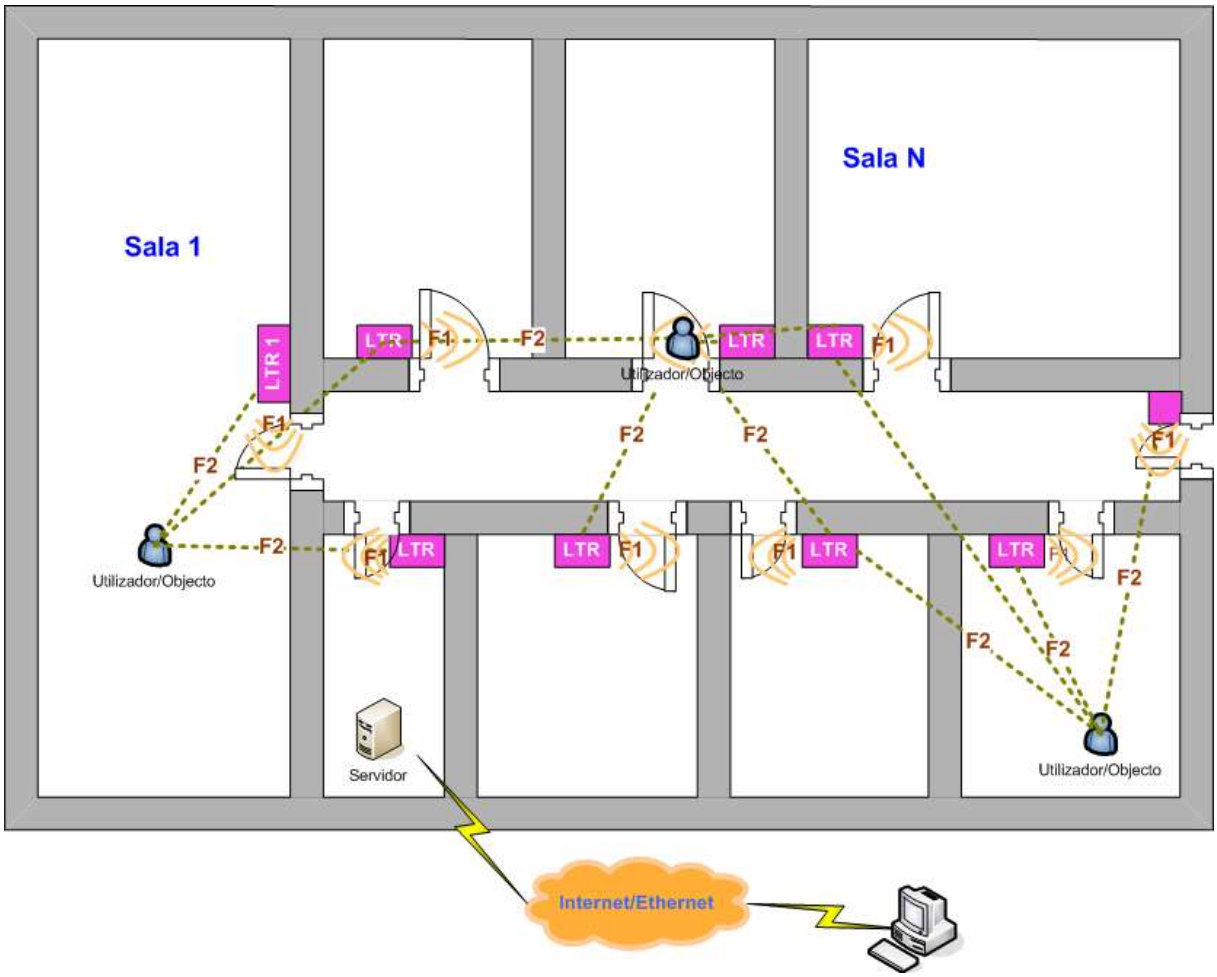



Figura 3-2 - Exemplo de configuração para um edifício de escritórios.

## 3.2. Arquitectura Física do Sistema

A arquitectura física proposta está representada na Figura 3-3 que identifica a utilização do conceito RFID activo. Para determinar a localização de uma determinada *tag* será utilizada uma rede de leitores que permite a cobertura do edifício e estará interligada com um servidor, designado por Servidor TraceMe, o que possibilita ajustar a distribuição dos equipamentos em função do grau de precisão pretendida.

[1] Utiliza como protocolo de rede o IP e como protocolo de acesso à rede *Ethernet* (IEEE802.3)

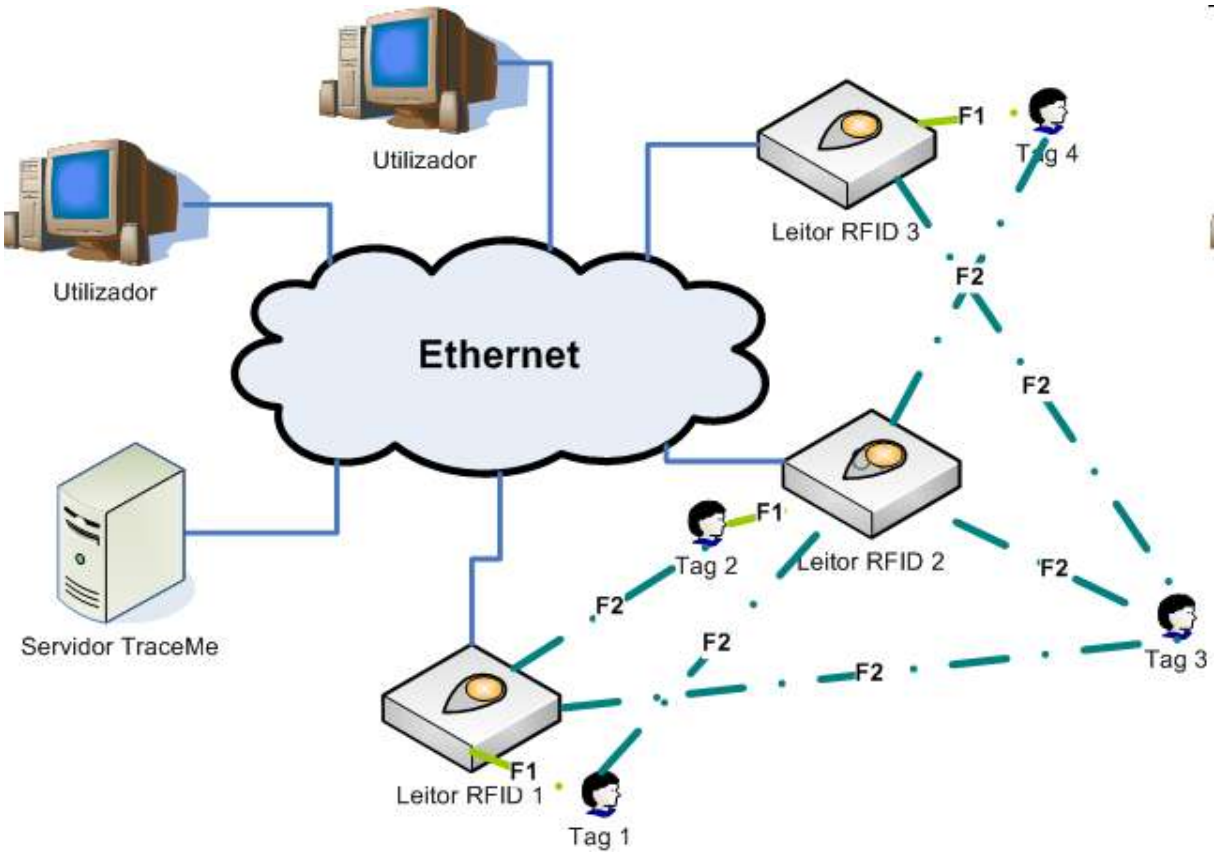



Figura 3-3 – Diagrama da arquitectura física.

O sistema proposto utiliza uma *tag* activa associada a uma pessoa ou a um objecto que transmite um sinal periódico na frequência F2 para os leitores que estiverem ao seu alcance. A *tag* também pode ser detectada por um determinado leitor numa frequência específica (F1) mas com um alcance inferior, mas responde com um sinal transmitido na frequência F2 a indicar o leitor que o detectou (através desta detecção é baseada o mecanismo de localização, detalhado na secção 3.6). O leitor ao receber informação de uma *tag* retransmite a mensagem para o Servidor TraceMe (em detalhe na Figura 3-4) através da *Ethernet*.

Com esta informação o servidor actualiza uma base de dados de referenciação onde é cruzada a informação referente á sala, antena e *tag*. Portanto a detecção da F1 indica-nos o local de passagem e a detecção da F2 indica-nos que a *tag* encontra-se no edifício. A frequência F1 deverá ter um alcance curto para cobrir pontos de passagem, a frequência F2 deverá ter um alcance superior que permite fazer a cobertura total do edifício. Para aceder as funcionalidades do sistema, os utilizadores acedem ao servidor *Web* através de um *browser* num computador ligado a *Ethernet*.

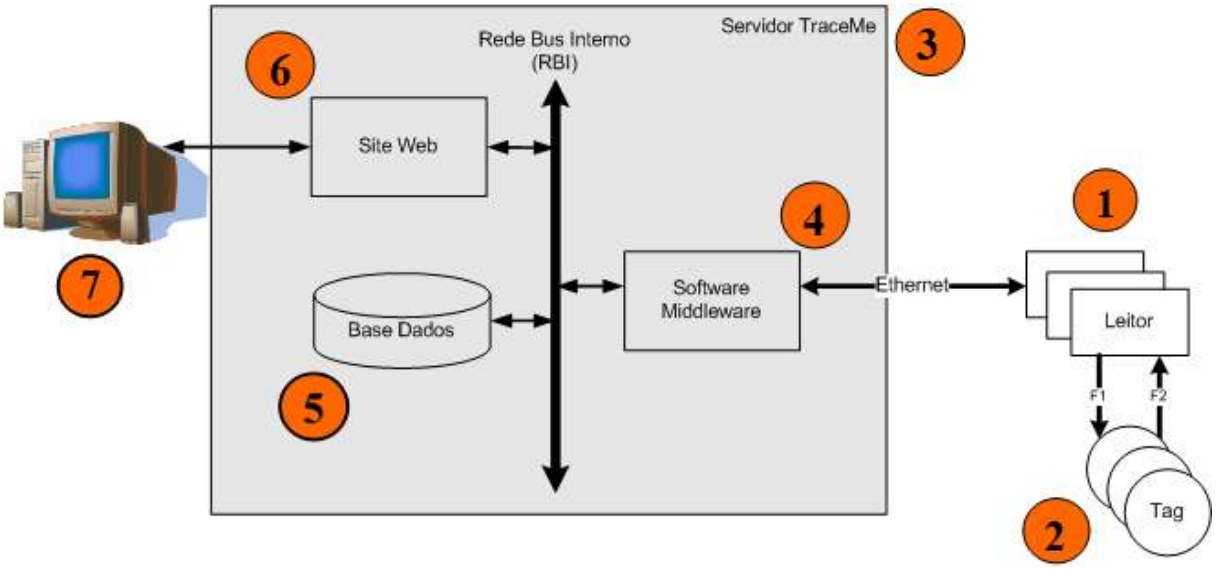



Figura 3-4 – Arquitectura do Sistema.

Através da Figura 3-4 pode verificar-se que aplicação é dividida por sete módulos:

- Modulo 1 – Equipamento leitor que pode transmitir sinais na frequência F1 e recebe sinais na frequência F2 (*hardware*);
- Modulo 2 – Equipamento *tag* que pode receber os sinais na frequência F1 e enviar sinais na frequência F2 (*hardware*);
- Modulo 3 – Equipamento Servidor TraceMe que recebe e envia os dados enviados pelos leitores e pelos *browsers* (*hardware*);
- Modulo 4 – *Software Middleware* que recebe os dados do leitor, insere e consulta a base de dados e comunica com o *Site Web* (*software*);
- Modulo 5 – Base de dados onde são armazenados todas as detecções registados pelo leitor, as informações e configurações dos equipamentos, configurações dos serviços, as informações dos utilizadores do sistema (*software*);
- Modulo 6 – *Site Web* com acesso restrito onde se podem visualizar todas as funcionalidades do sistema, como por exemplo a localização de uma determinada *tag* e configurar os equipamentos do sistema (*software*);
- Modulo 7 – Computador com acesso ao *site Web* através de um *browser* (Internet Explorer® e Firefox®) (*hardware*);

## 3.3. Arquitectura Lógica do Sistema

A arquitectura lógica do sistema encontra-se na Figura 3-5, esta arquitectura é baseada através [6] e [10], onde foi adaptado para suportar as funcionalidades pretendidas. A arquitectura proposta está dividida em quatro módulos, os "Equipamentos RFID", o "Software Middleware", a "Base Dados" e "Serviço de dados". O módulo "Equipamentos RFID" representa os leitores e as *tags* de RFID, o módulo "Software Middleware" é responsável pela interligação entre os módulos e pelos mecanismo das funcionalidades do sistema, o módulo "Base Dados" guarda todos os dados e as informações relevantes para o sistema funcionar correctamente e o módulo "Serviços de dados" é a aplicação que permite aceder e configurar as funcionalidades do TraceMe.

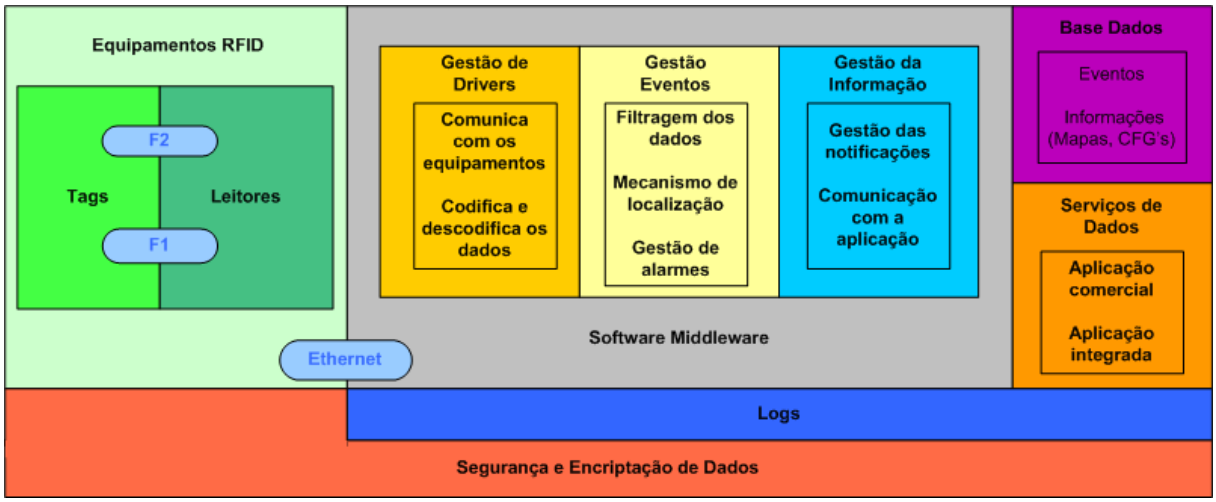



Figura 3-5 – Diagrama da arquitectura lógica.

- **Equipamentos RFID**:
  - ***Tags*** – Equipamento *tag* activa que transmita um sinal na frequência F2 para os leitores, podem ser detectados na frequência F1 (*Hardware*);
  - **Leitores** – Equipamento leitor que recebe os sinais das *tags* e transmite para o Servidor TraceMe, através da *Ethernet*, a indicação do sentido, a identificação da *tag* e a identificação do leitor. O leitor transmite um sinal na frequência F1 para excitar as *tags* (*Hardware*);
- **Software Middleware**:
  - **Gestão de Drivers**:
    - **Comunica com os equipamentos** – Identifica na comunicação qual é o leitor que está a transmitir e tem a noção da infra-estrutura da rede;
    - **Codifica e descodifica os dados** – Identifica o *driver* específico para a codificação e a descodificação da mensagem de um determinado leitor.

  - **Gestão Eventos**:
    - **Filtragem dos dados** – Permite filtrar os dados relevantes para não existir dados redundantes na base de dados;
    - **Mecanismo de localização** – Algoritmo que indica a localização do novo evento recebido;
    - **Gestão de alarmes** – Detecção de todos os alarmes configurados pelo sistema;
  - **Gestão da Informação**:
    - **Gestão das notificações** – Permite tratar das notificações recebidas;
    - **Comunicação com a aplicação** – Permite comunicar com aplicação para receber ou enviar notificações;
- **Base de Dados**:
  - **Eventos** – Guarda os eventos registados pelos leitores;
  - **Informações** – Guarda informações dos utilizadores, dos equipamentos, mapas e configurações;
- **Serviços de Dados**:
  - **Aplicação comercial** – *Site Web* para aceder as funcionalidades do sistema para utilizador comum;
  - **Aplicação integrada** – Integração da aplicação com os servidores base de dados e *Web*;
- **Logs** – Regista todos os passos importantes do sistema;
- **Segurança e Encriptação de Dados** – Permite segurança em todas as comunicações no sistema.

### *3.3.1.* Arquitectura do *Software*

Os módulos de *software* apresentados na secção anterior (secção 3.3) são os módulos "Software Middleware", que é divido pelo "Gestão de Drivers", "Gestão Eventos" e "Gestão da Informação", os restantes são "Base Dados" e "Serviços de Dados".

Através da Figura 3-6 pode analisar-se o módulo "Gestão de Drivers" consiste num só serviço divido por três camadas, o "Conector", a "Codificação e descodificação dos Dados" e o "Adaptador". A camada inferior é o "Adaptador" que permite a comunicação com os equipamentos, portanto estabelece ou recebe as ligações dos equipamentos, e permite ainda a encriptação dos dados. A camada a seguir é a "Codificação e descodificação dos dados", como o nome indica codifica ou descodifica os dados a enviar ou receber da camada "Adaptador". A camada superior é o "Conector"

que consiste na interligação com os outros serviços do *middleware*, permitindo receber ou enviar dados doutros serviços para os equipamentos.


Figura 3-6 – Diagrama da Gestão de Drivers.

O módulo "Gestão Eventos", ilustrado na Figura 3-7, está dividido por várias fases de execução, começando pelo "Receptor de dados" e "Filtragem dos dados". As fases seguintes são "Detecção de *tags*", "Mecanismo de localização", "Detecção de violação" e "Detecção de bateria". Estas fases acedem a base de dados através da "Integração com a Base Dados", e por fim encontra-se a fase "Conector".

A primeira fase "Receptor de Dados" corresponde a recepção de informação proveniente do "Gestão de Drivers". A fase "Filtragem dos dados" corresponde a filtragem da informação recebida, permitindo assim não haver dados e registos redundantes. A "Detecção de *tags*" verifica se existe alguma *tag* que não comunica durante um tempo pré-definido com o sistema (as *tags* não se encontram no alcance dos leitores, por exemplo se uma *tag* sair do edifício é detectada a sua ausência quando o tempo configurado for alcançado), caso detecta gera um alarme de detecção de ausência.

O "Mecanismo de localização" determina a localização de cada *tag* em tempo real, este mecanismo está descrito em pormenor na secção 3.6. A "Detecção de leitores" analisa se existe algum alarme de falta de comunicação com os leitores, caso existe é criado um alarme de falta de comunicação. A "Detecção de violação" verifica se as *tags* foram violadas, ou seja, se tentaram adulterar a *tag*, caso seja detectado é gerado um alarme de violação da *tag* (a *tag* deverá ter mecanismo que permite detectar a violação e informar o sistema). A "Detecção de bateria" verifica se as *tags* estão com bateria fraca, se for detectado alguma *tag* é criado um alarme de detecção de bateria.

A "Integração com a Base de Dados" como o nome indica é a interligação com a base de dados, permitindo assim uma independência entre os serviços e a base de dados. O "Conector" transmite toda a informação relevante para o servidor *Web*, esta informação pode ser dados sobre alarmes e eventos

correspondente a localização das *tags*. Neste módulo é necessário garantir segurança nos acessos a Base de Dados.


Figura 3-7 – Diagrama da Gestão Eventos.

A Figura 3-8 corresponde a "Gestão de Informação" que está dividida em três serviços, o "Teste de cobertura", o "Gestão de notificações" e o "ReaderWatcher". Os serviços são executados paralelamente mas pode haver comunicação entre os serviços.

O serviço "Teste de cobertura" corresponde à análise dos eventos recebidos pela camada inferior ("Gestão de Drivers") e indicar para uma determinada *tag* quais são os leitores que conseguem receber o sinal o seu valor RSSI e a percentagem de eventos recebidos de cada leitor. Esta informação é enviada pelo "Conector" para o servidor *Web*.

No serviço "Gestão de notificações" é iniciado pelo "Receptor de notificações" que recebe as notificações de vários serviços, de seguida consiste a própria "Gestão de notificações" que determinada qual é a notificação a efectuar e o seu mecanismo. Após ter efectuado o mecanismo utiliza o "Envia o resultado" para o serviço que enviou inicialmente a notificação.

Com o serviço "ReaderWatcher" verifica-se o funcionamento dos leitores, e utiliza as notificações para comunicar com os leitores, corresponde ao "Envia notificação". Neste módulo é necessário garantir segurança nos acessos a Base de Dados e ao Servidor Web.

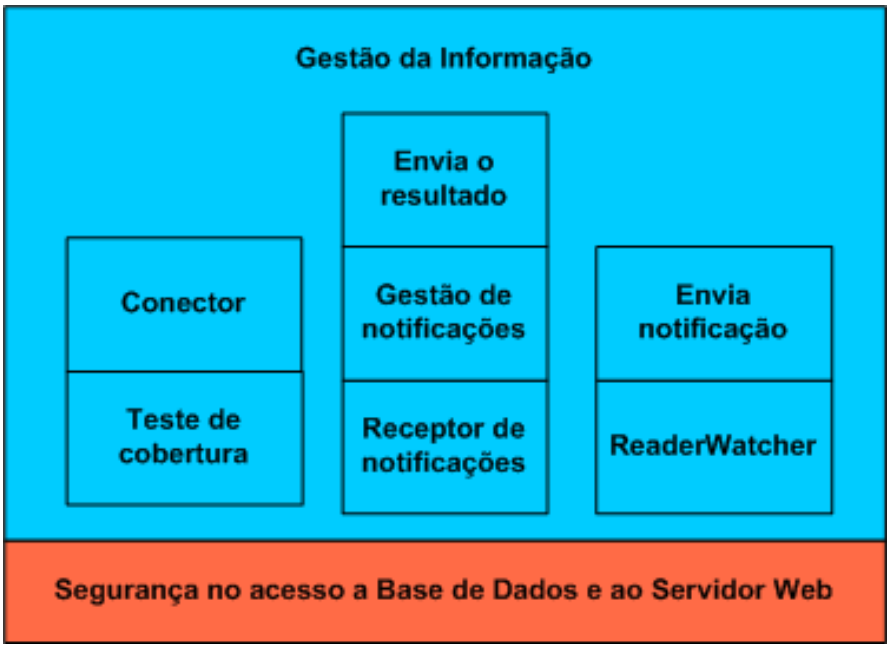



Figura 3-8 – Diagrama da Gestão de Informação.

O módulo "Base Dados" está dividido em quatro tipos de armazenamento, as "Informações", os "Eventos", "Características do sistema" e "Alarmes". O primeiro tipo divide-se em informações sobre os utilizadores e os equipamentos. O segundo guarda os eventos registados pelos leitores contendo a informação associada a esse evento, como a zona, a unidade e a *tag*. O terceiro guarda as características do sistema, que podem ser configurações, parâmetros e mapas predefinidos do edifício. O quarto guarda os alarmes gerados pelo sistema, como detecção de comunicação com os leitores, detecção de ausência das *tags*, detecção de bateria e detecção de violação das *tags*. Neste módulo é necessário garantir a protecção de dados na base de dados, o diagrama está representado na Figura 3-9.

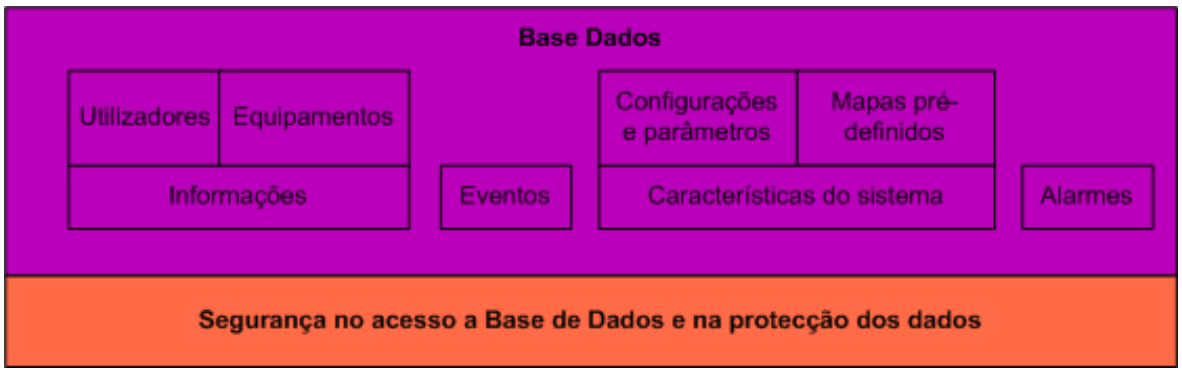



Figura 3-9 – Diagrama da Base de Dados.

O “Serviços de dados” é constituído pela “Aplicação comercial”, que se divide em duas partes, a “Aplicação integrada” e a “Aplicação Back end”. A primeira parte é o *interface* com as funcionalidades para o utilizador comum, a segunda parte é o *interface* para o utilizador Administrador com acesso as configurações de todo o sistema. Nas duas partes é possível criar notificações através da “Criação de notificações”. Neste módulo é necessário ter um nível de autenticação para assegurar o acesso restrito à aplicação. Este módulo está representado na Figura 3-10.

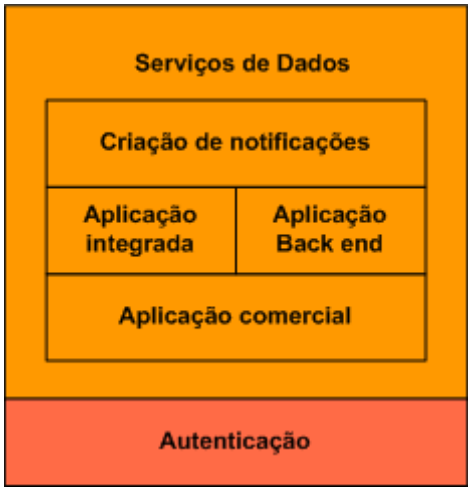



Figura 3-10 – Diagrama dos Serviços de Dados.

## 3.4. Características do *Software* e *Hardware*

As características do *software* e do *hardware* correspondem aos módulos apresentados na Figura 3-4. Os módulos “Software Middleware” (Módulo 4), “Base de dados” (Módulo 5) e “Site Web” (Módulo 6) correspondem as características do *software*. As características do *hardware* são os equipamentos *tag* (Módulo 2), leitor (Módulo 1), Servidor TraceMe (Módulo 3) e computador (Módulo 7).

De seguida, são apresentadas as características funcionais que cada módulo deve implementar para o *software* e *hardware*. Para facilitar a identificação das características estes serão designados por “Sx” para o *software* e “Hx” para o *hardware*, em que x começará em 1 e vai até ao número limite de características de cada. Algumas características podem corresponder a vários módulos mas só será inserido num módulo mais apropriado.

### 3.4.1. Características do *Software*

O *software* está dividido por três módulos, que são os seguintes:

- **Software Middleware** – permitirá a comunicação com os equipamentos do sistema RFID com as plataformas de aplicação e controla o fluxo de informação. As características deste módulo serão:

  **S1. Receber dados dos leitores** – permitirá a existência de conectividade entre os equipamentos e o *middleware*, quando um leitor detecta uma *tag* envia essa informação para os serviços, que incluí a direcção (o leitor deverá suportar a detecção da direcção), o ID da *tag*, ID do leitor, estado da bateria, estado da *tag* e o RSSI. Esta característica permitirá a ligação a vários equipamentos, criando um *driver* genérico para conseguir comunicar;

  **S2. Comunicar com a base de dados** – pretende a ligação entre os serviços e a base de dados para guardar as informações recebidas do leitor (S1) e a verificação das permissões de acesso;

  **S3. Comunicação com o servidor *Web*** – permitirá a existência de conectividade entre os equipamentos e aplicação *Web*, sem necessitar de consultar a base de dados, ou seja, a aplicação *Web* pode pedir pedidos directos aos equipamentos;

  **S4. Capacidade de verificar configurações** – pretende a conectividade com a base de dados para conseguir verificaras configurações efectuadas;

  **S5. Receber notificações** – permitirá notificar ao serviço, através do servidor *Web*, configurações a efectuar;

  **S6. Filtragem de dados –** pretende filtrar os dados relevantes para não existir dados redundantes na base de dados, como por exemplo, quando não existe alteração no posicionamento da *tag* não deverá estar sempre a guardar a mesma informação dessa *tag*;

  **S7. Transmitir dados para os leitores** – permitirá a existência de conectividade entre os equipamentos e o *middleware*, quando o *middleware* é notificado pelo Serviço *Web* de alguma alteração nas configurações dos equipamentos deverá enviar essas configurações aos respectivos leitores. Como por exemplo alterar o endereço IP (*Internet Protocol*) do leitor;

  **S8. Camada genérica para comunicar com os leitores** – permitirá distinguir os dados provenientes dos leitores e utilizar o *driver* correspondente para a codificação/descodificação da informação. Permitindo assim utilizar vários tipos de equipamentos;

**S9. Mecanismo de localização** – permitirá verificar os eventos gerados pelos leitores, utilizando um mecanismo para determinar a posição correcta da *tag* em tempo real;

**S10. Permissão de acesso** – pretende verificar se um determinado evento que está associado a uma pessoa ou a um objecto tem permissão de acesso na zona onde foi detectado;

**S11. Detecção de alarmes** – permitirá detectar e criar alarmes de falta de bateria de uma *tag*, falta de comunicação com as *tags* e leitores e violação da *tag*;

- **Base de Dados -** Este módulo corresponderá para o armazenamento da informação do sistema, as suas características deverão ser as seguintes:

  **S12. Perfis de utilizadores** – permitirá guardar perfis específicos para cada utilizador, permitindo controlar os acessos as funcionalidades do sistema;

  **S13. Utilizadores do sistema** – permitirá guardar e verificar os dados e as configurações dos utilizadores que têm acesso ao sistema;

  **S14. Configurações do sistema** – deverão armazenar as configurações dos equipamentos instalados no sistema;

  **S15. Localização dos equipamentos** – deverá guardar a localização dos equipamentos instalados no sistema;

  **S16. Informações do edifício** – deverão guardar as informações e características do edifício, e ainda mapas predefinidos indicando as coordenadas de cada divisão de uma determinada planta;

  **S17. Eventos** – permitirão armazenar os eventos detectados pelos leitores;

  **S18. Perfis de controlo de acesso** – permitirá guardarem perfis de controlo de acesso em relação ao utilizador do sistema (o portador da *tag*) com as zonas de acesso autorizado;

  **S19. Alarmes** – deverá guardar todos os alarmes e suas características;

- ***Site Web*** – Este módulo corresponderá ao servidor *Web* que disponibilizará a página aos utilizadores do sistema e receberá os dados do middleware. Este módulo deverá conter as seguintes características:

  **S20. Autenticação** – permitirá acesso só aos utilizadores registados e com direito de acesso, através de *login* e palavra-chave;

**S21. Gerir equipamentos** – permitirá listar, criar, editar, apagar, configurar os equipamentos do sistema;

**S22. Gerir pessoas/objectos** – permitirá listar, criar, editar, apagar, configurar/associar pessoas/objectos do sistema;

**S23. Perfil de utilizadores** – será implementado privilégios nas sessões no *browser*, criando assim vários tipos de modo de sessão;

**S24. Encriptação** – deverá ser implementada um método de encriptação nas palavras-chave que seja compatível com a programação a ser utilizada;

**S25. Localização em tempo real** – pretende visualizar em tempo real a localização de uma determinada *tag* (deverá estar associada a um utilizador ou objecto);

**S26. Gerir permissões de acesso** – pretende listar, criar, editar, apagar, configurar as permissões de acesso;

**S27. Detalhes do edifício** – possibilita verificarem as características do edifício, tipo de edifício, número de andares, possibilidade de ver as plantas do edifício;

**S28. Comunicação com o middleware** – permitirá receber e transmitir dados do middleware correspondente a localização, permissões, alarmes das tags, teste de comunicação com um determinado leitor, mudança de endereço IP de um leitor e reiniciar um leitor;

**S29. Comunicação com *browsers*** – deverá ser possível enviar dinamicamente ao cliente a informação relevante.

### 3.4.2. Características do Hardware

De seguida serão apresentados os requisitos funcionais que cada módulo deve implementar para o *hardware*. Os módulos do *hardware* deverão ser os seguintes

- **Tag** – etiqueta electrónica de RFID, as suas características deverão ser as seguintes:

  **H1. Transmitir periodicamente –** pretende que a *tag* envie a sua identificação periodicamente de modo a que um ou vários leitores a possa escutar e enviarem posteriormente essa informação para o servidor TraceMe, ficando o middleware a saber que a *tag* está no raio de acção de um ou vários leitores;

**H2. Detectar na frequência F1** – pretende que a *tag* seja detectada na frequência F1 para enviar essa informação ao Servidor TraceMe através da frequência F2, ficando o *middleware* a saber da localização da *tag*;

**H3. Transmitir estado** – pretende que a *tag* envie informação do estado da bateria e o seu estado (se foi forjado ou não);

- **Leitor –** Equipamento RFID que permite detectar as *tags* RFID, as suas características são as seguintes:

  **H4. Receber sinais RF** – deverá intersectar os sinais de identificação das tags na frequência F2;

  **H5. Módulo *Ethernet*** – deverá suportar internamente a adição de um módulo *Ethernet*.

  **H6. Comunicar com o servidor TraceMe** – deverá comunicar com o Servidor TraceMe através do módulo *Ethernet* (H5);

  **H7. Suportar entradas e saídas digitais –** permitirá interligar com outros sistemas;

  **H8. Suportar porta RS232** – deverá estar disponível um suporte para comunicação RS232, para que possam ser feitas algumas configurações locais ou interligação com outros dispositivos locais;

  **H9. Suportar antenas direccionais** – deverá ter suporte a antenas para identificar qual é o sentido da *tag*.

- **Servidor TraceMe** – Servidor que deverá conter os serviços do *middleware*, a base de dados e o servidor *Web*. Este módulo deverá ter as seguintes características:

  **H10. Módulo Ethernet** – deverá suportar internamente o módulo *Ethernet* para receber as comunicações dos leitores (H6);

  **H11. Servidor *Web*;**

  **H12. Servidor base de dados;**

  **H13. Suportar o funcionamento dos serviços** – deverá suportar a tecnologia que vai ser utilizada para o desenvolvimento dos serviços no servidor *Web* (H11);

Os requisitos mínimos (disco, processador e memória RAM) para o sistema funcionar depende do número de *tags* e de leitores.

