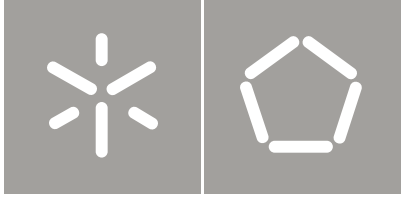


Universidade do Minho
Escola de Engenharia

António André Marques Ribeiro

RFID:
Middleware portável, genérico, autónomo e escalável

Tese de Mestrado de Informática

Trabalho efectuado sob a orientação dos
Professor Doutor António Nestor Ribeiro
Engenheiro Paulo Resende Almeida Lopes

Novembro de 2008

Dedicatória

Dedico este trabalho a todas as pessoas que fazem, ou fizeram, parte da minha vida, e que de alguma forma contribuíram para que convergisse neste ponto.

Agradecimentos

Não posso deixar de agradecer às pessoas que de alguma forma contribuíram para o bom rumo deste trabalho.

Assim, agradeço ao Professor António Nestor Ribeiro, pela paciência e esforço dispendido na minha orientação. Agradeço também ao Engenheiro Paulo Almeida Lopes, pela orientação, no contexto empresarial em que me inseri, para levar a cabo esta tese.

Não posso também, de forma alguma, esquecer os meus companheiros de mestrado que, em conjunto comigo, sofreram para a chegada a bom porto.

Quero ainda deixar um agradecimento, muito especial, para a entidade *Wipro Retail*, e a todos os seus colaboradores, que de diversas formas e por diversas vezes me deram conhecimentos e meios para atingir os meus fins.

A estes e a todos que ficaram fora destes agradecimentos, mas que igualmente merecem,

Muito Obrigado.

Resumo

O aumento da quantidade de objectos físicos ou abstractos com que temos de lidar no nosso dia-a-dia faz com que muitas vezes lhes percamos o rasto. Quantas vezes não damos por nós à procura, por exemplo, da chave do nosso carro? Quanto tempo se perde, diariamente, com esta procura?

Se isto acontece no mundo limitado dos objectos pessoais de um indivíduo, que dizer do enorme universo de objectos que podemos encontrar num espaço tão complexo como uma biblioteca? E que dizer quando se trata de, por exemplo, um entreposto comercial? Facilmente chegamos à conclusão que é necessário um grande esforço de coordenação para manter um tal conjunto de objectos organizados, de forma que seja simples e rápido localizar cada um deles individualmente.

Vamos agora imaginar que, de alguma forma, seria possível evitar todo esse esforço e ainda assim conseguir os mesmos (e melhores) resultados. Suponhamos que conseguíamos fazer com que, um objecto, quando procurado, tivesse a capacidade de nos comunicar a sua posição. Imaginemos ainda que seria possível, questionando um objecto, obter a sua identificação, historial, características, etc. Não será difícil imaginar as vantagens que, de semelhante situação, poderíamos retirar.

É precisamente aqui que a tecnologia RFID se revela importante. Esta tecnologia, através da colocação de dispositivos electrónicos, não só nos permite ter o conhecimento da localização de um objecto como ainda, e esta será mais uma enorme vantagem, sabermos univocamente de que objecto se trata, permitindo-nos dessa forma associar a este informação que apenas a ele pertence.

Naturalmente que, nem tudo são vantagens. Se imaginarmos um local onde milhares (ou muitas vezes milhões) de objectos estão constantemente a assinalar a sua presença, rapidamente se chega à conclusão que teremos de ter mecanismos adequados para a gestão de tamanho fluxo de dados.

É sobre este tema que se debruça este trabalho: no estudo de uma forma de controlar a gigantesca quantidade de dados proveniente de um sistema RFID, tirando o máximo partido das possibilidades desta tecnologia.

Palavras-Chave: RFID, Middleware RFID, Auto-identificação, Localização, EPC, Radiofrequência, JAVA, Gestão de Armazéns, Retalho.

Abstract

The growth of the amount of objects that we deal on a daily basis, very often, makes us lose their track. How many times do we find ourselves searching for a lost car key? And how much time is lost, every day, with this search?

If this happens in the limited world of personal objects, what can we say about the enormous universe of objects that we can find, let's say, in a library? And what about the amount of different objects we can find in a retailer's warehouse? Easily, we conclude on the great coordination effort that it is necessary to maintain such an amount of objects organized, in a way that we can, fast and easily, find any of them individually.

Let's now imagine that, somehow, we could avoid all this effort and still obtain the same (or even better) results. Let's suppose that, somehow, when searching for a specific object, this object would have the capability to announce his position. Imagine now that, questioning an object, it would be possible to obtain his identification, history, main characteristics, etc. It wouldn't be hard to find many advantages of a situation like this.

This is the exact point where the RFID technology comes in action. With this technology we can, not only have the information of the exact location of an object, but, also, we can know exactly what object it is, allowing us, this kind of identification, to keep information that belongs only to it.

Naturally, not everything is advantages. If we imagine a place where thousands (maybe millions) of objects are constantly announcing their presence, we soon conclude on the need of having adequate mechanisms to control such huge data flow.

This is exactly where this research is made, in the way to control the data flow originated in such a system without losing its enormous capabilities.

Key-Words: RFID, RFID middleware, Auto-identification, Tracking, EPC, Radio Frequency, JAVA, Warehouse-Management, Retail.

Índice de Conteúdos

1 Introdução 1
1.1 Enquadramento 2
1.2 Objectivos 3
1.3 Colaboração com a Wipro Retail 5
1.4 Este Documento 6
2 Tecnologia RFID 9
2.1 Introdução 9
2.1.1 Conceitos da Tecnologia 9
2.1.2 O Passado 11
2.1.3 O Presente 13
2.1.4 O Futuro 14
2.2 Funcionamento da Tecnologia 15
2.2.1 Princípios de Funcionamento 16
2.2.2 Principais Componentes 20
2.3 Caracterização das Etiquetas 29
2.3.1 UHF 30
2.3.2 HF 31
2.3.3 LF 32
2.3.4 Comparação das Caracterizações Apresentadas 33
2.4 Limitações tecnológicas 34
2.4.1 O problema da água 34
2.4.2 O problema do metal 36
3 Estado da Arte 39
3.1 Tipos de Envolvimento 39
3.2 Critérios de Avaliação 41
3.3 Sistemas existentes 42
3.3.1 SAP 43

3.3.2 Oracle 45
3.3.3 Comparação das soluções SAP e Oracle 48
4 Middleware RFID 49
4.1 Introdução 49
4.1.1 Enquadramento 49
4.2 Mote 52
4.2.1 Interfaces de comunicação 52
4.2.2 Interface *middleware* - clientes 53
4.2.3 Interface middleware - leitores 53
4.2.4 Interface de Administração 55
4.2.5 Outros requisitos 55
4.3 Concepção da solução 55
4.3.1 Tópicos relacionados com os clientes 56
4.3.2 Tópicos relacionados com os leitores 63
4.3.3 Administração do leitor 68
4.3.4 Funcionamento interno do *middleware* 69
4.4 Desenvolvimento do Middleware 76
4.4.1 Tecnologia utilizada 76
4.4.2 Interfaces de comunicação 77
4.4.3 Ordens 77
4.4.4 Leitores 83
4.4.5 Etiquetas 87
5 Discussão de Aplicabilidade 91
5.1 Mote 91
5.2 Alterações ao Processo 92
5.3 Justificação 93
5.4 Aplicação Prática 94
5.5 Outras possibilidades 98
6 Conclusões e trabalho futuro 99
6.1 Conclusões 99
6.1.1 Acerca da tecnologia RFID 99
6.1.2 Acerca do middleware proposto 100
6.2 Trabalho futuro 101

7 Bibliografia e referências 105
7.1 Artigos, estudos e manuais 105
7.2 Livros 108
8 Anexos 109
8.1 Diagrama de Classes 111
8.1.1 Diagrama Cortado 112
8.1.2 Diagrama Completo 116
8.2 Normas e Regulamentação 119
8.2.1 A EPC Global 119
8.2.2 O Código EPC 119
8.2.3 Outras Normas 124
8.3 Privacidade 127
8.3.1 Desactivação de Etiquetas 128
8.3.2 Etiqueta Bloqueadora 128
8.4 Outras Tecnologias de Identificação 129
8.4.1 Códigos de Barras 129
8.4.2 Reconhecimento Biométrico 131
8.4.3 Smart Cards 132
8.4.4 OCR 133

Índice de Ilustrações

FIGURA 2.1 - OS TRÊS ELEMENTOS DE UM SISTEMA RFID 17
FIGURA 2.2 - ESQUEMA DE UTILIZAÇÃO - IDENTIFICAÇÃO 18
FIGURA 2.3 - ESQUEMA DE UTILIZAÇÃO - LOCALIZAÇÃO POR ANTENA 19
FIGURA 2.4 - ESQUEMA DE UTILIZAÇÃO - LOCALIZAÇÃO POR TRIANGULAÇÃO 20
FIGURA 2.5 - EXEMPLO DE LEITOR E LIGAÇÕES COMUNS 21
FIGURA 2.6 - EXEMPLO DE LIGAÇÕES DE ANTENAS 22
FIGURA 2.7 – EXEMPLO DE LEITOR SDIO COM ANTENA EMBUTIDA 22
FIGURA 2.8 - EXEMPLO DE ANTENA 24
FIGURA 2.9 - EXEMPLO DE ANTENA EMBUTIDA EM LEITOR 25
FIGURA 2.10 - DIFERENÇAS DE ALCANCE ENTRE POLARIZAÇÕES 25
FIGURA 2.11 - ETIQUETAS UHF DE PAPEL, PARA MATERIAIS NÃO METÁLICOS E NÃO AQUOSOS. 27
FIGURA 2.12 - ETIQUETAS UHF RÍGIDAS, PARA MATERIAIS METÁLICOS OU AMBIENTES HOSTIS. 27
FIGURA 2.13 - REPRESENTAÇÃO DO EFEITO DA ÁGUA NAS ONDAS DE RF 35
FIGURA 2.14 - REPRESENTAÇÃO DA OPACIDADE DO METAL E EFEITOS INDESEJADOS 37
FIGURA 4.15 - ESQUEMA DE ARQUITECTURA GENÉRICA DE UM SISTEMA RFID 50
FIGURA 4.16 - INTERFACES DE LIGAÇÃO COM O *MIDDLEWARE* 53
FIGURA 4.17 - DIAGRAMA DE ESTADO DE UMA ORDEM 71
FIGURA 4.18 - GRÁFICO DA EQUAÇÃO DE EXCLUSÃO DE ETIQUETAS. APENAS SÃO CONSIDERADAS VÁLIDAS AS ETIQUETAS CUJA CONFIANÇA SE ENCONTRE NA ZONA COLORIDA. 75
FIGURA 4.19 - DIAGRAMA DE CLASSES DE CONDIÇÕES 78
FIGURA 4.20 - DIAGRAMA DE CLASSES DE ORDENS 80
FIGURA 4.21 - DIAGRAMA DE CLASSES DE COMANDOS 81
FIGURA 4.22 - DIAGRAMA DE CLASSES DE PERMISSÕES 82
FIGURA 4.23 - DIAGRAMA DE CLASSES DE NOVOS LEITORES 84
FIGURA 4.24 - DIAGRAMA DE CLASSES DE GESTÃO DE LEITORES 86
FIGURA 4.25 - DIAGRAMA DE CLASSES DE ZONAS 87
FIGURA 4.26 - DIAGRAMA DE CLASSES RELATIVAS A CONFIANÇA 88
FIGURA 5.27 - ESQUEMA TÍPICO DA RECEPÇÃO E DESPACHO DE ENCOMENDAS NUM ENTREPOSTO 92
FIGURA 5.28 – FUNCIONAMENTO COM A INTRODUÇÃO DA TECNOLOGIA RFID 93
FIGURA 5.29 - EXEMPLO DE DEFINIÇÃO DE ORDEM 95
FIGURA 5.30 - EXEMPLO DE RELATÓRIO PARA A ORDEM DA FIGURA 5.29 95
FIGURA 5.31 - EXEMPLO DE DEFINIÇÃO DE ORDEM 96
FIGURA 5.32 - EXEMPLO DE RELATÓRIO GERADO PARA A ORDEM DA FIGURA 5.31 97
FIGURA A.33 - EXEMPLO DE CÓDIGO DE BARRAS 130
FIGURA A.34 - EXEMPLO DE *SMART CARD* 132

Índice de Tabelas

TABELA 2.1 - FREQUÊNCIAS MAIS COMUNS 30
TABELA 4.2 - EXEMPLO DE DISTRIBUIÇÃO DE ZONAS LÓGICAS 54
TABELA A.3 - FORMATO DA CODIFICAÇÃO SGTIN-96 121
TABELA A.4 - FORMATO DA CODIFICAÇÃO SSCC-96 121
TABELA A.5 - FORMATO DA CODIFICAÇÃO SGLN-96 122
TABELA A.6 - FORMATO DA CODIFICAÇÃO GRAI-96 122
TABELA A.7 - FORMATO DA CODIFICAÇÃO GIAI-96 123
TABELA A.8 - FORMATO DA CODIFICAÇÃO DOD-96 123
TABELA A.9 - FORMATO DA CODIFICAÇÃO GID-96 124

Acrónimos

EPC	Electronic Product Code
EPCIS	Electronic Product Code Information Service
HF	High Frequency
IP	Internet Protocol
ISO	International Standards Organization
LAN	Local Area Network
LF	Low Frequency
RF	Radiofrequência
RFID	Radio Frequency Identification
UHF	Ultra High Frequency
DoD	United States Department of Defense
IFF	Identify Friend or Foe
CMOS	Complementary Metal-Oxide Semiconductors
MIT	Massachusetts Institute of Technology
GPS	Global Positioning System
API	Application Programming Interface
USB	Universal Serial Bus
SDIO	Secure Digital Input Output
PDA	Personal Digital Assistant
IO	Input and Output
SGTIN	Serialized Global Trade Item Number

SSCC	Serial Shipping Container Code
SGLN	Serialized Global Location Number
GRAI	Global Returnable Asset Identifier
GIAI	Global Individual Asset Identifier
CAGE	Commercial and Government Entity
EUA	Estados Unidos da América.
ALE	Application Level Events
LLRP	Low-Level Reader Protocol
OCR	Optical Character Recognition
EAN	European Article Number
RUP	Rational Unified Process
UML	Unified Modelling Language
WMS	Warehouse Management System
HTTP	Hypertext Transfer Protocol
SOAP	Simple Object Access Protocol

1 Introdução

Muitas vezes, não é a necessidade que provoca a evolução. Muitas vezes, mesmo sem qualquer necessidade, o acaso, ou qualquer outro factor, faz com que boas ideias surjam. Pois este é precisamente o caso da tecnologia que se relaciona com o tema aqui em estudo, a RFID.

Aqui, o grande culpado foi, sem dúvida, o acaso. O acaso fez com que boas ideias surgissem, e com isto fosse apresentada uma tecnologia que, apesar de não ser recente, tem visto a sua evolução e possibilidades de utilização serem amplamente exploradas no decorrer dos últimos anos [10]. Na grande maioria das vezes, a sua utilização não é incluída em novas funcionalidades, mas sim, em novas, e mais eficazes, formas de realizar tarefas existentes [10].

De forma natural, o aparecimento de uma nova tecnologia, como é o caso da RFID, exerce pressão sobre a comunidade, de modo a que tecnologias igualmente inovadoras surjam também, com o intuito de suportar e dar a possibilidade de melhor a integrar em aplicações reais.

É exactamente neste campo que se desenvolve este trabalho, ou seja, numa nova aproximação à criação de uma ferramenta de suporte a esta tecnologia. Mais concretamente, a criação de um *middleware* RFID que possibilite a interligação entre esta tecnologia e os sistemas existentes que, de alguma forma, possam tirar partido da informação gerada por esta.

Neste capítulo é feita uma primeira contextualização do problema em estudo, de forma a enquadrar o desenvolvimento do tema.

Durante os próximos capítulos, será apresentada de forma mais aprofundada esta tecnologia, assim como as suas necessidades e pontos fracos. Para além

disto, e este será o ponto nuclear desde trabalho, poder-se-á também assistir aos desenvolvimentos que com este trabalho se atingiram.

1.1 Enquadramento

Numa altura em que o mercado começa a demonstrar um superior interesse pela tecnologia RFID, sendo esta, actualmente, a tecnologia que se perspectiva quando o objectivo é a identificação de objectos singulares, vários são os desafios que se vislumbram. Basta prestarmos alguma atenção àqueles que são, actualmente os principais impulsionadores da tecnologia, ou seja, o grande retalho [12, 15, 25], em conjunto com as empresas que a estes fornecem serviços tecnológicos.

Alguns destes desafios são de ordem económica, outros de ordem política e outros ainda prendem-se com as limitações técnicas e físicas da tecnologia [62].

Neste estudo em concreto, o desafio que se propõe discutir é uma nova aproximação à forma como os diversos componentes físicos, e respectivos *firmwares*, de uma estrutura RFID, ou seja, os seus leitores, antenas e etiquetas, se podem ligar a uma outra estrutura, desta feita composta de *software.*

A estrutura de *software*, que será quem na realidade tirará o proveito da informação obtida pela estrutura física, pode ser lógica, de dados ou apenas de apresentação, não devendo isso influenciar a sua forma de actuação. Uma vez que o objectivo da solução a desenvolver é o de interligar dois sistemas distintos, este terá a designação de *middleware*.

Apesar de o principal objectivo de um *middleware* ser o de interligar dois, ou mais, sistemas, muito mais há a dizer. Como será facilmente compreensível, se a sua única função fosse a interligação de sistemas, não só pouco traria a ganhar como ainda acrescentaria um acréscimo de operações desnecessário. Assim sendo, pretende-se adicionar alguma lógica ao *middleware* para garantir que os dados não só são entregues, como são entregues da forma melhor e mais conveniente. Com isto, pretende-se que o *middleware* aqui em estudo seja capaz

de filtrar a informação, de forma que apenas a informação relevante seja entregue ao destinatário. É ainda uma boa opção fazer com que este *middleware* não fique limitado a apenas alguns tipos de destinatário, sendo idealmente possível que a este seja indiferente onde vai entregar os dados. Da mesma forma, a origem dos dados não deve ser de forma alguma limitada, a não ser nos seus extremos de ligação. Para além do já referido, existe ainda uma questão que deve sempre ser levada em consideração, ou seja, a enorme quantidade de dados envolvidos num sistema RFID. Não será certamente difícil de imaginar um sistema em que milhares (senão milhões) de etiquetas estão presentes e onde estas são constantemente lidas. Assim sendo, o factor escalabilidade pode-se tornar crucial na determinação da qualidade de um sistema como aquele que aqui se pretende desenvolver.

1.2 Objectivos

Tal como o tema em estudo indica, o foco deste trabalho reside na elaboração de um *middleware* para RFID portável, genérico, o mais autónomo possível e que este seja eficazmente escalável.

O *middleware* é uma peça de software com a função de integrar e proporcionar a interoperabilidade entre dois ou mais sistemas distintos, sejam eles *hardware*, *software* ou ambos [63, 72].

Para o caso em estudo, e tal como anteriormente mencionado, a função do *middleware* será a de fazer a ponte entre os leitores (ou o seu *firmware*) e uma qualquer outra aplicação (*software*), de forma que esta possa eficazmente tirar partido da informação recolhida pela infra-estrutura RFID, seja a identificação ou a localização de objectos.

Naturalmente, de forma ideal, evita-se ter de reescrever um *middleware* de cada vez que este é necessário para uma nova aplicação. Assim sendo, este deve ser construído independentemente de uma qualquer aplicação específica, com a

preocupação de o tornar o mais portável possível e, com isto, ser facilmente adaptável a qualquer nova aplicação ou situação.

Da mesma forma que, numa extremidade do *middleware*, podem estar diversas aplicações diferentes e, por isto, temos de o tornar portável, na outra extremidade podemos também ter diferentes tipos de leitor. Assim sendo, este *middleware* não se poderá prender a apenas um protocolo de leitor. Para além de não se prender a um protocolo, deve também possibilitar a rápida introdução de um novo tipo de leitor, sem que sejam necessárias modificações estruturais. Isto vai definir a capacidade de adaptação do middleware.

Não sendo já tão fulcral, este *middleware* deverá evitar, sempre que possível, a interacção humana. O grande objectivo deste atributo é que se chegue a um ponto em que, por exemplo, tudo o que é necessário para a instalação de um novo leitor seja ligá-lo ao sistema. A partir daqui, todas as configurações e, genericamente, a inclusão do leitor no sistema, seriam feitas autonomamente. Da mesma forma que com os leitores, este deverá disponibilizar uma ligação que dê a possibilidade de criar e ligar *software*, este próprio, autónomo.

Como é óbvio, num cenário com centenas, ou talvez milhares, de leitores, não é concebível a existência de uma instância de *middleware* por cada um destes. O *middleware* deverá ser capaz de funcionar com uma instância apenas, que seja capaz de estabelecer a ligação entre todos os leitores e as aplicações em questão, sem perdas de performance notáveis. Isto vai definir a escalabilidade do *middleware*.

Uma vez que esta é uma tecnologia que apresenta ainda bastantes falhas em termos físicos [35, 62], tentar-se-á sempre, tanto quanto possível, fazer uma compensação ao nível do *middleware*, para que esta falhas possam, no nível superior, ter o menor impacto possível.

Em concreto, o desenvolvimento desta solução deverá permitir atingir os seguintes objectivos:

1. estudar e compreender a tecnologia RFID, as suas potencialidades e perceber o estado actual da tecnologia;

2. perspectivar o papel de um componente integrador, que forneça capacidade de comunicação com os componentes de um sistema RFID;
3. idealizar os conceitos para um *middleware* extensível e configurável, que siga os conceitos de portabilidade, generalidade, autonomia e escalabilidade propostos;
4. desenvolver o *middleware* idealizado, de acordo com as melhores práticas de desenvolvimento actualmente existentes;
5. Verificar a possibilidade de integração do *middleware* proposto com as ferramentas de retalho utilizadas pela Wipro Retail.

1.3 Colaboração com a Wipro Retail

Uma vez que, para o desenvolvimento deste trabalho, é necessário um profundo conhecimento da tecnologia, assim como meios de experimentação adequados, este será desenvolvido em conjunto, e colaborando directamente, com a Wipro Retail.

A Wipro Retail é uma empresa com mais de dez anos de experiência na implementação de sistemas informáticos em todas as diversas áreas do retalho.

Sendo os principais interessados nas potencialidades desta tecnologia, precisamente, os comerciantes da área do grande retalho, ou do retalho de valor, os resultados, e aquisição de conhecimentos, resultantes do desenvolvimento de um projecto como este serão uma mais valia para a Wipro Retail.

Como tal, este trabalho poderá contar com a total disponibilidade da Wipro Retail, tanto a nível de conhecimento como de meios, para que deste trabalho resulte não só a aplicação em causa, como tudo o que a envolve.

Convém referir que, todo o desenvolvimento deste trabalho será feito nas instalações desta empresa, onde se acordou ser o melhor local para ter acesso aos meios disponibilizados pela empresa.

1.4 Este Documento

Este documento está organizado em seis capítulos principais. Começando pelo seu enquadramento, passando pelos conceitos teóricos e técnicos fundamentais, o desenvolvimento da solução, conclusões a tirar, até às sugestões de trabalho futuro.

Os seis capítulos indicados são:

1.Introdução: Pretende contextualizar o trabalho, apresentando o seu enquadramento e delineando os principais objectivos. Expõe quais os pontos-chave que aqui serão tratados.

2.A Tecnologia RFID: Detalha o que é a tecnologia RFID em todas as suas componentes. Entendido como fundamental, este capítulo parte das definições basilares das arquitecturas subjacentes, e restantes objectos constituintes, passando pela análise de cada componente da arquitectura e das suas particularidades. Mantendo especial foco nos assuntos de maior interesse para este trabalho, passa também, de forma mais superficial, pelas diversas áreas que à tecnologia dizem respeito.

3.Estado da arte: Pretende fazer uma análise crítica daquilo que actualmente se pode encontrar no mercado, no que diz respeito a soluções de middleware RFID actualmente existentes.

4.Middleware RFID: O núcleo deste trabalho. Debate as razões que estiveram na génese das abordagens sugeridas na dissertação, os princípios e a fundamentação do modelo descrito. Detalha o modelo nos componentes

constituintes, e da restante arquitectura. Completa-se o capítulo com a descrição da implementação prática do trabalho desenvolvido, assim como o conjunto de considerações e especificações colhidas no decurso do trabalho e sobre a passagem do modelo à fase de implementação.