- **Computador Cliente** – Computador que acede as funcionalidades do sistema através da *Web*, as suas características são as seguintes:

  **H14. Browser –** Suporte para os browsers Internet Explorer® 6.0 ou superior e Firefox® 2.0 ou superior.

## 3.5. Middleware RFID a desenvolver

O *middleware* proposto pretende receber dados provenientes dos equipamentos RFID, e reenvia para o servidor *Web* e para uma base de dados. A sua principal função deverá ser controlar o fluxo de dados entre os equipamentos RFID e os sistemas de integração (base de dados e servidor *Web*), permitindo funções de agregação e filtragem de informações inválidas ou incorrectas. A sua principal responsabilidade deverá ser a qualidade e utilidade da informação.

Nas comunicações com outros sistemas integrados, o *middleware* proposto deverá sempre utilizar o conceito do *driver* genérico, que consiste numa camada genérica específica só para comunicar com esse sistema, permitindo assim uma plataforma móvel, modelar e adaptável à implementação que seja proposta.

A comunicação com a base de dados deverá suportar ligações com o MySQL® [1], SQL Server® [2] ou PostgreSQL®. Pode inclusive comunicar com várias bases de dados ao mesmo tempo, utilizando a camada genérica específica para a comunicação com a base de dados. Através deste conceito o *middleware* não é dependente de uma determinada base dados e torna-se mais robusto, que é uma grande vantagem, porque permite escolher a base de dados para cada implementação. Mas para cada implementação é necessário adaptar os procedimentos para a base de dados específica. A camada genérica, que pode ser considerada como *driver* genérico de base de dados, precisa de saber qual é o *driver* específico da base de dados.

Outro sistema integrado deverá ser um servidor HTTP que também deveram suportar o mesmo conceito de *driver* genérico, mesmo acontece com as comunicações para os equipamentos RFID. Dependendo da aplicação, os sistemas base de dados e servidor HTTP, podem ser inseridos no mesmo computador ou distribuídos em vários computadores.

As funcionalidades fornecidas pelo *middleware* proposto são: instalação; actualizações; gestão da configuração dos leitores; notificações; excepções; gestão de *tags*; configuração do sistema; filtragem

[1] Sítio da MySQL®: *http://www.mysql.com/* (Ultima vez visitado em Dezembro 2008)

[1] Sítio da SQL Server®: *http://www.microsoft.com/sqlserver/2008/en/us/default.aspx* (Ultima vez visitado em Dezembro 2008)

de eventos; teste de cobertura; mecanismo de localização; alarmes e verificar o funcionamento dos leitores. Estas funcionalidades encontram-se representadas na Figura 3-11.

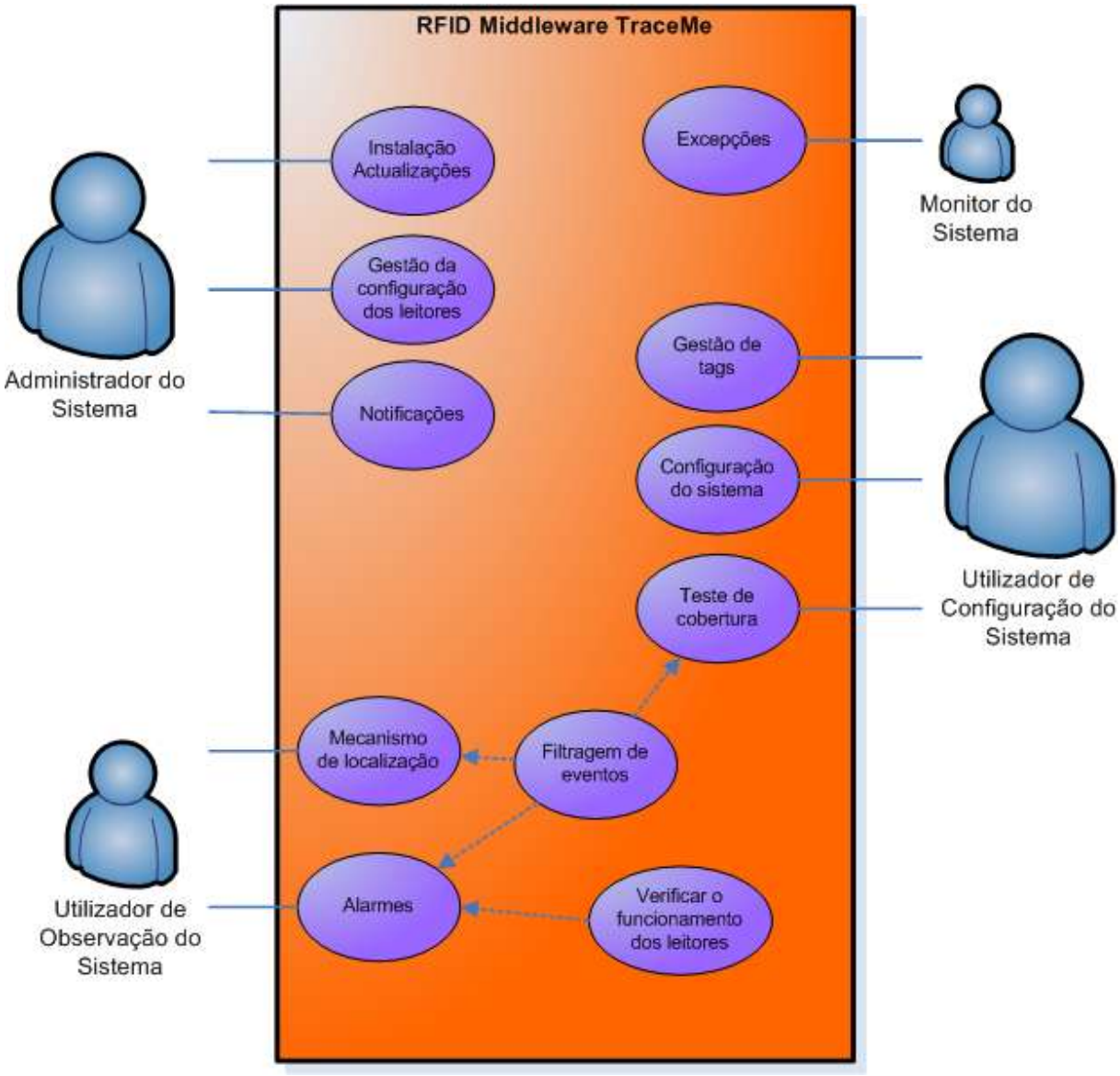



Figura 3-11 – Funcionalidades do *middleware* proposto.

Algumas funcionalidades referidas anteriormente são destinadas aos utilizadores, analisando a Figura 3-11 os "Administradores do Sistema" contêm as funcionalidades "Instalação", "Actualizações", "Gestão da configuração dos leitores" e "Notificações". O "Utilizador de Observação do Sistema" corresponde ao "Mecanismo de localização" e "Alarmes". O "Utilizador de Configuração do Sistema" pertence ao "Teste de Cobertura", Configuração do sistema" e "Gestão de *tags*". O "Monitor do Sistema" corresponde aos "Relatórios", "Estatísticas" e "Excepções". As características de cada uma das funcionalidades são:

- **Instalação** – todos os componentes do *middleware* devem ser fáceis de instalar, contendo um instalador rápido e fácil de interpretação;

- **Actualizações** – sempre que for necessário actualizar o *middleware* deverá ser de forma simples e com o menor tempo de interrupção possível;

- **Gestões da configuração dos leitores** – os parâmetros dos leitores deveram ser possíveis de configurar, permitindo a comunicação bidireccional entre os leitores e o *middleware* (associadas as características S7 e S28);

- **Notificações** – esta funcionalidade deverá permitir comunicar com os leitores para configurar o endereço IP, realizar testes de comunicação, reiniciar e configurar o intervalo de comunicação das *tags* (relacionado com as características S5 e S28);

- **Excepções –** todas as excepções deveram ser guardadas em ficheiros *logs* para que sejam analisadas. Para distinguir as excepções deverá ser implementado um nível de gravidade para as excepções;

- **Gestão de *tags*** – esta funcionalidade deverá permitir associar, desassociar e identificar uma determinada *tag* com uma pessoa ou objecto (pertencentes as características S14, S21 e S22);

- **Configuração do sistema** – todos os parâmetros do *middleware* deverá ser configurado num ficheiro de configuração, este ficheiro deverá ser possível alterado quando o sistema estiver a funcionar;

- **Filtragem de eventos** – através desta funcionalidade os eventos deveram ser filtrados para não existir eventos redundantes (associada a característica S6);

- **Teste de cobertura** – esta funcionalidade depende da "Filtragem de eventos", deverá permitir receber os eventos de uma determinada *tag* e determinar os leitores que se encontram ao seu alcance; (relacionado com as características S1 e H4);

- **Mecanismo de localização** – este mecanismo deverá localizar em tempo real todos as *tags* em movimento, também é dependente da "Filtragem de eventos" (pertence a característica S9);

- **Verificar o funcionamento dos leitores** – esta funcionalidade deverá analisar se existe equipamentos avariados, caso existe cria um "Alarme" do tipo comunicação de leitor; (relacionadas com as características S11 e S19);

- **Alarme** – esta funcionalidade pode ser criada através da "Filtragem de eventos" e do "Verificar o funcionamento dos leitores". Esta funcionalidade deverá ser possível detectar a falta de comunicação com os leitores, a detecção de ausência de comunicação das *tags*, bateria fraca das *tags*, violação das *tags* e detecção de acessos não autorizados (associadas as características S11 e S19).

Na Figura 3-12 encontra-se o *middleware* proposto para o TraceMe. Esta camada está dividida pelo "Adaptor" e pelo "Proxy". O "Adaptor" permite a comunicação com os equipamentos e com o "Proxy", recebe os dados dos equipamentos e transmite-os descodificados para o "Proxy". Recebe ainda os dados do "Proxy" e transmite-os codificados para um determinado equipamento.

O "Proxy" corresponde ao "ProxyEvent", "ProxyWeb" e "ReaderWatcher". O "ProxyEvent" recebe os dados enviados pelo "Adaptor". Suporta ainda a filtragem dos dados, mecanismo de localização, detecção de alarmes, teste de cobertura e gestão de *tags*. O "ReaderWatcher" permite verificar o funcionamento dos leitores. O "ProxyWeb" trata das notificações.

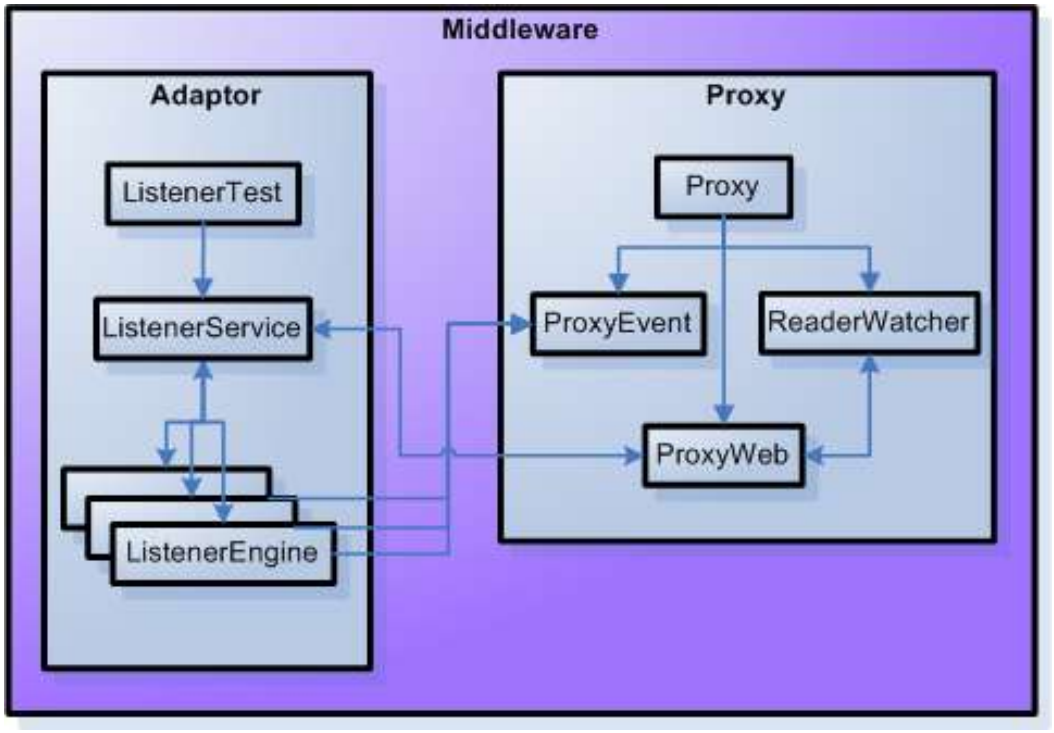



Figura 3-12 – *Middleware* proposto para TraceMe.

### 3.5.1. Comparação com outros *middlewares*

Após a descrição de alguns *middlewares* relacionadas com a tecnologia RFID e também a descrição do *middleware* proposto para o TraceMe, foi realizado uma comparação entre as características (Tabela 3-1) de cada um.

Tabela 3-1 – Comparação entre os middlewares.

| Middleware | Localização tempo real | Controlo de inventário | Características |
|---|---|---|---|
| Sun Java System RFID Software | | ✓ | Suportar níveis elevados de confiabilidade e de escalabilidade para as redes de EPC e desenvolvido em Java. |
| RFID Anywhere LIS® | ✓ | ✓ | Uma infra-estrutura flexível que integra a lógica do negócio, localização e rastreamento de objectos, RTLS e desenvolvido em .Net. |
| ALE Server® | | ✓ | Boa performance e fácil utilização, integração do *hardware* RFID com modelos de negócios existentes e desenvolvido em Java. |
| TraceMe | ✓ | | Conceito de *driver* genérico para comunicar os sistemas integrados, permitindo assim uma plataforma móvel, modelar e adaptável à implementação que seja proposta e desenvolvido em Java. |

Através da Tabela 3-1 pode analisar-se que cada *middleware* contém as suas próprias características de funcionamento. O *middleware* TraceMe proposto pretende implementar as melhores características de cada sistema, como por exemplo suportar níveis elevados de confiabilidade, infra-estrutura flexível que integra a lógica do negócio, boa performance e fácil utilização.

## 3.6. Algoritmos de localização

O algoritmo de localização permite determinar a localização em tempo real de pessoas ou objectos. Através deste é definido a precisão da localização e as zonas de controlo de acesso. O algoritmo influência as funções "Filtragem de eventos" e "Mecanismo de localização" anteriormente descritos (secção 3.5). Os algoritmos propostos são três e são os seguintes:

- **Simples** – recolhe os eventos gerados pela frequência F1 (descrito na secção 2.3.1) e ignora os eventos gerados pela frequência F2 (descrito na secção 2.3.1). Assim a localização só é feita quando as *tags* são excitadas pela F1, permitindo a localização por zona. Desta forma, um leitor está associado a uma só zona e uma zona pode estar associada a vários leitores. A precisão da localização é pela zona onde o leitor está posicionado;

  Portanto, na implementação "Filtragem de eventos" é essencial ignorar os eventos da F2 e filtrar os eventos da frequência F1 para obter o último evento. No "Mecanismo de localização" é necessário consultar a base de dados e verificar qual é a zona do leitor que detectou o evento. Na Figura 3-13 está apresentado um exemplo de funcionamento do algoritmo, indicando o alcance de cada leitor, e com três leitores só é possível distinguir três zonas (A, B e C);

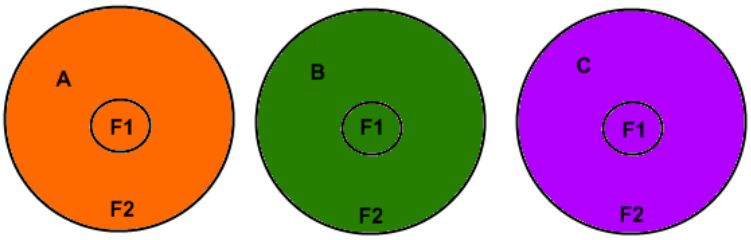



Figura 3-13 – Exemplificação do algoritmo Simples e Intermédio.

- **Intermédio** – neste algoritmo a localização também é baseada pela frequência F1 como no algoritmo Simples, mas utiliza a frequência F2 para actualizar o evento. Podendo assim detectar se alguma *tag* deixa de comunicar com o sistema. A precisão da localização é a mesma do algoritmo Simples;

  Na implementação a diferença para o algoritmo Simples é a utilização do tempo dos eventos da frequência F2 para actualizar o tempo do evento de localização. Na Figura 3-13 apresenta um exemplo do algoritmo, com três leitores só é possível distinguir três zonas (A,B e C);

- **Avançado** – o algoritmo é baseado através da triangulação dos eventos recebidos na frequência F1 e F2 (utilizando o método "Potência do sinal" descrito na secção 2.8). Neste algoritmo um leitor pode corresponder a várias zonas e uma zona pode corresponder a vários leitores. A precisão da localização é melhor e corresponde a área de intersecção dos eventos recebidos;

  A implementação "filtragem de eventos" é essencial distinguir entre os eventos da frequência F1 e F2 e ignorar os eventos redundantes. No "Mecanismo de localização" é mais complexo que os anteriores porque é necessário identificar a zona no qual os eventos intersectam entre si. Na Figura 3-14 está apresentado um exemplo de funcionamento do algoritmo, e neste exemplo com três leitores é possível distinguir dez zonas (A, B, C, D, E, F, G, H,I e J).

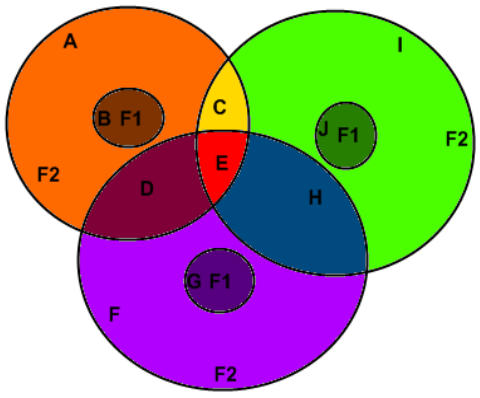



Figura 3-14 – Exemplificação do algoritmo Avançado.

## 3.7. Base de Dados

A concepção de uma base de dados é um dos componentes mais importante para o desempenho geral do sistema, através desta são armazenados todas as movimentações das *tags*, todos os alarmes gerados e todas as configurações precisas para o funcionamento correcto da aplicação.

Como principais características a base de dados deverá armazenar muitos dados, para tal deverá ser robusta, suportar tabelas com muitos registos, fácil de utilização e interpretação. Outra característica será licença gratuita para fins comerciais.

O diagrama da base de dados proposto está em anexo na secção A.2. As informações mais importantes para armazenar são as seguintes:

- **Alarmes** – todos os alarmes são armazenados, cada alarme tem um tipo de alarme associado e informações relevantes do alarme gerado;
- **Eventos** – todas as detecções de uma *tag* são guardadas na base de dados, cada evento tem uma data inicial e final, a unidade associada a este evento, a zona do evento e a identificação da *tag*;
- **Unidades** – cada unidade tem um tipo de unidade associado, para distinguir entre pessoas e objectos, contém ainda um tipo de alarme associado e guarda as informações específicas dessa unidade;
- ***Tags*** – todas as *tags* tem que ser inseridas na base de dados, senão as *tags* não são reconhecidas pelo sistema;
- **Leitores** – todos os leitores tem que ser armazenados, senão o mecanismo de localização não conseguir identificar a zona onde se encontra a *tag*, armazenada ainda as configurações de cada leitor e o seu tipo para o *middleware* conseguir distinguir leitores de marcas diferentes;
- **Zona –** cada zona está associado a uma planta, um tipo de zona, um tipo de alarme, a sua posição na planta e informações sobre o posicionamento dos elementos visuais;
- **Configurações das plantas –** as plantas são armazenadas e são associados a um determinado piso e edifício;
- **Utilizadores da página** – os utilizadores que têm acesso ao *browser* são armazenados com as suas informações e também o tipo de acesso à página;
- **Horários** – as sessões entre as *tags* e uma unidade podem ser determinados por horários, permitindo controlar os acessos dependendo da hora;

- **Perfis** – os perfis de unidade e utilizador são armazenados para conseguir melhor desempenho e organização.

## 3.8. Servidor Web

Para aceder às funcionalidades do sistema é necessário aceder a um *browser* que está alojada num servidor HTTP. Este servidor deverá suportar o "WebServer", exemplificado na Figura 3-15, para interligar os dados provenientes do *middleware* para as páginas *Web*. A licença do servidor para fins comerciais deverá ser gratuita. As principais funcionalidades deste servidor são as seguintes:

- **Identificar as opções de visualização** – Para cada acesso *Site Web* é criado um objecto "BrowserObject" para guardar as informações de cada página, como ID da sessão e identificar qual a funcionalidade que está a ser executada. Através desta informação o servidor só envia os dados relativos à funcionalidade em execução. O servidor deverá conseguir transmitir dados para uma determinada página sem precisar de receber pedidos;
- **Notificações** – o "WebServer" deverá receber notificações ("NotificationService") das páginas *Web* e guardar num objecto "EventObject" para ser tratado pelo *middleware*, depois de ser tratadas o servidor deverá transmitir o resultado da notificação para o respectivo *browser*.

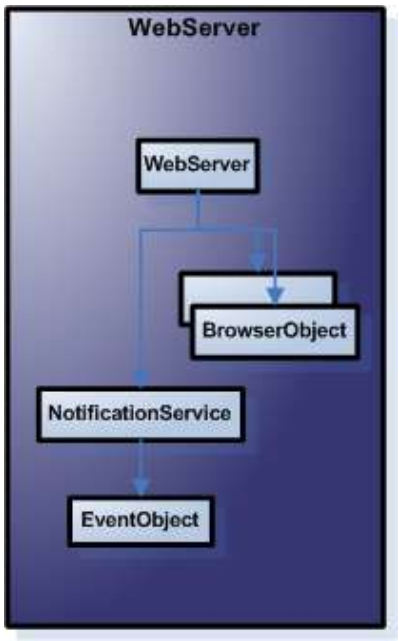



Figura 3-15 – WebServer proposto para o TraceMe.

## 3.9. Interligação entre camadas

O sistema pode dividir-se em quatro camadas, "Equipamento Físico", "Middleware", "WebServer" e "Browser". A interligação entre as camadas está representada na Figura 3-16. As interligações entre as camadas geram quatro fluxos principais. O primeiro fluxo (1.x) representa a detecção de um evento até a visualização no *browser*. O segundo fluxo (2.x) exemplifica uma notificação desde do *browser* até ao leitor e a sua resposta devolvida ao *browser* correspondente. O terceiro fluxo (3.x) demonstra o funcionamento do mecanismo de análise dos leitores e o quarto fluxo (4.x) corresponde a comunicação entre o "Browser" e o "WebServer".

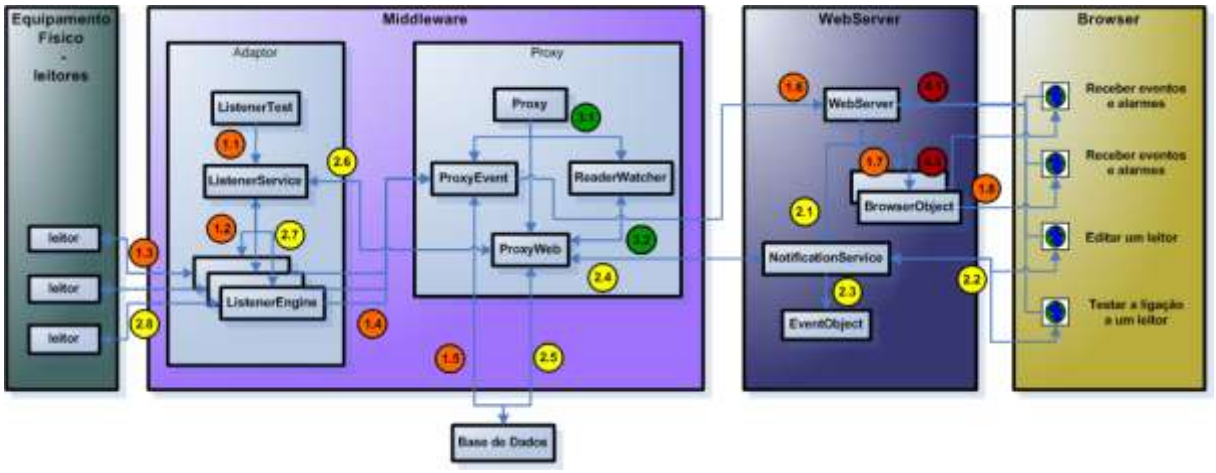



Figura 3-16 – Fluxo de dados entre as várias camadas.

Através do serviço "ListenerTest" inicia-se o primeiro fluxo que cria um processo designado por "ListenerService" (1.1). Este processo é um *driver* genérico em que para cada leitor que faz uma ligação cria um processo com o seu *driver* específico designado por "ListenerEngine" (1.2). Esse processo vai receber todas as informações do respectivo leitor (1.3). A informação recebida é descodificada e estruturada num formato específico, para ser enviada para o "Proxy", mas propriamente o serviço "ProxyEvent" (1.4). Este serviço processa e filtra todos eventos do leitor, recorre a base de dados (1.5) para armazenar a informação e também executa várias funcionalidades inseridas no *middleware*, como por exemplo o mecanismo de localização e a detecção de alarmes. Após de processar os eventos, a informação é enviada para o servidor "WebServer" (1.6). Este servidor analisa a informação recebida e verifica através do "BrowserObject" (1.7) quais as páginas que precisam dessa informação e de seguida transmitem a informação para as páginas correspondentes (1.8).

Pedidos e resposta de notificação corresponde ao segundo fluxo. Para pedir uma notificação é fundamental que o serviço "NotificationService" esteja em execução (2.1). A notificação é criada a

partir de um *browser* e comunica com o serviço "NotificationService" (2.2). Este serviço guarda a informação no objecto "EventObject" (2.3) e envia a notificação num formato específico para o serviço "ProxyWeb" (2.4). O "ProxyWeb" processa a notificação e caso seja necessário comunica com a "Base de Dados" (2.5) para obter mais informações. Após ter a notificação com todos os dados é enviado para o "ListenerService" (2.6), neste serviço verifica qual é o leitor correspondente e envia essa informação ao processo correspondente (2.7). O processo codifica a informação e transmite para o leitor (2.8). O fluxo da informação da resposta é o mesmo, mas começa pelo leitor e acaba no *browser*.

O terceiro fluxo consiste na funcionalidade detectar leitores que não responde aos testes de comunicação. Primeiro é necessário iniciar o serviço "ReaderWatcher" (3.1), o serviço envia uma notificação de teste de comunicação para cada leitor através do "ProxyWeb" (3.2). A partir deste ponto é considerado como se fosse uma notificação normal, mas a resposta é enviada para o "ReaderWatcher".

A configuração dos parâmetros das funcionalidades disponíveis para os utilizadores corresponde ao quarto fluxo. Sempre que uma configuração ou opção é alterada na página é transmitido essa alteração para o "WebServer" (4.1), de seguida o servidor guarda essas alterações no objecto "BrowserObject" associado a página (4.2). Através desta comunicação é possível filtrar os dados a transmitir para as páginas *Web*.

## 3.10. Conclusões

O sistema TraceMe deverá controlar automaticamente os acessos a diferentes áreas de um edíficio, a localização da zona de um utilizador específico ou de um equipamento em tempo real e accionar alarmes de detecção de ausência no edifício e falha de comunicação com os equipamentos.

A arquitectura física proposta é constituída por leitores e *tags* RFID activo, um servidor *Web*, uma base de dados, um *middleware* e pelo menos um computador para aceder as funcionalidades do sistema. A arquitectura lógica proposta é organizada por quatro módulos, os "Equipamentos RFID", o "Software Middleware*"*, a "Base Dados" e "Serviço de dados".

As características do *software* correspondem ao *middleware*, a base de dados e o *site Web*. No *middleware* são especificadas as comunicações com os equipamentos RFID e com as plataformas de aplicação. A base de dados é detalhado a informação que é necessária armazenar e o *site Web* é especificado as funcionalidades e a comunicação com os restantes sistemas da solução.

O *middleware* proposto pretende controlar o fluxo de dados entre os equipamentos RFID e os sistemas de integração, uma base de dados e um servidor *Web*, permitindo funções de agregação e filtragem de informações inválidas ou incorrectas.

Para determinar a localização em tempo real de pessoas ou objectos são definidos três algoritmos, o simples, o intermédio e o avançado. O algoritmo simples e intermédio permite localização por zonas, o avançado permite localização por triangulação dos sinais recebidos. O algoritmo intermédio e avançado permitem a detecção de ausência de comunicação das *tags*.

Os dados que são necessários armazenar numa base de dados deverão ser os eventos gerados pelos equipamentos, os alarmes accionados pelo sistema, informação relativas as unidades e utilizadores da página, as configurações dos equipamentos RFID, das plantas e zonas do edifício.

As principais funcionalidades do servidor *Web* proposto são a identificação das opções de visualização e a criação de notificações. A identificação permite caracterizar cada *browser* que esteja a aceder ao servidor e as notificações permite criar eventos de configuração do sistema.

As interligações entre as camadas do sistema proposto geram quatro fluxos principais, a detecção de um evento até à visualização no *browser*, uma notificação desde do *browser* até ao leitor e a sua resposta para o *browser* correspondente, o mecanismo de análise dos leitores e a comunicação entre o *browser* e o "WebServer".

Com a especificação do sistema detalhada podemos prosseguir com a implementação do projecto. No próximo capítulo são apresentados os equipamentos RFID, a base de dados, o servidor *Web*, o desenvolvimento do *middleware* e a integração de sistemas.

# 4. Implementação do Projecto

A implementação do projecto consiste na apresentação dos equipamentos RFID, da base de dados e do servidor *Web*, do desenvolvimento do *middleware* e da integração de sistemas.

## 4.1. Equipamento RFID

A escolha dos equipamentos reflectem nos requisitos identificados na secção 3.4.2. Os principais requisitos são os dois modos de funcionamento (frequência F1 e F2), o leitor deverá comunicar através do protocolo TCP/IP com o sistema e as *tags* deveram ser activos para o sistema de localização. Os equipamentos escolhidos que preenchiam todos os requisitos são de um parceiro integrador da ISA™. Na secção seguinte os equipamentos são apresentados e as suas características são descritas.

### 4.1.1. Equipamentos utilizados

Os equipamentos utilizados para o desenvolvimento do *software* e dos testes do sistema TraceMe foram os leitores e as *tags* do parceiro integrador da ISA™. Existem dois tipos de leitores e quatro tipos de *tags*. Os leitores são designados por LTR-001 (Figura 4-1) e LTR-002 (Figura 4-2). As *tags* são designadas por IDA-003 (Figura 4-3), IDA-004 (Figura 4-4), IDA-005 (Figura 4-5) e IDA-007 (Figura 4-3).


Figura 4-1 – Leitor LTR-001.

Figura 4-2 – Leitor LTR-002.

Figura 4-3 – *Tag* IDA-003 e IDA-007.

Figura 4-4 – *Tag* IDA-004.

Figura 4-5 – *Tag* IDA-005.

O leitor LTR-001 é receptor/emissor do sistema de leitura de *tags* RFID activas bidireccionais ou unidireccionais de longo alcance, com interface *Ethernet*. Através desta unidade é possível a detecção, configuração e interacção com as *tags*. Por existir a componente emissora, na frequência 125kHz, é possível o envio de telecomando a partir das *tags* bidireccionais e, por isso, uma melhor performance do sistema e uma maior precisão na localização das *tags*. Com este equipamento podem-se ainda ligar alarmes, sensores, barreiras de controlo de acessos, *displays* LCD ou teclados através de comunicações por RS232/485 ou via *Ethernet*, como é exemplificado no diagrama da Figura 4-6.

As principais características deste leitor são a interface *Ethernet*, alcance de 2 metros na transmissão, alcance de 30 metros na recepção, na frequência 868MHz e alta segurança por encriptação dos dados. As principais aplicações típicas são controlo de acessos, localização de pessoas em edifícios, limitador de acessos restritos, acesso proibido por ausência da *tag*, detecção de ausências de pessoas, prática de desportos, localização de máquinas em movimento, validação de identidade, controlo logístico, controlo de domótica e elevadores, protecção de idosos e crianças (mais informações ver Anexo A.3).

O leitor pode ter duas antenas externas em vez da antena interior de 125kHz, permitindo identificar o sentido das *tags*. A Figura 4-7 está exemplificado as antenas exteriores, neste caso as antenas são horizontais e são colocadas no piso para detectar a entrada ou saída de carrinhos.

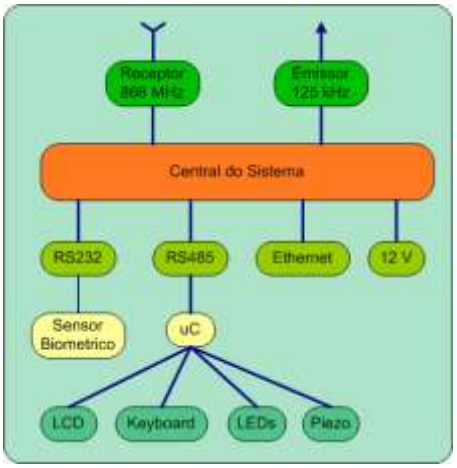



Figura 4-6 – Diagrama de blocos do leitor LTR-001.

Figura 4-7 – Exemplo prático das antenas exteriores.

Com o leitor LTR-002 só é possível receber sinais 868MHz. Este leitor tem todas as características e aplicações típicas do leitor LTR-001 menos o emissor de 125 kHz, como se pode verificar no diagrama na Figura 4-8 (para mais informações ver Anexo A.4).

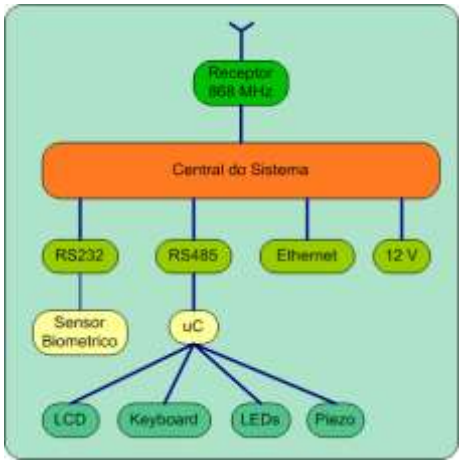



Figura 4-8 – Diagrama de blocos do leitor LTR-002.

A *tag* IDA-003 é uma etiqueta de identificação pessoal RFID, activo, bidireccional, de longo alcance e com elevada autonomia capaz de localizar uma pessoa num edifício. Por serem *tags* activas e com memória interna, consegue-se parametrizar, configurar e inserir novos dados em tempo real na *tag* e também enviar telecomandos (mensagens específicas de configuração) através da *tag*. Estes dados são então transportados pelo utente da *tag* interagindo com o sistema cada vez que é detectado.

Tem como principais características o alcance de 30 metros na transmissão, alcance de 2 metros na recepção, alta segurança por encriptação dos dados, sem manutenção ou recarga da bateria, dimensões standardizadas para *tags* RFID, completamente estanque a poeiras e água e uma autonomia de 5 a 10 anos. As aplicações típicas são as mesmas do leitor LTR-001. Na Figura 4-9 é ilustrado o diagrama de blocos da *tag* (para mais informações ver Anexo A.5).

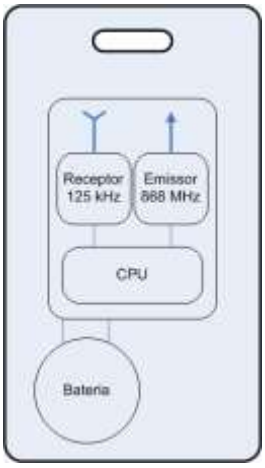



Figura 4-9 – Diagrama de blocos da *tag* IDA-003 e do IDA-004.

Em relação a *tag* IDA-004 tem as mesmas características da *tag* IDA-003 só que o formato é mais robustez e mais próprios para associar a objectos. As aplicações típicas são gestão de carrinhos, localização de bens em edifício, cronometragem em desportos, protecção de objectos contra roubos e validação de dados. O diagrama de blocos da *tag* pode-se verificar na Figura 4-9 (para mais informações ver Anexo A.6).

Outro tipo de *tag* é a IDA-005 que corresponde a uma etiqueta RFID activa, unidireccional, de longo alcance e com elevada autonomia, capaz de identificar objectos diferentes a longa distância. Tem as mesmas características da *tag* IDA-003 retirando o receptor 125 kHz. As aplicações típicas são controlo logístico de veículos, cargas e contentores, capturar informação sobre passagem de veículos, localização de veículos e objectos em grandes espaços, detecção de omissão de objectos, protecção de mercadorias em exposição e em *stock* (para mais informações ver Anexo A.7). A *tag* IDA-007 é muito idêntico ao IDA-003 até mesmo no formato, a diferença principal é que a IDA-007 não tem o receptor 125 kHz. As aplicações típicas são as mesmas do leitor LTR-001 (para mais informações ver Anexo A.8). O diagrama de blocos das *tags* IDA-005 e IDA-003 pode-se verificar na Figura 4-10.

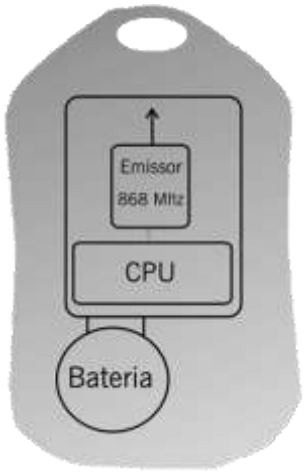



Figura 4-10 – Diagrama de blocos da *tag* IDA-005 e da IDA-007.

### 4.1.2. Funcionamento entre os equipamentos

O funcionamento dos equipamentos do parceiro integrador da ISA™ entre o leitor e *tag* são descritos da seguinte forma:

- O leitor excita periodicamente a *tag* na frequência 125 kHz, como é exemplificado na Figura 4-11;

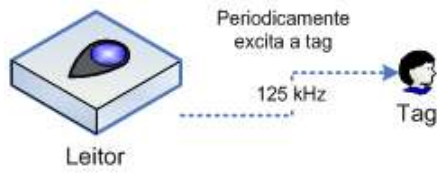



Figura 4-11 – Leitor excita a *tag* na frequência 125 kHz.

- Resposta da *tag* à excitação do leitor é enviar um sinal em 868MHz com a informação do RSSI, se a *tag* foi violada, o estado da bateria, o número de série do leitor que interrogou-o e o número de série da *tag*, como é exemplificado na Figura 4-12.


Figura 4-12 – Resposta da *tag* á excitação do leitor.

- Quando a *tag* está no alcance nos 125 kHz é possível programar a comunicação do *beacon* (se é periódico ou não e caso seja periódico pode-se indicar o intervalo de tempo), como é ilustrado na Figura 4-13.

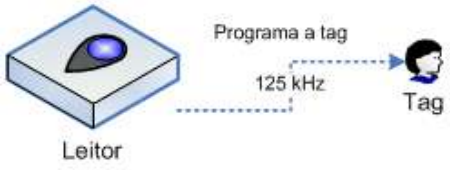



Figura 4-13 – Programar uma *tag*.

- A *tag* responde o resultado do comando através de um sinal em 868 MHz, como é exemplificado na Figura 4-14.


Figura 4-14 – Resposta da *tag* ao comando.

- Se a *tag* estiver programada para transmitir o *beacon* envia um sinal periódico em 868 MHz com a informação do RSSI, o estado da *flag* de violação, o estado da bateria e o seu número de série, como é ilustrado na Figura 4-15.

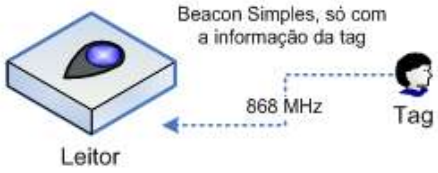



Figura 4-15 – Transmissão do *beacon* Simples.

- Com o modo funcionamento do duplo anel (explicado em detalhe no anexo A.3) activado, a *tag* só transmite um sinal em 868 MHz quando detectar uma transição de passagem. A Figura 4-16 é demonstrado o funcionamento do duplo anel, a "Tag 1" não transmite nenhum sinal porque não passou pelos dois *loops*, a "Tag 2" também não transmite porque encontra-se no meio dos *loops* e a "Tag 3" transmite porque efectuou uma transição pelos dois *loops*.

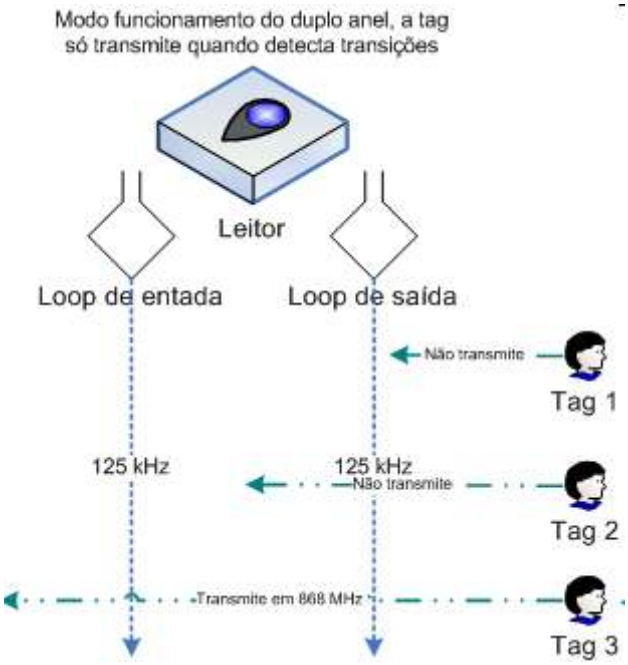



Figura 4-16 – Modo funcionamento do duplo anel.

## 4.2. Base de dados e servidor *Web*

Os sistemas de integração existentes no sistema são a base de dados e o servidor HTTP. No caso da base de dados inicialmente foi proposto ser o MySQL® por ser uma das base de dados mais utilizada e referenciada, mas como esta base de dados precisa de licença para fins comerciais é proposto a PostgreSQL® porque é robusta e não precisa de licença. Tem mais de 15 anos de desenvolvimento activo e uma arquitectura sólida que permite uma reputação forte na confiabilidade, integridade dos dados e precisão. Funciona em todos os sistemas operativos principais, incluindo Linux, UNIX® [1] (AIX® [2] - *Advanced Interactive eXecutive*, DEB - *Berkeley Software Distribution*, HP-UX® [3] - *Hewlett Packard UniX*, Solaris® [4], Tru64® [5]), e Windows® [6]. O tamanho máximo da base de dados é ilimitado [35].

No caso dos servidores HTTP foram inicialmente propostos o PHP [7] (*Hypertext Preprocessor*) e o Apache Tomcat® [8] por serem servidores que não precisem de licença para fins comerciais. Como os serviços são desenvolvidos em Java, o servidor proposto é o Apache Tomcat® porque suporta Java [9]. [36].

## 4.3. Desenvolvimento do *middleware*

A implementação do *middleware* consiste em duas partes distintas, o "Adaptor" e o "Proxy". A primeira parte corresponde à comunicação com os equipamentos físicos, portanto só recebe e envia mensagens dos equipamentos. A segunda parte, que é numa camada superior, consiste no cálculo de todas as funcionalidades do sistema, como por exemplo a filtragem de eventos, detecção de alarmes e detecção de acesso restritos. Nas secções seguintes são apresentados fluxogramas gerais das duas partes.

---

[1] Sítio da UNIX ®: *http://www-03.ibm.com/systems/p/os/aix/index.html* (Ultima vez visitado em Dezembro 2008)

[2] Sítio da AIX®: *http://www-03.ibm.com/systems/p/os/aix/index.html* (Ultima vez visitado em Dezembro 2008)

[3] Sítio da HP-UX®: *http://h20338.www2.hp.com/hpux11i/cache/324545-0-0-0-121.html* (Ultima vez visitado em Dezembro 2008)

[4] Sítio da Solaris ®: *http://www.sun.com/software/solaris/index.jsp* (Ultima vez visitado em Dezembro 2008)

[5] Sítio da Tru64®: *http://h30097.www3.hp.com/* (Ultima vez visitado em Dezembro 2008)

[6] Sítio da Windows ®: *http://www.microsoft.com/brasil/windows/default.mspx* (Ultima vez visitado em Dezembro 2008)

[7] Sítio da PHP: *http://www.php.net/* (Ultima vez visitado em Dezembro 2008)

[8] Sítio da Apache Tomcat®: *http://tomcat.apache.org/* (Ultima vez visitado em Dezembro 2008)

[9] Sítio da Java: *http://www.java.com/pt_BR/* (Ultima vez visitado em Dezembro 2008)

### 4.3.1. Interligação com os equipamentos

A interligação com os equipamentos RFID é através de um adaptador designado por "Adaptor". O fluxograma do "Adaptor" pode encontrar-se na Figura 4-17. O bloco inicial é o "ListenerTest" que precisa de comunicar com a base de dados para obter todas as informações dos leitores activos, de seguida inicia o serviço "ListenerService" e depois configura os parâmetros de inicialização da comunicação com os leitores. O serviço "ListenerService" fica a escuta de receber comunicações dos leitores ou então notificações do "Proxy". Este serviço comporta-se como um *driver* genérico, porque em cada ligação que recebe identifica-a e para cada caso específico determina o procedimento a fazer. Para cada ligação for um leitor então cria um processo específico para ele ("ListenerEngine"), este processo criado para cada leitor difere entre os modelos dos leitores, mas de uma forma geral tem um algoritmo semelhante. Se a ligação for do "Proxy" então verifica qual é o processo correspondente ao leitor que pertence a notificação, e indica-o da notificação que precisa de fazer. O processo recebe os dados da ligação e descodifica através de um protocolo específico do leitor, a seguir envia a informação para o "ProxyEvent" e ainda envia dados para o leitor caso tenha alguma notificação para enviar.

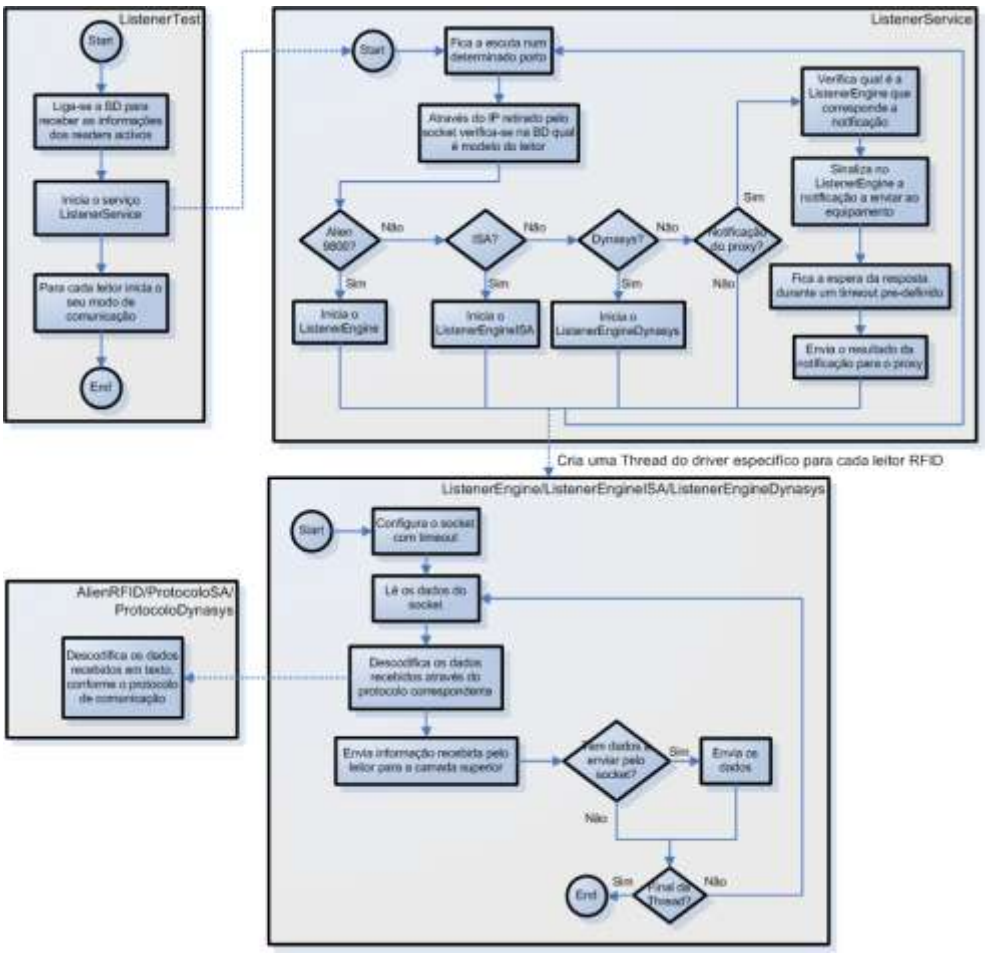



Figura 4-17 – Fluxograma do "Adaptor".

### 4.3.2. Proxy

Na Figura 4-18 é apresentado o fluxograma do "Proxy". O bloco inicial da figura é o "Proxy" que inicia os serviços "ProxyEvent", "ProxyWeb" e "ReaderWatcher". O serviço "ProxyEvent" fica a espera de receber ligações e cria um processo para cada uma, este processo é designado por "EventReceiver", a sua função é receber eventos do "Adaptor" e os alarmes dos leitores do "ReaderWatcher". Na primeira ligação recebida o serviço cria um processo designado "EventProcess". Este processo consiste na filtragem de todos eventos recebidos, no mecanismo de localização, na detecção dos alarmes de bateria fraca da *tag*, na detecção da violação da *tag*, na detecção de ausência da comunicação das *tags*, na detecção da falta de comunicação com os leitores e na detecção das *tags* em zonas não autorizadas. Ao longo destas funcionalidades o serviço acede a base de dados para inserir eventos e obter dados para o funcionamento dos mecanismos. Após de verificar todos os mecanismos envia a informação resultante para o "WebServer".

O serviço "ReaderWatcher" verifica se todos os leitores respondem aos testes de comunicação. Primeiro comunica com a base de dados para obter os leitores activos e depois para cada um faz um teste de comunicação através das notificações do "ProxyWeb". Caso detecta algum leitor a não funcionar envia essa informação para o "ProxyEvent" para que seja tratada e comunicada para o "WebServer".

Por fim, o serviço "ProxyWeb" fica à espera de receber notificações, que podem ser do "WebServer" ou do "ReaderWatcher", quando as recebe o serviço pode comunicar com a base de dados se precisar obter mais informações para criar um evento, que vai ser enviado para o "Adaptor". Após a transmissão do evento, o serviço fica a espera da resposta durante um tempo predefinido e quando recebe a resposta envia para quem estabeleceu a ligação.

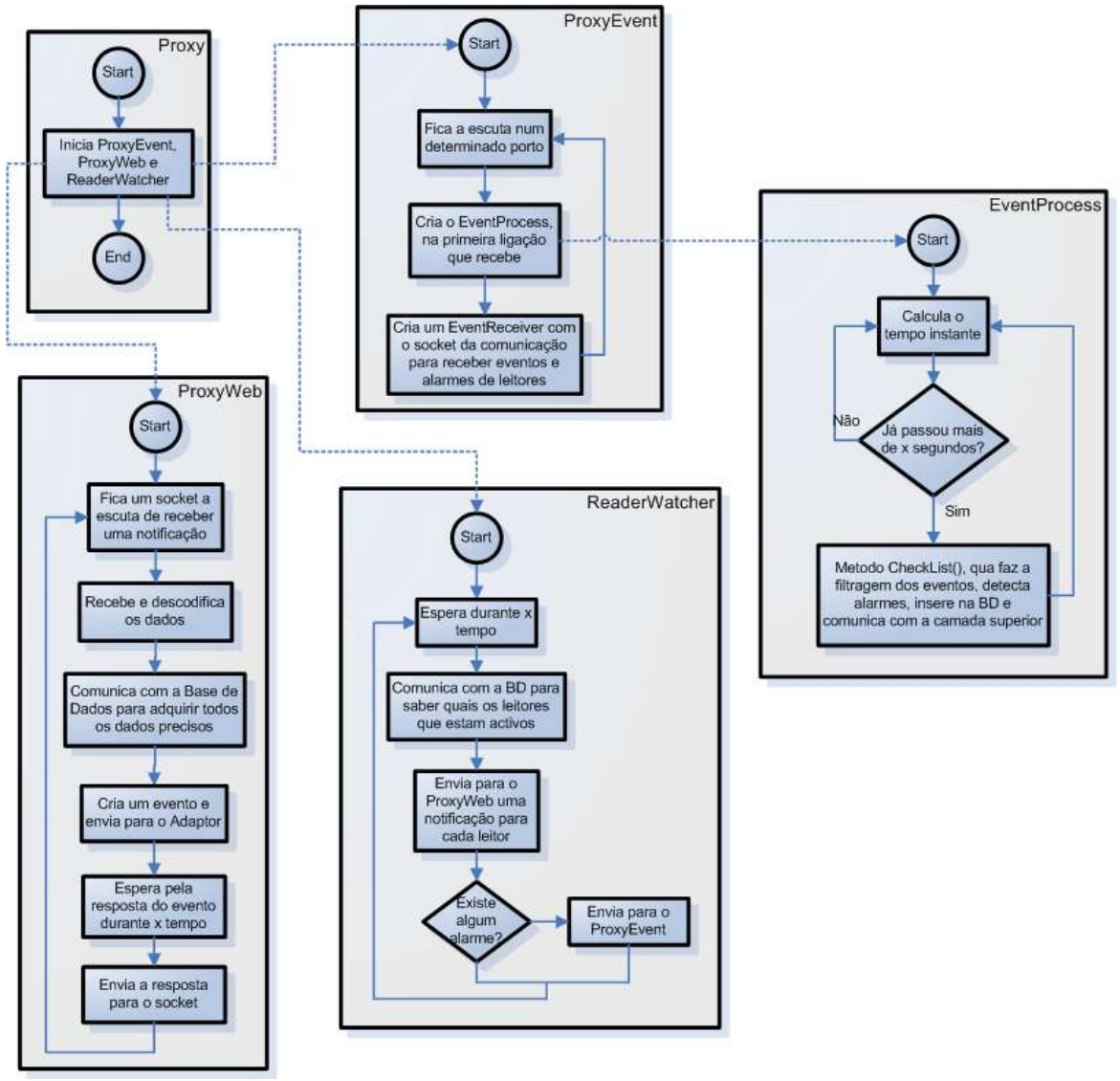



Figura 4-18 – Fluxograma do Proxy.

## 4.4. Integração de sistemas

A integração consiste no desenvolvimento de três sistemas, o servidor *Web*, o *middleware* e a base de dados. Com a comunicação entre o *middleware* e a base de dados é utilizado o JDBC e na comunicação entre o servidor *Web* e as páginas *Web* é implementado o Reverse Ajax (DWR).

Na implementação do servidor *Web* é proposto o "WebServer", que permite receber e processar dados relativos as localizações das *tags* e aos alarmes, comunicar dinamicamente essa informação para os respectivos *browsers*, receber configurações das funcionalidades e receber notificações para comunicar com os dispositivos RFID.

Quando um utilizador acede ao *browser* é criado uma ligação ao "Webserver", no método "initiation" (Figura 4-19) que permite criar um objecto "BrowserObject" com as informações do *browser*. Na primeira vez que esta função é executada cria um processo independente (Figura 4-20) para receber e

processar dados. O método "initiation" é sempre chamado quando o utilizador muda de ecrã de visualização, para que o servidor saiba quais as informações são precisas ser transmitidas, assim o servidor não precisa de transmitir todas as informações que recebe a todos os *browsers*.

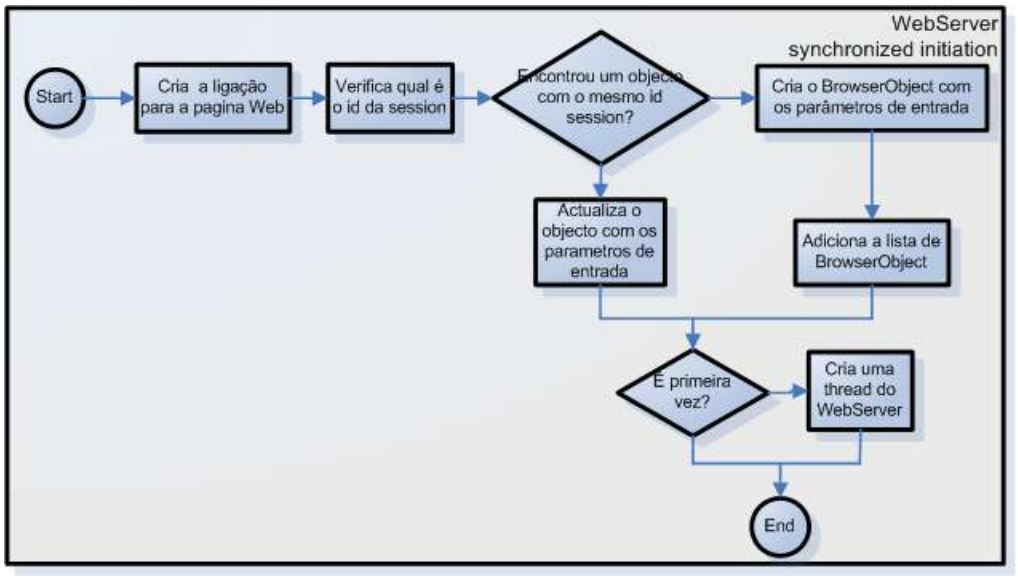



Figura 4-19 – Algoritmo do método "initiation".

Para receber e processar os dados é criado um processo independente, o seu principal método é o "run", exemplificada na Figura 4-20, que fica à espera de receber ligações do "Proxy", a informação transmitida para o servidor pode ser sobre a localização das *tags* e alarmes que foram detectados. Para cada informação é verificado quais são os *browsers* que precisam de receber os dados, que são transmitidos dinamicamente através do Reverse Ajax.

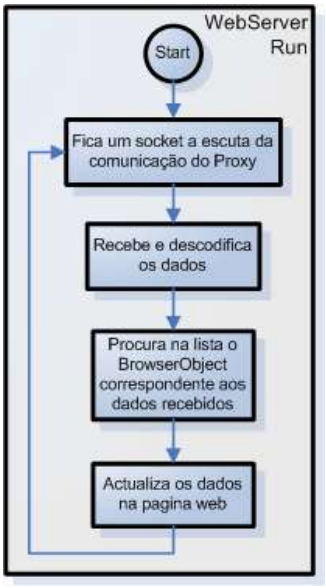



Figura 4-20 – Algoritmo do método "run".

Através do *browser* o utilizador pode criar notificações, que podem ser um teste de comunicação a um leitor, reiniciar um leitor, configurar o endereço IP do leitor e configurar o intervalo de transmissão do sinal das *tags*. Estas notificações são transmitidas através do Reverse Ajax e acedem ao método "AddEvent" do "WebServer", ilustrado na Figura 4-21. Este método permite criar um processo de cada vez para tratar da notificação, o processo é designado por "NotificationService" que transmite a notificação para o "ProxyWeb", e fica a espera da resposta para depois enviar para *browser* o resultado da notificação.

O servidor contém vários métodos independentes que podem ser executados a partir do *browser*, esses métodos correspondem a pedidos de informação e dados para o funcionamento do sistema. As funcionalidades que precisam de métodos independentes são: localização rápida; identificadores de pessoas ou objectos na planta; visualização dos alarmes na planta; configuração do filtro dos alarmes.

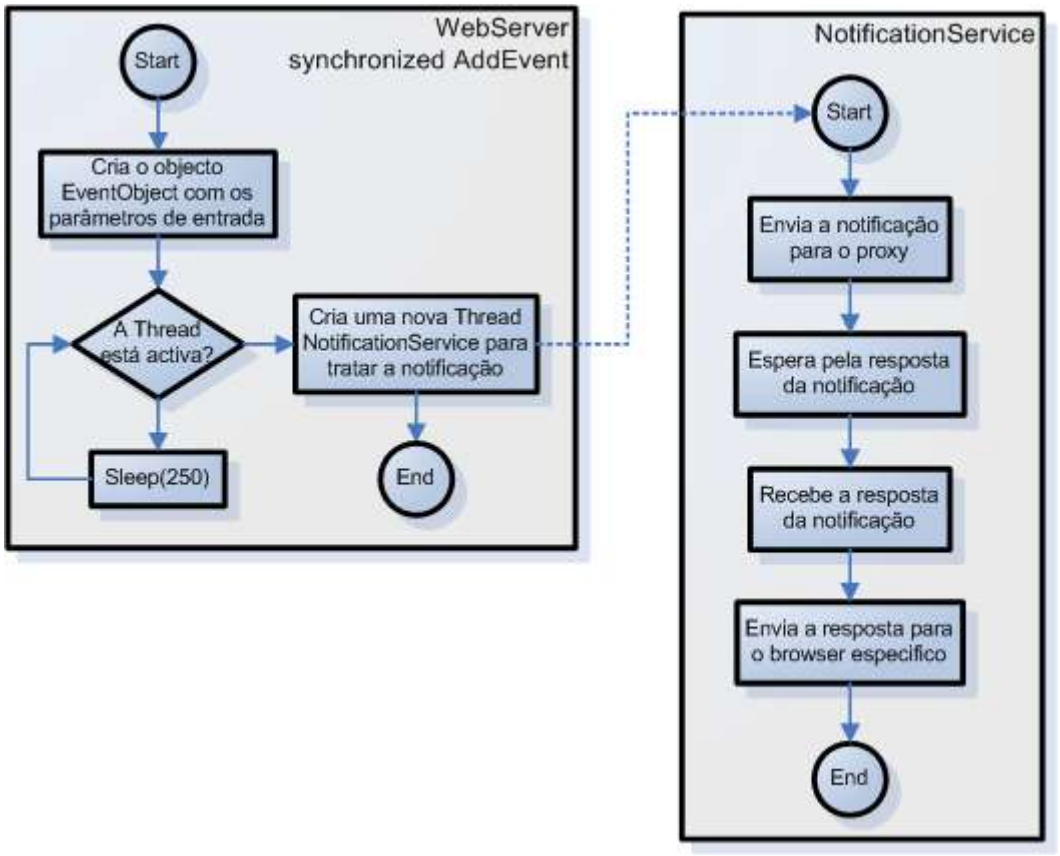



Figura 4-21 – Algoritmo do método "AddEvent".