5.Discussão de aplicabilidade: Neste capítulo discutem-se perspectivas de aplicação prática deste trabalho, assim como alguns exemplos práticos de possibilidades de implementação que se perspectivam.

6.Conclusões e trabalho futuro: Descreve as principais conclusões do trabalho efectuado, nas abordagens defendidas, no desenvolvimento efectuado, e nas possíveis aplicações futuras do sistema apresentado. Contextualiza os resultados alcançados no momento actual das Tecnologias de Informação, no que respeita à possibilidade de implementação de um sistema semelhante. São abertas ainda as oportunidades de investigação futuras na mesma linha condutora defendida neste trabalho.

O trabalho completa-se com os anexos, onde serão acrescentados os documentos de cariz técnico detalhado que por motivos de facilidade de leitura não foram incluídos no corpo textual do documento principal, nomeadamente a modelação

2 Tecnologia RFID

Com este primeiro capítulo pretende-se apresentar a tecnologia que veio motivar este trabalho, assim como, de certa forma, as próprias motivações.

O capítulo começa com uma introdução à tecnologia RFID que tem como objectivo fazer uma contextualização desta. Aqui vamos abordar o nascimento e primeiras incursões práticas, o estado e utilizações actuais e, finalmente, serão apresentadas as expectativas de diferentes visões desta tecnologia.

Posteriormente, será feita uma abordagem mais prática. Aqui se fará também uma descrição mais técnica e funcional da tecnologia, passando ainda pelas diferentes formas com que esta tecnologia se apresenta e suas limitações.

Será feita também uma abordagem às questões e preocupações relacionadas com a segurança desta tecnologia.

Para finalizar, serão apresentadas outras tecnologias de identificação, assim como um comparativo entre estas e a RFID.

Não se pretende, a não ser quando tal se mostre pertinente, nesta fase, apresentar qualquer tipo de elação com respeito ao trabalho a desenvolver.

2.1 Introdução

2.1.1 Conceitos da Tecnologia

O acrónimo RFID significa *Radio Frequency Identification*, descrevendo desta forma todo e qualquer sistema de identificação onde um dispositivo electrónico, através da recepção/emissão de ondas rádio ou da variação de campo magnético, se consegue identificar, quando interrogado [62, 64, 65, 67].

A tecnologia RFID, tal como já mencionado, oferece, através de peças físicas de *hardware*, a possibilidade de automatizar o processo de identificação de um objecto. Através de mecanismos, do ponto de vista da implementação, simples, consegue-se que um objecto ou entidade, quando questionado sobre a sua identidade, responda com informação útil, sem que para isso seja necessária interacção humana ou contacto físico directo [62, 64, 65].

Os três elementos mais representativos e característicos de um destes sistemas são, sem sombra de dúvida, as etiquetas, as antenas e os leitores. São exactamente estes três elementos que distinguem, à primeira vista, um sistema RFID de um qualquer outro sistema electrónico. Adiante, estes três elementos serão mais aprofundadamente detalhados.

Para já, é suficiente saber que um sistema RFID se desenvolve, habitualmente, colocando uma etiqueta, com um identificador único, no objecto que se pretende identificar. Posteriormente, esta etiqueta pode ser interrogada por um leitor, através das suas antenas, de forma a de novo se obter a sua identificação.

Grande parte do crescendo de interesse que se tem verificado nesta tecnologia deve-se, na realidade, às recentes apostas de grandes corporações e entidades públicas nesta tecnologia.

Os dois maiores exemplos podem ser encontrados na gigante privada *Wal-Mart* e no Departamento de Defesa dos Estados unidos da América (DoD) [19, 62, 65, 66].

A *Wal-Mart* é, nem mais nem menos, do que a maior cadeia retalhista do mundo. Esta exigiu que, a partir do ano de 2005, os seus 100 maiores fornecedores colocassem etiquetas RFID em todos as encomendas, quer nas paletes quer nas caixas que tivesse como destino os seus entrepostos. Apesar disto, recentemente esta exigência começou a cair no esquecimento visto os seus maiores fornecedores serem de bens de baixo valor (*The Coca-Cola Company*, *The Gillette company*, *Procter&Gamble*, etc.), o que levava a um grande incremento considerável nos custos do produto final.

O DoD, tendo uma incrivelmente grande cadeia de fornecimento e um muito complexo sistema de logística, onde se incluem objectos de elevado risco (armas, explosivos, etc.) e com possibilidade de necessidade imediata, viu nesta tecnologia uma excelente oportunidade para mais facilmente manter o estado dos seus bens sob controlo. Como tal, tem vindo nos últimos anos a implementar em toda a sua cadeia de fornecimento, esta tecnologia.

Existe, no entanto, uma grande diferença entre estes dois gigantes, o *Wal-Mart* e o DoD. Enquanto o *Wal-Mart* decidiu que seriam os seus fornecedores a tornar os objectos identificáveis por RFID, o DoD decidiu que isto seria feito internamente. As diferenças nos resultados obtidos são visíveis, não estando, no entanto, no âmbito deste estudo, a análise do sucesso possível com uma solução que inclua esta tecnologia [65].

2.1.2 O Passado

Aquela que foi, possivelmente, a primeira incursão naquilo que é hoje o mundo do RFID, foi efectuada em 1948, quando Harry Stockman publicou um artigo com o título "*Communication by Means of Reflected Power*". Neste artigo, o autor afirmava que um longo caminho havia a percorrer até que todos os problemas na "comunicação por energia reflectida" fossem resolvidos, e a aplicação desta tecnologia fosse possível em larga escala.

Pouco tempo depois, no inicio da década de 1950, surgiu aquela que é considerada a primeira aplicação prática de um sistema RFID, apesar de não ter ainda essa designação. No final da Segunda Grande Guerra, os alemães descobriram que, quando os seus pilotos giravam os seus aviões no ar, o sinal que era reflectido para os radares voltava modificado. Este método, apesar de simples, seria o suficiente para informar os operadores dos radares sobre a identidade dos aviões que se aproximavam das suas bases e se estes seriam amigos ou inimigos [10, 62].

Este é então considerado o primeiro sistema RFID passivo. Do mesmo modo, e na mesma altura, na força aérea inglesa, sob o comando de Sir Robert Alexander

Watson-Watt, foi desenvolvido um sistema semelhante, sob a designação *Identify Friend or Foe* (IFF). A grande diferença deste para o sistema alemão, residia no facto de o sistema inglês incluir a utilização de transmissores eléctricos que, recebendo um sinal de radar, transmitiam um novo sinal, numa frequência específica, que os identificava como *Friendly* (amigos). Este é, por sua vez, considerado o primeiro sistema RFID activo.

Na década de 1960 começaram os estudos sobre as possibilidades de utilização da radiofrequência para a localização/identificação, mais notavelmente por R. F. Harrington, que publicou, em 1964, o resultado do seu estudo "*Theory of Loaded Scatterers.*" e *J. H. Vogelman*, que ainda em 1959, registou a patente denominada "*Passive data transmission techniques utilizing radar echoes*" [10, 62].

Foi também na década de 1960 que surgiram as primeiras empresas com o intuito de comercializar produtos que, na sua essência, eram já RFID. Estes artigos eram umas, relativamente pequenas, etiquetas electrónicas, com capacidade de armazenamento de apenas 1 bit, assinalando apenas se aquela se encontrava ou não activa. Estas etiquetas tiveram uma aceitação massiva, especialmente no comércio, onde eram utilizadas para evitar o roubo de produtos valiosos [62].

Ainda hoje se pode encontrar este tipo de etiqueta, mais comuns em lojas de vestuário.

Dado o futuro promissor desta tecnologia, logo na década seguinte, de 1970, laboratórios, empresas, universidades, entre outras, encontravam-se em plena fase de exploração da tecnologia. Ficou então esta década marcada como sendo, no que diz respeito a esta tecnologia, a década da exploração [61].

Com o aparecimento da tecnologia CMOS, ainda no final da década de 1960, a evolução das etiquetas teve um salto enorme, devido à capacidade de estas serem fabricadas com mais funcionalidade e com menor tamanho, contribuindo ainda para uma significativa redução nos consumos energéticos dos circuitos [10, 62].

Já nas décadas de 1980 e 1990 começou a massificação da tecnologia em utilizações comerciais. O exemplo da identificação dos animais, ou as primeiras experiências para cobrança de estradas com portagem são disso exemplo.

Foi ainda na década de 1990 que começaram a surgir as primeiras normas e regulamentação para esta tecnologia[10, 62]. Exemplo disto foi o aparecimento do *Auto-ID Lab*, no MIT, que foi posteriormente, já na década de 2000, substituído pela EPC Global [10].

2.1.3 O Presente

Actualmente, a tecnologia RFID encontra-se em plena expansão comercial. Existem já diversos fabricantes e fornecedores desta tecnologia, assim como as já referidas entidades e organizações públicas ou privadas.

Visto já se justificar uma produção em série destes componentes, o preço da tecnologia desce a um ritmo elevado.

Em termos tecnológicos, frequentemente aparecem novas soluções que vêm ao encontro das necessidades e que, muitas vezes, criam novas formas de exploração desta tecnologia.

A uniformização de normas e regulamentação a nível internacional começa também a tornar-se uma realidade[1].

Com tudo isto começa já a ser possível vislumbrar a concretização de alguns objectivos que, ainda recentemente, seriam por muitos considerados irreais.

Novos estudos surgem diariamente, na sua grande maioria, enaltecendo as vantagens que esta tecnologia trouxe [7, 8, 12, 15, 16].

A colocação de etiquetas ao nível da unidade é já uma realidade em alguns sectores, com especial foco nos objectos valiosos [16].

Mesmo nos objectos de pouco valor, como objectos típicos de mercearia, esta tecnologia é também usada. Neste caso, não ao nível da unidade, pois o preço ainda não o justifica, mas ao nível da caixa ou palete de unidades [62]. Soluções

[1] Mais informações em anexo

de localização utilizando tecnologia RFID, umas vezes integradas com outros sistemas, como GPS, outras vezes por si só, encontram-se também a ser comercializadas [62].

Contudo, e apesar de o futuro desta tecnologia se apresentar prometedor, existem ainda diversos factores que contribuem para um abrandamento do desenvolvimento da tecnologia, alguns de ordem mais politica, outros de ordem técnica e, ainda outros, de ordem económica. Este será um tema para discutir em maior detalhe mais adiante neste documento.

2.1.4 O Futuro

O sonho de qualquer pessoa que esteja, positivamente, ligada ao RFID, passa pela colocação de etiquetas RFID em todos os objectos/entidades à face da terra, ganhando a possibilidade de interagir com estes. Naturalmente que, dada a actual conjuntura, tal não passará, pelo menos a curto prazo, disso mesmo: um sonho [37].

Claro que, dadas as vantagens que esta tecnologia apresenta, quando comparada com outros tipos de tecnologia de identificação e localização de objectos, a inevitabilidade de implantação, faseada, desta tecnologia, é evidente.

Não querendo fazer futurologia, pegando naquilo que presentemente se encontra já implementado, o mais previsível cenário em termos de futuro é que, o tipo de objectos por onde esta tecnologia se começará, ou já começou, a espalhar serão os mais valiosos. Depois disto, prevê-se que começará a progredir no sentido decrescente da cadeia de valor, para outros objectos mais comuns [37, 65].

Não é provável, contudo, que esteja para breve a identificação ao nível do item.

Aqueles que apostam nesta tecnologia, tal como em muitas outras, podem-se distinguir em três categorias distintas: implementadores imediatos, implementadores seguidores e implementadores lentos.

No que a este capítulo, do futuro, respeita, os implementadores imediatos já pouco têm a dizer. Estes são aqueles que apostam nesta tecnologia tendo a

consciência que, assim que esta esteja mais madura, serão eles os primeiros a arrecadar os dividendos. São estes que já no presente, se encontram a implementar e utilizar esta tecnologia. Com isto, serão estes ainda que terão a oportunidade de influenciar as normas, ou de fazer com que a investigação nesta tecnologia caminhe num sentido que lhes seja mais favorável.

Os implementadores seguidores são aqueles que, vendo um grande futuro nesta tecnologia, e sabendo que, mais tarde ou mais cedo, terão de apostar nela, têm ainda algum receio que esta não seja ainda a melhor altura para entrar. Estes vão esperar, mantendo-se no entanto atentos para, em qualquer momento que tal se justifique, entrar com toda a força na tecnologia.

Finalmente, os implementadores lentos, tal como o nome indica, são aqueles que, apesar de muitas vezes encontrarem certas vantagens na tecnologia, não pretendem apostar nela até que esta esteja totalmente estável, tanto em termos de preço como de normas, e seja possível verificar ganhos reais em implementações de industrias semelhantes à sua.

2.2 Funcionamento da Tecnologia

Nesta secção apresentar-se-á, de forma mais aprofundada, a tecnologia subjacente à RFID, começa com os princípios de funcionamento desta, onde tal será apresentado de uma forma mais abstracta. Nesta interessa realmente perceber os efeitos da tecnologia, visto que as causas serão mais aprofundadamente apresentadas nas sub-secções seguintes.

Convém salientar que esta é uma secção mais direccionada ao nível de hardware e, não sendo esse o foco deste trabalho, que está mais voltado para o nível de software, será então uma secção mais superficial. Será aqui feita também uma primeira incursão no software, mais naquilo que é necessário para constituir um sistema RFID completo que na sua especificação ou análise de funcionalidade.

Para finalizar a secção, alguns problemas e limitações da tecnologia serão também analisados, problemas estes que, apesar de derivarem na sua maioria do nível de hardware, são importantes para no futuro percebermos algumas das opções tomadas.

2.2.1 Princípios de Funcionamento

Tal como foi já dito anteriormente, a tecnologia RFID tem duas funções principais. São elas as de identificar e localizar objectos de qualquer natureza.

A forma como isso é conseguido passa, invariavelmente, pela utilização de três componentes chave da tecnologia: o leitor, a antena e a etiqueta.

Como é óbvio, sem um software adequado, que trate a informação recolhida, estes três elementos seriam absolutamente inúteis. No entanto, o software não é um elemento característico desta tecnologia, mas antes uma espécie de catalisador, que potencia a utilidade desta. Assim sendo, vamos considerar, para já, que um sistema RFID é composto apenas pelos três elementos anteriormente mencionado.

Para começar, devemos colocar cada conceito no seu respectivo lugar. Tanto ao nível do hardware como de software (neste caso *firmware*), o cérebro de um sistema RFID é o seu leitor. Em última análise, é o leitor quem dá as ordens das acções a tomar, sejam elas ler, escrever, apagar, etc.

O meio utilizado pelo leitor, de forma que as suas ordens sejam cumpridas, são as antenas. Estas encontram-se directamente ligadas ao leitor, normalmente através de um cabo. Podem, no entanto, encontrar-se embutidas no próprio leitor.

Pode-se considerar que as antenas são a forma de um leitor comunicar, ou mesmo de se expressar. Quando um leitor pretende executar qualquer acção, é através destas que o faz.

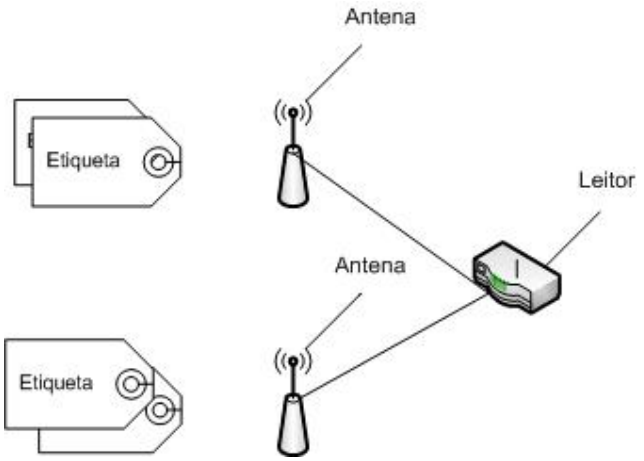


Figura 2.1 - Os três elementos de um sistema RFID

Finalmente, temos o ultimo elemento a ser apresentado que, apesar disto, não deixa de ser o mais importante de todos: as etiquetas. É através das etiquetas, que são colocadas nos objectos que pretendemos identificar, que conseguimos interagir com estes. São estas que possuem a identificação e que, quando questionadas por um leitor, através das suas antenas, respondem com a sua identificação.

Este esquema de funcionamento pode ser mais facilmente visualizado na Figura 2.1.

Temos então, de uma forma geral, que as etiquetas são colocadas nos objectos, e é apenas a estes que se encontram ligadas. As antenas encontram-se ligadas ao leitor, e apenas a este, e é através destas que o leitor comunica com as etiquetas. Ao acto de comunicar com as etiquetas, vamos chamar de interrogação, mesmo que muitas vezes não seja realmente uma interrogação o que esteja a suceder. Da mesma forma, para aquele que interroga, ou seja, o leitor, nem sempre os seus actos são de leitura.

Como foi também anteriormente referido, as principais utilizações desta tecnologia prendem-se com a identificação e localização de objectos. Para além disto, existe ainda uma conjugação destas duas utilizações, com a designação de rastreamento. A diferença entre o rastreamento de um objecto e a localização de

um objecto pode ser muito ténue. Na verdade, ambas nos indicam onde se encontra, em determinado momento, um objecto. Apesar disto, na prática, a localização é normalmente utilizada para encontrar objectos que se encontram imóveis (p.e. num armazém), enquanto o rastreamento é mais utilizados para localizar objectos que se encontram em trânsito (p.e. quando transportados).

Começando pela identificação, este serve exactamente o propósito de identificar um objecto, ou seja, aproximamos as antenas, tal como esquematizado na Figura 2.2, do objecto/etiqueta que pretendemos identificar, com o intuito de deste obter o seu código identificador.

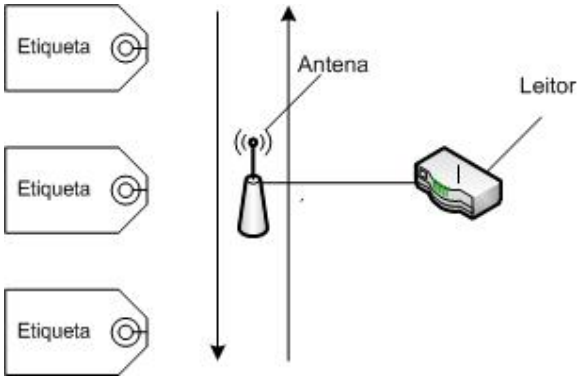


Figura 2.2 - Esquema de utilização - identificação

Tal como foi anteriormente mencionado, uma utilização comum desta tecnologia é a localização de objectos. Para tal, temos duas possibilidades distintas. A localização por antena e a localização por triangulação.

Como o próprio nome indica, a localização por antena consiste na localização de um objecto através da detecção do mesmo numa antena.

Colocando estrategicamente antenas, espalhadas por uma área, de tal forma que, preferencialmente, não interfiram umas com as outras, podemos identificar que, quando detectada por uma dessas antenas, uma etiqueta se encontra na área que a esta foi atribuída. O conceito é, na realidade, bastante simples e está esquematizado na Figura 2.3.

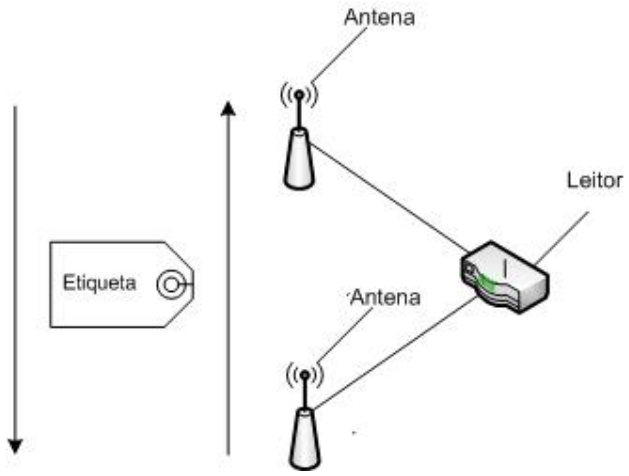


Figura 2.3 - Esquema de utilização - localização por antena

Apesar de a sua eficácia ser discutível, visto a localização de objectos ser, mesmo teoricamente, bastante imprecisa, este é, actualmente, o método mais seguro quando queremos localização.

Outro método possível, assegurando maior precisão, é a localização por triangulação.

Este método não é exclusivo desta tecnologia, existindo actualmente implementado em diversos tipos de redes de comunicações móveis, como as redes *WiFi* ou as redes de telemóvel.

Não querendo entrar em grande detalhe sobre a técnica, visto estar fora do âmbito deste trabalho, convém ter a noção que, para uma localização precisa, é necessário que as etiquetas a localizar estejam no alcance, a todo o momento, de pelo menos três antenas. Quanto maior o número de antenas a captar a mesma etiqueta, melhor a qualidade da localização. Podemos encontrar um esquema destes na Figura 2.4.

Apesar de, teoricamente, esta ser uma forma muito mais precisa de localização que a localização por antena, na prática, isto não se verifica, sendo esta extremamente falível dada a natureza sensível da tecnologia[2].

[2] A explorar na secção 2.4

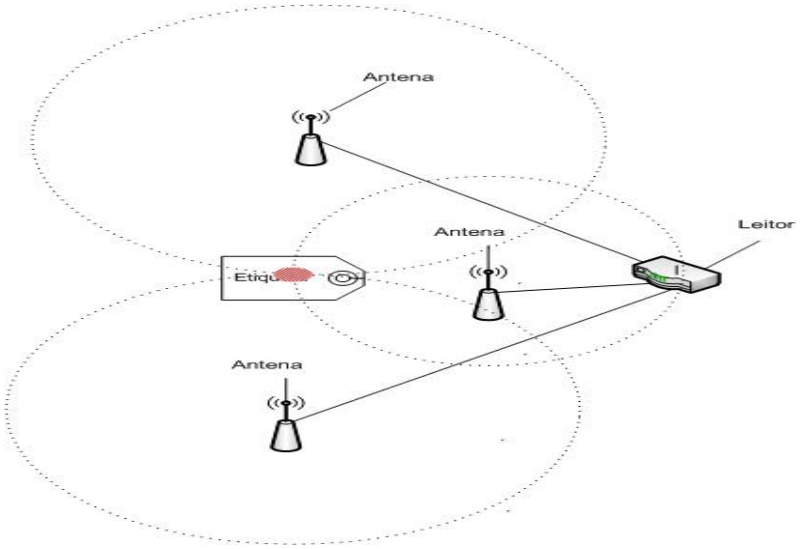


Figura 2.4 - Esquema de utilização - localização por triangulação

2.2.2 Principais Componentes

Vamos então agora passar ao aprofundamento de cada um dos componentes.

2.2.2.1 Leitores

Tal como foi anteriormente descrito, é ao leitor que cabe a tarefa de realizar as acções necessárias ou pedidas ao sistema.

Na Figura 2.5 podemos ver o exemplo de um leitor. Caracteristicamente, um leitor é, na verdade, um computador. Alguns possuem mesmo um sistema operativo como o *Windows XP* ou uma distribuição, específica ou genérica, do *Linux*.

Por norma, para além da necessária ligação de alimentação, estes têm ligações *ethernet*, que possibilitam uma maior flexibilidade na arquitectura da implementação, dando uso a tecnologias existentes e cuja funcionalidade é amplamente reconhecida.

É também normal encontrar num leitor, uma outra ligação, que poderá ter diversos formatos (*USB*, *RS-232C*, etc.). Esta ligação tem, normalmente, toda a funcionalidade que se consegue obter através da ligação ethernet, assim como

alguma funcionalidade mais, com o objectivo de facilitar a configuração do mesmo.

Existem ainda alguns leitores que oferecem a possibilidade de, a eles, ligar um comum monitor de computador.

Figura 2.5 - Exemplo de leitor e ligações comuns

Para além destas ligações, mais comuns na maioria dos sistemas, podem também encontrar-se os conectores destinados às antenas, tal como se pode ver na Figura 2.6.

Ainda na Figura 2.6, podemos encontrar uma ligação de cor dourada que, neste modelo de leitor específico, é utilizado para conectar uma antena que escuta o ambiente. Desta forma, é possível ao leitor fazer uma melhor gestão de colisões, visto passar este a ter o conhecimento das especificações das comunicações que se encontram, a cada momento, em desenvolvimento.

Figura 2.6 - Exemplo de ligações de antenas

Convém salientar que, apesar de isto ser bastante comum na maioria dos leitores, não é normativo. Ou seja, existem leitores de várias formas e para as mais diversas utilizações, pelo que o seu formato, ou o formato das suas ligações, podem divergir um pouco das Figuras 2.5 e 2.6 aqui apresentadas.

O exemplo da Figura 2.7 mostra um leitor RFID, com antena embutida, e cuja única ligação é SDIO.