## 4.5. Conclusões

Os equipamentos escolhidos que preenchiam todos os requisitos foram leitores e *tags* RFID de um parceiro integrador da ISA™, onde se destaca os dois modos de funcionamento na frequência F1 e F2.

Os sistemas de integração existentes no sistema são a base de dados e o servidor *Web*. A base dados proposta é a PostgreSQL® e o servidor *Web* proposto é o Apache Tomcat®.

A implementação do *middleware* consiste em duas partes distintas, o "Adaptor" e o "Proxy". A primeira parte corresponde à comunicação com os equipamentos físicos, a segunda parte pertence a uma camada superior, que consiste no cálculo de todas as funcionalidades do sistema.

A integração de sistemas consiste na comunicação entre o *middleware* e a base de dados, onde é utilizado o JDBC e na comunicação entre o servidor *Web* e as páginas *Web*, onde é implementado o Reverse Ajax (DWR).

O próximo capítulo são apresentados os testes realizados da implementação do projecto, também é apresentado uma análise dos resultados obtidos.

# 5. Testes de campo e análise de resultados

Concluídas as fases de estudos, das tecnologias envolvidas na solução, e da implementação da especificação proposta, deu-se início às realizações dos testes práticos. Os testes surgem com o principal objectivo de analisar, de forma prática e objectiva, o comportamento do sistema RFID e a validação das funcionalidades do sistema.

## 5.1. Metodologia dos testes

A metodologia dos testes foi projectada através de [37], com o objectivo de avaliar as funcionalidades especificadas, analisar as influências dos parâmetros na localização como também na detecção de alarmes e dos recursos disponíveis para a realização dos testes. A execução dos testes foi planeada em três fases, a preparação dos testes, a realização dos testes e o registo dos testes. A Figura 5-1 são apresentadas as fases descritas.

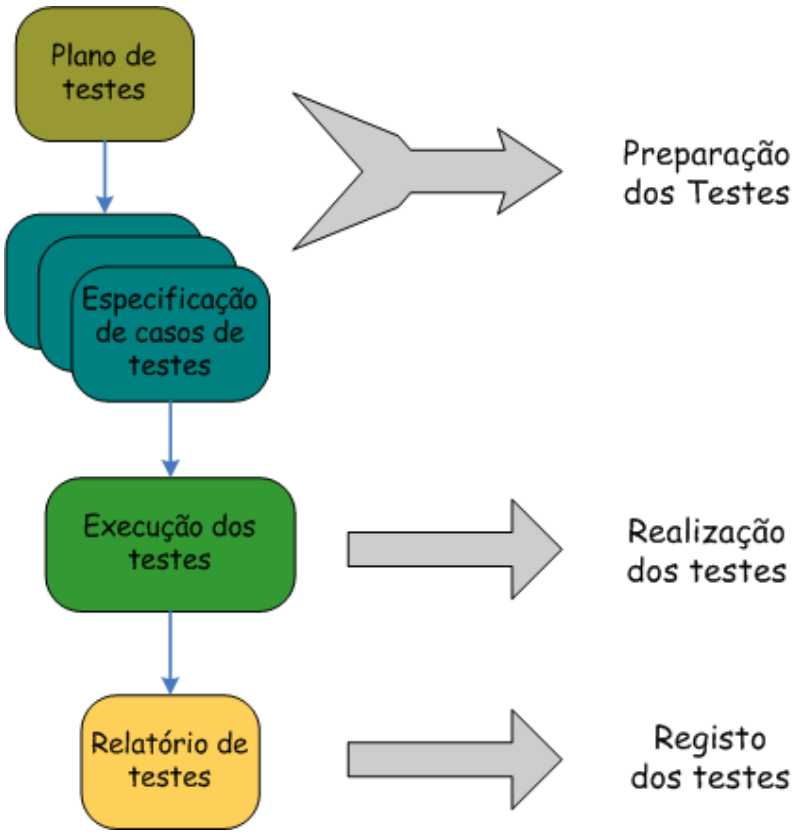



Figura 5-1 – Fases da execução dos testes.

A preparação dos testes são divida em duas partes, o plano de testes e a especificação de casos de testes. O plano de testes indica o planeamento da execução dos testes, incluindo os recursos dos testes, as funcionalidades a serem testadas, as tarefas a serem realizadas. A especificação de casos de testes consiste nos casos de testes, incluindo os parâmetros de entrada, resultados esperados e procedimentos para a execução dos testes. A realização dos testes corresponde a execução dos testes de campo.

O registo dos testes consiste no relatório de testes, onde apresenta os resultados obtidos dos testes e uma breve análise e conclusões referentes aos testes efectuados.

## 5.2. Descrição do equipamento utilizado

Os equipamentos utilizados na realização dos testes foram os equipamentos do parceiro integrador da ISA™ (leitores, antenas e *tags* - secção 4.1.1), um servidor com *middleware* proposto, base de dados PostgreSQL® e o Apache Tomcat®. O servidor é designado por "Servidor TraceMe". O diagrama dos equipamentos é igual ao diagrama da arquitectura física do sistema na Figura 3-3.

Os leitores utilizados foram o LTR-001 com uma antena de 125kHz incorporada e o LTR-001 com duas antenas exteriores de 125kHz, exemplificado na Figura 4-1. As *tags* utilizados foram os IDA-003 e IDA-004, representados na Figura 4-3 e Figura 4-4 respectivamente.

## 5.3. Testes realizados

Para cada cenário de teste foram realizados diferentes tipos de testes, como teste de cobertura de um leitor ou vários leitores, validação das funcionalidades da aplicação, do algoritmo de localização e visualização das *tags* em tempo real. Além destes cenários foi efectuado um teste ao funcionamento das antenas exteriores com o algoritmo em modo anel duplo (explicado no anexo A.3).

### 5.3.1. Funcionamento em modo duplo anel

Este modo de funcionamento consiste em determinar o sentido de passagem de uma *tag* pelas antenas externas. A *tag* só transmite o sinal quando detectar uma passagem, caso este algoritmo não estivesse activo a *tag* responde a todos os pedidos de 125 kHz feitos pelo leitor. Através deste algoritmo o leitor não envia mensagens em abundância, permitindo assim menos processamento de informação na parte do servidor TraceMe. Para activar este algoritmo são necessários configurar o modo do leitor para suportar este funcionamento e também o tempo de intervalo da transmissão de tramas por 125 kHz.

Na Tabela 5-1 é apresentado os resultados obtidos, por cada célula corresponde a passagem dos dois sentidos, ou seja, a entrada e saída de uma porta. Se o leitor enviar uma detecção de entrada é

considerado uma passagem "*in"*, se enviar uma detecção de saída é considerada uma passagem "*out"* e se não enviar nenhuma informação é considerada como "–". A passagem de cada sentido está dividido por "\", o primeiro da célula corresponde uma passagem de saída e o segundo corresponde a uma entrada.

Tabela 5-1 – Resultados obtidos no teste do funcionamento em modo duplo anel.

| **Posicionamento da *tag*** | **Modo de passagem da *tag* nas antenas** | | | | |
|---|---|---|---|---|---|
| | **Normal** | **Correr** | **Parar 5s no meio das antenas e continuar** | **5s no meio das antenas e voltar para trás** | **Junto da porta, sem sair/entrar na sala** |
| Bolso da camisa - posição vertical | *out* / *in* | *out* / *in* | *out* / *in* | -- / *in-out* | -- / -- |
| Bolso da camisa - posição horizontal | *out* / *in* | *out* / *in* | *out* / *in* | -- / *in-out* | -- / -- |
| Presa no bolso das calças vertical | *out* / *in* | *out* / *in* | *out* / *in* | -- / *in-out* | -- / -- |
| No bolso direito das calças | *out* / *in* | *out* / *in* | *out* / *in* | -- / *in-out* | -- / -- |
| No bolso esquerdo das calças | *out* / *in* | *out* / *in* | *out* / *in* | -- / *in-out* | -- / -- |
| Numa mochila vazia | *out* / *in* | *out* / *in* | *out* / *in* | -- / *in-out* | -- / -- |
| Na mala com um portátil | -- / -- | -- / -- | -- / -- | -- / -- | -- / -- |

Através da Tabela 5-1 verifica-se que a *tag* quando está numa mala com um portátil não é detectada nenhuma passagem e também quando o modo de passagem é 5s no meio das antenas e voltar para trás na passagem de entrada comunica duas vezes (*in e out*) em vez de não comunicar, apesar de comunicar o resultado final está correcto, ou seja, no início do teste está fora da sala e depois da passagem fica na mesma fora da sala.

### 5.3.2. Testes em campo aberto

Este caso de estudo consiste verificar o comportamento dos equipamentos RFID no exterior (espaço vazio). Neste conjunto de testes foram utilizados todos os leitores disponíveis. O primeiro cenário foi composto pelos leitores e uma *tag* colocada a cerca de 120 metros no exterior durante 3 horas. As conclusões obtidas neste teste foram que o RSSI varia bastante de receptor para receptor, aproximadamente 40%, a taxa de perdas foi inferior em todos os leitores a 20%.

De seguida reduziu-se a distância da *tag* para 60 metros e com esta alteração a taxa de perdas era inferior a 10%.

Também foi realizado um teste com uma *tag* a deslocar-se desde os 100 metros até junto ao receptor e verificou-se que o valor de RSSI por vezes desce ou mantém-se quando a *tag* é aproximada do leitor, apesar que em termos gerais o RSSI sobe á medida que a *tag* se aproxima do leitor.

### 5.3.3. Monitorização de uma sala

Neste caso de estudo consiste na validação do funcionamento do teste de cobertura e análise do funcionamento dos equipamentos RFID utilizados nos testes práticos. O cenário de testes (Cenário 1) está representado na Figura 5-2, a escala da figura é 1/80. Este cenário corresponde a uma sala do IPN [1] (Instituto Pedro Nunes), o único objecto representado na figura são as secretárias, mas existe em quase todas secretárias computadores, portáteis e cadeiras. Também é representado a localização dos leitores (300, 302, 303) que se encontram em cima das mesas e da *tag* (representado na figura em amarelo) que está posicionada por baixo de uma mesa. Os leitores são do mesmo modelo com uma antena de 125 kHz incorporada.

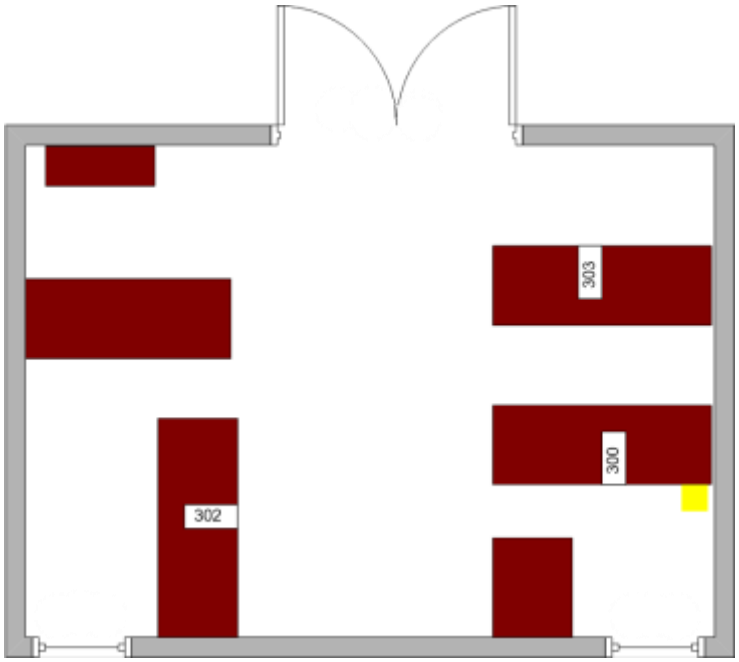



Figura 5-2 – Cenário 1 dos testes realizados.

A *tag* não se encontra no alcance dos 125 kHz dos leitores e o *beacon* foi configurado com o intervalo de 5s. O resultado do teste está representado na Tabela 5-2, o campo "Data" representa ao tempo quando foi escrito nos *logs* após ter sido processado pelo sistema e o "Recebido" na "Descrição" corresponde ao tempo que o evento foi recebido pelo sistema.

[1] Sítio do IPN: *https://www.ipn.pt/si/initapplication.do* (Ultima vez visitado em Dezembro 2008)

Tabela 5-2 – Resultados obtidos no primeiro teste.

| Data | Informação | Descrição |
|---|---|---|
| 2008-06-05 15:18:10.663 | Inicio do Proxy | Os três leitores ligaram-se ao Adaptor |
| 2008-06-05 15:18:13.427 | Evento | idLeitor = 302; Recebido: 2008-06-05 15:18:12.423 |
| 2008-06-05 15:18:13.428 | Evento | idLeitor = 300; Recebido: 2008-06-05 15:18:12.419 |
| | | |
| 2008-06-05 15:18:18.427 | Evento | idLeitor = 302; Recebido: 2008-06-05 15:18:18.183 |
| | | |
| 2008-06-05 15:18:24.427 | Evento | idLeitor = 300; Recebido: 2008-06-05 15:18:23.951 |
| 2008-06-05 15:18:24.428 | Evento | idLeitor = 302; Recebido: 2008-06-05 15:18:23.954 |
| | | |
| 2008-06-05 15:18:30.428 | Evento | idLeitor = 302; Recebido: 2008-06-05 15:18:29.696 |
| | | |
| 2008-06-05 15:18:36.429 | Evento | idLeitor = 300; Recebido: 2008-06-05 15:18:35.475 |
| 2008-06-05 15:18:36.429 | Evento | idLeitor = 302; Recebido: 2008-06-05 15:18:35.476 |
| | | |
| 2008-06-05 15:18:41.427 | Evento | idLeitor = 302; Recebido: 2008-06-05 15:18:41.227 |
| 2008-06-05 15:18:41.428 | Evento | idLeitor = 300; Recebido: 2008-06-05 15:18:41.225 |
| | | |
| 2008-06-05 15:18:47.428 | Evento | idLeitor = 302; Recebido: 2008-06-05 15:18:46.978 |
| 2008-06-05 15:18:47.428 | Evento | idLeitor = 300; Recebido: 2008-06-05 15:18:46.978 |
| | | |
| 2008-06-05 15:18:53.427 | Evento | idLeitor = 302; Recebido: 2008-06-05 15:18:52.728 |
| | | |
| 2008-06-05 15:18:59.427 | Evento | idLeitor = 300; Recebido: 2008-06-05 15:18:58.492 |
| 2008-06-05 15:18:59.428 | Evento | idLeitor = 302; Recebido: 2008-06-05 15:18:58.489 |
| | | |
| 2008-06-05 15:19:04.43 | Evento | idLeitor = 302; Recebido: 2008-06-05 15:19:04.23 |
| 2008-06-05 15:19:04.43 | Evento | idLeitor = 300; Recebido: 2008-06-05 15:19:04.233 |
| | | |
| 2008-06-05 15:19:10.429 | Evento | idLeitor = 302; Recebido: 2008-06-05 15:19:09.953 |
| 2008-06-05 15:19:10.43 | Evento | idLeitor = 300; Recebido: 2008-06-05 15:19:09.961 |

Através dos resultados verifica-se que o leitor 303 não enviou nenhum evento, o leitor 300 não enviou três eventos, o leitor 302 enviou todos os eventos. Através deste teste é possível validar o funcionamento correcto do serviço teste de cobertura.

### 5.3.4. Testes dos algoritmos de localização

Este caso de estudo pretende analisar os algoritmos de localização propostos (secção 3.6). O primeiro algoritmo é mais o simples, a localização é feita através da detecção nos 125 kHz e não é possível detectar quando a *tag* deixa de enviar o *beacon*.

De seguida foi realizado testes para analisar o algoritmo de localização através do valor de RSSI com o método de triangulação dos eventos recebidos. Através do RSSI recebido pelos eventos (sinal emitido pelas *tags* periodicamente) pretende-se avaliar a possibilidade de limitar o valor para definir a área de cobertura dos leitores, outra avaliação é verificar se existe relação entre os valores recebidos nos testes com a perda de eventos. O método de triangulação consiste na intersecção das áreas de cobertura dos leitores para definir zonas distintas, portanto, os leitores foram posicionados em salas diferentes para analisar se este método podia ser validado.

Todos os testes realizados foram de 60 segundos e a *tag* transmitia o *beacon* em 2,5 segundos e sem mobilização, teoricamente os eventos recebidos sem perdas poderiam ir de 23 a 26. Os leitores são todos do modelo com uma antena incorporada em cada um. Foram feitos testes de manha, entre as 11 horas e as 12 horas e 30 minutos, no inicio da tarde, entre as 15 horas e as 16 horas e 30 minutos e por fim, no final da tarde, entre as 17 horas e 30 minutos e as 19 horas.

Na Figura 5-3 é exemplificado o cenário de testes (Cenário 2), a sua escala é 1/25, o local correspondente a uma parte da antiga infra-estrutura das instalações da ISA™, indicando todos os pontos de teste (ao todo são 18 pontos) e a localização dos leitores (300 na Sala 2, 302 na Sala 3 e 303 na Sala 1). Os leitores são do mesmo modelo com uma antena de 125 kHz incorporada, cada um representa a sala que está associada. A localização dos pontos de teste corresponde aos cantos e ao ponto central de cada sala onde se encontra os leitores. A Figura 5-3 tenta também ilustrar o ambiente das salas, as secretárias (em todas as salas onde foram efectuados testes de cobertura) e um armário de metal (na Sala 2).

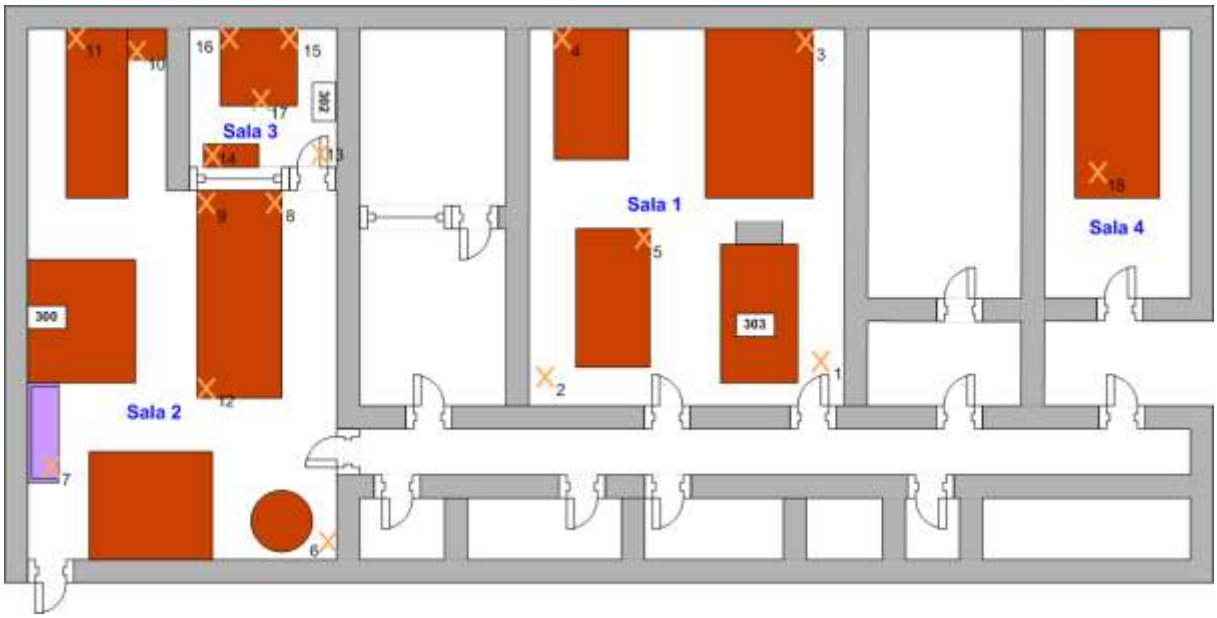



Figura 5-3 – Cenário 2 dos testes práticos realizados.

Os resultados obtidos deste conjunto de testes encontra-se na Tabela A-3 no Anexo A.10, onde o valor RSSI corresponde ao mínimo e ao máximo recebido (na tabela os valores estam separado pelo "\") e está convertido em dBm através da conversão explicada em anexo A.11.

Através dos resultados obtidos pode verificar-se que o mecanismo de triangulação, baseado pelo RSSI, não é válido porque é bastante incoerente e inconstante. Recorrendo às análises, os pontos que se consegue identificar uma zona através o método limitar RSSI por um valor pré-definido são os pontos 1 (limiar de -85dBm), 2 (limiar de -85dBm), 4 (limiar de -85dBm), 5 (limiar de -88dBm) e 18 (limiar de -87dBm). Através do método da diferença de RSSI foram só os seguintes pontos 13 (diferença de

10 dBm do leitor 302 em relação aos outros) e 15 (diferença de 10 dBm do leitor 302 em relação aos outros). Os restantes pontos não foi possível determinar uma zona em concreto em tempo real, porque na maioria dos casos existia muitas perdas o que origina no mecanismo de localização um mau funcionamento por ter dados incoerentes.

Por exemplo no ponto 17 no teste de manhã num certo instante o mecanismo de localização indicava que a *tag* estava na zona de intersecção entre os três leitores, noutro instante indicava que estava na zona de intersecção do leitor 300 e 302, e noutro instante de tempo indicava que estava na zona do leitor 302, portanto existe muitas oscilações de zonas e a *tag* nem se quer movimentou.

Após análise dos dados obtidos foi decidido reestruturar o mecanismo de localização em tempo real, de forma a este ser mais fiável e constante. A mudança consiste em localizar a *tag* através dos 125kHz, porque o alcance é menor, por isso o erro diminui e aumenta a precisão da localização, e utilizar os dados de 868MHz para actualizar o evento e indicar que a *tag* ainda está dentro do edifício, ou seja, caso a *tag* não comunica com o sistema durante um tempo determinado é gerado um alarme de detecção de ausência (mecanismo intermédio, secção 3.6).

### 5.3.5. Localização em tempo real num edifício

Este conjunto de testes pretende exemplificar a visualização da localização em tempo real e o controlo de acessos. O cenário de teste (Cenário 3) está representado na Figura 5-5, a escala da planta é 1/500, o percurso efectuado foi Área 1, Área 2 e por fim Área 3. Este cenário corresponde a um piso da infra-estrutura do IPN. Este é composto na mesma por 3 leitores, no entanto 2 leitores contém duas antenas externas de 125kHz em vez da antena incorporada. As antenas exteriores ficam posicionadas nas ombreiras das portas para determinar o sentido, exemplificado na Figura 5-4.

Figura 5-4 – Posicionamento das antenas externas.

O cenário está representado na Figura 5-5, onde analisamos os leitores 103, 6084 e 104. Os leitores 6084 e 104 são do mesmo modelo com duas antenas exteriores de 125 kHz (*in* e *out*), o leitor 103 corresponde ao modelo com uma antena incorporada. Com esta configuração é possível distinguir 3 áreas, "Área 1" (designado também por "Sala da ISA"), "Área 2" (designado também por "Sala 1") e "Área 3" (designado também por "Sala 2"). A "Área 1" é limitada pelo leitor 103, a "Área 2" é demarcada pelo leitor 6084 e a "Área 3" é definida pelo 104.


Figura 5-5 – Cenário 3 dos testes práticos realizados.

As *tags* RFID disponíveis para estes testes foram 3, as *tags* com o código 7415, 7358 e 3855. Para cada *tag* é associado uma unidade, o 7415 e o 7358 corresponde a duas pessoas e o 3855 corresponde a um objecto fixo (objecto que tem uma localização por omissão). Também são associados a cada unidade as áreas de acesso permitido. As configurações destes parâmetros encontram-se na Tabela 5-3.

Tabela 5-3 – Parâmetros dos testes de visualização da localização.

| Código da *tag* | Nome da unidade | Permissão de acesso |
|---|---|---|
| 7415 | Maciel | Área 2 e Área 3 |
| 7358 | Teresa | Área 1 e Área 2 |
| 3855 | Computador do Nuno Costa | Área 1 |

As três *tags* foram colocadas sensivelmente ao mesmo tempo na Área 1, onde deverá aparecer um alarme de permissão ao Maciel e o respectiva imagem a representar uma pessoa a vermelho, uma imagem a representar uma pessoa a verde corresponde à Teresa e uma imagem a indicar o objecto fixo a verde. Na Tabela 5-4 são representados os resultados obtidos na camada "Proxy" após a filtragem dos eventos, mecanismo de localização e detecção de alarmes.

Tabela 5-4 – Informação obtida quando as *tags* foram detectadas na Área 1.

| Data | Informação | Descrição |
|---|---|---|
| 2008-09-17 15:50:31.136 | Evento | idLeitor = 103; idCartão = 7358; Zona = Área 1 |
| 2008-09-17 15:50:33.839 | Evento | idLeitor = 103; idCartão = 7415; Zona = Área 1 |
| 2008-09-17 15:50:33.839 | Alarme | Alarme = Permissão; idCartão = 7415; Zona = Área 1 |
| 2008-09-17 15:50:34.745 | Evento | idLeitor = 103; idCartão = 3855; Zona = Área 1 |

Através da tabela pode verificar-se a detecção de três passagens pelo leitor 103 e uma detecção de um alarme de acesso. A visualização desta informação está apresentada na Figura 5-6, o alarme detectado encontra-se na tabela dos alarmes e as imagens correspondentes as passagens aparece na Área 1.

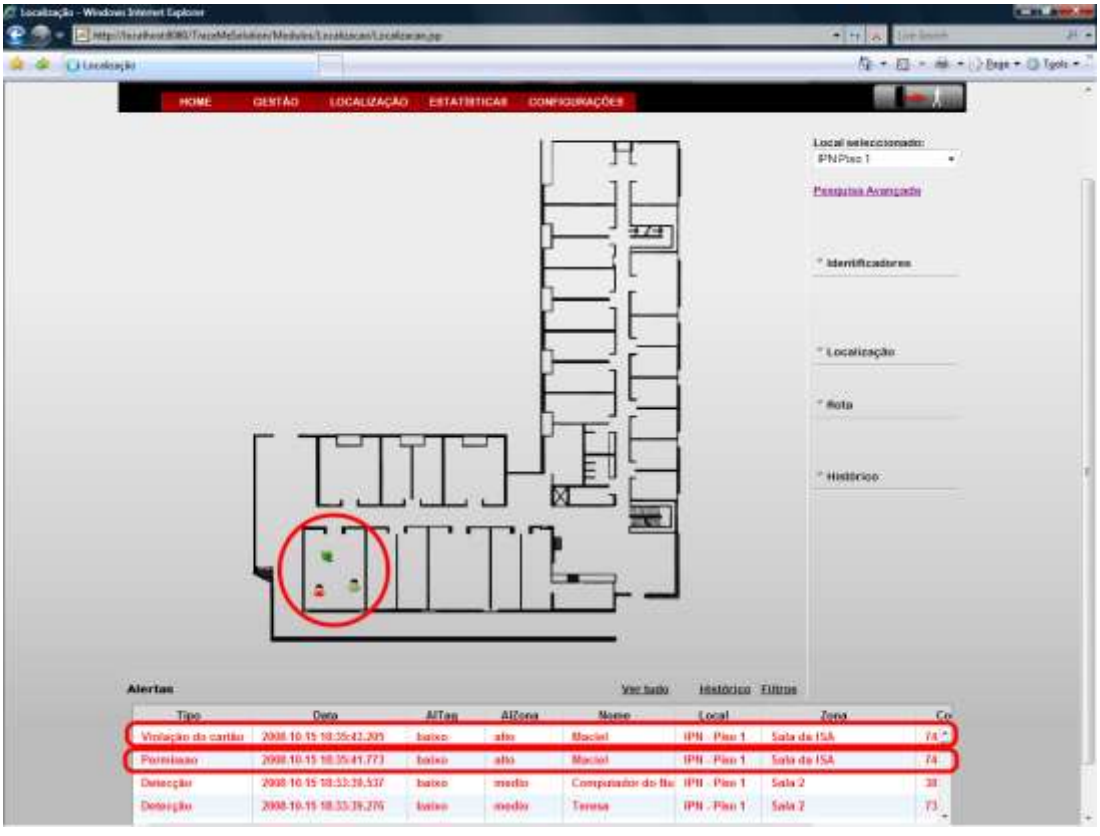



Figura 5-6 – Detecções das *tags* na Área 1 e da violação do estado da *tag*.

De seguida as *tags* foram transportadas para a Área 2, onde deverá detectar um alarme de permissão ao Computador do Nuno Costa e o respectiva imagem a representar um objecto, uma imagem a representar um grupo de pessoa a verde corresponde à Teresa e ao Maciel. Na Tabela 5-5 são representados os resultados obtidos.

Tabela 5-5 – Informação obtida quando as *tags* foram detectadas na Área 2.

| Data | Informação | Descrição |
|---|---|---|
| 2008-09-17 15:55:07.011 | Evento | idLeitor = 6084; idCartão = 7415; Zona = Área 2 |
| 2008-09-17 15:55:07.933 | Evento | idLeitor = 6084; idCartão = 3855; Zona = Área 2 |
| 2008-09-17 15:55:07.933 | Alarme | Alarme = Permissão; idCartão = 3855; Zona = Área 2 |
| 2008-09-17 15:55:08.198 | Evento | idLeitor = 6084; idCartão = 7358; Zona = Área 2 |

Através da tabela pode verificar-se a detecção de três passagens pelo leitor 6084 e uma detecção de um alarme de acesso. A visualização desta informação está apresentada na Figura 5-7, o alarme detectado encontra-se na tabela dos alarmes e a imagem correspondente as passagens aparece na Área 2.

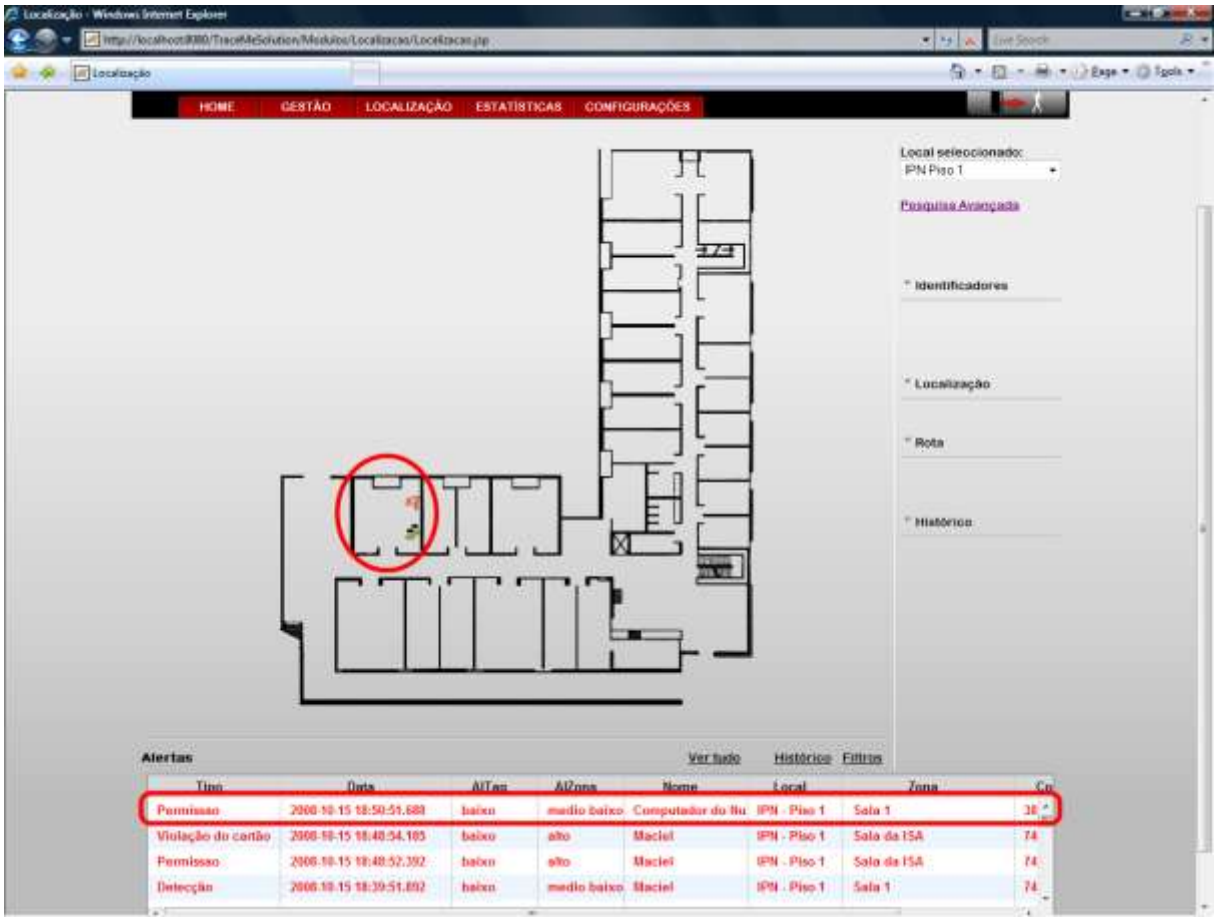



Figura 5-7 – Visualização das *tags* quando foram detectadas na Área 2.