Figura 2.7 – Exemplo de leitor SDIO com antena embutida

O tipo de leitor apresentado na Figura 2.7, é comummente utilizado em dispositivos móveis com uma entrada SDIO, como um PDA ou alguns telemóveis.

Para além das ligações de hardware que são oferecidas por um leitor, existem outras, muito mais comuns (senão mesmo generalizadas), que são oferecidas por um leitor. Estas ligações são compostas por *software* e existem normalmente na forma de uma API. É através destas APIs que é possível desenvolver peças de *software* capazes de interagir com os leitores.

Uma outra forma também disponibilizada pelos leitores que possuem uma ligação *ethernet* é a interacção com estes utilizando *sockets* e programação sobre estes.

Como se pode imaginar, as possibilidades são diversas. Dependendo da utilização pretendida para o leitor, existe uma extensa oferta que cobre grande parte das possibilidades.

Importa agora, depois de percebermos o que é um leitor RFID, perceber para que serve.

Normalmente, e apesar do que anteriormente foi dito, um leitor RFID não tem vontade própria. Ou seja, um leitor não é normalmente capaz de efectuar operações de IO por sua vontade, até porque tal não traria grandes vantagens. As ordens de leitura e escrita são dadas por uma peça de software externa, utilizando os meios disponibilizados pelo leitor, tal como anteriormente referido (APIs, *sockets*, etc.).

Da mesma forma, ao leitor pouco interessa aquilo que lê, ficando este apenas com a tarefa de cumprir a ordem que lhe é dada e, quando aplicável, responder a esta ordem.

Apesar de nem sempre se verificar, ao leitor nem sequer competiria ter uma noção de estado, uma vez que, tal como o nome indica, a este apenas compete ler e entregar à entidade responsável o resultado da sua leitura.

2.2.2.2 Antenas

É através destas que o leitor efectua as operações que lhe compete. As antenas são, por norma, dispositivos simples, sem qualquer lógica envolvida, constituídas apenas por um conector que possibilita a ligação a um leitor, e a antena em si, servindo esta para enviar os sinais RF recebidos do leitor e, posteriormente, receber os sinais RF reflectidos pelas etiquetas, de volta para o leitor.

Na Figura 2.8 temos um exemplo de antena, utilizada no leitor apresentado na Figura 2.6.

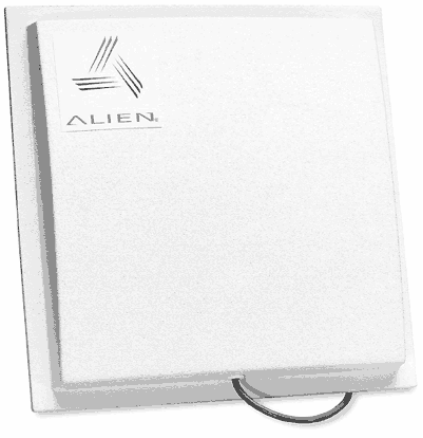


Figura 2.8 - Exemplo de antena

Tal como foi mencionado para os leitores, as antenas não terão, necessariamente, de ser como acima apresentadas.

A Figura 2.9 apresenta uma antena, já embutida no leitor, que é normalmente fixada em locais onde exista necessidade de alguma autonomia e modularidade, por exemplo, em veículos empilhadores.

Figura 2.9 - Exemplo de antena embutida em leitor

Um factor a não desprezar, apesar de ser de pouca relevância para este estudo, é o facto de podermos ter antenas de diferentes polarizações, tal como apresentado na Figura 2.10, sendo estas linear ou circular, podendo com estas obter diferentes campos e distancias de captura de etiquetas.

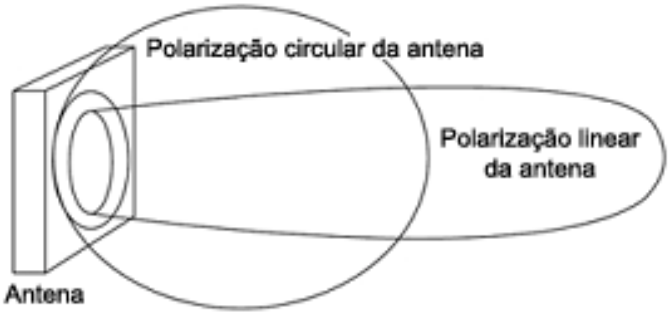


Figura 2.10 - Diferenças de alcance entre polarizações

2.2.2.3 Etiquetas

É nas etiquetas RFID que reside o principal foco, assim como as potencialidades e mais-valias, desta tecnologia.

Existem diversos tipos de etiqueta. Diversos tipos de frequências podem ser usadas nesta tecnologia. Entre elas, as mais comuns e, consequentemente,

normalizadas, são as LF, HF e UHF. Adiante, veremos quais as diferenças entre estas.

Para alem da frequência, pode também variar a forma como a resposta dada pelas etiquetas ocorre. Assim, podemos ter etiquetas activas, semi-passivas e passivas. Aquilo que difere entre estas etiquetas é o facto de algumas serem totalmente alimentadas e capazes de, mesmo sem receberem qualquer sinal, terem a iniciativa de comunicar para um leitor. Estas etiquetas designam-se etiquetas activas. Algumas destas etiquetas activas têm ainda a particularidade de serem capazes de efectuar processamento, incluindo algumas delas sensores ambientais, tal como termómetros, sensores de luminosidade, de pressão, etc.

Outras etiquetas, apesar de não terem a capacidade de, autonomamente, enviar o seu sinal, são também alimentadas. O motivo para tal é simples. Uma etiqueta que seja parcialmente alimentada tem a capacidade de amplificar o sinal que vai responder, quando questionada por um leitor, conseguindo com isto uma muito maior distância de leitura. Estas são as etiquetas semi-passivas.

Finalmente, as etiquetas passivas não têm qualquer tipo de alimentação própria. A energia que estas utilizam para responder, quando questionadas por um leitor, é a própria energia, gerada pelo campo magnético ou eléctrico criado pelo leitor através das antenas. Assim sendo, de certa forma, podemos considerar que estas etiquetas reflectem o sinal recebido do leitor, alterando-o de forma que este consiga identificar qual a etiqueta que reflectiu. Na Figura 2.11 podemos encontrar exemplos destas ultimas etiquetas, aplicadas em papel autocolante, mais comummente utilizadas em objectos de composição não metálica e não aquosa, como embalagens de cartão ou paletes de madeira.

Figura 2.11 - Etiquetas UHF de papel, para materiais não metálicos e não aquosos.

A Figura 2.12 apresenta mais dois exemplos de etiquetas UHF passivas, onde o microchip e as antenas se encontram resguardadas numa caixa rígida. Estas etiquetas são mais apropriadas, não só para ambientes mais hostis (como entrepostos ou transportes) como para objectos compostos de materiais metálicos ou aquosos, como bidões ou objectos de tara retornável.

Figura 2.12 - Etiquetas UHF rígidas, para materiais metálicos ou ambientes hostis.

A arquitectura característica de uma etiqueta é muito simples. Esta é composta unicamente por dois componentes principais: uma, ou várias, antenas e um pequeno chip. Grande parte daquilo que é visível numa etiqueta são as suas antenas, sendo o chip raramente maior que a cabeça de um alfinete. A estas antenas, de forma semelhante às antenas dos leitores, cabe a tarefa de recepção do sinal, a entrega deste ao microchip, que o transforma, e a sua posterior emissão. Uma vez que está também fora do âmbito deste trabalho, não nos vamos centrar na discussão sobre, por exemplo, o formato destas antenas, ou o material utilizado que, naturalmente, não é decidido ao acaso.

Outro aspecto das etiquetas, que aqui deve apenas ser abordado, mas que é mais detalhadamente pormenorizado em anexo, é a sua capacidade de armazenamento de dados, que se encontra normalizado na norma *EPC Tag Data Standard*. Esta norma, que trata da questão do armazenamento de dados em etiquetas, indica um armazenamento óptimo de 96 bits, extensível até 202 bits. Este conjunto de bits, quando codificado de acordo com a norma, tem a designação de código EPC. O valor atribuído a estes bits, segundo a norma, não é totalmente livre. A norma disponibiliza diversos formatos para várias utilizações distintas, sendo comum a todos, o facto de terem um cabeçalho de 8 bits, indicativos do formato escolhido.

Para além destes 8 primeiros bits, os restantes 88 poder-se-ão dividir, dependendo do formato, em código de companhia, código de produto, número de série, entre outras [52].

2.2.2.4 Software

Tal como anteriormente mencionado, aquilo que realmente caracteriza um sistema RFID são os três elementos anteriormente apresentados.

No entanto, e uma vez que os dados terão necessariamente de ter alguma utilidade, teremos de ter um sistema que seja capaz de, tal como foi já apresentado, receber estes dados num formato, quantidade e frequência que nem sempre são os ideais.

Dito isto, podemos considerar que uma outra peça fundamental para o bom funcionamento de um sistema RFID é o software que fica encarregue desta tarefa, ou seja, o *middleware* existente.

De facto, esta é a única peça de software que realmente teremos de ter em conta para um sistema RFID, pois é apenas até esta que, para praticamente todas as aplicações desta tecnologia, providencia um denominador comum.

A partir desta aplicação, existe já uma enorme diversidade de possibilidades de aplicação da informação recolhida, e dificilmente teremos uma peça de software, que seja superior a esta, inteiramente dedicada ao sistema RFID.

Assim sendo, a função do middleware num destes sistemas é a de fazer a ponte entre o baixo nível do firmware do sistema de leitores e o nível que realmente interessa, ou seja, o nível da aplicação que vai utilizar a informação recolhida.

Naturalmente que, dada a previsível dimensão de um sistemas destes, esta poderá não ser uma tarefa simples. Basta pensarmos num sistema de alguma dimensão, com algumas dezenas de leitores, cada um deles a ler algumas centenas de etiquetas dezenas de vezes por segundo, para termos um cenário em que a quantidade de etiquetas lidas, por segundo, ultrapasse as centenas de milhares.

Num cenário semelhante a este, o mais provável é que a percentagem de mudanças de estado, ou seja, o desaparecimento e aparecimento ou mudança de distância de etiquetas, em cada segundo, não ultrapasse uma parca percentagem. Com isto, seria um esforço inglório e inútil deixar esta informação ser tratada directamente pelo software que tira partido da informação.

Uma vez que o *middleware* é, precisamente, o foco deste trabalho, e que será desenvolvida uma grande quantidade de trabalho em torno deste, adiante veremos mais desenvolvimentos em volta deste assunto.

2.3 Caracterização das Etiquetas

Sendo a tecnologia RFID, tal como o próprio nome indica, desenvolvida em torno da radiofrequência, será esta certamente uma temática que valerá a pena abordar.

Uma vez que a área de física fica completamente fora do âmbito deste trabalho, vamos apenas referir brevemente as vantagens e desvantagens de cada uma das frequências características, e quais as implicações da utilização destas ao nível do sistema.

É de salientar que a caracterização aqui feita das diversas frequências se baseia na situação actual da tecnologia. Como é natural, a evolução desta, provocada

pelo passar do tempo e pela necessidade, certamente invalidarão algumas destas considerações no futuro.

Na Tabela 2.1 podemos encontrar as frequências mais comuns utilizadas nesta tecnologia.

Frequência (de referencia)	LF (125 kHz)	HF (13.56 MHz)	UHF (860 – 960 MHz)
Distância de Leitura	< 0.5 m	< 1m	< 10 m
Taxa de Leitura (teórico)	1 cod/sec	Aprox. 200 cod/sec	Aprox. 1000 cod/sec
Custo	Elevado	Médio	Baixo
Performance (água e metal)	Média	Baixa	Muito Baixa
Tamanho	Grande	Médio	Médio

Tabela 2.1 - Frequências mais comuns

Devemos, no entanto, ter a percepção que estes valores não são mandatórios, ou seja, praticamente qualquer frequência do espectro pode ser utilizada para esta tecnologia.

2.3.1 UHF

Tal como indicado na Tabela 2.1, a distância máxima, segura, utilizando a frequência de leitura UHF, será de aproximadamente 10m. Claro que isto estará dependente de várias condicionantes, como o ambiente em que está inserido. A quantidade de objectos metálicos influencia esta leitura de forma imprevisível, tanto podendo aumentar de forma indesejada o alcance, como diminuindo este.

Dependendo de diversos factores, como a distancia, a quantidade de ruído ou os materiais que compõe o ambiente, a taxa de leitura poderá atingir máximos teóricos de 1000 etiquetas por segundo.

O preço das etiquetas é um factor fundamental para o crescimento e eficaz alastramento da tecnologia.

Comparativamente, o preço das etiquetas UHF é baixo. Actualmente, é já possível adquirir etiquetas a preços inferiores a €0.10/etiqueta, prevendo-se que até 2010 atinjam preços inferiores a €0.05/etiqueta. Apesar disto, etiquetas especialmente desenvolvidas para ambientes desfavoráveis, ou para objectos de

composição desfavorável, rondam ainda preços relativamente elevados, na ordem dos €2.

Estudos demonstram que, mesmo com etiquetas especiais, a leitura em materiais desfavoráveis se revela altamente ineficaz, impossibilitando a sua leitura.

O tamanho das etiquetas que utilizam esta frequência é variável, mas geralmente médio, encontrando-se, com alguma facilidade, etiquetas a partir de $9cm^2$. Apesar disto, a tendência é que o tamanho deste tipo de etiqueta venha, a médio prazo, a ser reduzido.

2.3.2 HF

A frequência HF é, mais comummente, utilizada para leituras de proximidade sem contacto. Daí que as distâncias de leitura sejam dificilmente superiores a 1m, não oferecendo sequer vantagens o aumento desta distância.

Apesar da velocidade de leitura oferecida por esta frequência ser média, raras são as situações em que existe a necessidade real de taxas de leitura superiores a uma dezena de etiquetas por segundo. Contudo, este tem a capacidade teórica de oferecer taxas de leitura na ordem das centenas de códigos por segundo.

O custo das etiquetas que utilizam esta frequência situa-se, actualmente e para as mais económicas, nos €0.5, não sendo invulgar encontrar etiquetas com um custo igual ou superior a €2. A justificação para o preço relativamente elevado destas etiquetas prende-se com o facto de estas não serem projectadas para uma utilização massiva, não sendo tão urgente, como no caso anterior, uma drástica redução nos preços.

Quanto à performance, utilizando esta gama de frequências, e especialmente para os materiais considerados de risco, como a água ou o metal, os resultados são médios. A justificação para isto prende-se com motivos físicos, que fazem com que, quanto maior o comprimento de onda, e consequentemente mais baixa a frequência, mais facilmente esta consegue contornar objectos, sejam eles do material que forem.

O tamanho de uma etiqueta que utilize esta gama de frequências, considera-se de tamanho médio pois, não ultrapassando 10cm^2, dificilmente se poderá tornar mais pequena.

2.3.3 LF

Este tipo de frequência é por norma utilizada para identificação por contacto. Aquilo que a diferencia de outras tecnologias é a não necessidade de, uma etiqueta que utilize esta tecnologia, estar realmente visível.

Se pensarmos, por exemplo, na identificação subcutânea de animais, rapidamente vemos qual poderá ser uma das utilidades desta frequência.

Uma vez que a identificação de etiquetas com esta tecnologia é feita por contacto, não existe a necessidade desta ser capaz de obter taxas de leitura superiores a 1 código por segundo.

Quanto ao custo, por motivos semelhantes às etiquetas HF, é elevado. Outra agravante é o facto de ainda não ser possível utilizar, tal como nas restantes, circuitos impressos, fazendo com que o preço destas etiquetas continue muito elevado. Actualmente, podem-se encontrar etiquetas que utilizem esta gama de frequências a partir de €1.5, apesar deste valor depender bastante da utilização que lhe pretendemos dar.

Das frequências até agora apresentadas esta é, sem dúvida, a que melhores resultados obtém. A classificação atribuída na Tabela 2.1, média, prende-se com o facto de esta ainda assim não apresentar uma taxa de sucesso de leituras completamente confiável.

Em termos de tamanho, a classificação atribuída às etiquetas que utilizam esta frequência é grande. Apesar de estas serem, seguramente, as etiquetas mais pequenas de todas em termos de área, ao contrário das outras, nestas temos de considerar também o volume.

Uma vez que não podem, ainda, ser impressas, estas etiquetas provocam uma saliência que, dependendo da utilização em causa, pode impossibilitar o seu uso.

2.3.4 Comparação das Caracterizações Apresentadas

Não é possível dizer de forma segura qual destas tecnologias é a ideal. Todas elas oferecem vantagens, dependendo da funcionalidade que pretendemos.

Quando o que se pretende é uma identificação massiva de objectos o mais indicado será, seguramente, a utilização da frequência UHF.

Comparando com as restantes, utilizando a UHF, poderemos ter uma menor quantidade de leitores e antenas que, a maior distância, identificam uma grande quantidade de etiquetas. Podemos pegar no exemplo de grandes entrepostos comerciais, onde conceitos como o controlo de inventário ou localização são de extrema importância, mas, em oposição a isto, a quantidade de objectos a controlar é gigantesca.

Apesar de a taxa máxima teórica de leitura da HF ser de aproximadamente 200 códigos por segundo, na prática, será extremamente complicado ter 200 objectos no raio de alcance de uma antena, que é, em condições próximas do ideal, de 1m. Naturalmente que, dado o tamanho de uma etiqueta HF, não será possível identificar objectos de reduzidas dimensões.

Quando o pretendido é, com facilidade e rapidamente, identificar uma grande quantidade de objectos, um de cada vez, então a HF será a frequência mais aconselhável. Várias utilizações estão já implementadas na realidade, como por exemplo, na identificação de entradas em transportes públicos. Nestes, é possível obter permissão de entrada sem tirar sequer o cartão identificador da carteira, bastando que a aproximemos do leitor.

Quando o que pretendemos é a identificação de objectos em ambientes pouco favoráveis, onde a presença de metais ou água seja uma constante, então o mais aconselhável talvez seja a utilização da LF. Esta, independentemente do material em que se encontra, tem, comparativamente com as restantes, uma grande probabilidade de ser eficazmente lida.

O exemplo da identificação de animais é, para esta frequência, o mais elucidativo. Apesar de as etiquetas terem a possibilidade de serem colocadas debaixo da pele dos animais, em alguns aplicações, em peixes, a sua leitura é

quase sempre conseguida. Com as outras frequências em comparação, tal poderia muitas vezes não ser possível.

Com isto conclui-se facilmente que não haverá uma convergência para uma só frequência. Teremos, alternativamente, a convivência destas três frequências, adaptando-se e especializando-se cada uma delas em aplicações distintas.

Neste capítulo apenas se apresentam estas três gamas de frequência. Convém, no entanto, ressalvar a existência de muitas outras. Algumas delas em utilização e com bastante sucesso, como é o caso da gama de frequências micro-ondas. Estas são, no entanto, as frequências mais prometedoras em termos de um futuro relativamente próximo para esta tecnologia.

2.4 Limitações tecnológicas

Tal como foi já por diversas vezes mencionado ao longo deste documento, a tecnologia aqui em estudo apresenta algumas limitações tecnológicas que são, a curto/médio prazo, de difícil resolução.

Uma vez que também isto terá um impacto considerável no trabalho em desenvolvimento, é agora uma boa altura para aprofundarmos um pouco mais esta temática.

Aqui ficar-se-á a saber, ao certo, o porquê, assim como as principais implicações, de certos materiais terem um efeito indesejado nas leituras de etiquetas.

2.4.1 O problema da água

Um dos problemas comuns, e com o qual é necessário ter especial cuidado aquando da planificação de um sistema RFID, é o da água.

Por motivos de ordem física, quanto mais alta a frequência de uma onda de radiofrequência, maior é a absorção desta pela água [35, 62].

Para termos uma ideia da diferença que pode fazer a frequência no grau de absorção da água, a taxa de absorção da água numa frequência baixa, de 100KHz, é 100.000 vezes inferior à taxa de absorção obtida numa alta frequência de 1GHz. Isto significa que, enquanto em baixas frequências, a taxa de absorção é quase nula, em frequências muito altas, a absorção é quase total.

Na Figura 2.13 podemos ver uma representação, mais perceptível, daquilo que acontece quando uma onda de radiofrequência encontra água. Como se encontra representado, ao encontrar água (ou materiais aquosos), o feixe de rádio-frequência perde grande parte, senão toda, a sua potência. Isto vai, em grande parte dos casos, impossibilitar a recepção deste sinal pelas etiquetas, assim como a possibilidade de resposta destas.

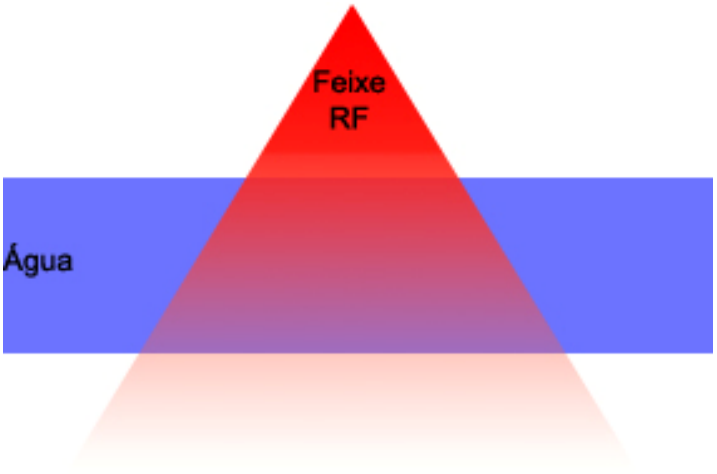


Figura 2.13 - Representação do efeito da água nas ondas de RF

Este aspecto terá um enorme impacto na planificação de um sistema RFID, visto a água ser um componente quase omnipresente no nosso ambiente.

Basta pensarmos num retalhista, que poderia ter, num futuro próximo, todo o interesse em colocar etiquetas em todas as unidades de produto das suas lojas. Isto incluiria, por exemplo, garrafas de água, ou fruta, cuja composição é maioritariamente água. Um problema como este pode impossibilitar, de todo, a implementação de tal sistema.

Infelizmente, as soluções actualmente possíveis para tal problema, não são muito animadoras. Algumas destas soluções, que de facto funcionam, utilizam etiquetas de um material especial que, sendo atingido por ondas rádio, por muito fracas que sejam, as vai acumulando até conseguir energia suficiente para emitir uma resposta visível [35]. Apesar de funcionar, esta estratégia aumenta o tempo de resposta das etiquetas, levando a que em certas situações não se consiga obter a informação desejada em tempo útil.

Outra solução passa por utilizar etiquetas especiais, onde as suas antenas e microchip se encontram separadas, através de materiais como o esferovite, do objecto em que se encontram colocadas, no espaço de alguns milímetros [35]. Isto será suficiente para tornar estas etiquetas completamente funcionais. No entanto, o preço destas é muito superior ao das etiquetas comuns.

2.4.2 O problema do metal

O metal, em maiores proporções que a água, provoca interferência num sistema RFID. Enquanto a água tem a particularidade de ser trespassável por ondas de radiofrequência, apesar de esta perderem potencia, o metal é completamente opaco a estas ondas.

Facilmente chegamos à conclusão que, colocando um objecto metálico no raio de acção de uma antena RFID, estamos a criar pontos cegos. Colocando nestes uma etiqueta, será virtualmente impossível obter leituras. Mesmo etiquetas colocadas entre uma antena e um objecto metálico, quando esta se encontra muito perto ou mesmo junta ao objecto, pode fazer com que o número de leituras correctas seja praticamente nulo. Na Figura 2.14 podemos observar uma representação deste fenómeno.

Uma das soluções apresentadas para a água está, actualmente, a ser estudada para aplicar no caso do metal. Esta seria a utilização de um material neutro, como a esferovite, capaz de manter uma certa distancia entre o objecto e a etiqueta, que fosse suficiente para proporcionar algumas leituras válidas. Com

isto, temos o mesmo problema encontrado no caso da água, ou seja, a subida do preço das etiquetas.

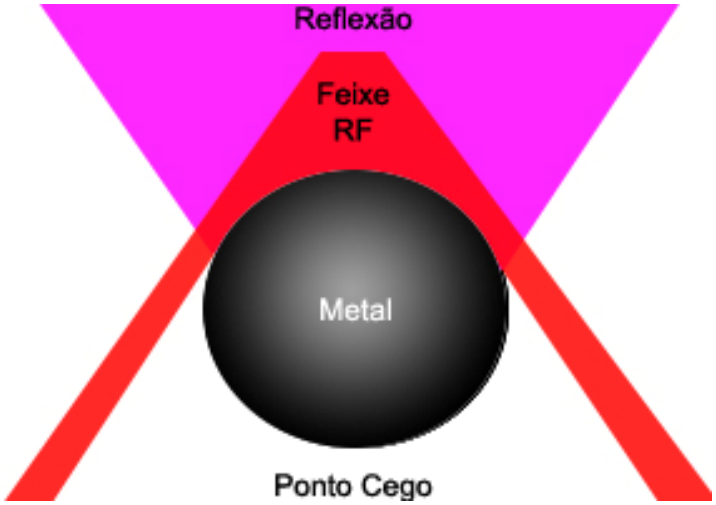


Figura 2.14 - Representação da opacidade do metal e efeitos indesejados

Outro problema, também provocado pelo metal, especialmente em aplicações RFID utilizadas para localização, é o facto de, em ambientes onde o metal seja abundante, existir uma grande quantidade de ruído provocado pela reflexão, e muitas vezes distorção, do sinal por parte do metal [62].

Em certas aplicações obtém-se leituras de distâncias muito superiores às desejadas, ou então o contrário [35].

3 Estado da Arte

Neste capítulo será feita uma análise das soluções, semelhantes àquela que aqui se encontra em estudo e desenvolvimento, actualmente existentes no mercado. Esta análise será tão detalhada quanto possível, dado que certos pormenores técnicos das soluções aqui em análise não são de domínio público. Assim sendo, esta análise basear-se-á em estudos publicados [38, 72], cujo público-alvo seriam empresas interessadas numa implementação desta tecnologia. Desta forma, será também possível ficar com uma perspectiva geral daquilo que é vulgarmente esperado de uma solução deste género.

Para a análise dos diversos sistemas de *middleware* aqui em discussão, tomam-se como passos de um processo característico os seguintes [72]:

- obtenção de dados gerindo os leitores RFID
- enriquecimento de dados para uso posterior
- entrega destes dados enriquecidos aos sistemas interessados

3.1 Tipos de Envolvimento

Existem diversas formas como qualquer empresa fornecedora, produtora ou integradora de software se pode encontrar ligada a esta tecnologia. Dependendo do seu alvo de mercado, a sua actuação e oferta pode variar num ângulo de acção bastante amplo. Será então, neste capítulo, apresentada a forma como os principais intervenientes que, actualmente, se podem encontrar no mercado, escolhem envolver-se com esta tecnologia[38].

- **Fornecedores dedicados**

Este tipo de empresa caracteriza-se pela sua dedicação total e única à tecnologia RFID.

As empresas que se encontram nesta classe são empresas que, tradicionalmente, emergiram das primeiras experiências piloto realizadas pelo *Auto-ID Center*. Outra característica destas é a sua participação activa no desenvolvimento de normas para esta tecnologia.

Alguns fornecedores aos quais se pode atribuir esta classificação são: a *ConnectTerra*, *GlobeRangers*, *OATSystems*, etc.

- **Fornecedores de aplicações**

Nesta classe encontram-se os fornecedores que, atendendo às regras impostas pelos grandes clientes, se foram, inadvertidamente, especializando nesta tecnologia. Como tal, aquilo que inicialmente não passava de um requisito ou característica das suas soluções, passou entretanto a ter um papel de destaque na sua oferta.

Alguns destes fornecedores escolheram construir as suas próprias soluções, de forma a incluir estas na sua actual oferta. Outros decidiram facilitar a integração de soluções já existentes no mercado, estreitando as relações com fornecedores dedicados.

Alguns exemplos de fornecedores com estas características são: *Provia Software*, *Manhattan Associates*, *RedPrairie*, etc.

- **Grandes fornecedores de plataformas**

Pode-se classificar com esta classe os fornecedores de grande dimensão, que fornecem grandes soluções e plataformas integradas de gestão. Nesta classe incluem-se empresas como a *Sun Microsystems*, a *IBM*, a *Microsoft*, etc.

A participação destas empresas nesta tecnologia caracteriza-se pela forte aposta em grandes soluções, com elevada escalabilidade e capacidade de integração. Muitos dos grandes desenvolvimentos de arquitectura de sistemas de software para esta tecnologia emergem de um grande investimento destes fornecedores.

- **Especialistas em integração**

Alguns especialistas em integração, como sejam a *webMethods*, a *TIBCO Systems* ou a *Ascential Software*, fazem, de forma semelhante aos grandes fornecedores de plataformas, grandes apostas em métodos de integração desta tecnologia com os sistemas actualmente existentes. O facto de, tradicionalmente, estes fornecedores apresentarem uma grande experiência com a integração de soluções que lidam com grandes volumes e fluxos de dados, dá-lhes uma vantagem no que concerne à integração de sistemas RFID.

3.2 Critérios de Avaliação

Para uma boa análise das diferentes soluções, alguns critérios devem ser definidos à partida. Aqui, os critérios analisados em cada um dos sistemas em discussão são os seguintes [72]:

- Escalabilidade: um incremento exponencial da quantidade de informação que simultaneamente chega dos leitores implica que o *middleware* tenha uma boa capacidade de balanceamento de carga.
- Normas: a utilização das normas por parte do *middleware* simplifica, não só os *upgrades* correctivos e evolutivos, como também a migração e escalamento de uma arquitectura já existente.
- Nível de processamento e enriquecimento de dados: para alem da recolha de dados, cabe também ao *middleware* o enriquecimento e filtragem destes. Um parâmetro a ter em consideração é, por exemplo, se determinado *middleware* é suficientemente configurável para que seja possível manipular a forma como os dados são entregues nos sistemas de destino.
- Partilha de funcionalidade do sistema: um *middleware* deve ser adaptativo, tanto quanto possível, uma vez que um dos seus desafios poderá ser a entrega de dados em sistemas completamente distintos, em localizações completamente díspares.

- Facilidade de integração: é obvio que não seria viável a reconstrução dos sistemas actualmente existentes nos interessados em implementar esta tecnologia. Assim sendo, deverá ser dada uma forma simples de proceder à integração de um sistema deste tipo com os sistemas que actualmente existem.
- Possibilidade de Customização: dada o amplo universo de potenciais utilizadores de uma solução deste tipo, e de certa forma complementando o anterior ponto, deve ser possível não só o *middleware* ser de fácil integração como também ser customizável para a obtenção daquilo que deste se pretende.

3.3 Sistemas existentes

As principais criticas e falhas dos sistemas existentes residem na sua fraca escalabilidade e dificuldades de integração que apresentam [38]. A grande maioria dos fornecedores de soluções aposta em sistemas com os quais se é capaz de fazer uma integração com as ofertas actuais. Isto leva a que, muitas vezes, seja de extrema complexidade, e muitas vezes impossível, a integração com sistemas ou processos de negócio mais específicos.

O exemplo do *SAP*, que é apontado como um dos fornecedores que mais fortemente tem investido em soluções para esta tecnologia, é a este título revelador [38]. Apesar de possuírem soluções capazes de suportar grandes cargas de trabalho, estas encontram-se ainda demasiado dependentes, estando muitas vezes completamente embutidas, nas soluções destinadas à cadeia de distribuição e logística.

Já no caso de fornecedores como a *Oracle*, a *IBM*, ou mesmo a *Microsoft* soluções prontas a integrar estão disponíveis para diversas plataformas. Apesar disto, em qualquer uma destas soluções existe um denominador comum, ou seja, todas elas oferecem soluções de integração desta tecnologia, mas nenhuma o oferece em exclusivo. A oferta destes fornecedores centra-se em plataformas de

integração, onde apenas um dos módulos dessa plataforma (normalmente dispendiosa e com um peso computacional elevado) é dedicado a esta tecnologia. No fundo, a aplicação das soluções destes fornecedores é realmente compensatória para quem: ou tem já uma plataforma desse fornecedor implementada, sendo apenas necessária a inclusão de um novo módulo, ou tem mais do que uma motivação para a implementação de uma dessas plataformas [72]. Nesta secção, vamos dar especial foco às duas principais soluções existentes, assim como aquelas em que mais informação existe disponível, sendo elas a solução *SAP* e a solução *Oracle*.

3.3.1 SAP

A *SAP* oferece uma solução integrável com os seus sistemas, denominada *SAP Auto-ID-Infrastructure*. Abstractamente, a solução SAP apresenta uma arquitectura de 4 extremidades, sendo elas as seguintes:

1. *Device Layer (DL)*: extremidade de comunicação com dispositivos de leitura/escrita (leitores, impressoras, etc.);
2. *Device Operation Layer (DOL)*: extremidade de gestão do sistema;
3. *Business Process Bridging Level (BPBL)*: extremidade que trata da negociação entre o *middleware* e o software interessado na informação (administração, customização, entre outras.);
4. *Enterprise Application Layer (EAL)*: extremidade que efectivamente trata da conexão entre o *middleware* e as aplicações.

Começando pelas duas primeiras extremidades, elas vão funcionar, naturalmente, em conjunto. O *DOL* tem, ao seu dispor, diversas instâncias de *DL*, estando cada uma destas responsável por um ou mais dispositivos. A cada uma das instâncias de *DL* está associado um *Device Controller* (*DC*), que tratará da comunicação efectiva com os dispositivos.

Um *DC* poderá ter dois modos de operação, o modo assíncrono, em que este espera por eventos vindos do dispositivo (como uma leitura, uma mudança de

temperatura, etc.), e o modo síncrono, em que o *DC* recebe ordens directas do *middleware*, e que transmite aos seus dispositivos.

Posteriormente, uma vez tendo dados para entregar, o *DL* passa-os ao *BPBL*, a quem compete o enriquecimento destes, recorrendo a regras, para posterior entrega ao EAL que se encarregará de os depositar no seu destino. Existe no *BPBL* um *Rules Engine* (*RE*), onde é possível predefinir as regras de tratamento dos dados, e que trata também da gestão e hierarquização destas regras [71].

Percorrendo agora os critérios sugeridos para a avaliação das soluções, temos o seguinte:

- Escalabilidade: O sistema oferecido pela *SAP* proporciona diversas formas para escalar todo o sistema. Uma das formas possíveis é conjugar diversos *DL*, de forma que estes usem um mesmo conjunto de regras no *BPBL*. É também possível juntar diversos *DL* e respectivos *DC* no *DOL*, para que o seu processamento não seja efectuado separadamente, conseguindo com isto alguma poupança de recursos. De forma a reduzir o tráfego na rede, tradicionalmente gerado por um sistema deste tipo, um primeiro controlo de erros pode ser inserido no primeiro nível de abstracção deste sistema. Assim, um controlo mais grosseiro poderá opcionalmente ser adicionado ao *DL*.
- Normas: Quanto ao cumprimento das normas, a *SAP* anunciou que todas as normas *EPC-Global* estão a ser seguidas. Apesar disto, a sua arquitectura não aponta para tal.
- Nível de processamento e enriquecimento de dados: O enriquecimento de dados é feito, tal como já referido, recorrendo a regras. Estas regras são geridas pelo *RE*, a quem compete organizar e hierarquizar a forma como estas são aplicadas. É ainda possível adicionar aos dados recolhidos, meta-dados provenientes de sistemas externos.
- Partilha de funcionalidade do sistema: De forma a partilhar a arquitectura do sistema, várias instâncias do mesmo sistema podem ser postas a correr paralelamente, mantendo-se simultaneamente uma

integração entre estas. Como tal, cada uma destas pode ser configurada para que se aplique a um sistema específico.

- Facilidade de integração: Teoricamente, existem diversas formas de integrar este sistema com outros sistemas existentes. Ferramentas tanto proprietárias da *SAP* como livres (Java, .NET) são disponibilizadas de forma a facilitar o desenvolvimento de soluções de integração.
- Possibilidade de Customização: Em termos de customização, todo o sistema é configurável através da interface administrativa, sendo ainda possível a adição de módulos desenvolvidos para efeitos específicos.

Em conclusão, apesar deste módulo ter sido desenvolvido com o intuito de se tornar independente da arquitectura do sistema onde se integra, tal nem sempre é conseguido. O ponto mais negativo é o facto de não existir explicitamente nenhum suporte para protocolos de comunicação ou integração independentes da plataforma em q este se insere. Apesar disto, quando o que se pretende é uma integração com soluções *SAP*, esta solução atinge os seus objectivos [72].

3.3.2 Oracle

A *Oracle* fornece uma solução mais modular que a anterior. Esta solução, de nome *Oracle Sensor Edge Server* (*OSES*), é um modulo de uma *framework* mais abrangente, a *Oracle Sensor-Based-Services*, direccionada ao processamento de dados provenientes de sensores.

A *OSES* é distribuída em duas variantes diferentes, a *EPC Compliance Enabler* e a *RFID Pilot*. Como os próprios nomes indicam, a primeira é recomendada para sistemas que pretendes seguir as normas EPC, enquanto a segunda é mais genérica, e portanto recomendada para sistemas mais específicos [71].

Esta solução apresenta uma arquitectura de três níveis:

1. *Device Driver Layer* (*DDrL*): que trata da gestão dos dispositivos de leitura e escrita (sejam leitores, impressoras, pilhas de luzes, ou outros).

2. *Data Processing Layer* (*DPL*): que trata da limpeza e uniformização dos dados recolhidos pelo *DDL*. A este nível, ainda não é feito o enriquecimento dos dados.
3. *Data Dispatching Layer* (*DDiL*): que trata da entrega da informação aos sistemas existentes. Existe também, aqui, um armazém de dados, onde estes são guardados para entrega posterior caso uma ligação não se encontre activa.

Todas estas camadas são geridas por um componente externo, de nome *Enterprise Manager*.

Cada uma destas camadas está construída como uma framework de desenvolvimento de plugins que, customizaveis recorrendo ao *Edge Developer Kit*.

O *DDrL* encontra-se implementado para ser uma *framework plug-and-play*, conseguindo-se com isto ter alguma facilidade de implementação e adição de novos dispositivos. Este tráz já algumas implementações para alguns dos dispositivos mais comuns.

Quanto ao *DPL*, este contém uma série de mecanismos predefinidos para limpeza e formatação dos dados. Neste nível, diversos agrupamentos lógicos podem ser concretizados, por exemplo, fazer com que todos os leitores das entradas norte de um armazém sejam tratados como um leitor apenas.

É através do *DDiL* que a entrega dos dados anteriormente tratados é feita. Este disponibiliza diversas formas de comunicação, como *HTTP*, *SOAP*, *Java Message Services* e *Oracle Streams*. Como em todos os níveis, novos *DDiL* podem ser criados e incluídos para aplicações específicas.

Avaliando agora esta solução segundo os critérios acima sugeridos, temos:

- Escalabilidade: Em termos de escalabilidade, esta é uma solução mais completa que a anterior. Uma vez que a computação paralela em *GRID* é suportada nativamente [57], varias instancias desta solução podem ser

conjugadas. É também possível, tal como foi já referido, agrupar leitores de forma que os seus dados sejam tratados conjuntamente.

- Normas: Quanto ao cumprimento de normas, uma vez que uma das distribuições é especifica para soluções que sigam as normas EPC-Global, esta tem a capacidade não só de consumir como gerar dados de e para etiquetas, cumprindo grande parte das normas sugeridas [57]. É ainda de salientar a utilização de formas de comunicação normalizadas e amplamente utilizadas, onde os já referidos *SOAP* e *HTTP* têm grande representatividade.
- Nível de processamento e enriquecimento de dados: Tratando da capacidade de processamento e enriquecimento de dados, este é completamente aberto a possíveis customizações, fornecendo já no entanto algumas soluções pré-formatadas. É ainda importante referir a possibilidade de adicionar e facilmente integrar um repositório de dados, onde se poderão armazenar meta-dados posteriormente utilizados para o enriquecimento dos dados recolhidos pelos leitores.
- Partilha de funcionalidade do sistema: Existem, no que à partilha de sistema concerne, duas formas de a concretizar. A primeira delas, não sendo contudo uma forma directa de o fazer, é ainda possível obter resultados semelhantes, recorrendo à já referida computação em GRID. Apesar de as três camadas terem de correr na mesma máquina lógica, esta pode ser um sistema GRID [57], acabando por ser possível a dispersão do sistema. Outra forma de o conseguir é recorrendo à partilha dos dados, tanto através das suas interfaces de comunicação, como de um repositório de dados, ou ainda uma conjugação de ambas as soluções.
- Facilidade de integração: Uma vez que esta solução suporta a comunicação utilizando meios bastante conhecidos e normalizados, a integração com soluções existentes, mesmo sendo soluções de difícil integração, torna-se possível. No entanto, o facto de ser possível, não implica que seja simples.

- Possibilidade de Customização: Para a customização, e como foi já referido, a própria arquitectura foi feita a pensar nesta possibilidade. Uma vez que todas as camadas lógicas da aplicação foram construídas para se tornarem, elas próprias, frameworks de desenvolvimento de *plug-ins*, fornecendo para isso o *Edge Developer Kit*, a customização e adição de novas funcionalidades torna-se uma grande mais-valia para esta solução [72].

Concluindo, esta solução apresenta duas grandes vantagens: o suporte de interfaces de comunicação de uso comum e a capacidade de modelação de uma solução customizada, partindo da solução *standard* fornecida.

Ainda assim, um factor negativo a apontar a esta solução, tal como no caso da anterior, é a aposta pouco abrangente no cumprimento das normas disponíveis para esta tecnologia.

3.3.3 Comparação das soluções SAP e Oracle

Torna-se aqui indispensável, depois desta análise, a existência de uma conclusão comparativa entre as duas soluções apresentadas. Em termos gerais, ambas as aplicações são semelhantes na sua oferta.

Ambas as soluções fazem uma aposta clara nas possibilidades de integração, sendo, no entanto, essa aposta mais visível na solução oferecida pela *Oracle*. Quanto às possibilidades de customização, estas são também claramente superiores na oferta da *Oracle*, sendo necessária uma referência à maior complexidade que esta solução vem trazer, visto não serem, ao contrário da solução *SAP*, parametrizações simples.

Estes são os aspectos mais diferenciadores das duas soluções, sendo que ambas têm falhas nas genericidade das suas soluções, visto ambas serem desenvolvidas para uma integração preferencial com soluções próprias. Também de referir a fraca aposta de ambas as soluções no cumprimento das normas estipuladas para este tipo de solução. Com isto, é posta em causa a potencialidade de globalização da tecnologia RFID.

4 Middleware RFID

Como valor acrescentado, e uma vez que apenas mais um *middleware* nada traria de novo, é também objectivo deste trabalho que o *middleware* desenvolvido seja o mais portável, genérico, autónomo e escalável possível. Adiante neste trabalho, será possível termos uma descrição mais pormenorizada da forma como pretendemos obter com cada um destes pontos de interesse.

4.1 Introdução

As duas próximas sub-secções dão inicio à descrição do caminho que se pretende seguir e objectivos a atingir, com o enquadramento, e qual a linha de desenvolvimento pela qual se vai este trabalho guiar.

4.1.1 Enquadramento

Por definição, um *middleware* é uma peça de *software* com a responsabilidade de facilitar a interligação de um sistema distribuído [63]. Os intervenientes desta ligação podem ser de diversas naturezas. Podem ser apenas peças de *hardware*, apenas peças de *software*, ou ainda uma mistura de ambos. Na situação aqui em estudo será este ultimo caso, ou seja, uma mistura de ambos.

Na realidade, a interligação é apenas de *software* mas, uma vez que se está aqui a considerar que o *firmware* ou controlador do leitor é parte integrante do *hardware*, pode-se afirmar que, a interligação aqui em causa, é entre peças de *hardware* e *software*.

Tal como anteriormente mencionado, o *middleware* RFID é uma peça fundamental da tecnologia, apesar de não ser uma componente característica do sistema. Podemos dizer que este é fundamental pois, basta que o sistema assuma proporções médias, com algumas dezenas de leitores e poucos milhares de etiquetas em simultâneo, para que o volume de dados assuma dimensões incomportáveis para um sistema que não possua este nível na sua hierarquia. No entanto, não se pode considerar que este seja um componente característico da tecnologia uma vez que, este tipo de software, não é exclusivo destes sistemas, sendo utilizado em diversas arquitecturas e diferentes tecnologias.

Podemos ver, na Figura 4.15, qual a posição que o middleware deverá assumir numa arquitectura para a tecnologia RFID.

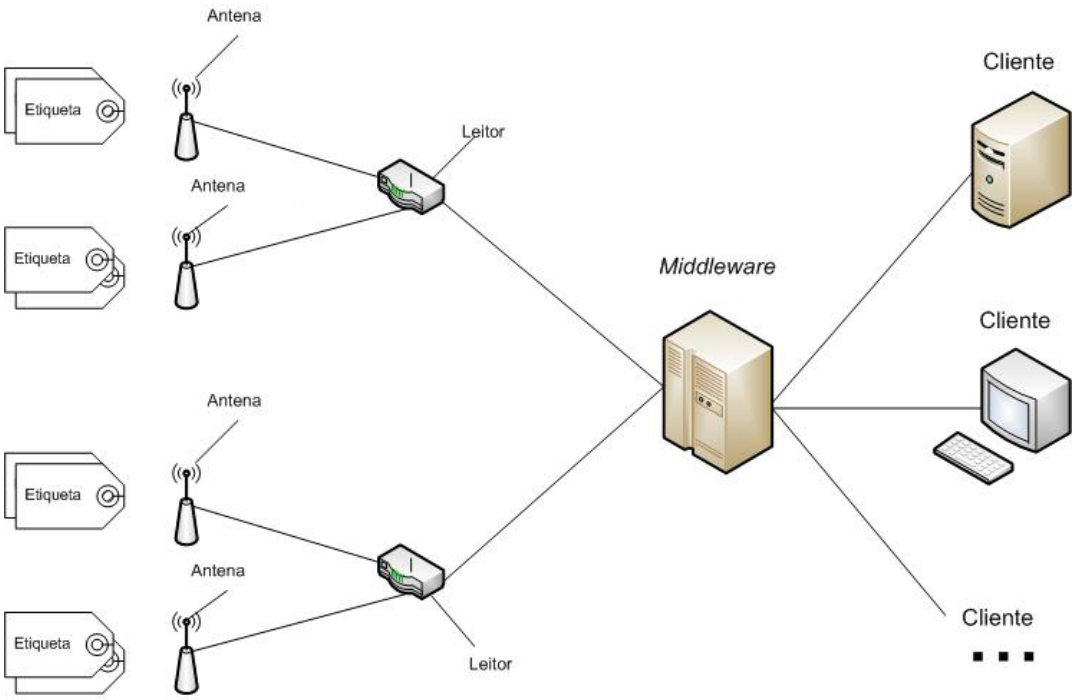


Figura 4.15 - Esquema de arquitectura genérica de um sistema RFID

Como foi já também, comentado, ao *middleware* não deverá competir apenas a simples interligação de componentes. Este deverá também, entre outras,

possuir a capacidade de homogeneização dos dados recolhidos. O que isto significa é que num sistema, como é o comum sistema RFID, poderemos ter diversos tipos diferentes de leitores, tendo cada um deles diferentes formatos de entrega de mensagens, utilizando mesmo diferentes protocolos para o conseguir. Naturalmente que não poderemos delegar no nível superior ao middleware, ou seja, a aplicações clientes deste, a tarefa de estar a descobrir o formato em que estes dados vêm, ou a forma como eles chegam. Assim sendo, fica também esta tarefa a cargo do *middleware*. Este, deverá receber dados dos diferentes leitores do sistema, que poderão ser leitores completamente diferentes utilizando diferentes formatos e protocolos para as suas mensagens, e entregar estes de forma normalizada, apenas com a informação necessária, e num formato que seja facilmente perceptível.

Outra tarefa também atribuída ao middleware é, dado o já mencionado elevado volume de dados que um sistema RFID poderá gerar, o de filtrar esta informação. Convém ser mais explícito, nesta fase, quanto ao conceito de filtragem a que aqui nos referimos. De forma alguma, num sistema fiável, interessa à aplicação cliente receber, de forma quase constante, a informação de que determinada etiqueta se encontra no alcance de determinado leitor. Basta que multipliquemos algumas dezenas de leitores por algumas centenas de etiquetas para que, imediatamente, nos apercebamos que o volume de dados gerado por um sistema semelhante poderá, facilmente, consumir mais dos recursos disponibilizados do que a aplicação objectivo, que tirará proveito desta informação. Tal cenário, apesar de viável, não fará de todo sentido. Assim sendo, a filtragem a que nos referimos, passa por apenas transmitir à aplicação cliente os dados que para esta sejam relevantes.

Estes são os requisitos tecnológicos fundamentais, nos quais o desenvolvimento deste trabalho assentará. Adiante teremos oportunidade de, não só aprofundar estes, como também apresentar novos pontos e desafios cuja adição se torne uma mais-valia para este trabalho.

4.2 Mote

É então, agora, enunciado o ponto de partida para este trabalho. Aqui pretende dar-se a conhecer o mote prático deste trabalho, ou seja, em termos abstractos, aquilo que é pretendido que este middleware seja capaz de fazer.