Por fim as *tags* foram transportadas para a Área 3, onde deverá aparecer dois alarme de permissão, ao Computador do Nuno Costa e o respectiva imagem a representar um objecto e a Teresa com a

respectiva imagem a identificar uma pessoa a vermelho, uma imagem a representar uma pessoa a verde corresponde ao Maciel. Na Tabela 5-6 são apresentados os resultados obtidos.

Tabela 5-6 – Informação obtida quando as *tags* foram detectadas na Área 3.

| Data | Informação | Descrição |
|---|---|---|
| 2008-09-17 15:57:00.198 | Evento | idLeitor = 104; idCartão = 7415; Zona = Área 3 |
| 2008-09-17 15:57:00.339 | Evento | idLeitor = 104; idCartão = 7358; Zona = Área 3 |
| 2008-09-17 15:57:00.339 | Alarme | Alarme = Permissão; idCartão = 7358; Zona = Área 3 |
| 2008-09-17 15:57:00.902 | Evento | idLeitor = 104; idCartão = 3855; Zona = Área 3 |
| 2008-09-17 15:57:00.902 | Alarme | Alarme = Permissão; idCartão = 3855; Zona = Área 3 |

Através da tabela pode verificar-se a detecção de três passagens pelo leitor 104 sendo duas são detecções de acesso não permitido. A visualização desta informação está apresentada na Figura 5-8, os alarmes detectados encontram-se na tabela dos alarmes e a imagem correspondente as passagens aparecem na Área 3.

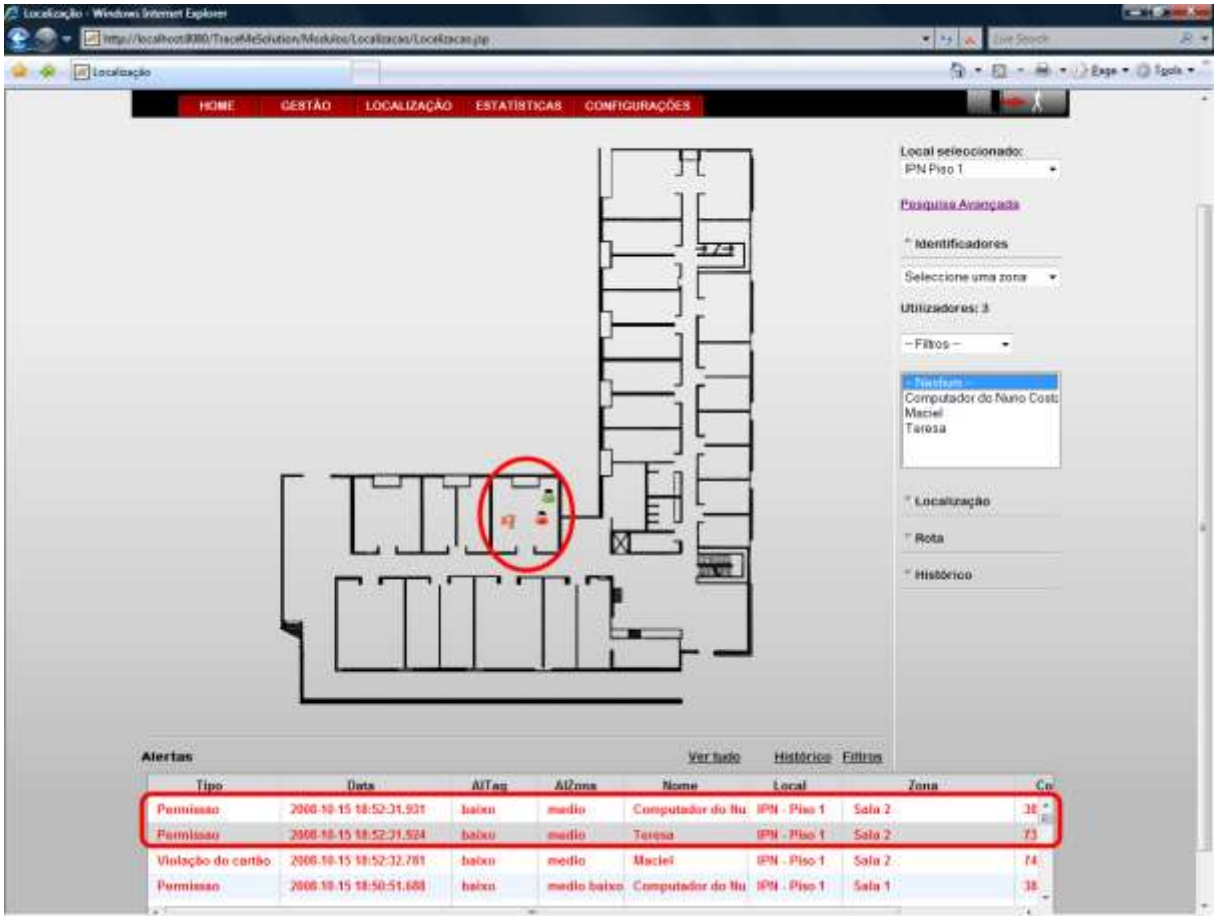



Figura 5-8 – Visualização das *tags* quando foram detectadas na Área 3.

Outra forma de visualizar a localização pode ser por zonas, onde se pode escolher uma determinada zona de um piso e analisar somente essa área. Na Figura 5-9 é indicado onde é que se pode escolher a zona de um determinado piso.

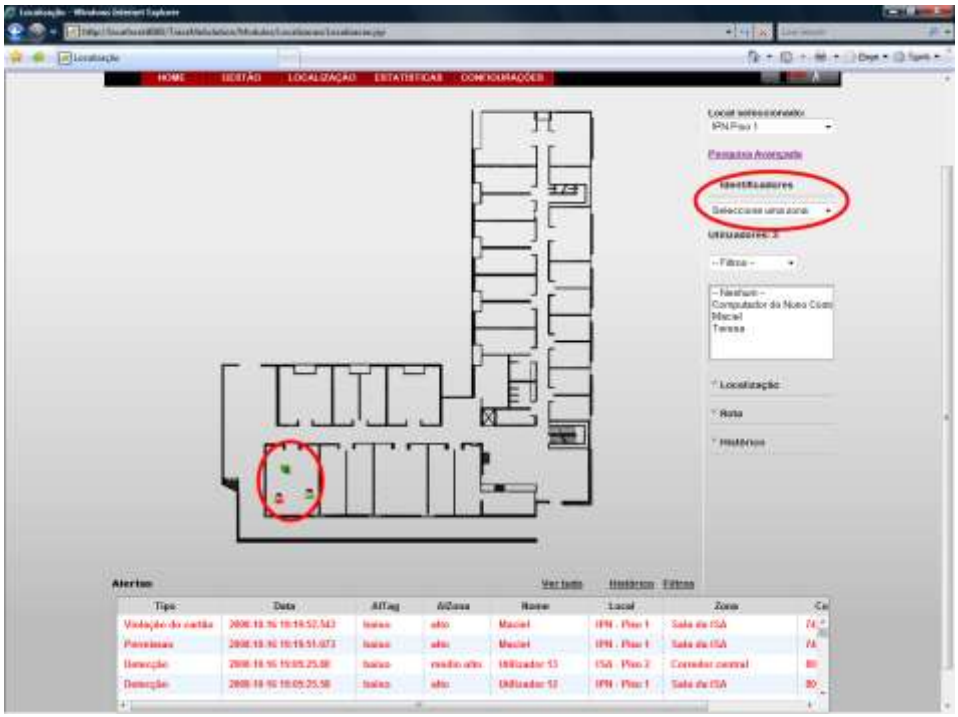


Figura 5-9 – Seleccionar uma determinada zona de um piso.

Após ter seleccionado a zona, a planta do piso passa para o canto direito inferior indicando só os alarmes gerados nas zonas definidas. Na zona central da página mostra a zona seleccionada, como é exemplificado na Figura 5-10, onde a zona seleccionada é a "Área 1" e na planta do piso em minúsculo parece a indicação que houve pelo menos um alarme na zona "Área 1".

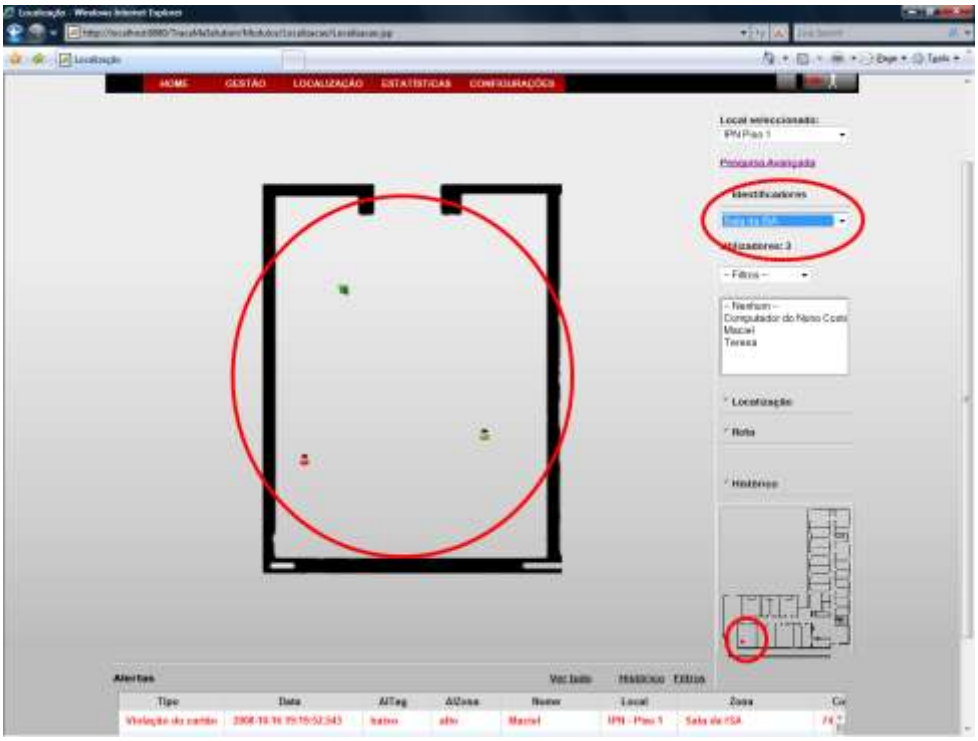


Figura 5-10 – Visualização da "Área 1" e do piso a indicar que existe pelo menos um alarme.

Caso as *tags* sejam transportadas para outras zonas não vai ser possível analisar, porque só é possível visualizar as *tags* que se encontram na zona "Área 1". Mas na planta do piso no canto direito é possível ver se existe alarmes noutras zonas. Na Figura 5-11 as *tags* foram movimentadas para a "Área 2", portanto a "Área 1" está vazia mas na planta do edifício indica que existe pelo menos um alarme na "Área 2".

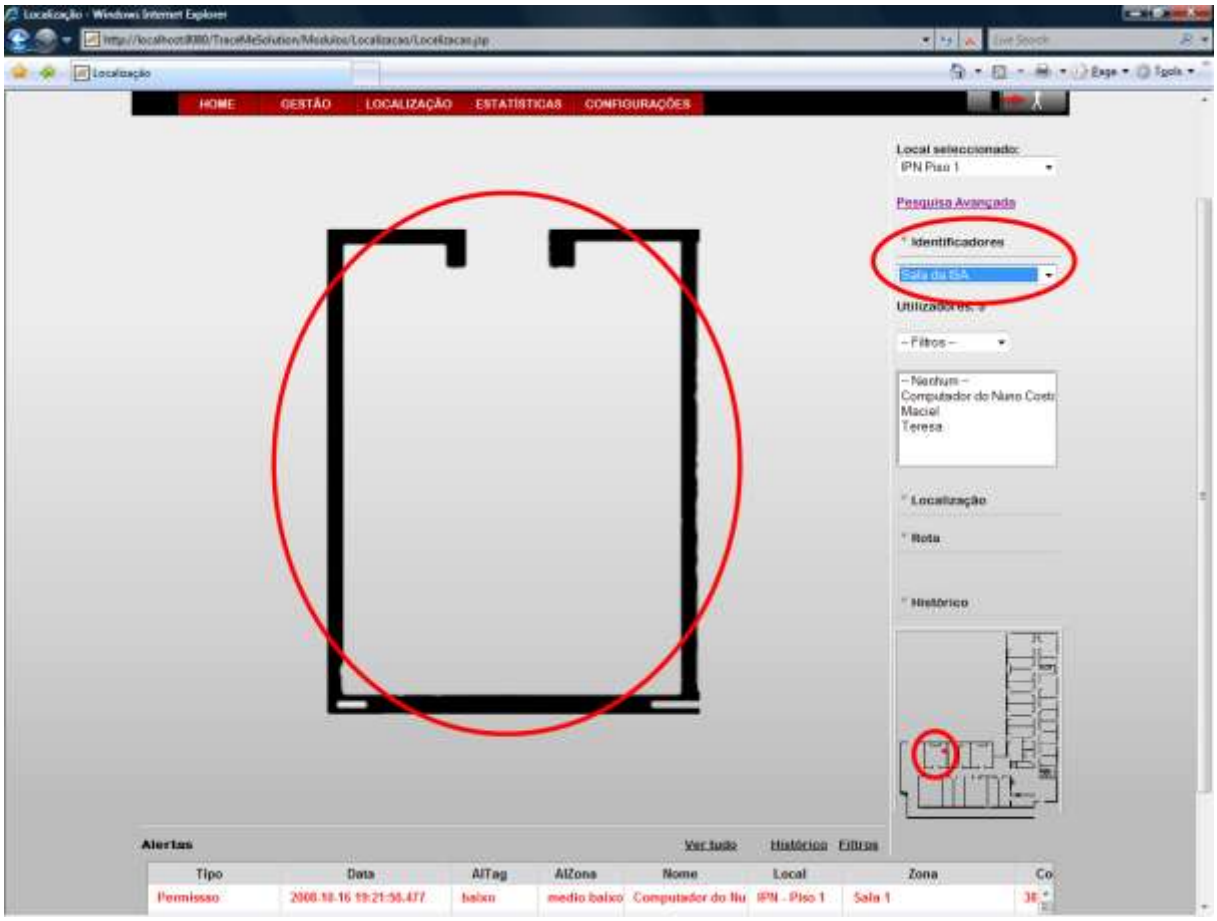



Figura 5-11 – Visualização da "Área 1" sem *tags* e do piso a indicar um alarme na "área 2".

Para visualizar as *tags* será necessário seleccionar a zona "Sala 1" nos identificadores, como é exemplificado na Figura 5-12.

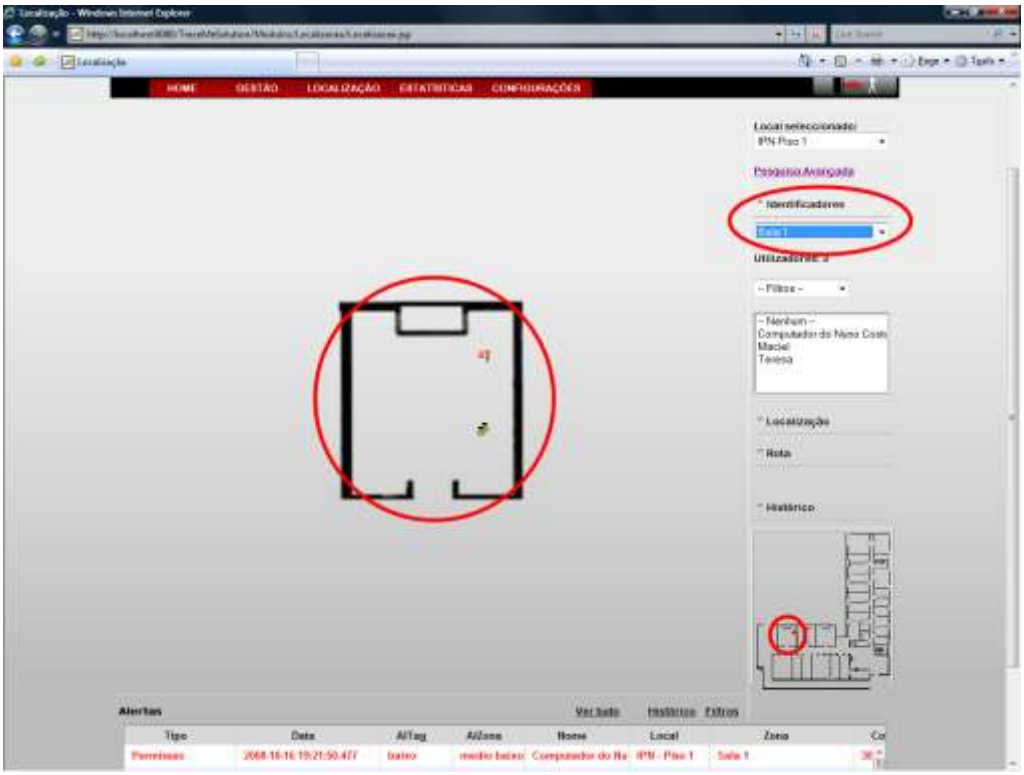



Figura 5-12 – Visualização da "Área 2".

Para voltar a visualizar de novo a planta do edifício só é necessário seleccionar a opção "Todas" nos identificadores, como é exemplificado na Figura 5-13.

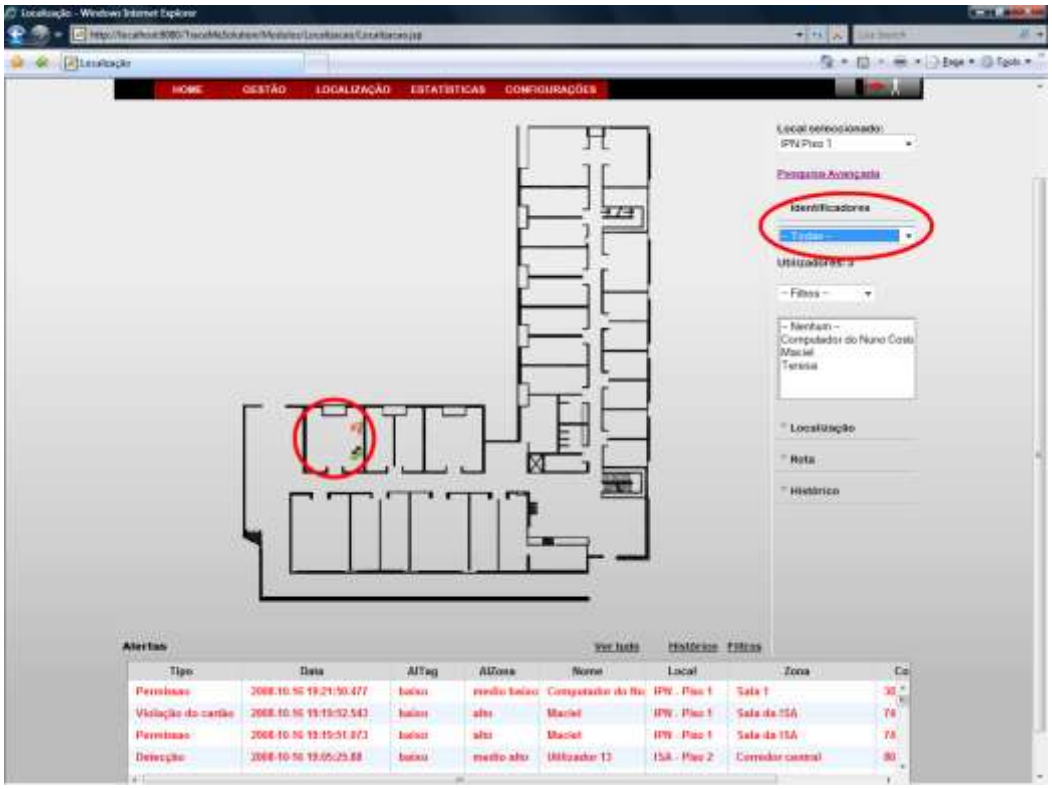



Figura 5-13 – Visualização de todas as zonas de um determinado piso.

### 5.3.6. Aferição das funcionalidades do sistema

Este conjunto de testes corresponde a verificação das funcionalidades do sistema, o cenário (Cenário 3) utilizado está exemplificado na Figura 5-5. As *tags* RFID disponíveis para estes testes foram os mesmos do caso de estudo anterior (secção 5.3.5) e as configurações dos parâmetros encontram-se na Tabela 5-3. As funcionalidades testadas neste conjunto foram as seguintes:

- **Detecção da violação da *tag*** – Esta detecção indica se a *tag* foi adulterada. A *tag* 7415 tem a *flag* de violação activa. Sempre que a *tag* é detectada, o sistema cria um alerta e é visível na tabela dos alertas, como exemplifica na Figura 5-6.
- **Identificadores** – Esta funcionalidade permite indicar o número de utilizadores presentes na planta, permite ainda destacar uma determinada unidade na planta, metendo a imagem da respectiva unidade a piscar. A Figura 5-14 é exemplificado a identificação do Computador Nuno Costa, a Figura 5-15 é ilustrado a identificação do Maciel e a Figura 5-16 é mostrado a identificação da Teresa.

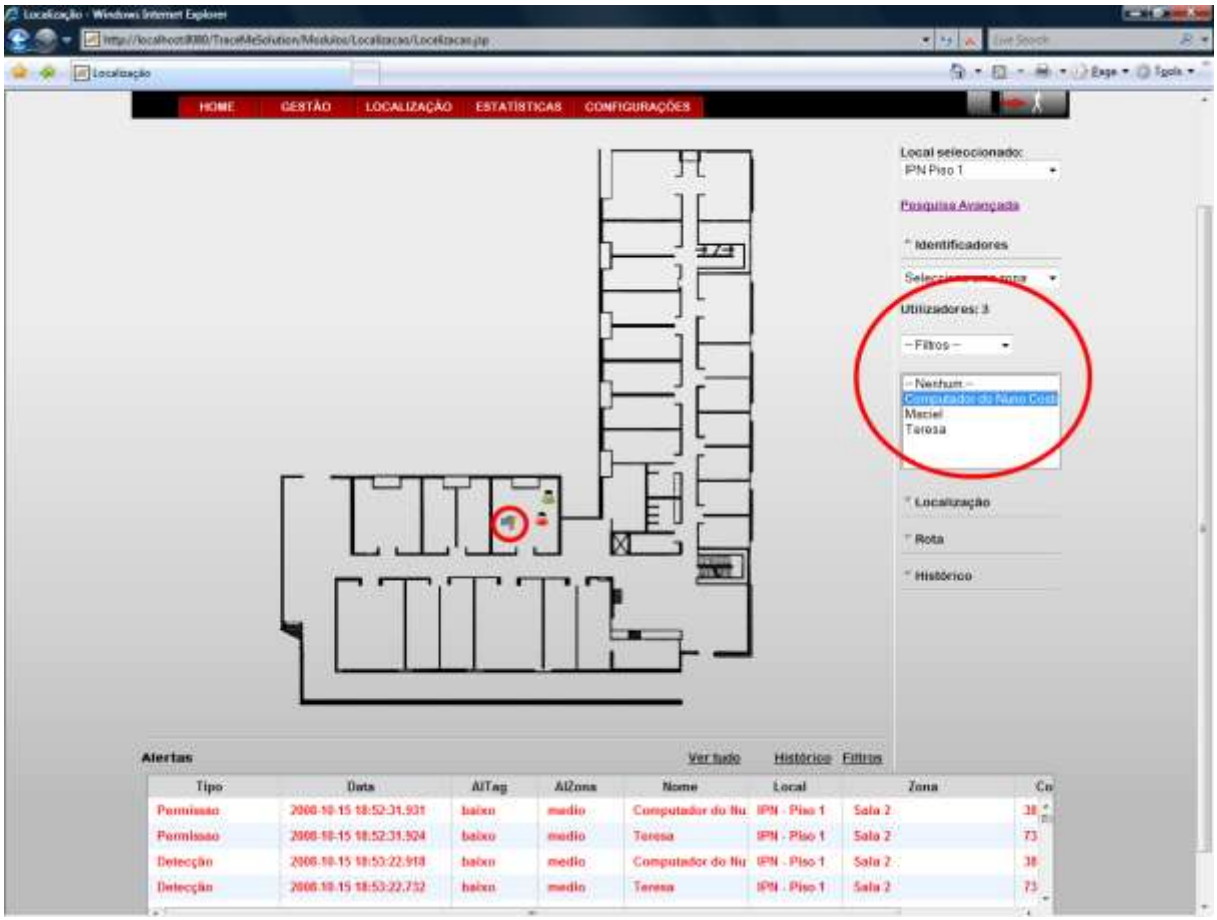



Figura 5-14 – Identificação do Computador Nuno Costa na planta.

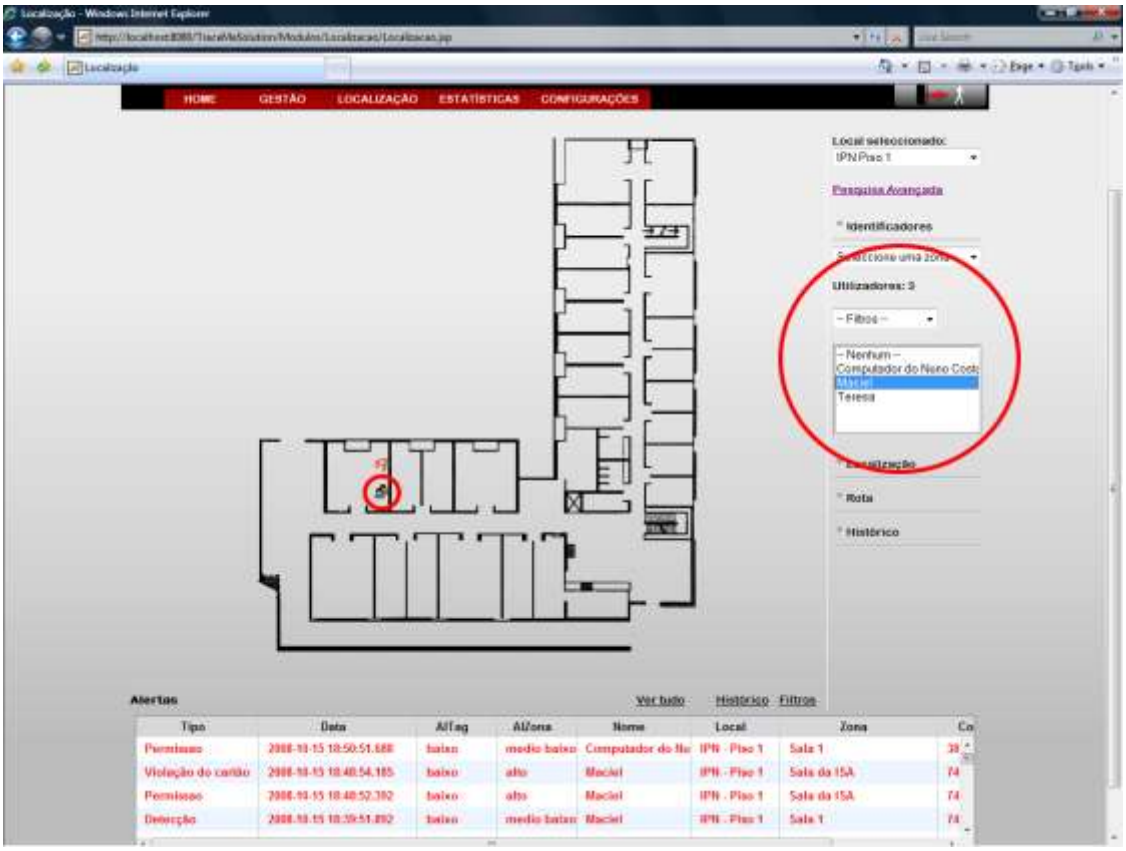



Figura 5-15 – Identificação do Maciel na planta.

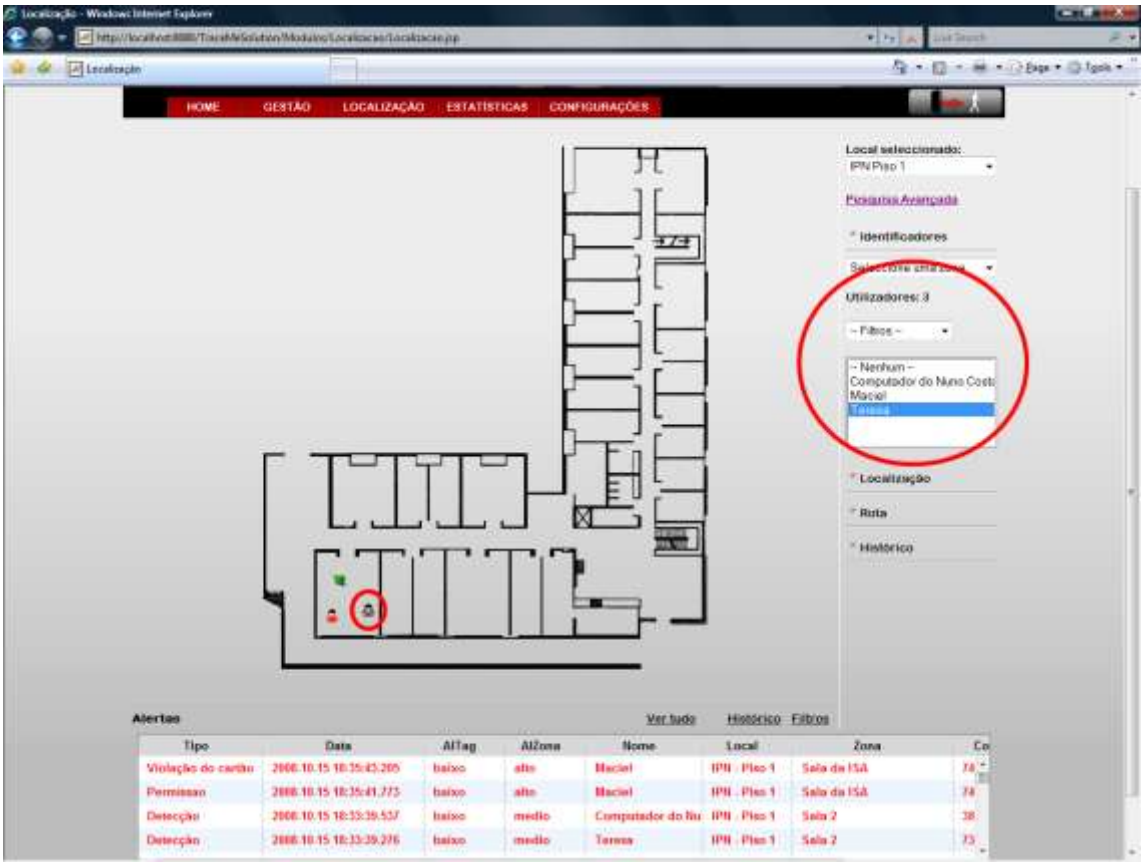



Figura 5-16 – Identificação da Teresa na planta.

- **Localização rápida** – Através desta funcionalidade é possível escolher uma determinada pessoa ou objecto e saber a sua localização. Através do reverse ajax indicamos qual é a pessoa ou objecto que pretendemos localizar ao servidor *Web* e a partir desse momento só acompanha a localização escolhida. Na Figura 5-17 é apresentado o menu da localização rápida que se encontra no separador Localização.