4.2.1 Interfaces de comunicação

Este middleware deve ter três tipos distintos de interface de ligação: uma para os seus clientes, outra para os leitores com os quais comunica e, finalmente, uma outra para uso administrativo.

A primeira destas interfaces, dos clientes, terá a função de aceitar ligações dos potenciais interessados na informação que por este é fornecida. Estes podem ser interfaces gráficas, que comunicam directamente com o *middleware*, como sistemas integrados de gestão, etc.

De seguida temos a interface de ligação aos leitores. A utilidade desta é obvia, mas deverá manter a maior genericidade possível, ou seja, não deverá ser direccionada para qualquer marca ou modelo de leitor especifico.

Por último, temos a interface de administração. Esta servirá para que seja possível executar e carregar as configurações a partir de um qualquer sistema (uma base de dados relacional, ficheiros xml, ficheiros de texto, etc) para o sistema middleware. Na Figura 4.16 podemos encontrar uma representação gráfica do que se acabou de descrever.

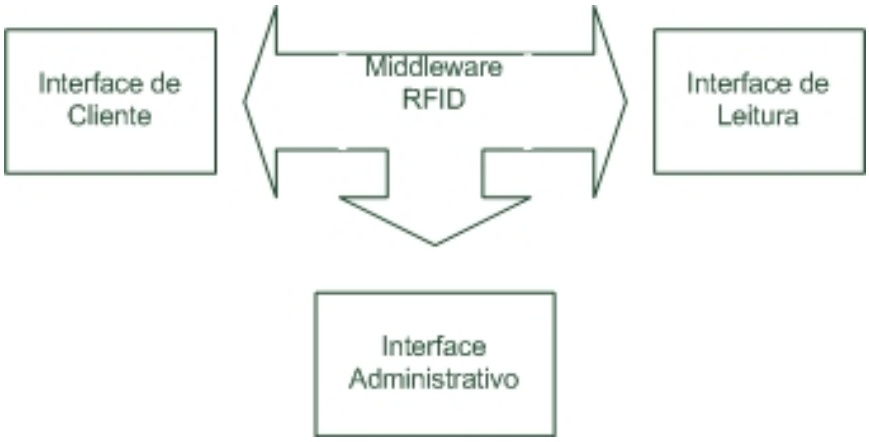


Figura 4.16 - Interfaces de ligação com o *middleware*

4.2.2 Interface *middleware* - clientes

Através da interface de cliente, anteriormente descrita, o *middleware* deverá ter a capacidade de aceitar ordens vindas dos clientes e, baseando-se nestas, construir respostas e relatórios que vai devolver, de novo, aos clientes.

Dada a diversidade de possíveis clientes que este *middleware* se propõe servir, impõe-se a utilização de um qualquer sistema que assegure que nem todos os clientes possam concretizar todas as ordens.

É também conveniente que, mais uma vez, dada a diversidade de possíveis situações onde este *middleware* se propõe ser capaz de intervir, seja possível aos clientes definir determinadas condições de leitura de forma a obter determinados efeitos. Alguns exemplos desta ultima característica poderiam ser: “quando a Etiqueta X se encontrar no Local Y, verificar se a Etiqueta Z se encontra no Local W;” ou “assim que a Etiqueta X esteja visível, apagar esta;”.

4.2.3 Interface middleware - leitores

É objectivo que, independentemente do tipo de leitor associado ao sistema, todas as operações pretendidas sejam possíveis de realizar. Como tal, as ordens serão dadas pelo cliente ao middleware sem que este tenha de ter a noção do tipo de leitor ao qual se está a dirigir. É mesmo desejável que, aos olhos dos

clientes, os leitores não sejam sequer visíveis, mas sejam apenas visíveis áreas lógicas, que de agora em diante vamos designar por zonas, que podem conter um ou mais leitores. Estas zonas não terão de ser constituídas, necessariamente, por leitores contíguos.

Da mesma forma, um leitor não tem de ser exclusivamente de uma zona, podendo pertencer simultaneamente a diversas destas zonas.

Podemos ver, na Tabela 4.2, um exemplo de distribuição de zonas lógicas.

Leitores					
Rede a que pertence	A	A	B	B	C
Zonas a que Pertence	Zona X Zona Y Zona Z	Zona X Zona W	Zona X	Zona W Zona Y Zona X	Zona X Zona Z

Tabela 4.2 - Exemplo de distribuição de zonas lógicas

Outro aspecto importante a ter em consideração, no que toca à comunicação entre os leitores e o *middleware*, é que, idealmente, este deverá ter capacidade de assimilar a entrada de um novo leitor no sistema. Em termos práticos, isto deverá reflectir-se na não necessidade de, estando o leitor ligado ao sistema, o utilizador ter de configurar o *middleware* de forma que este o passe a reconhecer. O que deverá acontecer é que o *middleware*, uma vez tendo o leitor ligado no sistema, deverá autonomamente localizar este e atribuir-lhe um protocolo de comunicação compatível efectuando, posteriormente, uma ligação com este e passar a utiliza-lo conforme seja necessário.

4.2.4 Interface de Administração

Como é comum a qualquer sistema, existe um conjunto de configurações que são necessárias para o seu conveniente funcionamento.

Este não será um sistema diferente, pelo que será conveniente a existência desta interface. No entanto, convém ressalvar que, segundo o pretendido, a interacção humana, uma vez estando o sistema a correr, deve ser a mínima possível. Assim sendo, é objectivo deste sistema que estas configurações sejam carregadas inicialmente, com o arranque do sistema, sendo estas definidas de forma que se mantenham válidas durante todo o tempo de execução deste.

4.2.5 Outros requisitos

Dada a natureza frágil desta tecnologia, é comum a existência de falhas, especialmente na leitura de etiquetas. Durante uma leitura, uma etiqueta pode deixar de ser lida sem que se nada no ambiente tenha mudado. Assim, é conveniente que o *middleware* consiga, de alguma forma compensar estas falhas. Naturalmente que não lhe será possível corrigir completamente estas. No entanto, poder-se-á encontrar alguma forma de minimizar o impacto que tais fragilidades possam provocar no sistema, tendo como base, por exemplo, o histórico de leituras de determinada etiqueta até à actualidade.

Um outro requisito a ter em consideração é a arquitectura física destes sistemas. A forma mais comum de interligar uma rede de leitores, middleware e seus clientes é, tal como explicado anteriormente neste documento, através de uma rede ethernet. Assim sendo, tal deverá ser levado em consideração na concepção da solução de forma a desta característica tirar algum partido.

4.3 Concepção da solução

Para conceber uma solução capaz de cumprir com os requisitos anteriormente propostos, será feita uma abordagem ponto-a-ponto, tocando em cada um dos

possíveis tópicos para cada um dos intervenientes deste sistema e definir qual a solução a adoptar.

4.3.1 Tópicos relacionados com os clientes

Pelo simples facto de serem os clientes os principais intervenientes desta solução, serão também estes os primeiros a ser abordados. Os conceitos a abordar para estes serão:

- Ordens;
- Relatórios;
- Identificação do cliente.
- Permissões;

4.3.1.1 Ordens

A forma como um cliente vai poder executar operações sobre determinada etiqueta será definida através de ordens.

Tal como foi já anteriormente referido, das ordens deve constar um conjunto de **condições** que, quando cumpridas, deverão originar a execução de um conjunto de **consequências**. Será este par condições/consequências o núcleo principal das ordens, podendo representar-se da seguinte forma:

$$Condições \rightarrow Consequencias\ ,$$

ou seja, um conjunto de condições que se revele verdadeiro implicará um conjunto de acções aqui denominadas por consequências. Apesar disto, as consequências poder-se-ão revelar verdadeiras ou falsas, dependendo do sucesso ou insucesso destas.

A informação relevante para o condicionamento das ordens prende-se com a localização onde determinada etiqueta se encontra ou não em determinado momento.

Convém explicitar o conceito de localização aqui aplicado. Tal como foi anteriormente indicado, as localizações traduzir-se-ão num conjunto de zonas

lógicas. A uma zona lógica não tem, necessariamente, de corresponder uma área física, mas antes um conjunto de leitores. A temática das zonas será adiante aprofundada em secção própria, bastando para já ter a noção que, quando se refere aqui uma localização, esta pode não corresponder, na verdade, a uma área ou espaço físicos.

Assim sendo, as condições serão definidas por um conjunto de pares etiqueta/zona, que passaremos a denominar de condição atómica, visto serem a menor divisão possível de uma expressão lógica, e que se representarão da seguinte forma:

$$Etiqueta\ @\ Zona\ ,$$

ou seja, quando determinada etiqueta se encontra em determinada zona, esta condição atómica assumirá o valor verdadeiro.

Uma vez que pode também ser pretendido executar determinada consequência exactamente quando uma etiqueta se encontra fora de determinada zona, de forma a existir, no *middleware,* alguma capacidade de localização, surge a necessidade de implantar a negação de condições atómicas. A negação de uma condição atómica poderá tomar a seguinte forma:

$$\neg(Etiqueta\ @\ Zona)$$

ou seja, esta condição será verdadeira quando a condição atómica que a constitui for falsa.

Ambas as condições anteriormente apresentadas, seja a condição atómica ou a condição atómica negada, caem na mesma classe, ou seja, ambas são condições pois ambas assumem valores de verdade ou falsidade.

Surge assim a necessidade de compor diversas condições. Para tal, introduz-se então, nesta fase, os operadores lógicos de conjunção e disjunção, ou seja, `E` e `OU` respectivamente. Estes podem-se representar da seguinte forma:

$$a \wedge b \quad e \quad a \vee b$$

respectivamente para a conjunção de condições e para a disjunção de condições, onde a e b representam condições atómicas. Mais uma vez, quer a conjunção

quer a disjunção de condições representam, elas próprias, novas condições. Estas passarão a ser designadas de condições compostas. O valor lógico de uma conjunção de duas condições será verdadeiro exclusivamente quando ambas as condições se revelem também verdadeiras, sendo falso em todas as outras situações. O valor lógico de uma disjunção de duas condições será verdadeiro quando pelo menos uma das suas condições for verdadeira, sendo falsa em todas as outras situações.

Uma vez que também estas caem na classe das condições, também a estas será possível aplicar a negação, obtendo então os seguintes resultados:

$$\neg(a \wedge b) \quad e \quad \neg(a \vee b)$$

ou seja, analogamente às condições atómicas, as condições apresentadas serão verdadeiras apenas quando as condições compostas forem falsas.

Evoluindo naturalmente, a composição de expressões compostas deverá também ser possível, bastando para isso que nas condições anteriormente apresentadas substituamos os valores de a e b, que representavam então condições atómicas, por condições compostas.

Com isto, poderemos facilmente obter condições compostas como a que, a título de exemplo, se segue:

$$\neg(\neg(a \wedge b) \vee (f \vee b))$$

onde a, b e f representam condições atómicas.

Assim sendo, um conjunto de condições será um conjunto de uma ou mais condições compostas de complexidade infinitamente variável.

Uma vez que um conjunto de condições pode ter uma cardinalidade infinitamente grande, urge a criação de mecanismos que facilitem o agrupamento e criação de conjuntos de condições. Para tal, surge a divisão do conjunto de condições em três subconjuntos que de seguida se explicitam:

- **Todos:** este subconjunto de condições é verdadeiro quando todas as condições que o compõe são também verdadeiras;

- **Qualquer:** este subconjunto de condições é verdadeiro quando pelo menos uma das condições que o compõe é também verdade;
- **Nenhum:** este subconjunto de condições é verdadeiro quando nenhuma das condições que o compõe é verdadeira.

Resumidamente, estes três conjuntos representam, respectivamente, uma grande conjunção, uma grande disjunção e ainda uma grande negação.

Com estes subconjuntos definidos, o conjunto de condições passa a ser verdadeiro apenas quando todos os seus subconjuntos são também verdadeiros.

Também para facilitar a criação de condições será introduzido o conceito de *Wild-Card*, ou seja, caracteres que podem assumir o papel de qualquer outro caracter ou conjunto de caracteres. Estas *Wild-Cards* serão possíveis de utilizar apenas nas etiquetas. O motivo pelo qual não será necessária a sua utilização nas zonas será mais adiante clarificado. Assim sendo, as *Wild-Cards* permitidas são as seguintes:

- **Um caracter:** Representado pelo símbolo **?**, este pode substituir apenas um caracter na identificação de uma etiqueta;
- **Vários caracteres consecutivos:** Representado pelo símbolo *****, este pode substituir uma qualquer cadeia de caracteres consecutivos na identificação de uma etiqueta.

Depois de termos o conjunto de condições, passamos então ao conjunto de comandos. Este, comparativamente ao conjunto de condições, é de complexidade muito inferior.

O conjunto de comandos disponíveis pode-se dividir em dois grupos principais: comandos de escrita e comandos de leitura de etiquetas. Dentro do conjunto de comandos de escrita temos três comandos:

- **Matar[3]:** com a utilidade de inutilizar permanentemente uma etiqueta;
- **Bloquear:** com a utilidade de inibir futuras codificações;

[3] Que vem do termo técnico com que se designa o comando na língua inglesa: *Kill command*.

- **Codificar:** com a utilidade de inserir dados numa etiqueta.

Estes são comandos simples, onde basta nos dois primeiros indicar a etiqueta, e no terceiro indicar a etiqueta e os dados a codificar nesta.

O segundo conjunto de comandos disponível, de leitura, conta apenas com um comando: precisamente, o que permite ler os dados de uma etiqueta.

Neste, é possível definir quais as etiquetas que pretendemos ler e em que zonas. Numa ordem de leitura, podem ser indicados diversos comandos, cada um deles com um par etiqueta/zona, indicando que etiquetas pretendemos ler em que zona. Também nestes é possível utilizar *Wild-Cards* nas etiquetas. É de salientar ainda que a execução das ordens é sequencial, ou seja, os comandos definidos primeiro serão os primeiros a ser executados.

Uma vez que a utilização das ordens pode ser especifica para determinado tipo de cliente, estas deverão conter um nível de acesso permitido, que fará com que os apenas alguns clientes as possam executar. Adiante será melhor explicitado o funcionamento das permissões.

4.3.1.2 Relatórios

Os relatórios são de funcionamento muito simples. Quando, numa ordem, todas as condições se verificadas verdadeiras, todas as consequências são executadas. Da execução de cada uma das consequências resulta um valor lógico de verdade ou falsidade, que representa o sucesso ou insucesso de cada uma delas.

Também a data e hora, em que cada uma das condições se tornou verdadeira, é armazenada, consoante seja ou não pedido, na ordem, que o relatório venha com o registo destes tempos.

A ordem é, posteriormente, integralmente enviada ao criador de relatórios que, baseando-se na informação desta, cria o relatório a ser entregue aos clientes interessados.

O formato escolhido para a recepção destes relatórios pode variar de cliente para cliente.

4.3.1.3 O Cliente

O cliente é representado no sistema pela sua identidade e endereço. A composição da identidade é livre, enquanto o seu endereço é a sua identificação de rede IP.

É através da sua identificação que um cliente é associado a um conjunto de ordens, as quais este subscreve.

Para além destes atributos, é também atribuída uma lista de níveis a cada um dos utilizadores. Estes níveis representam a permissão que estes têm para executar ordens. A forma como são lidadas as permissões é de seguida apresentada.

4.3.1.4 Permissões

As permissões têm sido até agora referidas como a forma de limitar o acesso, da parte dos clientes, a certas funcionalidades do middleware, mais concretamente, às ordens.

Na verdade, esta é mais uma forma de manter uma estrutura organizada na forma como os acessos às ordens são feitos.

Cada uma das ordens terá um nível de acesso, com a seguinte forma:

nível_a . nivel_b . nivel_c . etc.

Da mesma forma, um cliente deverá ter uma lista de permissões, num formato semelhante ao anterior, que lhe vai permitir o acesso ou não a determinada ordem.

A forma como a verificação das permissões é feita pode considerar-se relativamente simples. Seja o exemplo seguinte:

Se o nível de acesso à Ordem X é definido pela seguinte frase:

a.a.b.c

e, uma das permissões constantes da lista de permissões do Cliente A é:

a.a.b

então, este utilizador terá permissão para subscrever esta ordem. No caso da Ordem Y, que tem o seu nível definido por:

a.a

então, o Cliente A já não teria permissão. Também para a Ordem Z, com o nível definido por:

a.a.c.d

o Cliente A não tem permissões suficientes.

No entanto, temos também o Cliente B, que na sua lista de permissões tem a frase:

a

Significa isto que, o Cliente B, terá permissão para utilizar quer a Ordem X, quer a Y, quer a Z.

Explicando o exemplo apresentado, o Cliente A terá acesso à Ordem X pois a frase de permissão que este apresenta encaixa na totalidade na frase de nível da Ordem X. Para a Ordem Y, os primeiros dois campos encaixam, no entanto, para o terceiro tal já não se aplica. Com a Ordem Z, os primeiros dois campos encaixam mas, o terceiro, é diferente. Assim sendo o Cliente A não terá permissão para subscrever a Ordem Z.

No caso do Cliente B, este terá permissão para subscrever qualquer uma das três ordens aqui apresentadas visto que o seu primeiro campo, ou seja "**a**", encaixa quer na Ordem X, quer na Y, quer na Z.

É de salientar que o encaixe de campos não é móvel, ou seja, o primeiro campo da permissão do utilizador terá de encaixar com o primeiro campo do nível da ordem, o segundo campo do utilizador com o segundo da ordem e assim sucessivamente. Ou seja, se um Cliente C tiver na sua lista de permissões a frase:

d.e.g

e o nível de acesso da Ordem W for:

c.d.e.g.r

o Cliente C não terá acesso à Ordem W.

Tal como mencionado inicialmente, esta não se pode considerar uma medida de segurança, pois no caso de um cliente realmente pretender aceder a determinada ordem, ele vai consegui-lo, bastando-lhe para isso adicionar uma nova permissão à sua lista de acessos.

A real utilidade e vantagem deste controlo de permissões reside na redução de ordens acidentais, dificilmente acontecendo que, especialmente ordens de leitura, sejam dadas por engano.

4.3.2 Tópicos relacionados com os leitores

Relativamente aos tópicos relacionados com os leitores, existem dois que realmente valerá a pena referir e aprofundar. São eles:

- Descoberta e ligação de novos leitores
- Definição de zonas lógicas

4.3.2.1 Descoberta e ligação de novos leitores

Uma das funcionalidades propostas para este *middleware*, é o facto de não ser necessária a reconfiguração do *middleware* de cada vez que lhe queremos acrescentar um novo leitor.

Assim, existem algumas premissas necessárias de forma a obter o efeito desejado. Apesar da não necessidade de configuração do *middleware*, existe a necessidade da configuração prévia dos leitores, anterior à sua ligação ao sistema. A principal configuração, necessária à correcta adição de leitores ao sistema, de forma a tirar o máximo partido deste, prende-se com a atribuição de um endereço de rede ao leitor. Isto poder-se-ia ultrapassar utilizando ferramentas e protocolos de rede adequados mas, ainda assim, nem toda a funcionalidade ficaria disponível. A razão para tal tem, mais uma vez, a ver com as zonas, temática que será abordada no seguimento.

Estando um novo leitor devidamente configurado e ligado ao sistema, via ligação *ethernet*, impõe-se agora a descoberta do mesmo pelo *middleware*. Para que tal seja possível, o *middleware* vai possuir uma ferramenta, com a qual, varrimentos sucessivos são feitos a intervalos pré-estabelecidos de endereços de rede.

Assim que um dos endereços questionados responda, imediatamente esse endereço é entregue a um novo bloco que se vai encarregar de o questionar acerca da sua identidade e protocolo de comunicação.

Caso a identidade ou protocolo de comunicação sejam reconhecidos como característicos de um leitor, é feita uma entrega imediata dessa comunicação numa "piscina" de leitores, onde todos os leitores activos da rede se encontrarão armazenados.

O sistema de varrimento, tal como enunciado, poderá trazer alguns problemas de congestionamento de rede e performance. Assim sendo, um limitador de tempo é incluído (e configurável), de forma que apenas seja interrogado um endereço por intervalo de tempo.

Naturalmente que, se tentarmos interrogar todos os endereços disponíveis no espectro de endereços de rede, poderia levar anos até finalmente encontrar um novo endereço. Como tal, gamas de endereços a analisar devem ser configuradas pelo administrador, de forma a limitar a quantidade de endereços questionáveis.

É ainda de salientar que, quando a um endereço corresponde já um leitor, este não voltará a ser questionado. Esta verificação é sempre feita antes de todas as questões efectuadas, e é feita à a cada um dos leitores.

4.3.2.2 Definição de zonas lógicas

Como foi dito, a criação de zonas lógica é, de facto, independente da existência de zonas ou áreas físicas correspondentes.

Uma zona é, na verdade, definida como um conjunto de leitores que não têm necessariamente de se encontrar contíguos ou, sequer, no mesmo espaço ou edifício físico.

Para o *middleware*, não existe o conceito de definição prévia das zonas disponíveis onde podemos adicionar ou retirar leitores. O que realmente sucede é que, de uma ordem, fazem parte diversas condições que, por definição, são constituídas pelo par etiqueta/zona. A zona, referida neste par, é representada pela tradução de 32 valores lógicos, em 32 bits (verdade = 1 e falsidade = 0) que, por sua vez, são divididos em 4 grupos de 8 bits e, posteriormente, traduzidos em números decimais. Exemplificando:

Partindo do conjunto de 32 valores lógicos (em que V representa Verdade e F representa Falsidade) *{VVVVVVVVVFVFFFVFFFFFVVVVVFVFFVVF}* obtemos a sequencia binária, já dividida em grupos de 8 bits *{11111111.10100010.00001111.10100110}* que, traduzindo para a numeração decimal, se pode representar por *{255.162.15.166}*, representando, então, este conjunto de valores decimais, aquilo a que, de agora em diante, passamos a designar por máscara de zona a que esta condição se refere.

Como se poderá, então, associar a máscara de zona às zonas definidas pelos leitores? Prestando alguma atenção à estrutura destas máscaras de zona definidas nas condições, rapidamente se chega à conclusão que esta é bastante semelhante àquela pela qual se regem os actuais endereços IP. Isto não acontece por acaso, pois será exactamente através da realização de algumas operações entre a máscara de zona e o endereço IP dos leitores que se poderá definir se um leitor pertence ou não a determinada zona.

Seja o seguinte exemplo:

Determinada condição tem, como mascara de zona, o valor *{0.136.3.128}*. Esta mascara de zona, traduzir-se-á na cadeia binária *{00000000.10001000.00000010.10000000}*, da qual resultará a seguinte sequência de valores lógicos:

{FFFFFFFFVFFFVFFFFFFFFFVFVFFFFFFF}.

O Leitor X, terá como endereço de rede *{192.168.10.184}*. Deste endereço resulta a cadeia binária *{11000000 10101000 00001010 10111000}*, da qual podemos extrair a seguinte sequência de valores lógicos:

{VVFFFFFFFVFVFVFFFFFFFFVFVFVFVVVFFF}.

Se conjugarmos estas duas sequencias de valores lógicos:

{FFFFFFFF VFFFVFFF FFFFFFVF VFFFFFFF}

^

{VVFFFFFF VFVFVFFF FFFFVFVF VFVVVFFF}

=

{FFFFFFFF VFFFVFFF FFFFFFVF VFFFFFFF}

ou seja, da conjugação da mascara de zona com o endereço do leitor obtemos, novamente, a mascara de zona definida na condição. Assim sendo, este leitor inclui-se na zona à qual esta condição se refere.

Se analisarmos um novo leitor, que vamos designar por Leitor Y, que terá o endereço *{10.201.98.172}*, o que resulta na cadeia binária *{00001010 11001001 01100010 10101100}* e que, por sua vez, se traduz na sequência lógica:

{FFFFVFVF VVFFVFFV FVVFFFVF VFVFVVFF}

Mais uma vez, conjugando esta sequência com a mascara de zona da condição anteriormente referida, obtemos:

{FFFFFFFF VFFFVFFF FFFFFFVF VFFFFFFF}

^

{FFFFVFVF VVFFVFFV FVVFFFVF VFVFVVFF}

=

{FFFFFFFF VFFFVFFF FFFFFFVF VFFFFFFF}

Ou seja, um leitor com um endereço de rede completamente diferente, que poderá estar numa zona completamente diferente do globo, mas que acaba por pertencer à mesma zona lógica do leitor anterior.