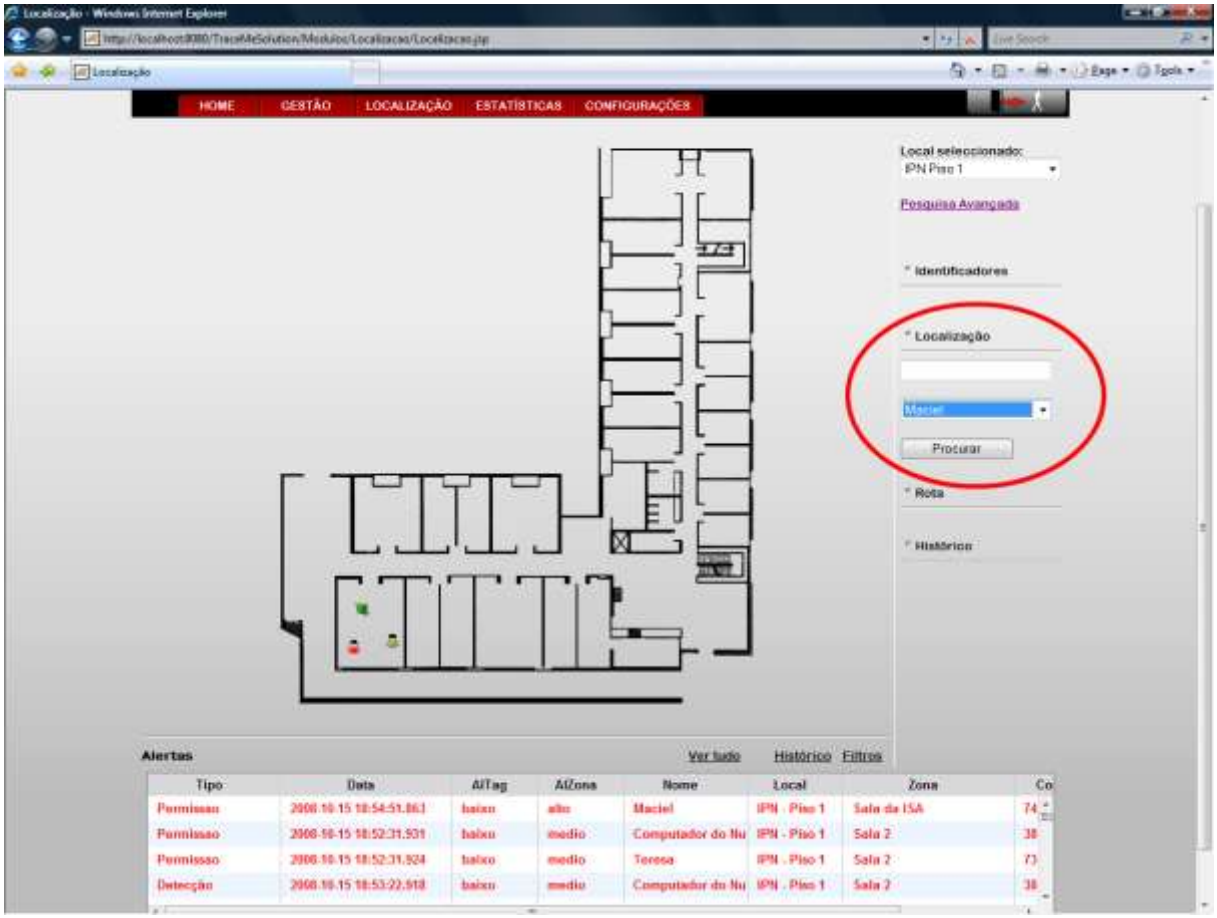



Figura 5-17 – Menu da localização rápida.

A partir deste menu pode-se escrever o nome ou escolher a pessoa ou objecto para procurar a sua localização. Na Figura 5-18 é mostrado o resultado da procura pelo Maciel.

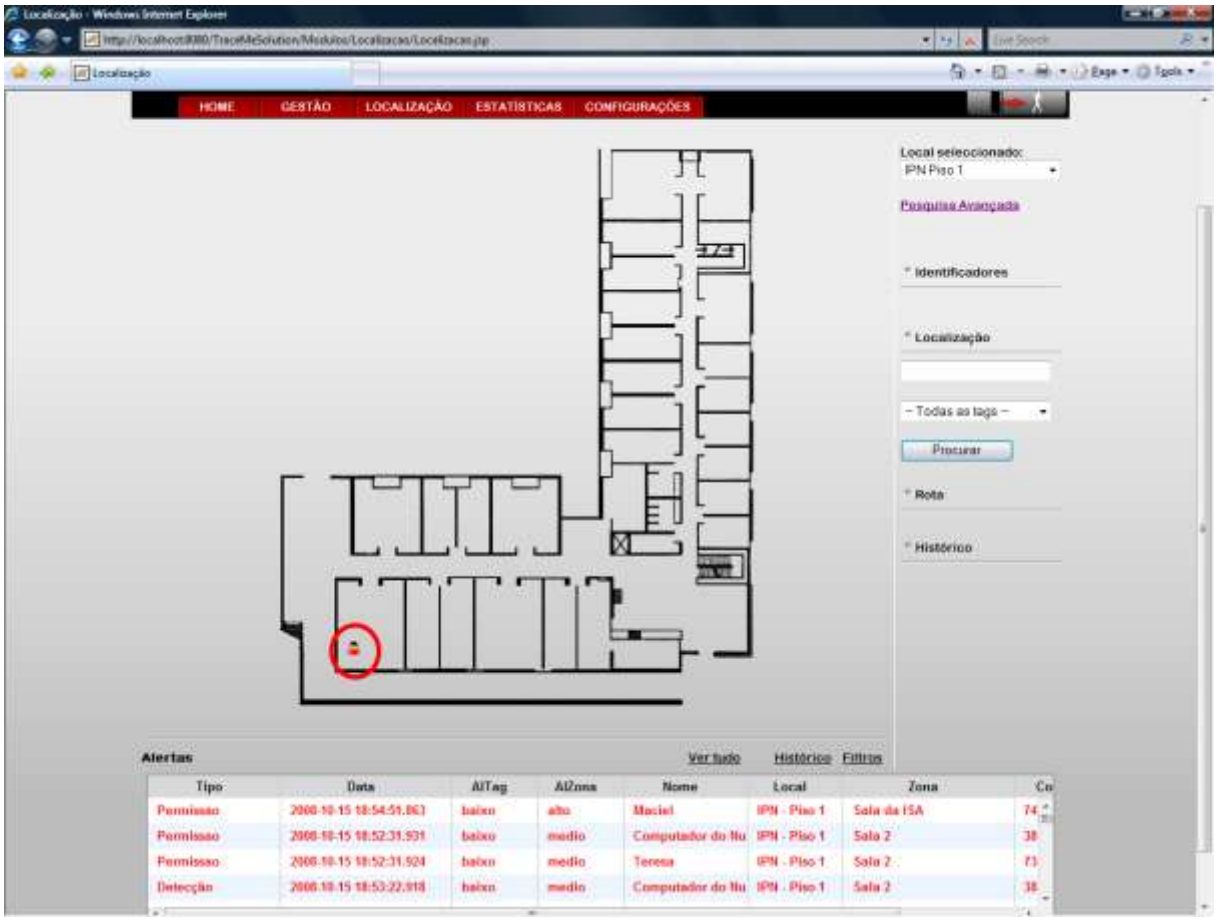



Figura 5-18 – Localização rápida da *tag* pertence ao Maciel.

De seguida as *tags* foram transportadas para a Área 2, onde deverá aparecer um alarme de permissão ao Computador do Nuno Costa e uma imagem a representar uma pessoa a verde corresponde ao Maciel. A Tabela 5-7 é representada a informação recebida.

Tabela 5-7 – Informação obtida quando as *tags* foram para Área 2.

| Data | Informação | Descrição |
|---|---|---|
| 2008-09-17 16:26:06.433 | Evento | idLeitor = 6084; idCartão = 7415; Zona = Área 2 |
| 2008-09-17 16:26:07.058 | Evento | idLeitor = 6084; idCartão = 3855; Zona = Área 2 |
| 2008-09-17 16:26:07.058 | Alarme | Alarme = Permissão; idCartão = 3855; Zona = Área 2 |
| 2008-09-17 16:26:06.886 | Evento | idLeitor = 6084; idCartão = 7358; Zona = Área 2 |

Através da tabela pode verificar-se a detecção de três passagens pelo leitor 6084 e uma detecção de um alarme de acesso. Mas visualização desta informação será só a do alarme e a informação relativa à *tag* do Maciel, exemplificado na Figura 5-19, o alarme detectado encontra-se na tabela dos alarmes e a imagem correspondente a passagem do Maciel está na Área 2.

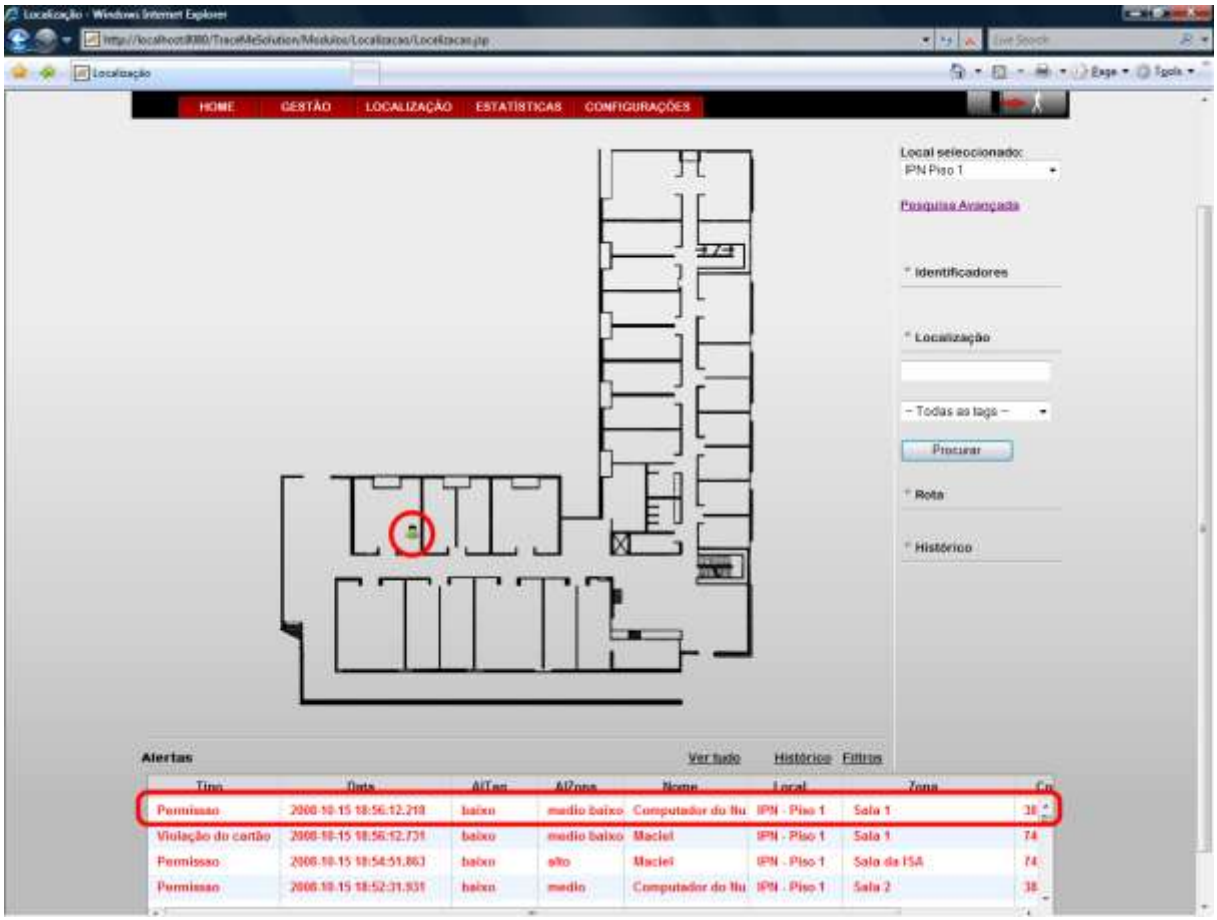



Figura 5-19 – Localização rápida da *tag* pertence ao Maciel na Área 2.

De seguida as *tags* foram transportadas para a Área 3, onde deverá aparecer dois alarme de permissão, ao Computador do Nuno Costa e a Teresa e a imagem a representar o Maciel. Na Tabela 5-8 são apresentados os dados obtidos pelo sistema.

Tabela 5-8 – Informação obtida quando as *tags* foram para a Área 3.

| Data | Informação | Descrição |
|---|---|---|
| 2008-09-17 16:28:23.058 | Evento | idLeitor = 104; idCartão = 7415; Zona = Área 3 |
| 2008-09-17 16:28:23.902 | Evento | idLeitor = 104; idCartão = 3855; Zona = Área 3 |
| 2008-09-17 16:28:23.902 | Alarme | Alarme = Permissão; idCartão = 3855; Zona = Área 3 |
| 2008-09-17 16:28:23.573 | Evento | idLeitor = 104; idCartão = 7358; Zona = Área 3 |
| 2008-09-17 16:28:23.573 | Alarme | Alarme = Permissão; idCartão = 7358; Zona = Área 3 |

Através da tabela pode verificar-se a detecção de três passagens pelo leitor 104 sendo duas são detecções de acesso não permitido. Mas as visualizações destas informações deverão ser os alarmes e a informação relativa à *tag* do Maciel, exemplificado na Figura 5-20, os alarmes detectados encontram-se na tabela dos alarmes e a imagem correspondente a passagem do Maciel está na Área 3.

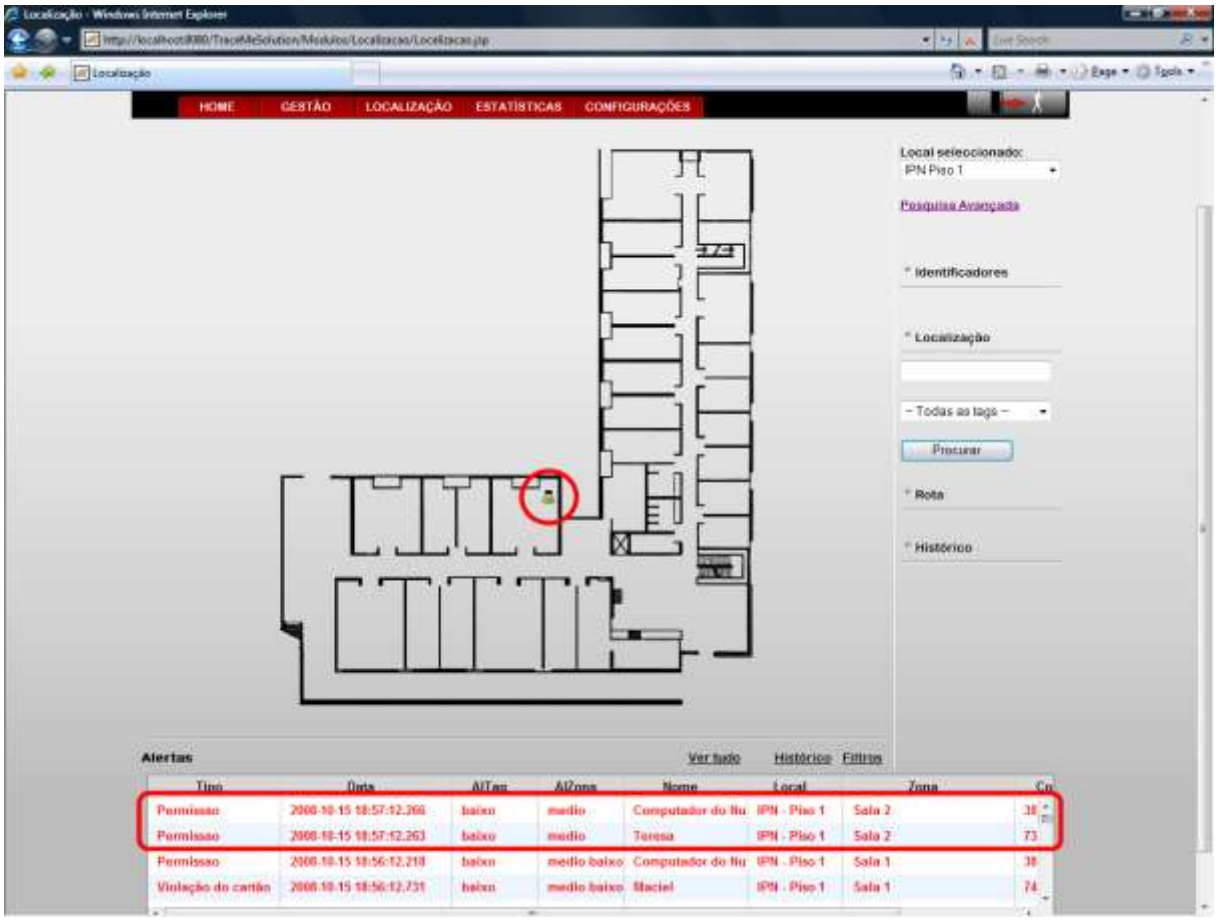



Figura 5-20 – Localização rápida da *tag* pertence ao Maciel na Área 3.

- **Detecção de ausência da *tag*** – Através da ausência da comunicação da *tag* durante um tempo pré-definido é criado um alerta de detecção na lista dos alertas. Portanto as *tags* que saírem do edifício (não se encontram ao alcance dos leitores) são detectadas quando o tempo pré-definido for expirado. As *tags* foram configuradas para não transmitirem *beacons*. Os alertas criados são apresentados na Figura 5-21, na parte da visualização da localização só aparece o Maciel porque a localização rápida ficou activada.

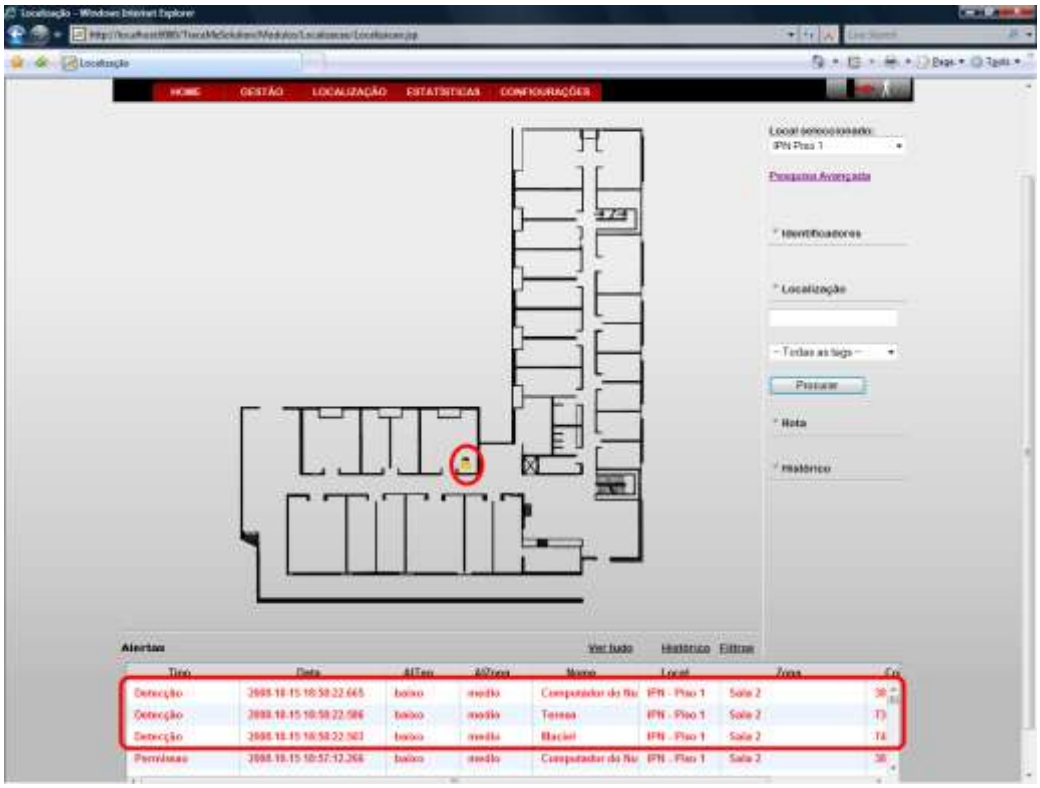



Figura 5-21 – Detecção de ausência da *tag* do Maciel.

De seguida foram activados os intervalos de *beacon* e foram transportadas para a Área 2, como é exemplificado na Figura 5-22.

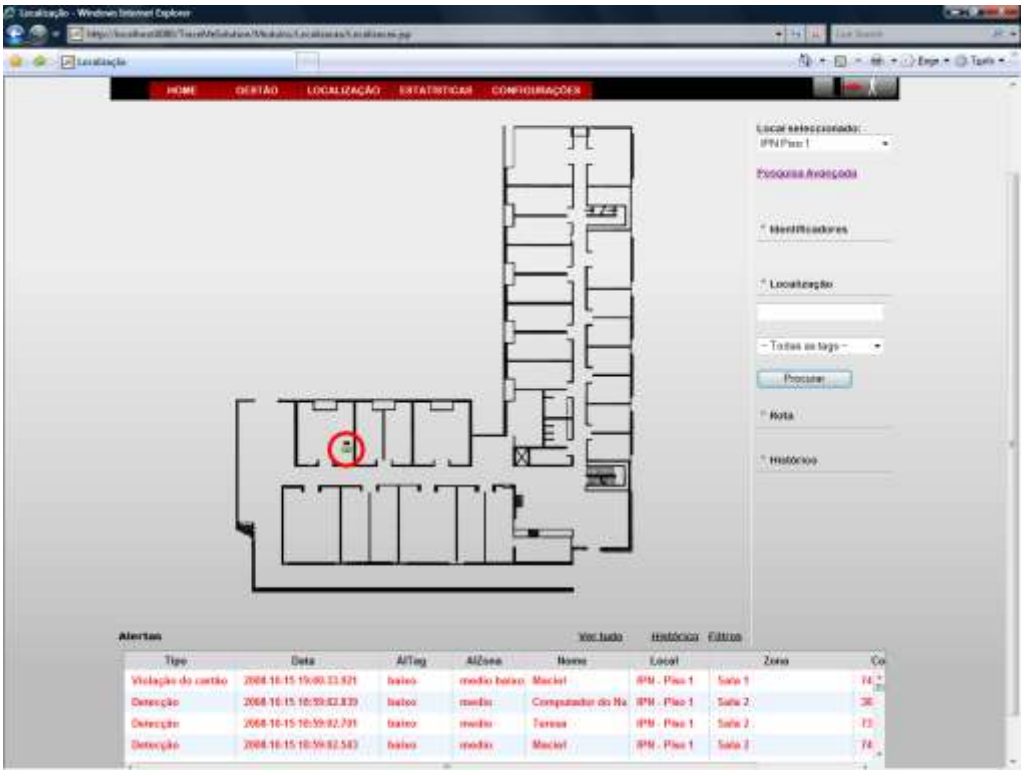



Figura 5-22 – Activação do intervalo dos *beacons* nas *tag*.

- **Detecção de falha do leitor** – Através do serviço "ReaderWatcher" é possível criar alertas de falha de comunicação com os leitores. Para este teste foi retirado o cabo de rede do leitor 103, o alerta criado é inserido na lista dos alertas, como é exemplificado na Figura 5-23.

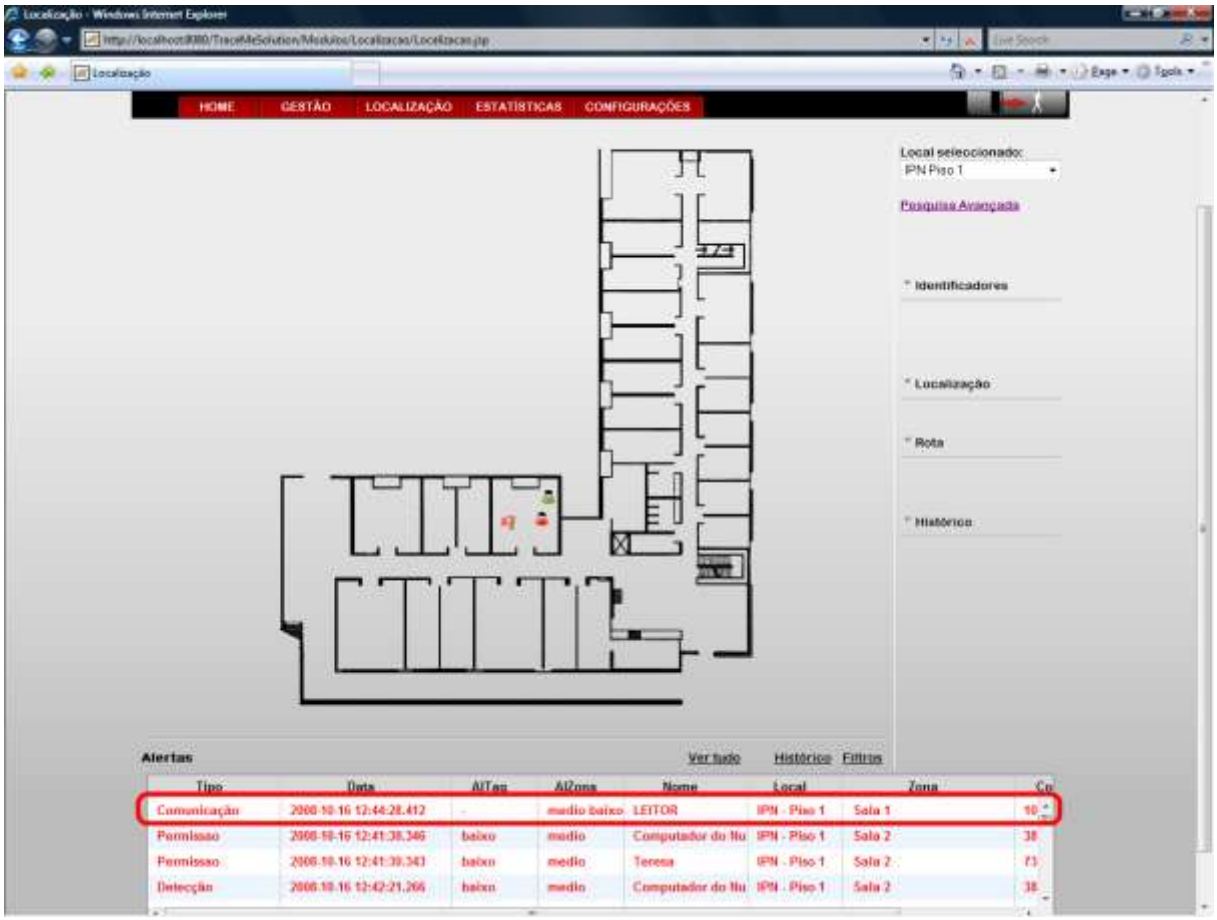



Figura 5-23 – Detecção de falha de comunicação com o leitor 103.

- **Detecção de bateria** – Em cada sinal que a *tag* transmite indica o estado da bateria, através dessa informação é possível detectar quando a *tag* está com pouca bateria e gerar um alarme de bateria. Para criar este cenário foi simulado uma *tag* com pouca bateria, como é exemplificado na Figura 5-24.

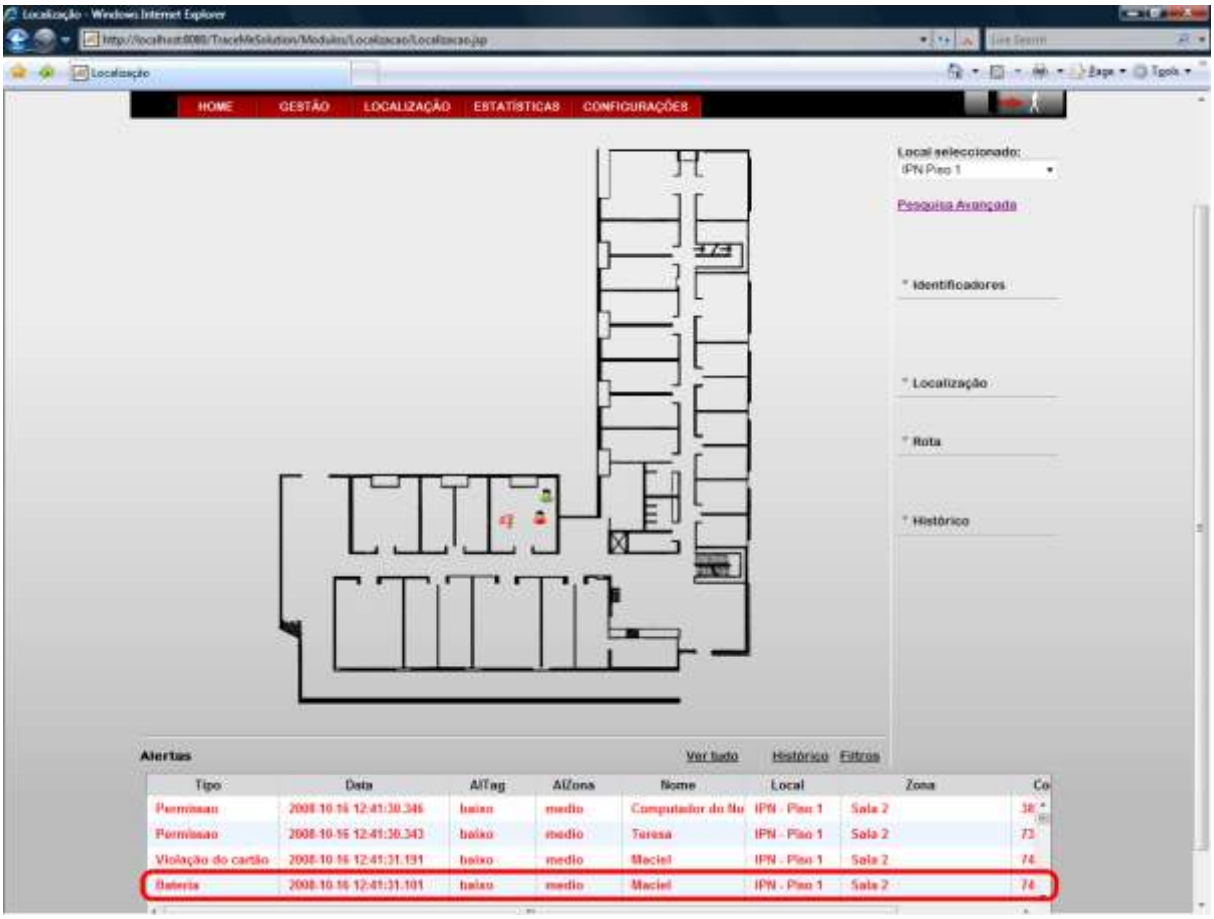



Figura 5-24 – Detecção de uma *tag* com pouca bateria.

Para cada alarme gerado é possível visualizar a sua localização através do ícone "Localização do alarme", como é exemplificado na Figura 5-25 a localização do alarme e falha de comunicação com o leitor.

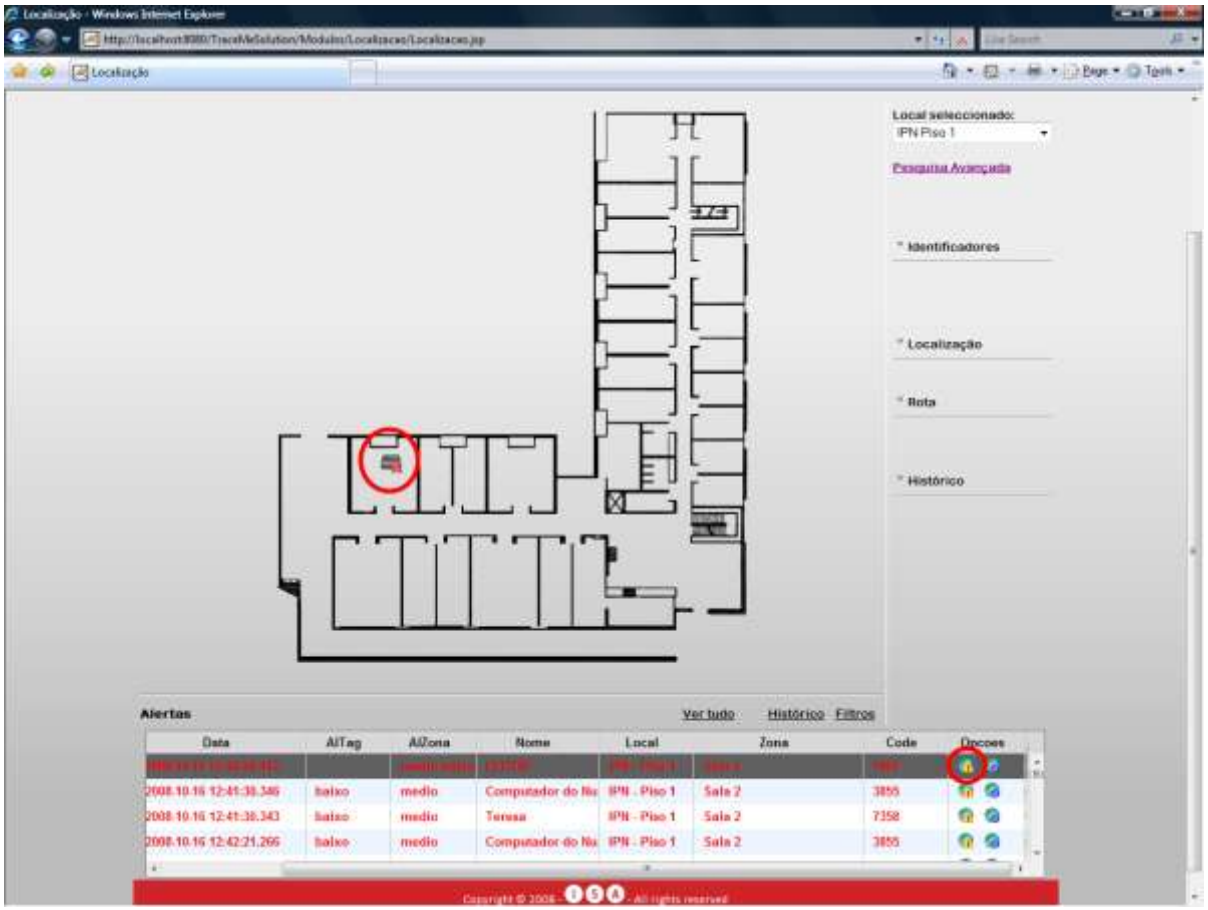



Figura 5-25 – Localização do alarme de comunicação com o leitor.