Seja agora o Leitor Z, que tem o endereço de rede *{192.168.10.10}*. Deste vai resultar a cadeira binária *{10101000 00001010 00001010}* que, por sua vez, produz a seguinte sequência lógica:

{VVFFFFFF VFVFVFFF FFFFVFVF FFFFVFVF}

Conjugando esta última com a máscara de zona definida na condição acima, obtemos o seguinte resultado:

{FFFFFFFF VFFFVFFF FFFFFFVF VFFFFFFF}

^

{VVFFFFFF VFVFVFFF FFFFVFVF FFFFVFVF}

=

{FFFFFFFF VFFFVFFF FFFFFFVF FFFFFFFF}

Podemos observar que, o resultado obtido desta conjugação não é, desta vez, igual à máscara de zona da condição. Assim sendo, este não é, apesar de poder pertencer à mesma rede que o Leitor X, um leitor que pertença à zona explicitada na condição.

4.3.3 Administração do leitor

Serão aqui descritos alguns tópicos de importância com respeito à administração do leitor. Os dois principais tópicos a considerar aqui serão:

- Ordens
- Zonas
- Permissões

4.3.3.1 Ordens

Tal como pode já ter ficado, de alguma forma, implicitamente perceptível, grande parte da administração do leitor passa pela definição das ordens que ficarão disponíveis para subscrição dos clientes.

Assim sendo, nunca um cliente poderá, em tempo real, criar uma ordem completamente nova. Na verdade, todas as ordens se devem encontrar já previamente definidas, devendo isto ser feito apenas e só pelo administrador (ou equipa de administração) do sistema.

Estando a definição das ordens concluída, estas devem ser carregadas para o leitor que os colocará numa *pool* de ordens.

Quando um cliente pretender, e tendo permissão para tal, poderá subscrever qualquer uma dessas ordens disponíveis, sobre a qual passará a receber relatórios sempre que uma mudança de estado no ambiente provoque a veracidade das condições da ordem.

4.3.3.2 Zonas

Em termos administrativos, a definição de zonas passa pela correcta configuração dos endereços de rede dos leitores e, simultaneamente, correcta utilização da definição de máscaras de zona nas condições das ordens.

Esta definição será o suficiente para obter o resultado que se pretende do sistema.

Também baseando-se nas potenciais zonas existentes, o administrador deverá definir um ou vários intervalos de endereços de rede questionáveis, de forma a facilitar a inclusão de novos leitores no sistema.

4.3.3.3 Permissões

A definição de permissões é relativamente simples. Deve ser criada, utilizando a definição de permissões anteriormente apresentada, uma hierarquia de potenciais clientes do *middleware*. Posteriormente, deverão ser atribuídos um ou mais desses níveis aos clientes envolvidos no sistema e, também, baseando-se na hierarquia definida, devem ser atribuídos níveis de acesso às ordens oferecidas pelo sistema.

4.3.4 Funcionamento interno do *middleware*

Para além das questões relacionadas com a comunicação entre os diversos intervenientes e o sistema, existe ainda a forma de organização interna.

Em concreto, são os conceitos relativos aos seguintes temas:

- Gestão de ordens
- Gestão de leitores
- Confiança de leitura

4.3.4.1 Gestão de ordens

Tal como foi já referido, internamente, o *middleware* carrega e armazena todas as ordens possíveis numa *pool* de ordens. No fundo, esta piscina de ordens não é mais que um conjunto organizado de ordens.

Inicialmente, todas as ordens se encontram desactivadas, uma vez que inicialmente nenhum cliente subscreveu ainda ordens.

Ao longo deste trabalho, referimos por diversas vezes o acto de subscrição de uma ordem por um cliente. Uma ordem, na verdade, não é propriedade de um cliente que a pretenda executar. Na verdade, um cliente não pode ou consegue, directamente, executar uma ordem. A única forma de um cliente ficar ligado a

uma ordem é subscrevendo esta ordem. Internamente, todas as ordens estão inactivas e são ignoradas pelo sistema até que pelo menos um cliente as subscreva.

Assim que um cliente subscreve uma ordem que se encontra inactiva, esta passa ao estado activo verificando quais das suas condições são já, no momento da activação, verdadeiras.

A partir desta altura, sempre que um novo evento ou mudança do estado do ambiente é recebida, é verificado o interesse desta mudança para esta ordem. Caso esta mudança seja relevante para uma ordem, as condições que esta afecta modificarão o seu valor lógico de acordo com a mudança do estado do ambiente.

O facto de um novo cliente subscrever uma ordem que se encontra já activa, não trará um acrescento significativo em termos de peso computacional, visto que este, ao subscrever uma ordem, está apenas a declarar que pretende também receber o resultado, na forma de um relatório, da execução dessa ordem.

Voltando agora aos eventos, estes podem ser essencialmente de dois tipos:

- A Etiqueta X foi detectada na Zona A
- A Etiqueta X deixou de ser detectada na Zona A

Interessa analisar qual o impacto que estes eventos têm na *pool* de ordens. Para ambos os eventos, o tratamento relativamente às ordens é semelhante. Quando um evento ocorre, este é comunicado à "piscina" de ordens que, posteriormente, verificará quais as ordens a quem esta etiqueta poderá interessar. Sempre que seja encontrada, numa ordem, uma condição que referencie esta etiqueta, a ordem em questão será colocada num conjunto de ordens a analisar. Depois de encontradas todas as ordens às quais este evento possa interessar, serão novamente analisadas as condições em causa em cada uma das ordens. Caso este novo evento faça com que exista uma mudança de valor lógico em qualquer condição de uma ordem, o valor lógico do conjunto de condições dessa ordem é avaliado e, caso seja verdadeiro, a ordem é executada.

Uma vez executada a ordem, esta passa novamente ao estado inactivo, ficando de novo a aguardar a subscrição por parte de clientes.

Um cliente que tivesse subscrito esta ordem, assim que esta é executada e o seu relatório enviado para todos os clientes subscritores, perde essa subscrição, necessitando de, caso deseje, voltar a fazer a mesma.

Como é óbvio, a qualquer momento, um cliente poderá querer abandonar a subscrição de uma ordem, pelo que essa opção foi contemplada.

Na Figura 4.17 podemos ver um diagrama com os possíveis estados e transições de estado de uma ordem.

Figura 4.17 - Diagrama de estado de uma ordem

4.3.4.2 Gestão de leitores

De forma análoga às ordens, também a informação dos leitores se encontra internamente armazenada numa "piscina" de leitores.

A forma como os leitores são adicionados ao sistema foi já, na secção 4.3.2, analisada. No entanto, depois de um leitor ser adicionado ao *middleware* pelo sistema de localização de leitores, a gestão destes passa a ser da responsabilidade da citada *pool* de leitores. A principal função que aqui deve ser accionada, relativamente aos leitores, será a manutenção destes, ou seja, da mesma forma que a adição de leitores é automática, a subtracção destes deve ter um comportamento semelhante. Assim que uma tentativa de comunicação com determinado leitor falhe, este leitor, juntamente com as etiquetas que a ele

estavam associadas, são descartados. Esta solução pode parecer algo radical mas, se nos lembrarmos que este endereço voltará a ser visitado pelo questionador de endereços, não será grave o descartar do leitor mesmo que a falha tenha sido circunstancial, pois se este responder novamente, voltará a ser adicionado ao sistema.

4.3.4.3 Confiança de leitura

Como vem sendo indicado ao longo deste documento, com maior expressão nos capítulos que referiam os problemas e fragilidades ainda presentes nesta tecnologia, será feito um esforço para que, de certa forma, seja incluída no *middleware* alguma compensação para os referidos problemas.

Assim, a forma pensada para obter esta compensação foi a inclusão de graus de confiança das leituras. De forma abstracta, estes graus de confiança podem ser descritos como: até que ponto podemos ou não confiar numa leitura, ou não-leitura, de determinada etiqueta, com base no histórico de leituras desta, para podermos considerar correcto o resultado obtido por essa leitura.

Assim sendo, a cada etiqueta que é inserida no sistema, é associado um grau de confiança que, com o passar do tempo, vai evoluindo.

Para que este grau de confiança possa ter alguma validade prática, será necessário alguma quantidade de leituras e falhas, sobre as quais poderemos construir este cálculo.

Naturalmente que, dada a possível quantidade de etiquetas num sistema destes, e visto que muitas delas se poderão manter no sistema por durante uma grande quantidade de leituras, não será viável, apesar de ser o ideal, o armazenamento de todo o historial de leituras de cada uma das etiquetas.

Assim, considera-se uma quantidade suficientemente representativa, e atribui-se por omissão, um historial de 50 leituras. Apesar de esta ser a quantidade aconselhada por defeito, esta quantidade poderá ser configurada pelo administrador do sistema.

Desta quantidade de leituras, que passaremos a designar por fila de confiança, teremos de designar uma percentagem que representará, desta fila, qual a quantidade de leituras que serão utilizadas para calcular a confiança histórica e qual a quantidade que será utilizada para o cálculo da confiança recente. Por defeito, é dado o valor de 80%, ou seja, das 50 leituras, 40 representarão a fila de confiança histórica de cada etiqueta e as restantes 10 vão representar a fila de confiança recente.

Existe ainda uma outra quantidade, dentro da fila de confiança, que sobrepõe as leituras das filas de confiança histórica e recente. Em número de leituras igual às existentes na fila de confiança recente, são seleccionadas, da fila de confiança histórica, a leitura mais recente e, da fila de confiança recente, todas as leituras excepto a última. Este conjunto de leituras será designado por fila de confiança recente anterior.

Temos então, até agora, os seguintes conceitos:

- **Fila de confiança:** onde existem os dados ordenados do sucesso ou insucesso das últimas tentativas de leitura de cada etiqueta;
- **Fila de confiança histórica:** que representa uma percentagem das tentativas de leitura mais antigas armazenadas na fila de confiança;
- **Fila de confiança recente:** que representa as tentativas de leitura, na fila de confiança, que não sejam incluídas na confiança histórica;
- **Fila de confiança recente anterior:** que representa o conjunto de leituras, em igual número à quantidade de leituras da confiança recente, compostas pela última leitura da fila de confiança histórica e todas as leituras da fila de confiança recentes excepto a última.

Como podemos então trabalhar estes valores de forma a compensar as falhas da tecnologia?

Para cada uma das filas de confianças apresentadas, é feito o seguinte cálculo:

$$Confiança = \frac{quantidade\_de\_leituras\_positivas}{quantidade\_de\_leituras\_total}$$

obtendo, desta forma, a confiança associada a cada uma das filas.

Assim sendo, por exemplo, uma etiqueta que tenha 40 leituras positivas em 40 tentativas de leitura na sua fila de confiança histórica, terá uma confiança histórica de 1. Uma etiqueta que tenha 20 leituras positivas em 40 tentativas de leitura na sua fila histórica, ficará com uma confiança histórica de 0.5.

Começando pelo mais simples, as confianças recente e recente anterior têm uma utilidade bastante linear. Considera-se que, sempre que a confiança de determinada etiqueta se encontra em crescimento, ou seja:

$$Confiança\_recente > Confiança\_recente\_anterior$$

uma etiqueta nunca deverá ser eliminada. Parece-me razoável pensar que, se o número de leituras positivas de uma etiqueta está a aumentar, não será sensato elimina-la.

Uma outra questão associada à confiança recente é que, nenhuma etiqueta com uma confiança recente inferior a 0.125 deverá ser considerada. O motivo para isto é que, com taxas de leitura tão baixas, o mais provável é que esta etiqueta esteja a ser esporadicamente lida numa zona que não é a sua.

No que respeita à confiança histórica, convém apresentar mais um conceito, que é a quantidade de leituras falhadas da fila de confiança recente. A esta quantidade vamos designar, convenientemente, por leituras falhadas recentes.

Assim sendo, uma outra condição que levará à exclusão de etiquetas de leitura será:

$$Confiança\_histórica > M \times Leituras\_falhadas\_recentes + B$$

onde M e B representam factores de sensibilidade. Mais uma vez, aos valores de M e B é atribuído um valor por omissão que poderá ser configurável pelo administrador do sistema.

Para os valores por omissão, M = -0.083333 e B = 1.166667, obtemos a seguinte inequação:

$$Confiança\_histórica > -0.083333 \times Leituras\_falhadas\_recentes + 1.166667$$

Se olharmos com atenção para esta fórmula, podemos verificar que esta representa a equação de um recta. Ao representarmos esta recta graficamente, obtemos o gráfico apresentado na Figura 4.18.

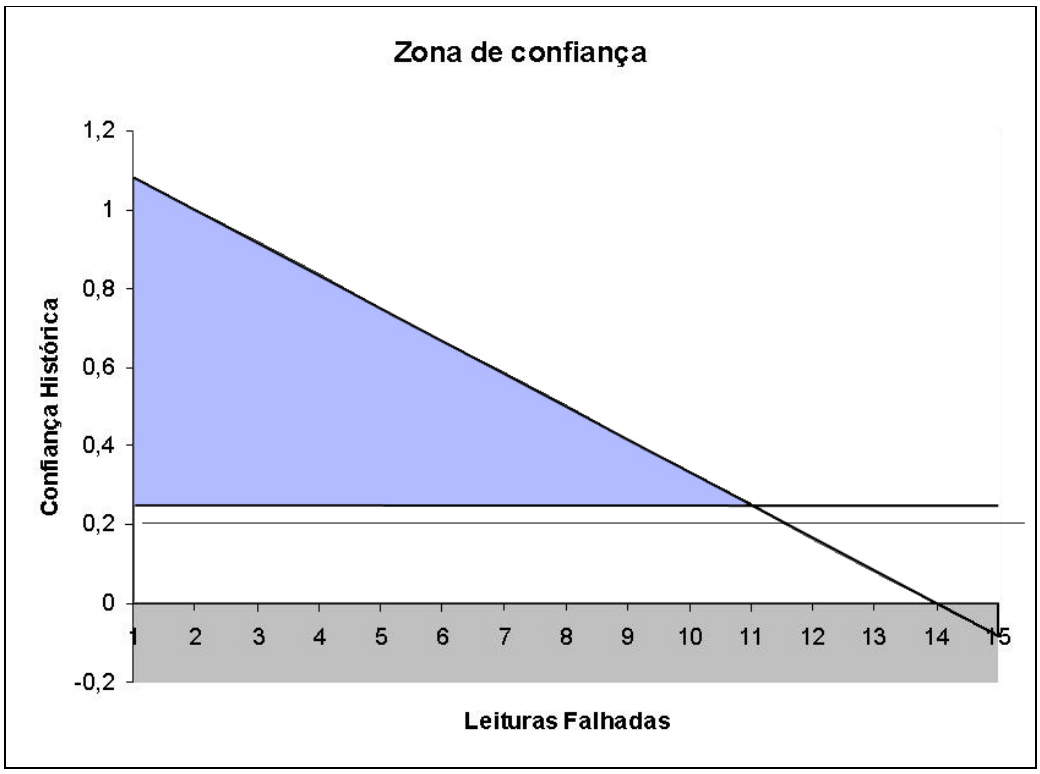


Figura 4.18 - Gráfico da equação de exclusão de etiquetas . Apenas são consideradas válidas as etiquetas cuja confiança se encontre na zona colorida.

Analisando o gráfico concluímos que a exclusão de determinada etiqueta acontece quando, relacionando a quantidade de vezes que, recentemente, uma leitura falhou, com a sua confiança histórica, os valores de ambos surgem acima desta recta.

Explicando a mais alto nível, é feita uma comparação do comportamento recente de cada uma das etiquetas com o seu comportamento histórico, ou seja, apenas se houver uma grande disparidade, no sentido negativo da percentagem de sucesso de leitura actual de determinada etiqueta para a histórica, assumimos que esta etiqueta deixa de estar presente.

Um exemplo prático disto será:

Suponhamos que determinada etiqueta teve, como confiança histórica, uma percentagem de leituras positivas de 100%. Para esta etiqueta, podemos considerar que, à primeira ou segunda falha consecutiva de leitura, esta etiqueta já não se encontrará no alcance de leitura do leitor.

Suponhamos agora que, determinada etiqueta, tem uma taxa de sucesso de leitura histórica de 30%. Quando, para esta etiqueta, temos duas ou três falhas consecutivas de leitura, nada podemos concluir, visto que, baseando-nos no seu historial, será comum a não leitura desta etiqueta.

4.4 Desenvolvimento do Middleware

Passa-se, nesta fase, à descrição da forma como foram implementados os principais recursos deste *middleware*. Visto que não é intenção, neste capítulo, abordar todo o desenvolvimento, mas apenas o desenvolvimento dos módulos principais, em anexo será possível visualizar o diagrama de classes completo para esta solução.

4.4.1 Tecnologia utilizada

A tecnologia utilizada na implementação deste *middleware* foi o *Java 2 – SDK 1.6*.

Os principais motivos que justificam esta escolha foram, em primeiro lugar, o facto de esta tecnologia ter suporte nativo multi-plataforma. Isto, dado que era pretendido que este middleware fosse o mais genérico possível, era uma característica incontornável. Para além deste, existiram factores como a quantidade e qualidade de documentação existente para esta tecnologia, ou a quantidade de recursos disponíveis, que a torna impar no meio das tecnologias possíveis.

O principal óbice apresentado por esta tecnologia é, de facto, a sua ineficiência aparente. O facto de esta tecnologia correr em cima de uma máquina virtual faz com que operações que, em tecnologias que compilam código máquina, são simples e rápidas, aqui se tornem desnecessariamente complexas e lentas.

4.4.2 Interfaces de comunicação

As interfaces de comunicação (clientes, leitores e administrativa) destacam-se não pela complexidade do seu desenvolvimento mas antes pelo cariz genérico que lhes é conferido.

Estas, surgem como extremidades inacabadas do nosso *middleware*, visto não se encontrarem, de todo, implementadas. A ideia foi criar estas pontas soltas, na forma de *Interfaces Java*, de forma que, assim que se pretenda adicionar este middleware a um sistema especifico, baste criar classes que implementem estas interfaces de acordo com os sistemas que pretendemos interligar com o nosso *middleware*.

4.4.3 Ordens

Existem várias implementações pertinentes para a questão das ordens. As que aqui serão discutidas são:

- Condições
- Gestão de ordens
- Comandos

- Permissões

4.4.3.1 Condições

A implementação das condições foi feita segundo o modelo de classes que se pode ver no diagrama de classes exposto na Figura 4.19.

Assim sendo, foram criadas 4 classes que representam os 4 tipos de condição disponível: o conjunto de condições, `ExpressionSet`, as duas condições compostas, `ExpressionAnd` e `ExpressionOr`, e ainda a condição atómica, `ExpressionAtom`. Tal como se pode verificar no diagrama, todas elas implementam uma mesma interface, `iExpression`, que as distingue como sendo expressões e que força a implementação do método:

```
public boolean expressionLogicValue()
```

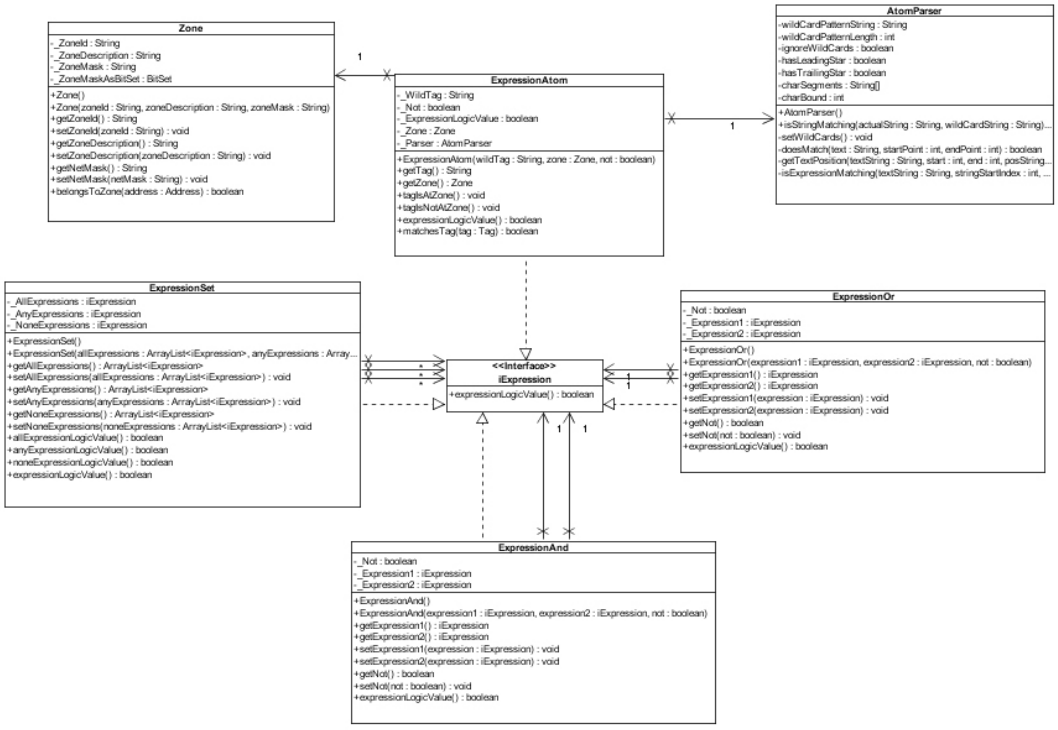


Figura 4.19 - Diagrama de classes de Condições

Como se constata, tanto as classes que representam as condições compostas como a classe que representa o conjunto de condições, aceitam como seus componentes `iExpressions`. Isto significa que, por exemplo, os componentes de uma condição composta podem ser uma outra condição composta e um conjunto de condições.

A classe `AtomParser` é uma classe auxiliar com a utilidade de fazer o reconhecimento da identificação das etiquetas mesmo utilizando *Wild-Cards*.

Como também é visível, as condições atómicas contêm uma instância da classe `Zone`, não contendo no entanto da classe `Tag`. O motivo para esta decisão prende-se com o facto de, a representação da etiqueta na condição atómica é sempre tomada como uma `String` de caracteres que poderá conter *Wild-Cards*. Assim sendo, e visto a classe `Tag` apenas poder representar uma etiqueta específica, e não um conjunto delas, não é incluída uma instância desta nas condições atómicas.

4.4.3.2 Gestão de ordens

A gestão de ordens é feita tal como anteriormente mencionado. Podemos ver na Figura 4.20 uma representação das classes que compõe a gestão de ordens na forma de um diagrama de classes.

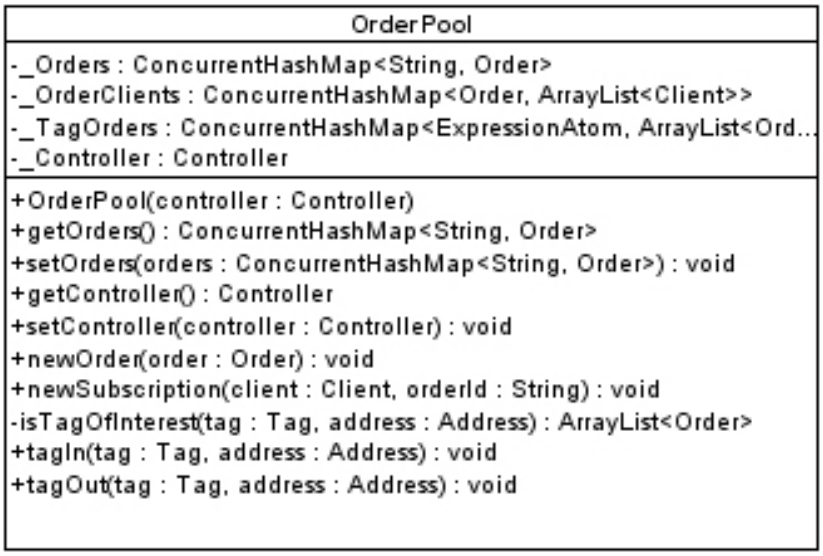


Figura 4.20 - Diagrama de classes de Ordens

Aqui, pode-se ver que a *pool* de ordens está representada pela classe `OrderPool`. Esta, contém três conjuntos, denominados por `_Orders`, `_OrderClients` e `_TagOrders`. O primeiro destes conjuntos contém, obviamente, as ordens que compõe esta *pool*, indexadas pela sua identificação. O segundo conjunto contém todos os clientes que têm subscrita alguma ordem. A indexação é feita não pela identificação de uma ordem mas pela ordem em si. Finalmente, temos o terceiro conjunto onde as ordens estão indexadas pelas etiquetas que contêm. Ou seja, se pretendemos saber a que ordens o evento de determinada etiqueta pode interessar, recorremos a este conjunto.

Também visíveis estão os seguintes métodos:

```
public boolean tagIn(Tag tag, Address address)
```

```
public boolean tagOut(Tag tag, Address address)
```

É através destes métodos que são comunicadas, à *pool* de ordens, as alterações no estado do ambiente, ou seja, quando entra ou sai determinada etiqueta do alcance de determinado leitor.