### 5.3.7. Teste num hospital

O objectivo deste teste de cobertura é para analisar o alcance dos leitores num hospital, fazendo um *site survey*. O cenário corresponde a uma ala de um edifício principal de uma unidade hospitalar, exemplificada na Figura 5-26, a escala da figura é 1/500. Através da figura pode analisar-se que é composto por 1 leitor. O leitor corresponde ao modelo com uma antena de 125 kHz incorporada. Os pontos assinalados correspondem a localização dos testes efectuados neste cenário. Existe mais ponto (22) que não está representado na figura porque é no piso superior na mesma posição do leitor, este ponto serve para analisar se o sinal consegue alcançar o piso superior.

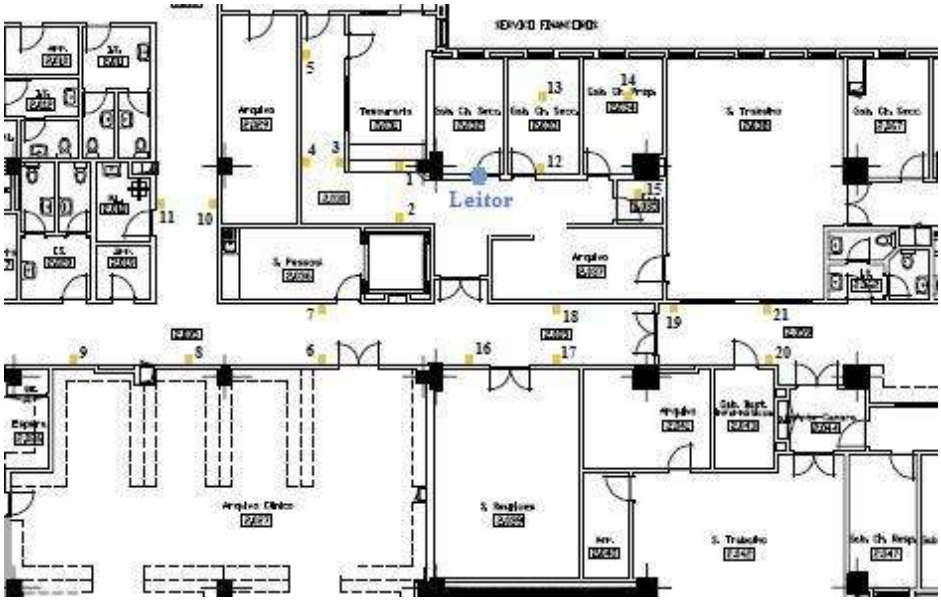



Figura 5-26 – Cenário 4 dos testes práticos realizados.

O leitor ficou posicionado na ombreia vertical da porta (ponto azul), em todos os testes efectuados as *tags* (pontos amarelos) estavam na posição vertical e eram fixos sem mobilização. A duração de cada teste foi de 60 segundos e a *tag* transmitia o *beacon* em 2,5 segundos, teoricamente os eventos recebidos sem perdas poderiam ir de 23 a 26. Na Tabela 5-9 são apresentados os resultados obtidos.

Tabela 5-9 – Resultados obtidos no teste de cobertura num hospital.

| Nome | Resultado |
|---|---|
| Teste 1 | Os eventos foram todos recebidos pelo sistema. |
| Teste 2 | Os eventos foram todos recebidos pelo sistema. |
| Teste 3 | Os eventos foram todos recebidos pelo sistema. |
| Teste 4 | Os eventos foram todos recebidos pelo sistema. |
| Teste 5 | Os eventos foram todos recebidos pelo sistema. |
| Teste 6 | Os eventos foram todos recebidos pelo sistema. |
| Teste 7 | Os eventos foram todos recebidos pelo sistema. |
| Teste 8 | Os eventos foram todos recebidos pelo sistema. |
| Teste 9 | O sistema não recebeu nenhum evento. |
| Teste 10 | O sistema não recebeu nenhum evento. |
| Teste 11 | Só um evento foi recebido pelo sistema. |
| Teste 12 | Os eventos foram todos recebidos pelo sistema. |
| Teste 13 | Os eventos foram todos recebidos pelo sistema. |
| Teste 14 | Os eventos foram todos recebidos pelo sistema. |
| Teste 15 | Os eventos foram todos recebidos pelo sistema. |
| Teste 16 | Os eventos foram todos recebidos pelo sistema. |
| Teste 17 | Os eventos foram todos recebidos pelo sistema. |
| Teste 18 | Os eventos foram todos recebidos pelo sistema. |
| Teste 19 | Os eventos foram todos recebidos pelo sistema. |
| Teste 20 | Os eventos foram todos recebidos pelo sistema. |
| Teste 21 | Só quatro eventos foram recebidos pelo sistema. |
| Teste 22 | Os eventos foram todos recebidos pelo sistema. |

Através dos testes efectuados pode-se afirmar, tendo em conta todos os parâmetros e condições dos testes, que as áreas que estão coloridas na Figura 5-27 têm cobertura por 868 MHz. Através do Teste 22 (posicionado no piso superior) verifica-se que o sinal consegue ultrapassar o piso.

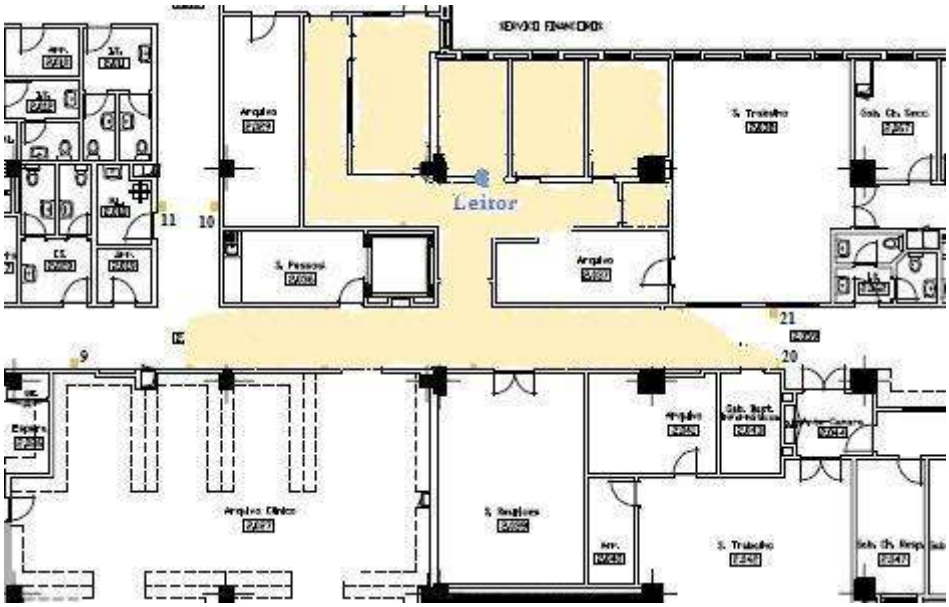



Figura 5-27 – Área de cobertura do leitor.

# 6. Conclusões finais

Nesta secção resumem-se algumas das conclusões mais importantes deste trabalho. As conclusões serão apresentadas em 3 secções distintas, embora algumas delas possam estar relacionadas entre si. Contudo, procurou agrupar-se as conclusões de acordo com os principais alvos de estudo, deste trabalho. As conclusões centram-se no desenvolvimento do *middleware*, no desenvolvimento do WebServer e nos testes práticos realizados.

No final da secção enumeram-se tópicos representativos de sugestões para eventuais trabalhos futuros, relacionados com os estudos desenvolvidos neste trabalho.

## 6.1. Desenvolvimento do *middleware*

O objectivo principal deste trabalho foi propor um sistema de localização de pessoas e objectos em ambientes interiores, através da tecnologia RFID. Para alcançar este objectivo foram propostos uma especificação e uma implementação de um *middleware*, a sua arquitectura baseada por camadas foi especificada através do estudo da tecnologia RFID e dos *middlewares* existentes no mercado. A implementação desta arquitectura alcançou o principal objectivo, através do mecanismo de localização proposto, onde a localização é feita através da detecção dos movimentos entre zonas predefinidas.

Além da localização foi implementado com sucesso o mecanismo de controlo de acessos, permitindo configurar os tipos de acesso e detectar acessos não autorizados, sempre que este é detectado é gerado um alerta no sistema para notificar os utilizadores da aplicação.

Outros alarmes propostos também foram desenvolvidos, as detecções de ausência de comunicação da *tag* durante um determinado tempo, a detecção de falha na comunicação dos leitores com o sistema, a detecção da violação da *tag* e a análise do estado da bateria. Estes alarmes são activados em tempo real e para cada um é gerado uma alerta no sistema para informar os utilizadores.

As funcionalidades de teste de cobertura e processamento de notificações geradas pela aplicação *Web* também foram desenvolvidas com sucesso, através de serviços pertencentes ao *middleware*. Na parte da implementação em si, foram alcançados os objectivos de comunicação com os leitores através de um *driver* genérico, permitindo assim maior flexibilidade e versatilidade nos equipamentos RFID, comunicação com uma base de dados, utilizando também o conceito do *driver* genérico e a comunicação com um servidor *Web* para actualizar automaticamente os campos no *browser*.

## 6.2. Desenvolvimento no servidor *Web*

A actualização dinâmica do *browser* foi conseguida através da implementação do Reverse Ajax, nomeadamente a biblioteca DWR. Através deste é possível actualizar dinamicamente um determinado *browser* sem precisar de fazer pedidos, permitindo assim a visualização da localização de pessoas e objectos e dos alertas gerados pelo sistema em tempo real. Também é possível fazer pedidos, designados por notificações, a partir do *browser* para o servidor. Estas notificações permite configurar os equipamentos do sistema RFID, testar o funcionamento dos leitores, comunicar com a base de dados e identificar qual a funcionalidade em execução e os seus parâmetros.

A partir das características do DWR foi possível implementar outras funcionalidades, como a localização rápida de uma determinada *tag*, visualizar os movimentos das *tags* entre zonas pré-definidas, mostrar o histórico das passagens das *tags*, identificar a localização de um determinado objecto ou pessoa numa planta e visualizar a localização de um determinado alarme.

## 6.3. Testes práticos realizados

Foram realizados testes no sentido de avaliar o comportamento e o funcionamento das funcionalidades do sistema. Os testes realizados foram testes de cobertura de um leitor ou vários leitores, validação das funcionalidades da aplicação, do algoritmo de localização e visualização das *tags* em tempo real e ao funcionamento das antenas exteriores com o algoritmo em modo anel duplo.

Em relação ao funcionamento em modo duplo verifica-se que a *tag* só não é detectado quando está numa mala com um portátil, nos restantes casos de observação os resultados obtidos foram os esperados.

As primeiras análises feitas foram nos testes em campo aberto, que permitia verificar o funcionamento dos equipamentos no espaço vazio, onde conclui-se que o valor do RSSI varia bastante de leitor para leitor, que as taxas de perdas para uma distância de 120 metros e 60 metros entre os leitores e as *tags* são inferiores a 20% e 10% respectivamente.

A monitorização de uma sala, que consiste na validação do funcionamento do teste de cobertura e análise do funcionamento dos equipamentos RFID, conclui-se o correcto funcionamento do serviço e a sensibilidade de detecção dos leitores dos sinais periódicos enviados pelas *tags*, esta sensibilidade depende bastante das condições do cenário e da localização dos leitores em relação ao posicionamento da *tag*.

No testes dos algoritmos de localização, que pretende validar os algoritmos de localização propostos, conclui-se que o algoritmo mais consistente e fiável é o método de localização através das detecções por 125 kHz, porque o seu alcance é inferior em relação aos sinais recebidos periodicamente por 868

MHz, o que permite diminuir o erro e aumentar a precisão da localização. Os sinais enviados periodicamente por 868 MHz actualiza a localização, permitindo indicar que a *tag* encontra-se dentro do edifício, ou seja, o sistema gera um alerta quando detecta que a *tag* não comunica durante um determinado tempo pré-definido, permitindo assim a detecção de *tags* que saiam do edifício sem autorização.

A localização em tempo real num edifício pretende analisar a visualização da localização em tempo real e o controlo de acessos não autorizados. Através do resultados obtidos conclui-se que o sistema cria um alerta quando uma *tag* é detectado numa zona que não tem acesso e insere na lista de alertas, conclui-se também a visualização dos movimentos da *tag* entre zonas.

Com aferição das funcionalidades do sistema pretende-se apresentar a validação de algumas funcionalidades importantes do sistema, como detecção da violação da *tag*, identificar uma determinada pessoa ou objecto numa planta, localização rápida de uma pessoa ou objecto, acompanhar a localização de um determinado objecto ou pessoa, detecção da falha de comunicação das *tags* com o sistema e detecção de falha de comunicação dos leitores.

Por fim, o teste num hospital pretende-se analisar o comportamento dos leitores num ambiente hospitalar. Recorrendo aos resultados obtidos, mas tendo em conta todos os parâmetros e condições dos testes, conclui-se uma área de cobertura do leitor e também que o sinal é detectado no piso superior.

## 6.4. Trabalho Futuro

O projecto TraceMe deverá ser comercializado, tendo como principal mercado os Hospitais, os principais objectivos são a detecção de roubos de bebés e de bens, o controlo de acesso e o sistema de localização dos funcionários e dos visitantes do hospital.

Algumas funcionalidades que podem ser adicionadas em versões futuras do sistema TraceMe são as seguintes sugestões:

1. Aperfeiçoar o mecanismo de localização, através de mais informação das zonas e dos seus possíveis vizinhos, analisar as movimentações detectadas pelo sistema são válidos e ao mesmo tempo avalia o funcionamento do mecanismo;
2. Adicionar o módulo de controlo de *stocks*, integrando *tags* e leitores passivos no sistema e também interligar com um sistema ERP para a gestão do *stock*;
3. Analisar o impacto da quantidade dos leitores e das *tags* no desempenho do sistema, desenvolver uma aplicação para simular leitores e *tags*;
4. Analisar o impacto do número de utilizadores a aceder aplicação no desempenho do sistema;

5. Adaptar aplicação para que a visualização das funcionalidades sejam suportadas nos telemóveis e PDA's;
6. Integrar com sistemas de domótica, como por exemplo o controlo de portas, janelas, sensores, iluminação, climatização e aumento da eficiência energética;
7. Integração com sistemas de videovigilância, permitindo associar uma câmara a um controlo de acesso e integrar também no sistema de localização.

Por fim, referir que o projecto TraceMe deverá ser comercializado, tendo como principal mercado os Hospitais, os principais objectivos são a detecção de roubos de bebés e de bens, o controlo de acesso e o sistema de localização dos funcionários e dos visitantes do hospital.

# Referências


[1] Kleder Miranda Gonçalves, "Um Framework para Comunicação Baseada em Localização", Dissertação de Mestrado, Universidade Católica do Rio de Janeiro**,**14 de Abril de 2005.

[2] Fernando Henrique Gines e Thiago Tadeu Tsai, "Projecto e Implementação de um sistema de identificação por RFID para uma aplicação de automação residencial", Relatório de Projecto de Final de Licenciatura, Escola Politécnica Universidade São Paulo, 2007.

[3] Harry Stockman, "Communication by Means of Reflected Power", Proceedings of the IRE, Volume 36, Issue 10, Outubro de 1948, páginas: 1196-1204.

[4] Hugo Miguel Cravo Gomes, "Construção de um sistema de RFID com fins de localização especiais", Dissertação de Mestrado, Universidade de Aveiro, 2007.

[5] Jeremy Landt, "The history of RFID", Potentials, IEEE, Volume 24, Issue 4, Outubro/Novembro 2005, páginas: 8-11.

[6] Tiago Palma e Ana Sofia Caetano, "Estado da Arte em RFID", Sybase, 12 de Outubro de 2007

[7] Alien Technology – Products – RFID Tags, *http://www.alientechnology.com/tags/index.php*

(Ultima vez visitado em Outubro 2008)

[8] Bonfield Group – Advanced Solution, *http://www.bonfieldgroup.com/index.php?p=standard_tag&m=p*

(Ultima vez visitado em Outubro 2008)

[9] Óscar Filipe Correia Brilha e Ricardo Filipe Leiria Veríssimo, "Smart Shopping Checkout Counter", Relatório de Projecto de Final de Licenciatura, Escola Superior de Tecnologia e Gestão – Instituto Politécnico de Leiria, Fevereiro 2008

[10] Sandip Lahiri, "RFID Sourcebook", IBM Press, Setembro 2005

[11] Raj Bridgelall, "RADAR Technology for Commodity Goods", Janeiro 2004, *http://www.ieee.li/pdf/viewgraphs_rfid.pdf*

(Ultima vez visitado em Outubro 2008)

[12] Klaus Finkenzeller, "RFID Handbook", 2ª edição da editora John Wiley & Sons, Ltd, 2003

[13] Alessandro de Souza Oliveira e Milene Franco Pereira, "Estudo da tecnologia de identificação por radiofrequência - RFID", Projecto de Graduação, Faculdade de Tecnologia – Universidade de Brasília, 12 de Dezembro de 2006.

[14] Hui Liu, Houshang Darabi, Pat Banerjee, e Jing Liu, "Survey of Wireless Indoor Positioning Techniques and Systems", Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions, Volume 37, Issue 6, Novembro 2007, páginas: 1067 – 1080.

[15] Martin Vossiek, Leif Wiebking, Peter Gulden, Jan Wieghardt, Clemens Hoffmann e Patric Heide, "Wireless Locating Positioning", Microwave Magazine, IEEE, Volume 4, Issue 4, Dezembro 2003, páginas: 77 – 86.

[16] Fernando Bergamos Mariani e Marcelo Nakamotome, "Simulador de Fluxo de Pessoas em Ambientes Monitorados", Relatório de Projecto de Final de Barchelato, Instituto de Ciências Exatas – Universidade de Brasília, 27 de Junho de 2008.

[17] Himanshu Bhatt e Bill Glover, "RFID Essentials", 1ª edição da editora O'Reilly Media, Inc, Janeiro 2006

[18] Sun-rfid: Sun Java System RFID Software 3.0,

*https://sun-rfid.dev.java.net/SJS_RFID_Software.html*

(Ultima vez visitado em Agosto 2008)

[19] RFID Anywhere –Data Management Software Application –Radio Frequency Identification System – Sybase Inc, *http://www.sybase.com/products/rfidsoftware/rfidanywhere*

(Ultima vez visitado em Agosto 2008)

[20] RFID Software Applications – Radio Frequency Identification Systems & Solutions Supplier – Sybase Inc, *http://www.sybase.com/products/rfidsoftware*

(Ultima vez visitado em Agosto 2008)

[21] logicAlloy.com – rfid made easy, *http://www.logicalloy.com/*

(Ultima vez visitado em Agosto 2008)

[22] logicAlloy.com – rfid made easy, *http://www.logicalloy.com/ale_screenshots.cfm*

(Ultima vez visitado em Agosto 2008)

[23] Java SE Technologies – Database, *http://java.sun.com/javase/technologies/database/*

(Ultima vez visitado em Agosto 2008)

[24] Huiwei Guan, Horace H. S. Ip e Vanchun Zhang, "Java-based approaches for accessing databases on the Internet and a JDBC-ODBC implementation", Computing & Control Engineering Journal, Volume 9, Issue 2, Abril 1998, páginas: 71 – 78.

[25] "JSP Architecture", *http://java.sun.com/developer/Books/javaserverpages/Chap12.pdf*

(Ultima vez visitado em Outubro 2008)

[26] JSP architecture – JSP Tutorial, *http://www.visualbuilder.com/jsp/tutorial/pageorder/4/*

(Ultima vez visitado em Outubro 2008)

[27] Google Maps .Net Control, *http://gmapsdotnetcontrol.blogspot.com/2006/08/exploring-reverse-ajax-ajax.html*

(Ultima vez visitado em Agosto 2008)

[28] Overview of DWR | Direct Web Remoting, *http://directwebremoting.org/dwr/overview/dwr*

(Ultima vez visitado em Setembro 2008)

[29] AeroScout MobileView | AeroScout, *http://www.aeroscout.com/content/mobileview*

(Ultima vez visitado em Agosto 2008)

[30] AeroScout MobileView 4.0 Data Sheet, *http://www.aeroscout.com/leadcapture/files/AeroScout+MobileView+Data+Sheet.pdf*

[31] Ekahau – Ekahau RTLS, *http://www.ekahau.com/?id=4200*

(Ultima vez visitado em Dezembro 2007)

[32] Ekahau RTLS Brochure, *http://www.ekahau.com/file.php?id=99414*

(Ultima vez visitado em Outubro 2008)

[33] Healtrax Asset and Patient Tracking Solution, *http://www.infologix.com/pdf/infologix-healthtrax.pdf*

(Ultima vez visitado em Outubro 2008)

[34] WiseTrack is Asset Management for Tracking Assets using RFID, Bar Code, Moble Scanners and the Web, *http://www.wisetrack.com/*

(Ultima vez visitado em Dezembro 2007)

[35] PostgreSQL: About, *http://www.postgresql.org/about/*

(Ultima vez visitado em Agosto 2008)

[36] Apache Tomcat – Apache Tomcat, *http://tomcat.apache.org/*

(Ultima vez visitado em Agosto 2008)

[37] The Institute of Electrical and Electronics Engineers, “IEEE Standard for Software Test Documentation”, IEEE Std 829-1998, 16 de Dezembro de 1998.

# Anexos

## A.1. Empresas do sector

A lista aqui apresentada indica as principais fabricantes da tecnologia RFID.

Tabela A-1 – Lista de Fabricantes da tecnologia RFID

| Fabricante | Produtos | Fonte |
|---|---|---|
| Applied Wireless ID | Antenas, *tags* e leitores | *http://www.awid.com* |
| Alien Technology | *Tags* passivas, semi-passivas e leitores | *http://www.alientechnology.com* |
| Avery Dennison | *Tags* Gen1 e Gen2 | *http://www.rfid.averydennison.com* |
| Biomark | Antenas, *tags* e leitores | *http://www.biomark.com* |
| Brooks Automation | Antenas e leitores | *http://www.brooks.com* |
| Confidex | *Tags* passivas e interrogadores | *http://www.confidex.fi* |
| Datamax Corporation | *Tags*, leitores, impressoras | *http://www.datamaxcorp.com* |
| Dynasys | *Tags* e leitores activos | *http://www.dynasys.pt/* |
| Ensyc | Dispositivos leitura/escrita | *http://www.ensycrfid.com* |
| FEIG | Leitores e impressoras | *http://www.feig.de* |
| Impinj | *Tags* e leitores | *http://www.impinj.com* |
| Intermec | *Tags*, leitores, antenas e impressoras RFID | *http://www.intermec.com* |
| Mifare | Leitores e *smart cards* | *http://www.mifare.net/about/* |
| Omron | Antenas, *tags* e leitores | *http://www.omronrfid.com* |
| OrganicID | *Tags* passivas orgânicas | *http://www.organicid.com* |
| Paxar | *Tags* passivas, *smart labels* e impressoras RFID | *http://www.paxar.com* |
| Philips | *Tags* e leitores | *http://www.philips.com* |
| Printronix | Impressoras RFID | *http://www.printonix.com* |
| Psion Teklogix | Leitores e impressoras | *http://www.psionteklogix.com* |

| | | |
|---|---|---|
| | RFID | |
| Reva Systems | Processadores de aquisição de dados de RFID | *http://www.revasystems.com* |
| RF Code | *Tags* e leitores | *http://www.rfcode.com* |
| Sato | *Tags*, leitores e impressoras RFID | *http://www.satoamerica.com* |
| SAVR Communication | *Tags* e leitores | *http://www.savrcom.com* |
| Symbol | *Tags*, leitores, *inlays* e até mesmo portais completos prontos a instalarem | *http://www.symbol.com* |
| TagSys | Todo o tipo de material | *http://www.tagsysrfid.com* |
| Texas Instruments | Todo o tipo de produtos | *http://www.ti.com/rfid* |
| Verichip | Aplicações médicas (neonatal e idosos) | *http://www.verichip.com* |
| Zebra | Todo o tipo de dispositivos relacionados com RFID | *http://www.zebra.com* |

## A.2. Diagrama da Base de Dados

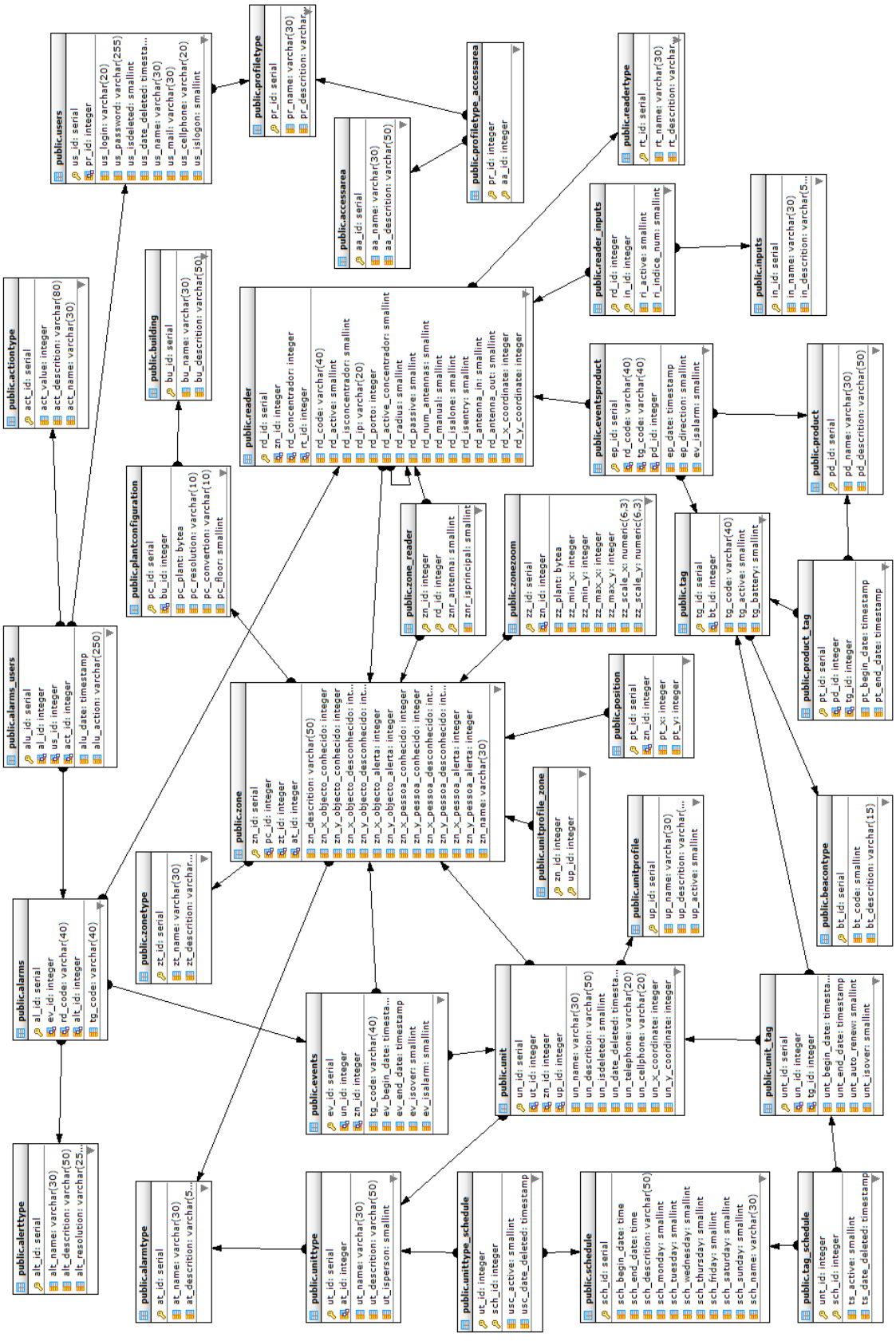



Figura A-1 – Diagrama da Base de Dados

## A.3. Leitor-001

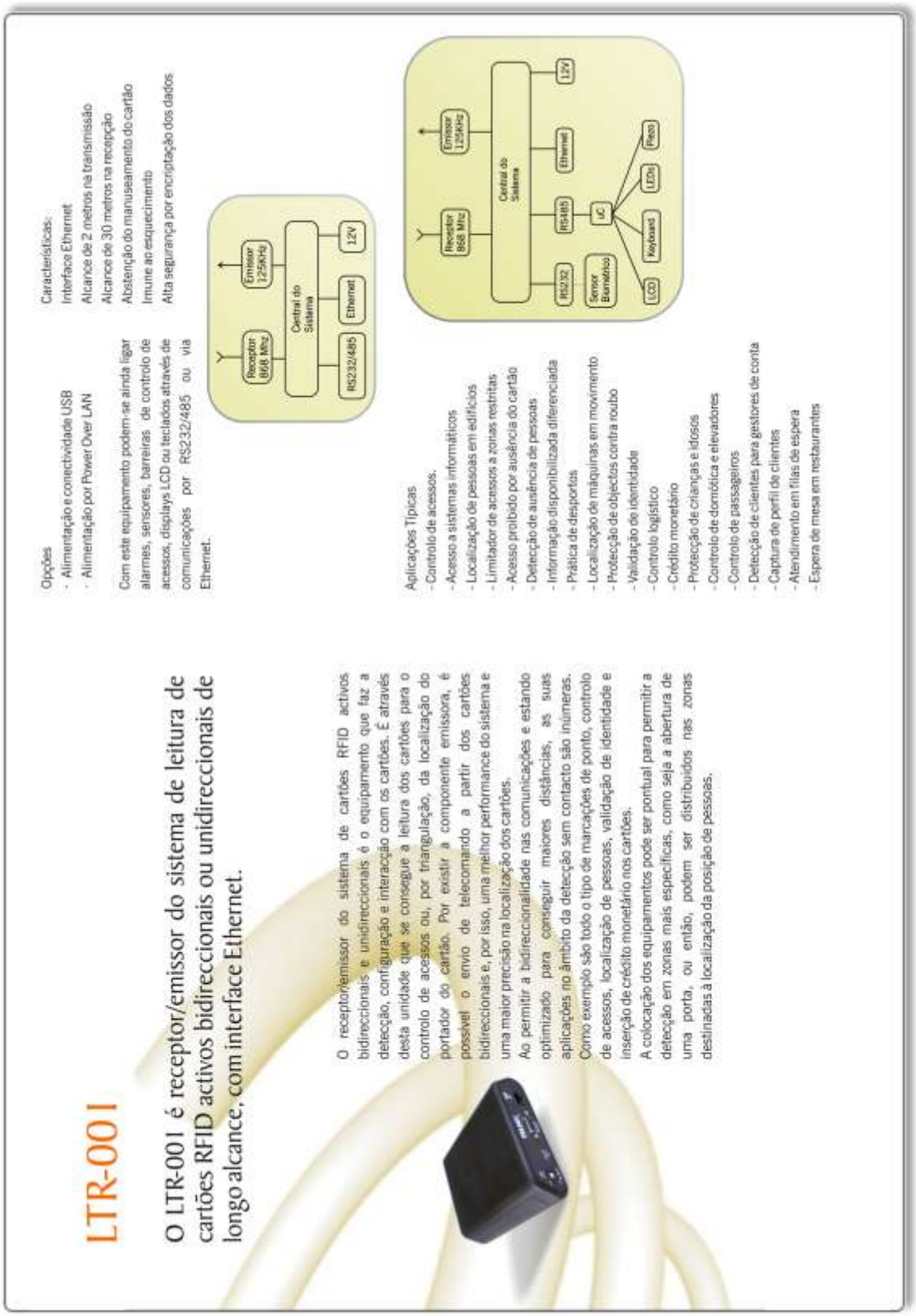



Figura A-2 – Descrição do Leitor-001.