4.4.3.3 Comandos

A Figura 4.21 evidencia a representação das classes que se encontram directamente ligadas aos comandos disponíveis.

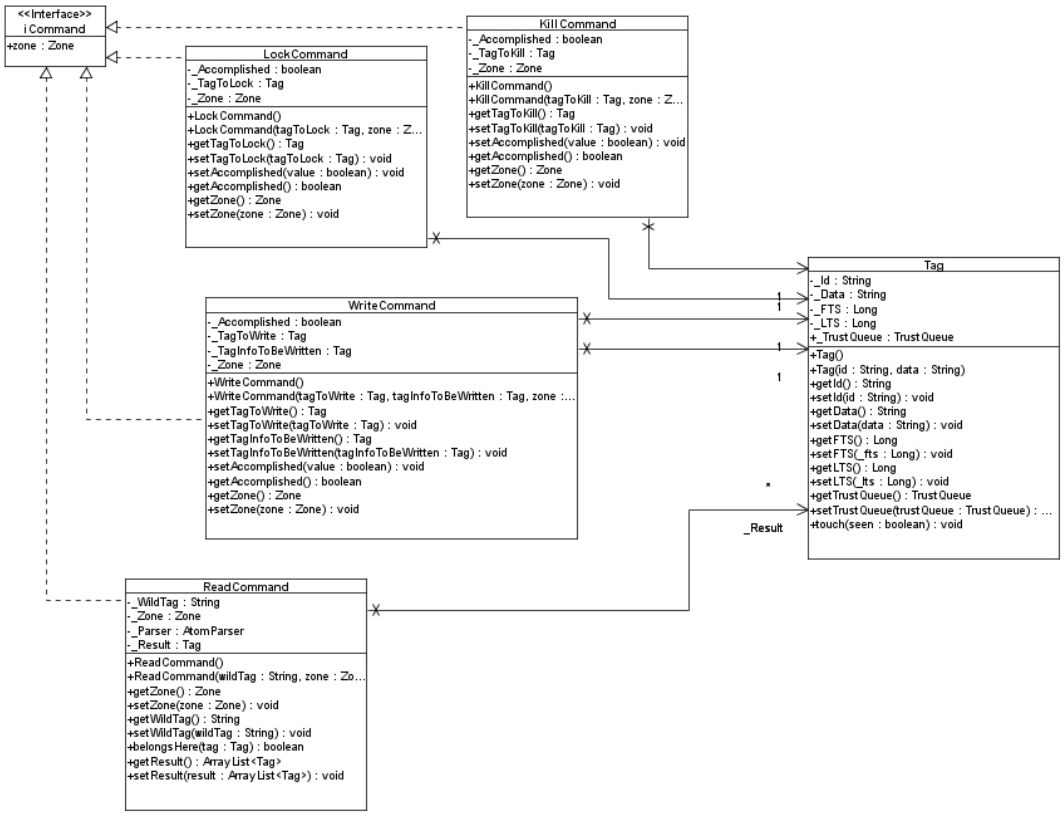


Figura 4.21 - Diagrama de classes de Comandos

Como podemos verificar, existe uma classe para cada um dos 4 comandos possíveis, implementando todas elas a Interface Java `iCommand`, que as identifica como sendo comandos.

Podemos ainda verificar que todas elas contêm um campo denominado `_Zone`, assim como o campo `_Tag`. Para um comando, contrariamente ao que acontecia

nas condições, quando queremos executar um comando é sobre uma etiqueta específica, daí a inclusão da classe `Tag`. O campo `_Zone` indica qual a zona de acção do comando.

4.4.3.4 Permissões

Na Figura 4.22 é apresentado o diagrama de classes representando as classes envolvidas no controlo de permissões.

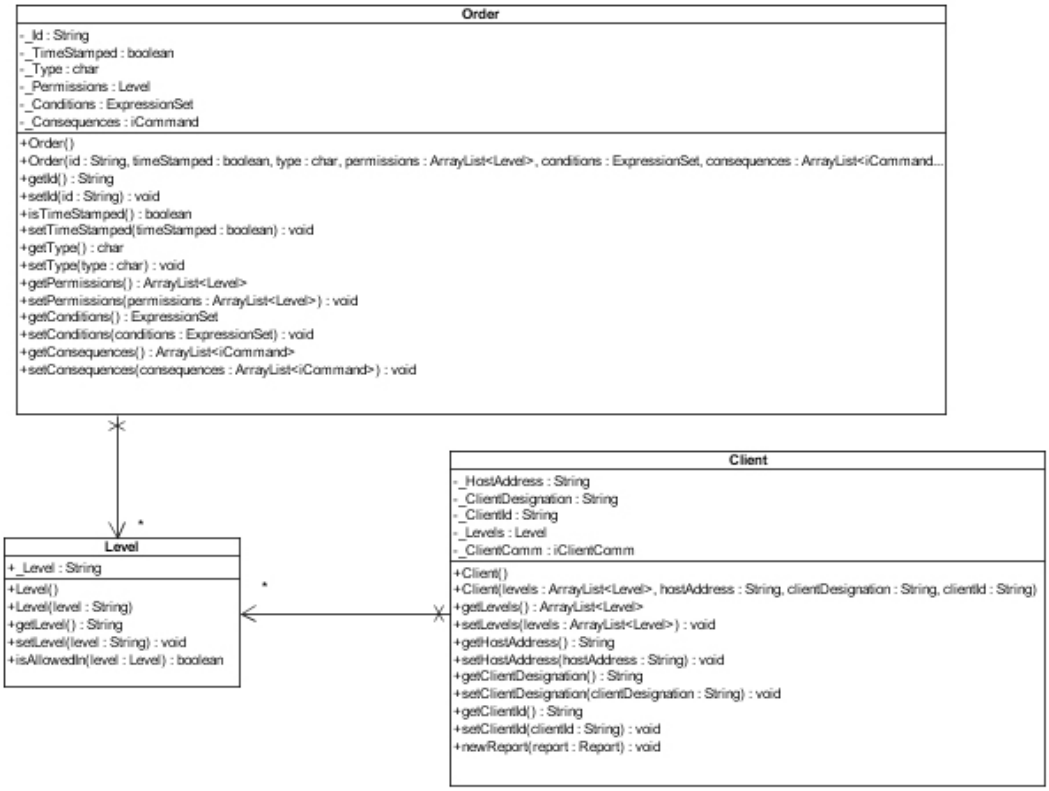


Figura 4.22 - Diagrama de classes de Permissões

Na realidade, e como se pode verificar no diagrama apresentado, as permissões não serão mais do que atributos das classes de cliente e de ordem. É de salientar ainda, na classe `OrderPool`, cuja classe está presente na Figura 4.20, a existência do seguinte método:

```
public void newSubscription(Client client, String orderId)
```

Será este o método que vai garantir, ou negar, o acesso de um cliente a determinada ordem.

É também importante referir que, caso o cliente `n` tenha permissão suficiente para subscrever esta ordem, será lançada a excepção `NotAllowedException`. Da mesma forma, quando é tentada uma subscrição de uma ordem inexistente, é lançada a excepção `UnkownOrderException`.

4.4.4 Leitores

Quanto aos leitores, existem também diversos pontos de interesse, mais concretamente relacionados com os seguintes temas:

- Inquisição de endereços e criação de novos leitores
- Gestão de leitores
- Zonas

4.4.4.1 Criação de novos leitores

As classes envolvidas na adição automática de novos leitores podem encontrar-se explícitas na Figura 4.23.

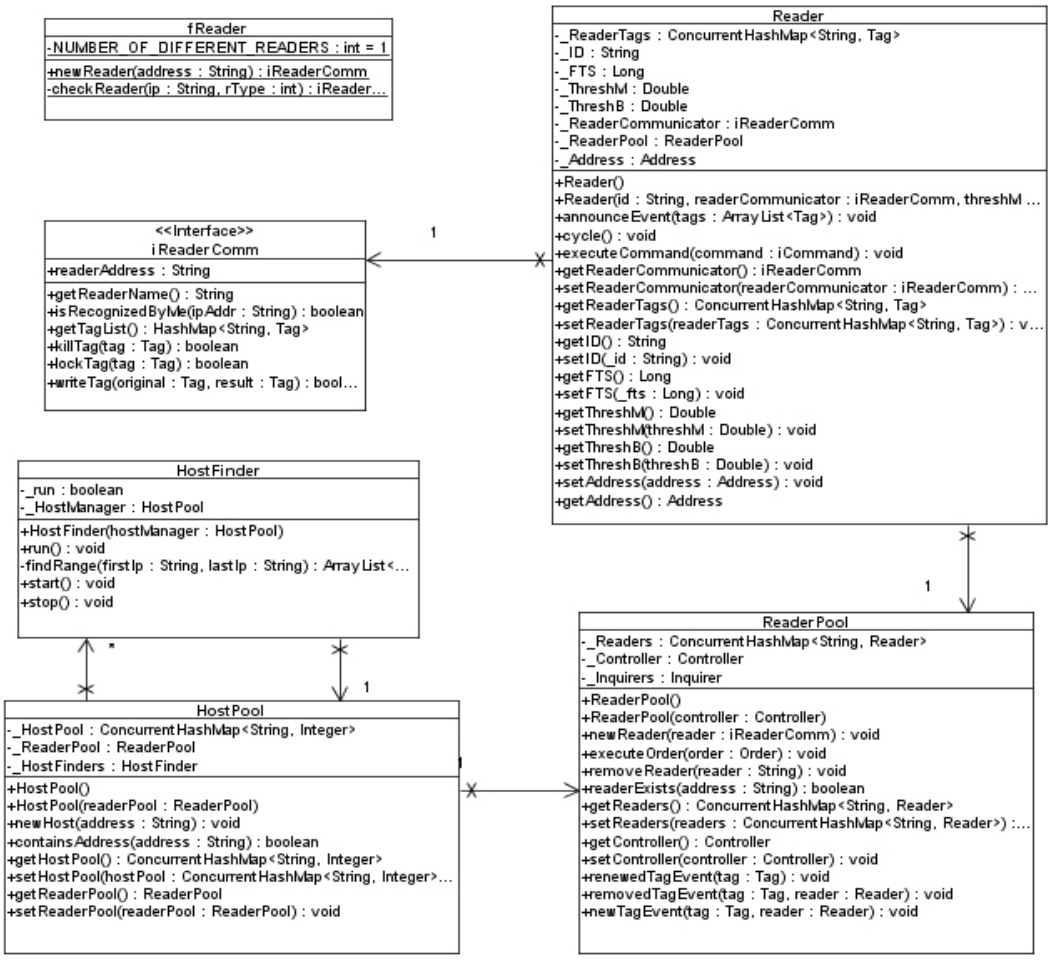


Figura 4.23 - Diagrama de classes de Novos Leitores

Nesta figura, podemos verificar a existência de duas *pools*, nomeadamente, a `HostPool` e a `ReaderPool`. A classe `HostPool` é a classe que trata da gestão de endereços pesquisáveis. Assim, esta contém, como se pode verificar na figura, o campo `_HostPool` que é onde são armazenados os endereços que poderemos ignorar (visto estarem já atribuídos) e um inteiro que indica durante quantos ciclos este endereço deve ser ignorado. Ao fim dessa quantidade de ciclos, uma nova visita a esse endereço será forçada, a bem da integridade.

Uma vez iniciado o sistema, e dependendo da quantidade de processadores disponíveis, são postas a correr uma ou mais *threads* de pesquisa de endereços,

que vão seguir o algoritmo apresentado para esta funcionalidade. Uma vez encontrado um endereço que responda, este é enviado para o método estático

```
public static void newReader(String address)
```

da classe `fReader`. Esta classe responsabiliza-se por verificar se existe alguma implementação da Interface `iReaderComm` capaz de lidar com o equipamento deste endereço. Caso encontre, retorna uma nova instância deste, retornando um valor nulo (`null`) caso contrário.

Assumindo que foi possível encontrar uma implementação de `iReaderComm` capaz de lidar com o equipamento do endereço em questão, é criada uma nova instância de `Reader`, onde é adicionada a instância existente da classe que implementa `iReaderComm`, sendo posteriormente adicionada à piscina de leitores, ou seja, à classe `ReaderPool`.

4.4.4.2 Gestão de leitores

Na Figura 4.24 encontra-se exposto o sub-diagrama das classes responsáveis pela gestão dos leitores. Considera-se como gestão de leitores, a manutenção dos leitores em si e ainda a pesquisa de etiquetas que se encontram no alcance destes.

Assim, tal como foi já mencionado, a gestão dos leitores prende-se com a eliminação destes assim que uma falha de comunicação ocorra. Mesmo que o leitor não tenha, realmente, desaparecido, tal não será grave pois o endereço deste será, a curto prazo, revisitado pelo sistema de adição de novos leitores.

No que concerne à manutenção das listas de etiquetas, isto é feito através da classe `Inquirer`. Uma instância de `ReaderPool` dispõe de uma ou mais instâncias da classe `Inquirer`, a qual ficará encarregue de questionar um conjunto de leitores pelas suas etiquetas. Cada `Inquirer` corre numa `Thread`

diferente, ficando o número de *threads*, e consequentemente o número de `Inquirers`, dependente da quantidade de processadores disponíveis.

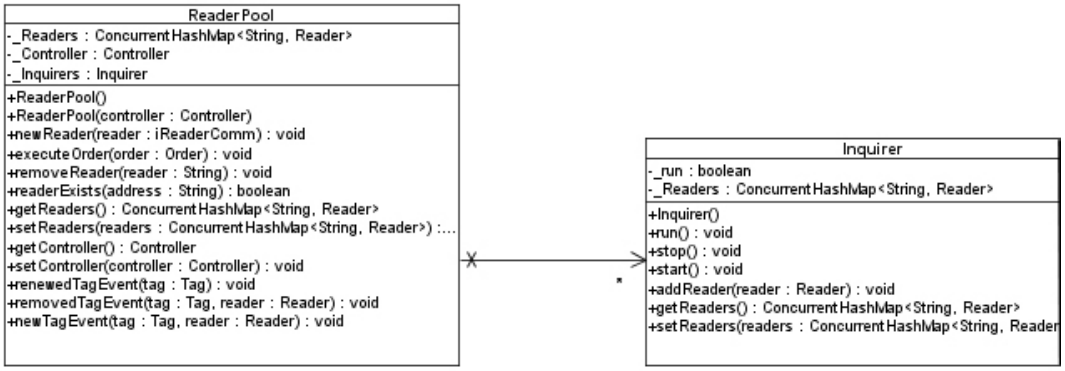


Figura 4.24 - Diagrama de classes de Gestão de Leitores

4.4.4.3 Zonas

A Figura 4.25 apresenta o diagrama de classe respeitante às classes que gerem o conceito de zona anteriormente descrito.

Como se pode verificar, cada instância da classe `Reader` contém uma instância da classe `Address`, que por sua vez terá uma instância da classe `BitSet`.

Será através destas duas classes que a decomposição do endereço de um leitor em cadeia de bits e, posteriormente, numa sequência de valores lógicos, será efectuada.

Quando se pretende saber se determinado endereço pertence, ou não, a uma zona, pode ser utilizado o método:

```
public boolean belongsToZone(Address address)
```

que dará exactamente essa indicação.

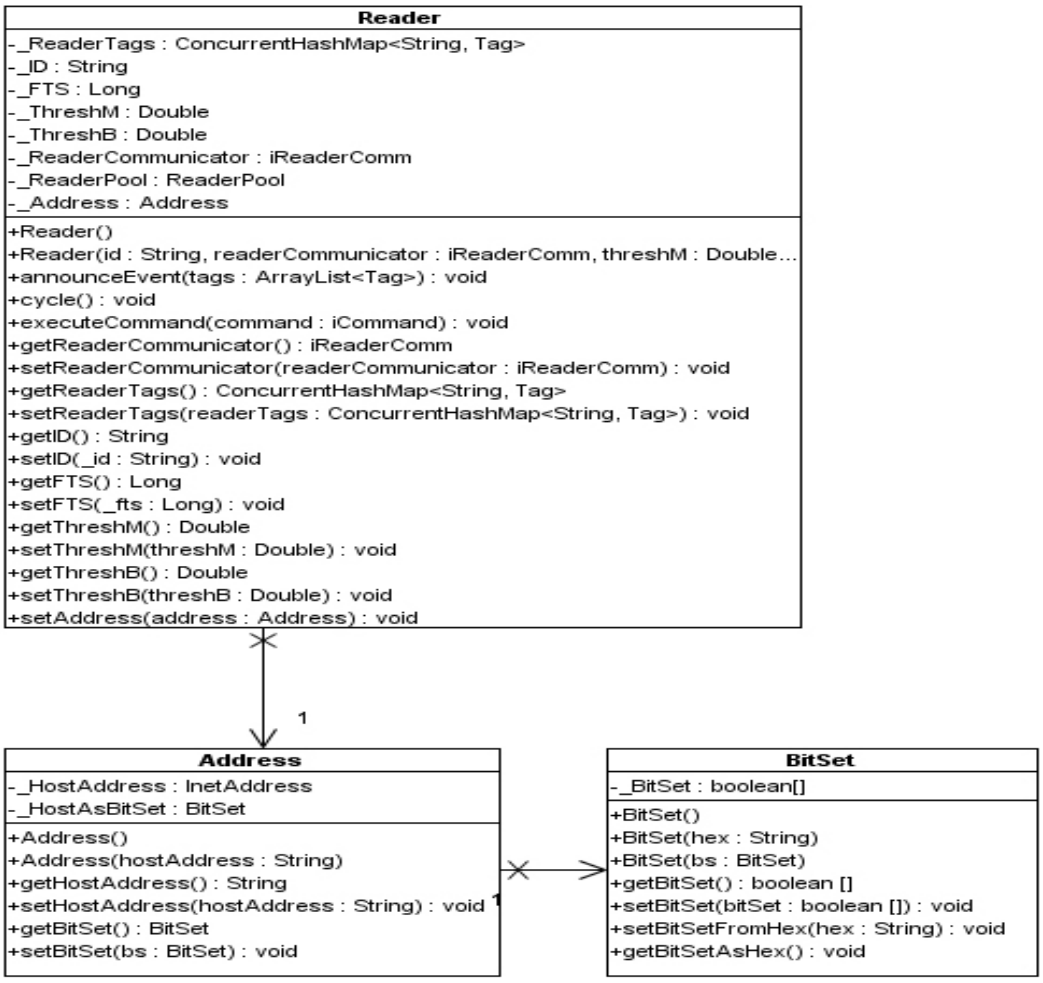


Figura 4.25 - Diagrama de classes de Zonas

4.4.5 Etiquetas

Relativamente às etiquetas, é nestas que se encontra uma das mais distintivas funções deste middleware, ou seja, a questão da confiança de leitura. Será em torno deste tema que esta secção se debruçará.

4.4.5.1 Confiança

No diagrama exposto na Figura 4.26 podemos encontrar as principais classes relacionadas com a definição de confiança acima tratada.

Aqui, podemos verificar que, tal como exposto em cima, cada instância de etiqueta, representada pela classe `Tag`, contém uma instância de fila de confiança, aqui representada pela classe `TrustQueue`.

De cada vez que o inquiridor de leitores questiona determinado leitor, é chamada, em todas as instâncias de `Tag` pertencentes a esse leitor, o método:

```
public void touch(boolean seen)
```

onde o parâmetro `seen` representa um valor lógico que será `true` caso a etiqueta tenha sido lida ou `false` caso contrário. O valor inserido no parâmetro deste método será, de seguida, inserido na fila de confiança.

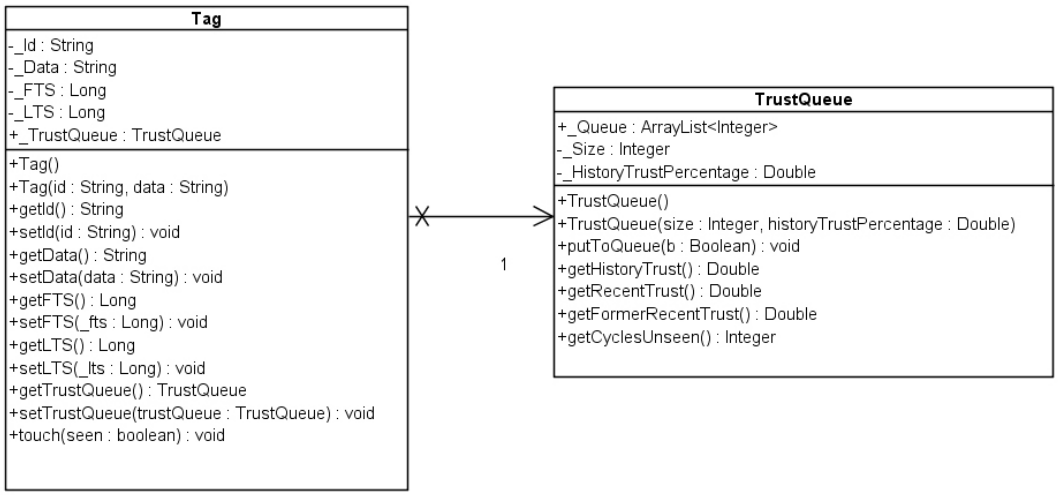


Figura 4.26 - Diagrama de classes relativas a confiança

Estando isto feito, a `ReaderPool` tratará de, de seguida, analisar os dados da `TrustQueue` da `Tag` de forma a verificar, baseada nos métodos já enunciados, se esta etiqueta deve ou não ser eliminada do sistema.

5 Discussão de Aplicabilidade

Neste capítulo será feita uma descrição do protótipo desenvolvido para avaliar a aplicabilidade desta solução, assim como a análise de algumas possibilidades de utilização da solução desenvolvida com este trabalho. Pretende-se explicar com maior detalhe em que circunstâncias surgiu a ideia e quais as premissas para o desenvolvimento deste trabalho.

5.1 Mote

Tal como referido inicialmente, este foi um projecto realizado em conjunto com a *Wipro Retail*, contribuindo esta, sobretudo, com os meios técnicos e conhecimentos específicos da área de retalho.

Existindo, na Wipro Retail, um grupo que se dedica em exclusivo à criação de projectos inovadores, a aplicação que tira partido deste *middleware* surgiu na sequência de um desses projectos.

O mote dessa ideia foi a utilização da tecnologia RFID na recepção e despacho de produtos pelos entrepostos dos retalhistas, de forma a agilizar estas operações.

Actualmente, esta operação é concretizada manualmente. Quando uma nova encomenda é recebida, a sua verificação e catalogação é feita recorrendo à tecnologia de código de barras. Esta verificação e catalogação, como se poderá compreender, é feita apenas até ao nível da caixa, não sendo viável aprofundar esta tarefa até ao nível do item.

Como facilmente se concluirá, esta é uma tarefa que se poderá tornar muito morosa, valendo a pena investir numa solução que de alguma forma consiga reduzir significativamente, não só o tempo destas operações, como a mão-de-obra necessária. Na Figura 5.27 podemos ver um esquema representativo deste tipo de configuração. Tanto a recepção como o despacho de encomendas são feitos manualmente, sendo de imediato registados no sistema de gestão do entreposto.

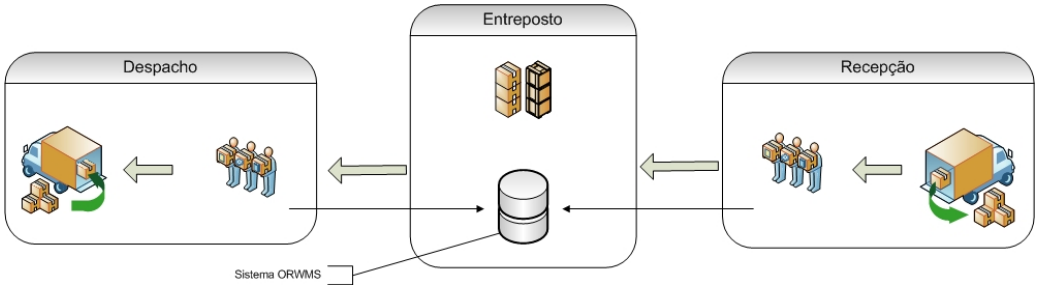


Figura 5.27 - Esquema típico da recepção e despacho de encomendas num entreposto

Assim sendo, foi pensado que a substituição dos códigos de barras por etiquetas RFID, não só nas caixas e paletes mas também em todas as unidades de produto, poderia ser uma solução a médio prazo. Naturalmente que, para um projecto de inovação, o factor custo não deve ser relevante, mantendo-se o foco, neste caso, na inserção de uma nova tecnologia num processo já existente.