## A.4. Leitor-002

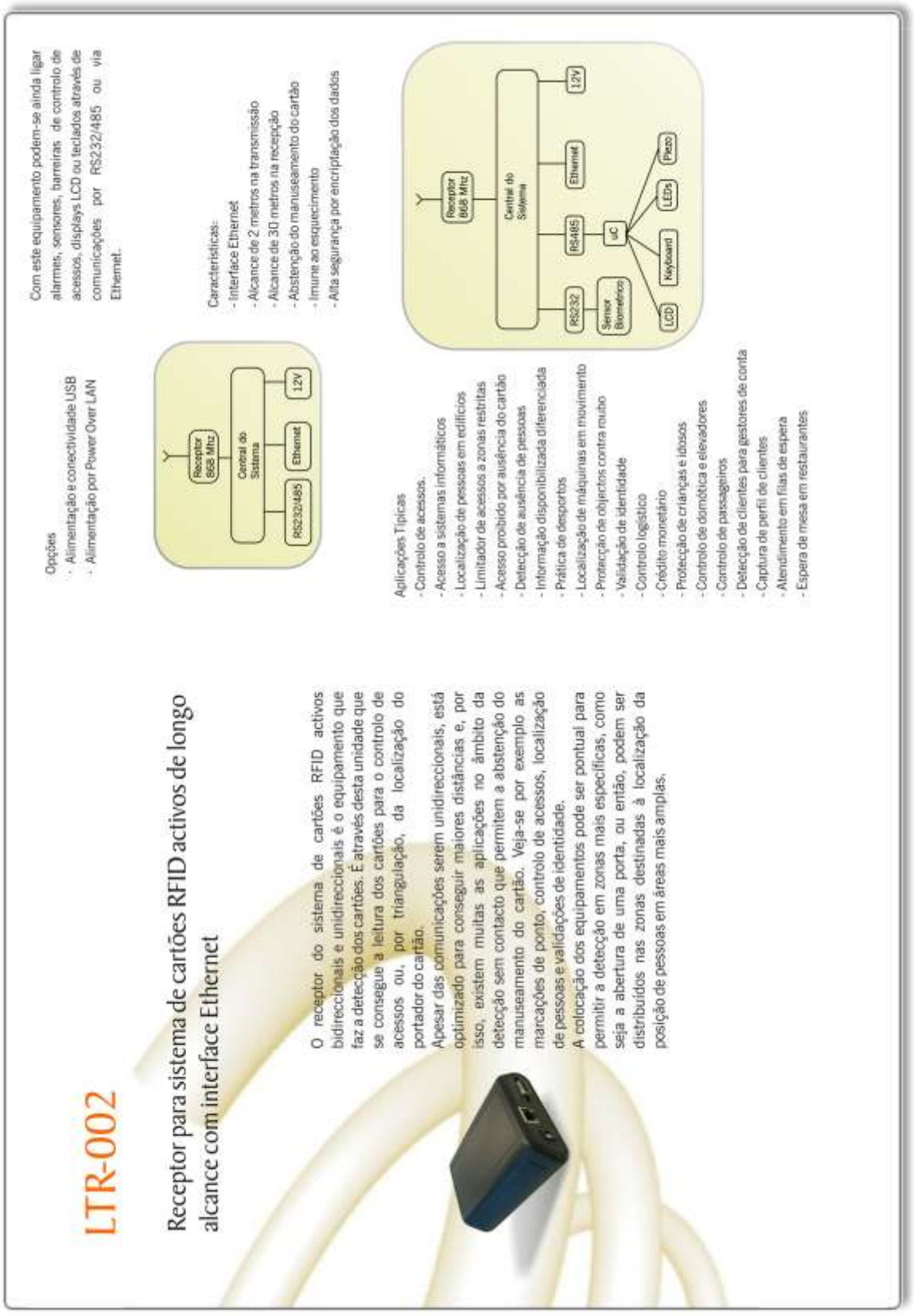



Figura A-3 – Descrição do Leitor-002.

## A.5. *Tag* IDA-003

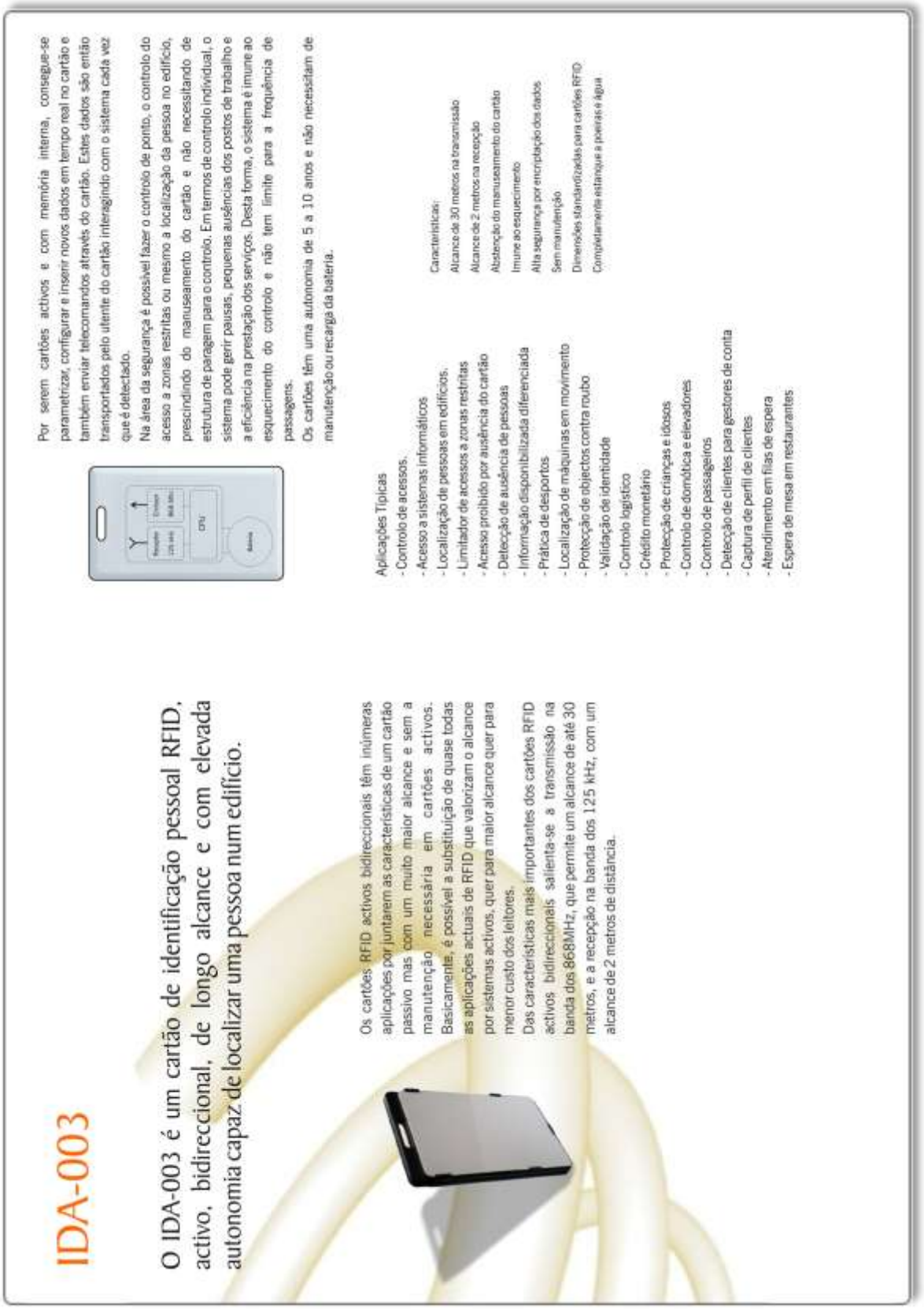



Figura A-4 – Descrição da *tag* IDA-003.

## A.6. *Tag* IDA-004

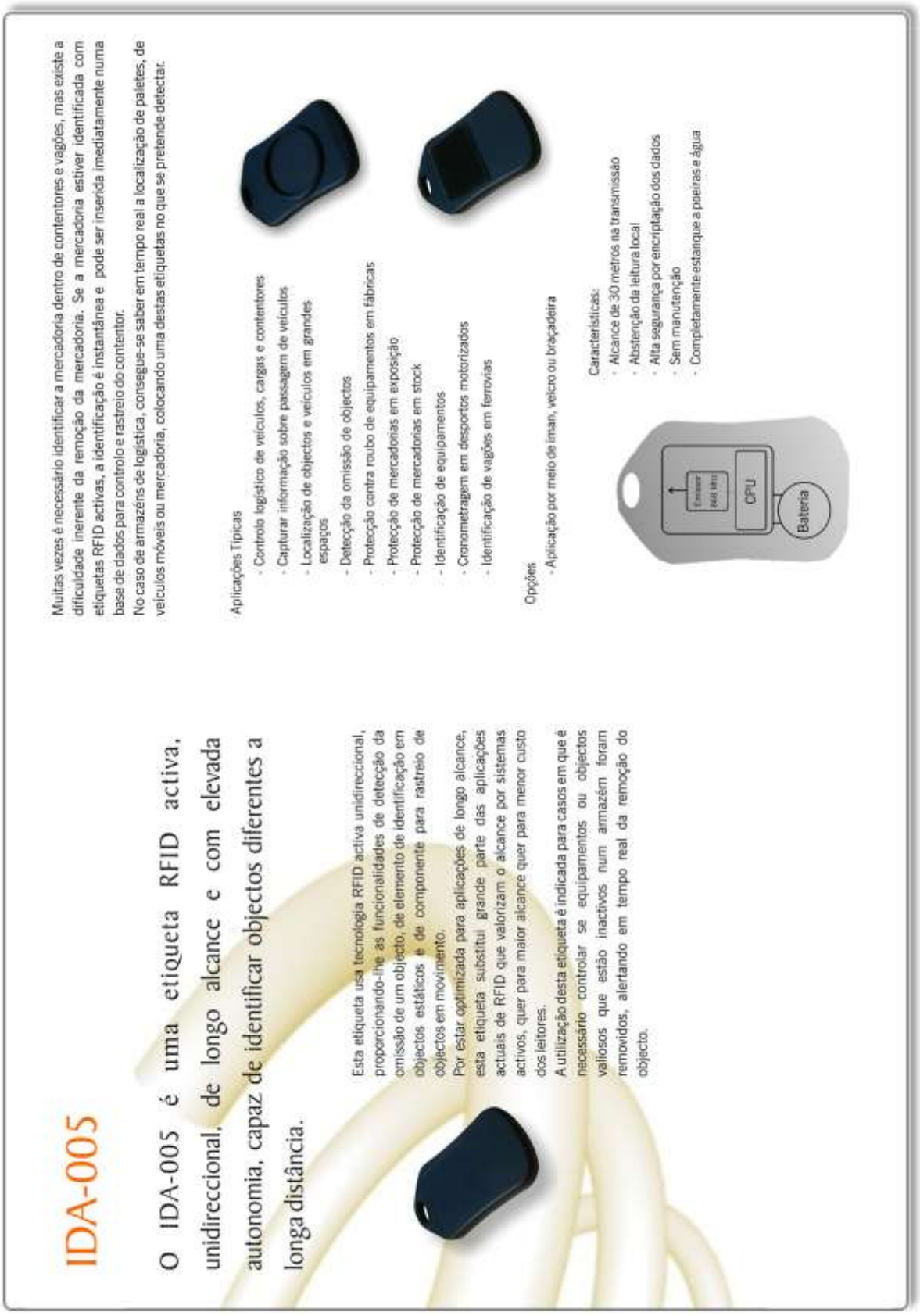



Figura A-5 – Descrição da *tag* IDA-004.

## A.7. *Tag* IDA-005

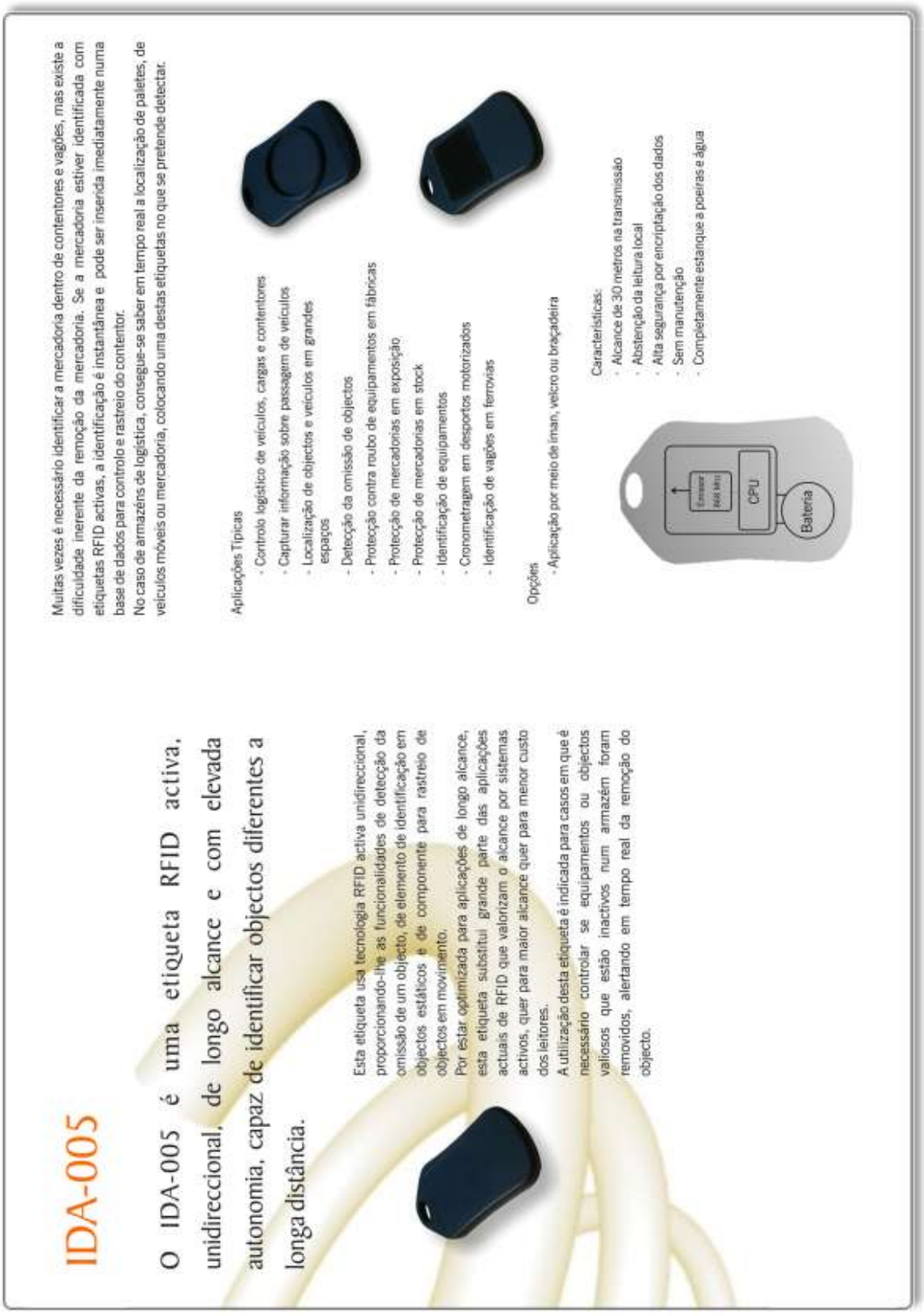



Figura A-6 – Descrição da *tag* IDA-005.

## A.8. *Tag* IDA-007


Figura A-7 – Descrição da *tag* IDA-007.

## A.9. Explicação sobre modo duplo anel

Este modo de funcionamento tem como objectivo permitir determinar o sentido de passagem de uma *tag* (IDA003 ou IDA004) usando um leitor. Para isso são colocados dois anéis justapostos, um ligado na saída *LOOP IN* e o outro no *LOOP OUT* do leitor, sendo este configurado para que a *tag* só transmita quando detecta transições entrada/saída ou saída/entrada.

Uma *tag* quando recebe informação de um anel com a indicação de ser um duplo anel, não executa uma transmissão por cada recepção de tramas mas apenas quando detecta mudança de estado (entrada para saída ou saída para entrada) caso não detecte não transmite nada. Quando transmite envia quatro tramas iguais espaçadas aleatoriamente no tempo. O leitor quando recebe essas tramas envia uma única trama para o servidor.

Este modo é activado através de um parâmetro de configuração e neste modo o leitor passa a sinalizar nas tramas enviadas por 125KHz a indicação que é um anel duplo e a *tag* que recebe essa informação altera o seu normal comportamento de resposta. O algoritmo da *tag* garante os seguintes pontos:

1. Ao fim de cerca de 2 segundos de ter recebido informação que lhe permite aferir que houve alteração de estado transmite essa informação por 868 MHz.
2. No caso de receber dos dois anéis considera como estando no interior.
3. Possui os seguintes estados:

Tabela A-2 – Estados da *tag* em modo de anel duplo.

| ***Tag* IDA003 ou IDA004** | | | | |
|---|---|---|---|---|
| **Estado** | **Estado actual** | | **TX** | **Estado** |
| **Inicial** | **RX *loop IN*** | **RX *loop OUT*** | **868 MHz** | **Final** |
| Exterior | Não | Não | Não | Exterior |
| Exterior | Não | Sim | Não | Exterior |
| Exterior | Sim | Sim | Sim | Interior |
| Exterior | Sim | Não | Sim | Interior |
| Interior | Não | Não | Não | Interior |
| Interior | Sim | Não | Não | Interior |
| Interior | Sim | Sim | Não | Interior |
| Interior | Não | Sim | Exterior | Exterior |

Para que a detecção seja eficaz há que garantir que a *tag* passa sobre os anéis a uma velocidade máxima que permite receber nas zonas em que recebe de um só anel pelo menos 3 tramas (distancia *d* da Figura A-8). Isto porque se for o tempo de uma única transmissão de 125 kHz pode ter um número de falhas incomportável para a fiabilidade de detecção pretendida.

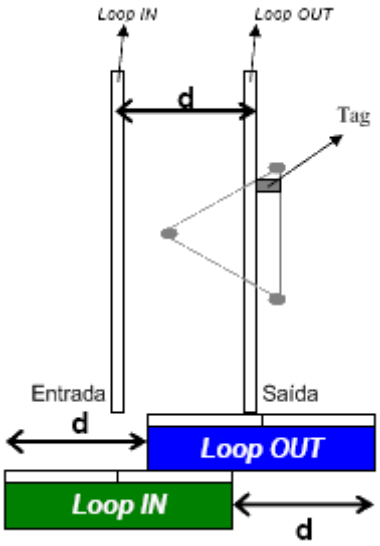



Figura A-8 – Radiação dos anéis duplos.

Os leitores executam as transmissões de dados por 125 kHz a um ritmo constante, o valor de fábrica é de 200 ms, ou seja no caso do anel duplo transmite por um anel 100 ms depois de ter transmitido pelo outro anel. Mas este tempo pode ser configurado através de parâmetro de configuração, em que pode ser colocado até um mínimo de cerca de 40 ms. Assim sendo a velocidade máxima que o sistema detecta é dada por *d* a dividir pelo parâmetro configurado, no entanto por razões de fiabilidade é conveniente considerar *d* a dividir por três vezes o parâmetro configurado.

## A.10. Testes ao algoritmo de localização através do método de triangulação

Tabela A-3 – Resultados obtidos do algoritmo de localização através do método de triangulação.

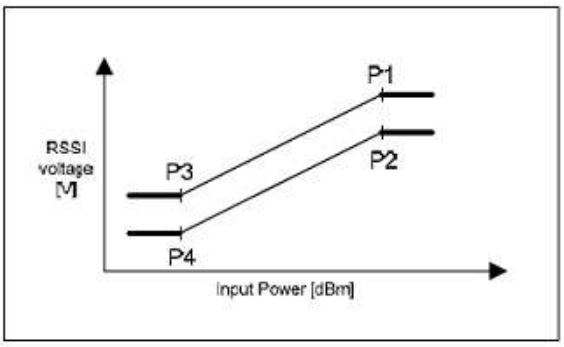
| Caracterização do cenário | | | Equipamentos | | |
|---|---|---|---|---|---|
| **Nome** | **Altura do teste** | **Informação obtida** | **Leitor 300** | **Leitor 302** | **Leitor 303** |
| Teste 1 | Manhã | Eventos (%) | 0 | 0 | 100 |
| | | RSSI (dBm) | - | - | -83,15/-82,67 |
| | Início da tarde | Eventos (%) | 0 | 0 | 100 |
| | | RSSI (dBm) | - | - | -82,42/-79 |
| | Final da tarde | Eventos (%) | 0 | 16 | 100 |
| | | RSSI (dBm) | - | -94,14/-93,65 | -78,27/-76,81 |
| | | | | | |
| Teste 2 | Manhã | Eventos (%) | 0 | 100 | 100 |
| | | RSSI (dBm) | - | -92,68/-86,33 | -82,18/-77,05 |
| | Início da tarde | Eventos (%) | 0 | 20 | 100 |
| | | RSSI (dBm) | - | -94,14/-90,97 | -69/-68,99 |
| | Final da tarde | Eventos (%) | 0 | 100 | 100 |
| | | RSSI (dBm) | - | -88,28/-86,08 | -68,99/-68,26 |
| | | | | | |
| Teste 3 | Manhã | Eventos (%) | 0 | 100 | 100 |
| | | RSSI (dBm) | - | -81,69/-79,74 | 83,40/-79,98 |
| | Início da tarde | Eventos (%) | 0 | 20 | 80 |
| | | RSSI (dBm) | - | -92,67-86,33 | -93,16/-82,91 |
| | Final da tarde | Eventos (%) | 0 | 12 | 100 |
| | | RSSI (dBm) | - | -93,9/-92,92 | -84,62/-82,42 |
| | | | | | |
| Teste 4 | Manhã | Eventos (%) | 0 | 100 | 100 |
| | | RSSI (dBm) | - | -81,69/-79,74 | -82,67/-79,98 |
| | Início da tarde | Eventos (%) | 0 | 100 | 100 |
| | | RSSI (dBm) | - | -81,45/-77,78 | -82,42/-80,96 |
| | Final da tarde | Eventos (%) | 3,85 | 100 | 100 |
| | | RSSI (dBm) | -90,97 | -78,03/-74,61 | -82,67/-78,52 |
| | | | | | |
| Teste 5 | Manhã | Eventos (%) | 0 | 100 | 100 |
| | | RSSI (dBm) | - | -70,46/-69,55 | -70,46/-66,55 |
| | Início da tarde | Eventos (%) | 3,85 | 100 | 100 |
| | | RSSI (dBm) | -91,21 | -87,3/-75,34 | -77,05/-76,07 |
| | Final da tarde | Eventos (%) | 12 | 100 | 100 |
| | | RSSI (dBm) | -92,68/-92,19 | -79,74/-78,03 | -75,1/-73,88 |
| | | | | | |
| Teste 6 | Manhã | Eventos (%) | 86 | 100 | 14 |
| | | RSSI (dBm) | -73,39/-72,17 | 84,38/-79 | -94,87/-92,92 |
| | Início da tarde | Eventos (%) | 58 | 67 | 100 |
| | | RSSI (dBm) | -84,38/-79,74 | -94,14/-83,87 | -92,43/-88,28 |
| | Final da tarde | Eventos (%) | 92 | 100 | 100 |
| | | RSSI (dBm) | -83,89/-80,47 | -90/-87,55 | -87,55/-78,02 |
| | | | | | |
| Teste 7 | Manhã | Eventos (%) | 32 | 100 | 0 |
| | | RSSI (dBm) | -92,19/-90,72 | -92,68/-86,33 | - |
| | Início da tarde | Eventos (%) | 22 | 91 | 0 |

| | | | | | |
|---|---|---|---|---|---|
| | | RSSI (dBm) | -89,5/-88,53 | -93,41/-91,7 | - |
| | Final da tarde | Eventos (%) | 35 | 100 | 0 |
| | | RSSI (dBm) | -92,43/-91,46 | -90,23/-88,28 | - |
| | | | | | |
| Teste 8 | Manhã | Eventos (%) | 100 | 100 | 100 |
| | | RSSI (dBm) | -82,42/-79,74 | -70,70/-68,26 | -88,28/-85,12 |
| | Início da tarde | Eventos (%) | 71 | 100 | 96 |
| | | RSSI (dBm) | -82,91/-80,71 | -76,32/-70,95 | -87,79/-83,15 |
| | Final da tarde | Eventos (%) | 69 | 100 | 96 |
| | | RSSI (dBm) | -82,42/-79,98 | -70,21/-66,55 | -93,16/-86,08 |
| | | | | | |
| Teste 9 | Manhã | Eventos (%) | 73 | 100 | 96 |
| | | RSSI (dBm) | -90,72/-83,4 | -72,66/-70,46 | -94,63/-90,97 |
| | Início da tarde | Eventos (%) | 88 | 100 | 84 |
| | | RSSI (dBm) | -86,08/-84,86 | -80,47/-78,76 | -94,38/-90,97 |
| | Final da tarde | Eventos (%) | 80 | 100 | 100 |
| | | RSSI (dBm) | -77,29/-73,88 | -74,85/-71,44 | -90,72/-84,34 |
| | | | | | |
| Teste 10 | Manhã | Eventos (%) | 65 | 100 | 4 |
| | | RSSI (dBm) | -77,78/-75,34 | -71,92/-68,75 | -92,43 |
| | Início da tarde | Eventos (%) | 92 | 100 | 100 |
| | | RSSI (dBm) | -83,25/-76,56 | -69,97/-67,53 | -89,06/-87,06 |
| | Final da tarde | Eventos (%) | 88 | 100 | 96 |
| | | RSSI (dBm) | -81,2/-74,37 | -68,26/-66,55 | -93,16/-85,1 |
| | | | | | |
| Teste 11 | Manhã | Eventos (%) | 58 | 100 | 0 |
| | | RSSI (dBm) | -87,06/-84,38 | -89,5/-85,84 | - |
| | Início da tarde | Eventos (%) | 40 | 100 | 16 |
| | | RSSI (dBm) | -90,97/-87,06 | -91,94/-83,4 | -95,36/-94,63 |
| | Final da tarde | Eventos (%) | 92 | 100 | 0 |
| | | RSSI (dBm) | -87,55/-83,64 | -84,38/-81,69 | - |
| | | | | | |
| Teste 12 | Manhã | Eventos (%) | 100 | 100 | 12 |
| | | RSSI (dBm) | -77,29/-75,34 | -77,05/-73,39 | -94,87/-93,4 |
| | Início da tarde | Eventos (%) | 100 | 100 | 33 |
| | | RSSI (dBm) | -77,54/-74,85 | -75,1/-71,92 | -95,12/-91,94 |
| | Final da tarde | Eventos (%) | 81 | 100 | 73 |
| | | RSSI (dBm) | -75,34/-72,17 | -78,03/-72,66 | -94,38/-90,72 |
| | | | | | |
| Teste 13 | Manhã | Eventos (%) | 75 | 100 | 33 |
| | | RSSI (dBm) | -87,3/-84,38 | -65,09/-64,84 | -93,9/-87,79 |
| | Início da tarde | Eventos (%) | 23 | 100 | 100 |
| | | RSSI (dBm) | -91,94/-90,23 | -65,09/-64,6 | -88,03/-84,38 |
| | Final da tarde | Eventos (%) | 76 | 100 | 96 |
| | | RSSI (dBm) | -86,08/-83,89 | -64,36/-64,11 | -94,14/-89,5 |
| | | | | | |
| Teste 14 | Manhã | Eventos (%) | 92 | 100 | 96 |
| | | RSSI (dBm) | -80,71/-78,03 | -73,39/-71,44 | -90,48/-88,28 |
| | Início da tarde | Eventos (%) | 92 | 100 | 48 |
| | | RSSI (dBm) | -83,4/-80,71 | -78,27/-72,17 | -94,38/-90,23 |
| | Final da tarde | Eventos (%) | 78 | 100 | 100 |
| | | RSSI (dBm) | -79,74/-78,76 | -72,17/-71,44 | -89,5/-85,84 |
| | | | | | |
| Teste 15 | Manhã | Eventos (%) | 62,5 | 100 | 91,67 |
| | | RSSI (dBm) | -84,62/-82,42 | -69,73/-68,51 | -87,79/-86,08 |

| | | | | | |
|---|---|---|---|---|---|
| | Início da tarde | Eventos (%) | 4,17 | 100 | 100 |
| | | RSSI (dBm) | -91,46 | -65,09/-64,6 | -91,21/-88,28 |
| | Final da tarde | Eventos (%) | 68 | 100 | 100 |
| | | RSSI (dBm) | -90,23/-86,33 | -66,06/-65,58 | -90,97/-88,28 |
| | | | | | |
| Teste 16 | Manhã | Eventos (%) | 87 | 100 | 100 |
| | | RSSI (dBm) | -81,69/-80,22 | -81,2/-79,25 | -89,5/-87,79 |
| | Início da tarde | Eventos (%) | 80 | 100 | 4 |
| | | RSSI (dBm) | -88,52/-84,86 | -73,39/-70,21 | -94,87 |
| | Final da tarde | Eventos (%) | 92 | 100 | 23 |
| | | RSSI (dBm) | -77,54/-75,83 | -72,17/-70,46 | -94,87/-92,92 |
| | | | | | |
| Teste 17 | Manhã | Eventos (%) | 95 | 100 | 52 |
| | | RSSI (dBm) | -86,57/-84,86 | -75,83/-72,9 | -94,87/-92,68 |
| | Início da tarde | Eventos (%) | 69 | 100 | 42 |
| | | RSSI (dBm) | -90,97/-87,55 | -64,36/-64,11 | -94,87/-91,21 |
| | Final da tarde | Eventos (%) | 88 | 100 | 100 |
| | | RSSI (dBm) | -87,79/-83,64 | -64,6/-64,36 | -94,38/-89,75 |
| | | | | | |
| Teste 18 | Manhã | Eventos (%) | 0 | 77 | 100 |
| | | RSSI (dBm) | - | -94,14/-90,72 | -83,89/-80,71 |
| | Início da tarde | Eventos (%) | 0 | 68 | 100 |
| | | RSSI (dBm) | - | 93,65/-90,23 | -85,35/-83,15 |
| | Final da tarde | Eventos (%) | 0 | 92 | 100 |
| | | RSSI (dBm) | - | -92,68/-89,26 | -82,91/-81,69 |

## A.11. Conversão do valor RSSI

A Figura A-9 indica a relação entre o valor analógico da tensão (V) e a potência recebida do RF (dBm), através da figura pode verificar-se que o RSSI possui uma excursão 700 mV. O *firmware* do leitor após *reset* estima o valor de *offset* (P3 ou P4) e quando recebe dados válidos converte para digital o valor de RSSI e estima o seu valor médio, subtraindo o valor do *offset*. Este valor médio sem *offset* corresponde ao valor disponibilizado em cada trama, podendo ser convertido para tensão se multiplicar por 5/1024 (5 Volts, 1024 *steps*) obtêm-se o resultado em Volt.




| | | |
|---|---|---|
| P1 | -65 dBm | 1300 mV |
| P2 | -65 dBm | 1000 mV |
| P3 | -100 dBm | 600 mV |
| P4 | -100 dBm | 300 mV |

Figura A-9 - Relação entre o valor analógico da tensão e a potência recebida da RF.

A expressão para converter o valor médio para tensão é a seguinte:

$$y = x \times \left(\frac{5}{1024}\right) \tag{1}$$

onde, *y* corresponde o valor em V e *x* o valor do RSSI lido e convertido pela ADC (*Analog-to-Digital Converter*). Para ter o valor em mV é necessário multiplicar por 1000. O declive das rectas representadas na Figura A-9 é 1/20. Recorrendo a expressão da recta,

$$y = ax + b \tag{2}$$

sendo, *a* o declive da recta e o *b* o ponto de intersecção com o eixo do *y*, a expressão da recta entre os valores P3,P1 e P4,P2 é a seguinte:

$$y = \frac{1}{20} \times \left(\left(x \times \frac{5}{1024}\right) \times 1000\right) - 100 \tag{3}$$

onde, *y* corresponde o valor em dBm e o *x* o valor médio do RSSI.