5.2 Alterações ao Processo

Em termos de infra-estrutura, algumas alterações são necessárias no entreposto. Há a necessidade de serem colocados, nos portos de entrada e saída do entreposto, portais com leitores RFID, com uma configuração física adequada. Estes serão capazes de captar a identificação de cada uma das unidades de produto que por estes passem, em contraste com o que actualmente acontece, ou seja, apenas é feita a leitura ao nível da caixa de unidades, uma por uma. Com esta alteração, não só a informação seria consideravelmente mais detalhada como a captação e armazenamento de

informação se pode concretizar de forma praticamente instantânea. Na Figura 5.28 encontra-se esquematizado o

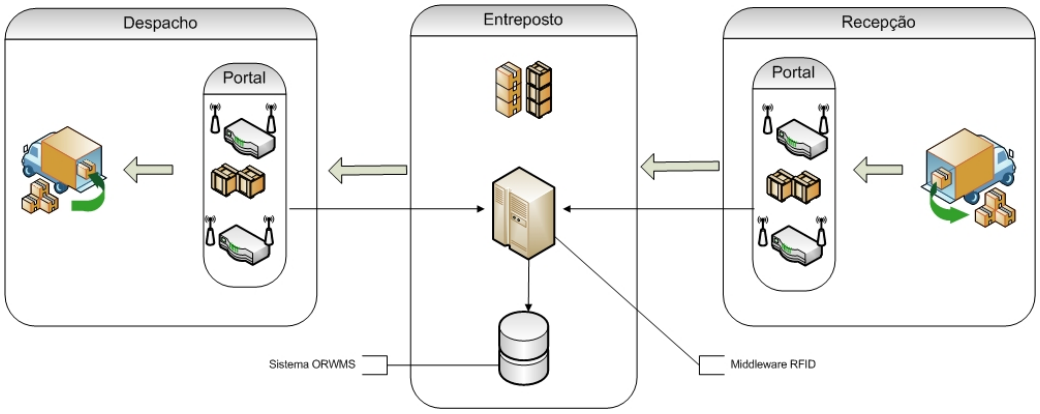


Figura 5.28 – Funcionamento com a introdução da tecnologia RFID

5.3 Justificação

No ponto em que se planeava a integração dos diversos dispositivos de leitura com os sistemas actualmente presentes num entreposto, identificaram-se diversos problemas:

- Como tratar a grande quantidade de dados proveniente de um sistema deste tipo?
- Como ultrapassar os problemas físicos da tecnologia?
- Como tornar este sistema suficientemente genérico e portável, de forma a promover a reutilização?
- Sendo a oferta de especialistas nesta área muito reduzida, como garantir que em futuras implementações seriam executadas de forma simples?

Resolveu-se que o ideal seria fazer uma aposta um pouco maior na integração, em oposição à possibilidade de fazer uma integração comum e específica, passando esta a ser, por si só, uma solução independente. Assim surgiu a necessidade de criação deste *middleware* que, tal como o tema deste trabalho indica, teria de ser portável, de forma a ser simples de incluir numa qualquer

arquitectura, onde fizesse sentido a inclusão da tecnologia RFID. Teria também a necessidade de ser genérico, para que a sua integração com a arquitectura legada dos sistemas de destino fosse possível. Outro ponto seria a autonomia, de forma a evitar a necessidade de especialistas numa tecnologia tão pouco explorada como é o caso da RFID. Finalmente, num sistema propenso a tomar grandes dimensões, a escalabilidade é um factor a ter em atenção, dada a potencial quantidade de dados que teriam de ser geridos por esta ferramenta.

5.4 Aplicação Prática

Tal como referido, ao *middleware* desenvolvido foi atribuída a responsabilidade pela interligação dos novos sistemas de leitura com os sistemas legados. Sendo o foco da *Wipro Retail* as soluções para retalho disponibilizadas pela *Oracle*, a integração é feita, neste caso, com o *ORWMS*[4].

Desta forma, e aplicando este caso ao middleware desenvolvido, as interfaces de cliente encontram-se ligadas ao *ORWMS*, sendo este o seu único cliente e tendo, naturalmente, todas as permissões de acesso às ordens. Na interface de leitura encontram-se ligados os leitores instalados nos portais, tanto de entrada como de saída do entreposto, passando estes a gerar os dados de entrada no sistema cliente.

Em termos das zonas, definiu-se que cada uma destas seria representada por um portal de entrada ou saída. Assim, cada uma destas zonas estará associada a um portal de entrada ou saída, podendo cada um destes portais conter 1 ou mais leitores.

Uma possível definição de ordem será, por exemplo, a apresentada na Figura 5.29.

[4] *Oracla Retail Warehouse Management System* . Sistema de gestão de armazéns. É comummente utilizado em entrepostos de retalhistas.

- Ordem: **wms_1**
- Permissão: **wms**
- Condições: {
  - Todos: {}
  - Qualquer: {***@192.168.10.0;**}
  - Nenhum: {}}
- Consequências:{
  - Comando de Leitura: {

Ler(*@192.168.10.0);}}

Figura 5.29 - Exemplo de definição de ordem

Supondo que os endereços dos leitores existentes nos portais de recepção de encomendas servem a máscara `192.168.10.0`, o que esta ordem indica é que se pretende ler todos as etiquetas que sejam detectadas nestes portais.

Um exemplo de relatório para esta ordem seria o apresentado na Figura 5.30.

- Relatório: **rep_1**
- Ordem: **wms_1**
- Zona: **192.168.10.0**
  - Comandos:
    - **Ler(*@192.168.10.0) {**

310EB6AF97737B0A7C000000

310080B1559D895BDC000000

3105C60F6B9769B4D3000000

31092882D089808144000000

3108FB4B2773425716000000

3105DF168BFCC12D02000000}

Figura 5.30 - Exemplo de relatório para a ordem da Figura 5.29

Neste relatório, da Figura 5.30, temos visíveis as etiquetas que foram lidas, segundo as regras definidas na ordem `wms_1`. É de salientar que, uma vez que da ordem apenas constava uma zona, também no relatório aparece apenas uma zona. Isto acontece independentemente do número de leitores que esta inclua.

Encontra-se, na Figura 5.31, outro exemplo de ordem possível para o sistema descrito.

- Ordem: **wms_2**
- Permissão: **wms**
- Condições: {
  - Todos: {
    (3108*@192.168.11.0 OU 3105*@192.168.11.0);}
  - Qualquer: {}
  - Nenhum: {
    Não(3108*@192.168.11.0 OU 3105*@192.168.11.0);}}
- Consequências: {
  - Comando de Escrita: {
    Matar(*@192.168.11.0);}}

Figura 5.31 - Exemplo de definição de ordem

O pretendido com esta ordem é que, supondo que à zona `192.168.11.0` correspondem os portais de saída do entreposto, em todas as etiquetas começadas por `3108` ou `3105` seja executado o comando `Matar`. Este comando deve ser executado quando apenas e só produtos com estas etiquetas sejam detectados na zona referida, devendo ser executado, então, em todas as etiquetas que estejam no alcance. Um dos possíveis motivos para uma ordem deste género seria, por exemplo, o facto de as etiquetas estarem a ser utilizadas para uma encomendas específica apenas a nível interno. Daí resulta a necessidade de, aquando da saída dessa encomenda, as etiquetas desta sejam desactivadas.

Para esta ordem, um possível relatório seria aquele que se encontra patente na Figura 5.32.

- Relatório: **rep_2**
- Ordem: **wms_2**
- Zona: **192.168.11.0**
    - Comandos:
        - **Matar(*@192.168.11.0)** {
          3108B6AF97737B0A7C000000
          310880B1559D895BDC000000
          3105C60F6B9769B4D3000000
          31052882D089808144000000
          3108FB4B2773425716000000
          3105DF168BFCC12D02000000}

Figura 5.32 - Exemplo de relatório gerado para a ordem da Figura 5.31

Na Figura 5.32 podemos verificar quais as etiquetas afectadas pelo comando dado pela ordem da Figura 5.31. É de salientar que, caso existissem mais comandos, estes apareceriam descriminados no relatório.

Com a aplicação disto num entreposto real estas tarefas passariam a realizar-se de forma praticamente instantânea, implicando isto não só uma redução no tempo de resposta de um entreposto a novas encomendas de saída, como um menor tempo de retenção dos transportadores. De facto, uma alteração deste tipo poderia, no caso de novos entrepostos, justificar uma arquitectura do espaço físico completamente diferente. Outro factor a ter em conta é a redução de mão-de-obra, passando os processos de recepção e despacho de encomendas a ser feitos de forma automática. Disto não só advêm as vantagens directas, como a redução de custos, como também outras como uma inferior taxa de erro, natural de uma tarefa realizada manualmente.

5.5 Outras possibilidades

Ainda com base no exemplo anterior, outra possibilidade de aplicação deste sistema, ainda num entreposto de retalhista, seria na gestão do stock deste. Mesmo em operações mais sensíveis, como no *put-away*[5] ou no *picking*[6], se pode imaginar uma infinidade de possibilidades para a utilização desta solução. Se pensarmos na possibilidade de ter um entreposto totalmente coberto por leitores RFID, poderíamos, a partir desse momento, abandonar grande parte das restrições existentes para as acções de *put-away*, passando a ter a possibilidade de utilizar um armazenamento quase completamente *had-hoc*.

Quando fosse a altura de realizar o *picking*, o sistema saberia, recorrendo ao conhecimento adquirido através do nosso *middleware*, exactamente em que zona se encontraria cada uma das unidades de produto, podendo assim entregar as ordens de *picking* aos funcionários sem perdas derivadas de uma acção de *put-away* errada.

Mesmo em termos de gestão de stock, na própria loja, este poderia ser controlado em tempo real, sem necessidade de existir uma grande equipa de gente a fazer esse controlo e contagem manualmente, para obter resultados apenas no final do dia.

As possibilidades de implementação deste sistema não se cingem, no entanto, exclusivamente à área do retalho.

[5] Acção desenvolvida num entreposto, que consiste na arrumação ordenada e planeada das encomendas recebidas no armazém .

[6] Acção desenvolvida num entreposto, que consiste na recolha de produtos do armazém de forma a criar as encomendas que posteriormente serão enviadas para as lojas.

6 Conclusões e trabalho futuro

Passa-se agora à descrição das principais conclusões do trabalho efectuado, nas abordagens defendidas, no desenvolvimento efectuado, e nas possíveis melhorias futuras do sistema apresentado.

Aqui contextualizam-se também os resultados alcançados no momento actual das Tecnologias de Informação, no que respeita à possibilidade de implementação de um sistema semelhante.

6.1 Conclusões

6.1.1 Acerca da tecnologia RFID

Grande parte das conclusões a retirar em relação à tecnologia RFID foram sendo apresentadas ao longo deste trabalho. Apesar disto, poder-se-ão fazer algumas considerações finais, de forma resumida.

A grande questão relativa à evolução desta tecnologia prende-se, sobretudo, com a sua capacidade de ultrapassar os diferentes níveis de obstáculos com que, actualmente, se defronta, tal como foi apresentado em capítulo próprio.

Apesar disto, e dada a grande potencialidade desta tecnologia, é opinião comum que grandes avanços se verificarão a curto prazo. No entanto, convém ressalvar que, entre as três possíveis posições quanto a esta tecnologia: a negativista, a moderada e a optimista, todas apresentam motivos válidos. Convém, no entanto, não cair em extremismos, ou seja, nem assegurar que, a curto/médio prazo as etiquetas RFID serão completamente ubíquas, nem garantir que esta tecnologia

entrará num beco sem saída tecnológico que, eventualmente, a conduzirá à extinção. Existe já a certeza razoavelmente generalizada que, a curto/médio prazo, esta tecnologia poderá trazer grandes benefícios para certas áreas específicas mas, dificilmente, atingirá a globalidade das áreas de negócio. Os motivos para tal prendem-se, tal como já discutido, com factores económicos que inviabilizam que a curto ou médio prazo seja possível a inclusão desta tecnologia em bens de pouco valor. Também a questão, a quem caberá a colocação das etiquetas nos objectos, tem retardado em algumas áreas, como a do retalho, que a implementação aconteça naturalmente.

6.1.2 Acerca do middleware proposto

Como foi proposto desde início, o objectivo deste trabalho foi a concepção e desenvolvimento de um *middleware* RFID que fosse portável, genérico, autónomo e escalável. Em termos de portabilidade, para a obtenção deste objectivo, ou seja, para que não ficasse condenado à utilização com aplicações específicas, nem apenas em determinadas áreas de negócio, foram criadas diversas interfaces de ligação com o *middleware*. Naturalmente que, quanto mais portável pretendemos que determinada aplicação seja, mais complexa será a sua implementação em qualquer sistema. No entanto, certos sacrifícios têm de ser feito em prol de um bem maior que é o da portabilidade total e capacidade de integração com qualquer aplicação.

Em relação ao segundo objectivo, a generalidade do *middleware*, que é caracterizada pela capacidade de adicionar qualquer tipo de leitor ao sistema, o mesmo tipo de sacrifício é necessário para obter o mesmo tipo de resultado, ou seja, será necessário mais esforço de implementação para tipos de leitores novos, ficando no entanto aberta a possibilidade de inclusão de qualquer tipo de leitor, actual ou futuro. Convém referir o facto de, não estando fechada a possibilidade de novas operações, com o desenvolver da tecnologia, virem a ser possíveis, tornar-se-ão então necessárias alterações ao núcleo do *middleware*.

Quanto à interacção dos utilizadores com o sistema, em termos de configuração, esta nunca esta é feita directamente no *middleware*. Todo o tipo de interacção passa pelas configurações iniciais, quer do *middleware* em si quer do leitor. Assim sendo, é possível afirmar que este se encontra num nível razoável de autonomia.

No que concerne à escalabilidade do *middleware*, que esteve sempre identificado como sendo potencialmente o objectivo mais difícil de atingir, actualmente a solução proposta poderá, de facto, não ter atingido completamente os requisitos definidos. Existe, para tal, a necessidade de efectuar testes em ambiente real, onde a quantidade de eventos de um sistema e situações de carga reais fossem possíveis. Assim sendo, e dada a impossibilidade de a curto prazo efectuar estes testes, este objectivo fica no plano de trabalhos futuros a efectuar sobre o *middleware*. Contudo, existem já sugestões com vista no possível melhoramento deste ponto. Uma solução possível, no sentido de fazer a distribuição de carga, não aumentado ao número e capacidade dos servidores, seria, alternativamente, ir distribuindo a carga pelos leitores, visto estes terem a maior parte do tempo a sua capacidade de processamento e armazenamento subaproveitadas. De qualquer das formas, como esta opção não é ainda norma, terá de ser então, ainda, o *middleware* a suportar com toda a carga.

6.2 Trabalho futuro

É importante, nesta fase, perspectivar o que se prevê que possam constituir como próximos passos, novas ideias e possíveis melhoramentos para uma futura versão:

- Inclusão da possibilidade de utilizar a tecnologia para localização por triangulação. Com o *middleware* actual, apenas é possível utilizar a localização por antena. Seria interessante, com o avanço da tecnologia, vir a incluir a possibilidade de localização por triangulação;

- Possibilidade de executar comandos mais complexos. Actualmente, as ordens de escrita são feitas recorrendo, apenas, ao código identificador da etiqueta. Seria interessante dar a possibilidade de, por exemplo, dar a possibilidade de apenas se escrever numa etiqueta X se uma outra etiqueta Y se encontrar em determinado local. Outro exemplo, este que trará maior complexidade a uma possível solução, seria a possibilidade de ordens de leitura encadeadas. Um exemplo seria, apenas escrever numa etiqueta X, se também for possível escrever numa etiqueta Y. Uma vez que as ordens são sequenciais, haveria sempre a possibilidade de, quando as condições fossem testadas, o resultado ser verdadeiro, mas aquando da escrita, este já não se verificar. Em determinadas situações, como o exemplo de matar uma etiqueta, as possibilidades de reverter um comando podem ser praticamente nulas;
- Inclusão da possibilidade de utilizar estruturas lógicas na definição das condições, de forma a facilitar a criação de condições mais complexas, dar a possibilidade de utilizar estruturas do tipo "se, então, senão", ou mesmo ciclos, seria uma boa inclusão neste *middleware*;
- Desenvolvimento de um sistema melhorado de resistência a falhas, em especial de comunicação, que adicione maior robustez a este sistema. Actualmente, e visto o real foco deste trabalho ser lógica central do *middleware*, a preocupação com a robustez deste foi deixada para uma próxima versão. Um exemplo de falha possível é no caso em que, por falha de comunicação, um relatório não seja entregue ao seu destinatário. Nesta situação, não existe nenhum mecanismo de recuperação;
- Implementação da distinção entre ordens activas e passivas, possibilitando de forma controlada que um cliente possa, directamente, dar uma ordem que lhe seja exclusiva;

- Adicionar a capacidade de distribuição do sistema;
- Optimização da paralelização de tarefas, com vista no já mencionado melhoramento do objectivo escalabilidade, a ser levada a cabo numa próxima iteração.

Certamente que muitas outras melhorias, algumas delas, quem sabe, óbvias, ficam aqui por referir. No entanto, estas foram as identificadas conforme o desenvolvimento e respectivos testes foram feitos. Assim, com efeito, enquanto alguns melhoramentos fazem sentido apenas numa próxima versão, outros devem ser considerados em iterações de desenvolvimento desta.

7 Bibliografia e referências

7.1 Artigos, estudos e manuais

[1] Kourouthanassis, P., Can technology make shopping fun?, ECR Journal Vol. 3 No. 2, 2003

[2] Kourouthanassis, P., Roussos, G., Consumers and pervasive retail, Athens University of Economics and Business

[3] Kourouthanassis, P., Pervasive Retail as a Means of Enhancing Consumers Shopping Experience, ECR Journal, 2003

[4] Roussos, G., Kourouthanassis, P., Gryazin, E., Systems Architecture for Pervasive Retail, 2003

[5] Ondrus, J., Pigneur, Y., Coupling Mobile Payments and CRM in the Retail Industry

[6] Palmer, M., Seven Principles of Effective RFID Data Management, Progress Software, 2005

[7] Freeman, A., Heschk, T., The Cost Effective Innovation of Radio Frequency Identification in the Supply Chain Management Network , University of Pittsburg, 2007

[8] Alexander, K. et al, Focus on Retail: Applying Auto-ID to Improve Product Availability at the Retail Shelf, Auto-ID Center, MIT, 2002

[9] Hargraves, K., Shafer, S., Radio Frequency Identification (RFID) Privacy: The Microsoft Perspective, Microsoft, 2004

[10] Landt, J., The History of RFID, IEEE Potentials, 2005

[11] Wamba, S. F., Lefebvre, L.A., Integrating RFID Technology and EPC Network Into a B2B Retail Supply Chain: A Step Toward Inteligent Business Processes, Journal of Technology Management & Innovation, Vol. 2 No. 2, 2007

[12] Chappel, G. et al, Accenture Auto-ID in the Box: The Value of Auto-ID Technology in Retail Stores, Auto-ID Center, MIT, 2003

[13] Active and Passive RFID: Two Distinct, But Complementary, Technologies for Real-Time Supply Chain Visibility , Savi Technologies

[14] Active RFID: Selecting the Optimal Frequency for Global Applications , Savi Technologies

[15] Záboj, M., Using RFID in supply chain and retail store unit, Mendel University of Agriculture and Forestry Brno, Czech Republic, 2005

[16] Freescale Semiconductor Manufacturing Case Study, AeroScout,

[17] Siegemund, F., Flörkmeier, C., Interaction in Pervasive Computing Settings using Bluetooth-Enabled Active Tags and Passive RFID Technology together with Mobile Phones, Institute for Pervasive Computing, Zurich, Switzerland , 2003

[18] Harrison, M., EPC™ Information Service – Data Model and Queries, Auto-ID Center, MIT, 2003

[19] Estevez, A., RFID Vision in the DoD Supply Chain, Defense AT&L, 2005

[20] Doran, J., Compensating for less than 100% case Read Rates, Business Action Group, EPCGlobal, 2005

[21] Lee, Y., Antenna Circuit Design for RFID Applications, Microchip, 2003

[22] Brock, D. L., The Electronic Product Code™ (EPC™) as a Meta Code, Auto-ID Center, MIT, 2003

[23] Gross, S. et al, Requirements and Technologies for Ubiquitous Payment

[24] Mallinson, S. et al, Determining a Better Metric for RFID Performance in Environments with Varying Noise Levels

[25] Pisello, T., Shrinking the Supply Chain Expands the Return: The ROI of RFID in the Supply Chain, Alinean, 2006

[26] Khandelwal, G., ASAP: A MAC Protocol for Dense and Time-Constrained RFID Systems, EURASIP Journal onWireless Communications and Networking, Volume 2007, Article ID 18730, 2007

[27] Flöerkemeier, C., Lampe, M., RFID middleware design - addressing application requirements and RFID constraints, Joint sOc-EUSAI conference, 2005

[28] Wamba, F. et al, Enabling Intelligent B-to-B eCommerce Supply Chain Management Using RFID and the EPC Network: A Case Study in the Retail Industry, ICEC'06, 2006

[29] Wang, F., Temporal Management of RFID Data, Integrated Data Systems Department, Siemens Corporate Research, Proceedings of the 31st VLDB Conference, Trondheim, Norway, 2005

[30] Lee, Y. et al, EXPLORING THE IMPACT OF RFID ON SUPPLY CHAIN DYNAMICS, Proceedings of the 2004 Winter Simulation Conference, 2004

[31] Chawathe, S. S. et al, Managing RFID Data, Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004

[32] Wang, N. et al, RFID Personal Object Tracking Device, 2008

[33] Lee, J., Analysis of RFID AntiCollision Algorithms using Smart Antennas, 2004

[34] Page, R., A Low Power RF ID Transponder, 1993

[35] Esteves, D., Simões, R., Tests results performed with the RFID system, reference ALR - 8800, Wipro Retail – Universidade do Minho Consortium, 2007

[36] Results of the Public Online Consulation on Future RFID Technology Policy, Comission of the European Communities, 2007

[37] Viehland, D., Wong, A., The Future of Radio Frequency Identification, Journal of Theoretical and Applied Electronic Commerce Research, Vol. 2 Issue 2, 2007

[38] Leaver, S., Evaluating RFID Middleware, Forrester, 2004

[39] GS1 Glossary

[40] Technical Report: 860MHz–930MHz Class I Radio Frequency Identification Tag Radio Frequency & Logical Communication Interface Specification Candidate Recommendation, Version 1.0.1, Auto-ID Center, MIT, 2002

[41] Application Level Events 1.1 FAQ, EPCGlobal

[42] Application Level Events 1.1 Overview, EPCGlo bal, 2008

[43] Application Level Events Specification, Version 1.1 Part I: Core Specification, EPCGlobal, 2008

[44] Application Level Events Specification, Version 1.1 Part II: XML and SOAP Bindings, EPCGlobal, 2008

[45] Reader Interface Guide – All Fixed Readers, Alien Technologies, 2008

[46] Tabela de Atribuição de Frequências, Anacom, 2007

[47] Reader Protocol Standard, Version 1.1, EPCGlobal, 2006

[48] Rules for the structure and drafting of International Standards, ISO/IEC Directives, Part 2, 2004

[49] Decisão Da Comissão, sobre a harmonização do espectro de radiofrequências para os dispositivos de identificação por radiofrequências (RFID) que funcionam na banda de frequências ultra-elevadas (UHF), Jornal Oficial da União Europeia, 25/11/2006

[50] ISO/IEC 18000-6 Specification

[51] Regulatory status for using RFID in the UHF spectrum, EPCGlobal, 2007

[52] Tag Data Standards Version 1.3.1, EPCGlobal, 2006

[53] Rádio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz – 960 MHz, Version 1.1.0, EPCGlobal, 2005

[54] Low Level Reader Protocol 1.0.1 Conformance Requirements Document, EPCGlobal, 2007

[55] Low Level Reader Protocol FAQ, EPCGlobal , 2007

[56] Low Level Reader Protocol, Version 1.0.1 Ratified Standard with Approved Fixed Errata, EPCGlobal, 2007

[57] Kalischnig, E.: RFID: Making sense of sensor-based technology in Manufacturing & Logistics, Julho 2004.

7.2 Livros

[58] Gratton, Dean A., Developing Practical Wireless Applications, Elsevier Digital Press, 2007

[59] Kou, W., Yesha, Y., Enabling Technologies for Wireless E-Business, Springer, 2006

[60] Muller, M. Essentials of Inventory Management, Amacom, 2003

[61] Dowla, F., Handbook of RF and Wireless Technologies, Elsevier, 2004

[62] Hunt, V. D., Puglia, A., Puglia, M., RFID - A Guide to Radio Frequency Identification, John Wiley & Sons, 2007

[63] Mahmoud, Q. M., Middleware for Communications 1 Edition, Wiley, 2004

[64] Bhatt, H., Glover, B., RFID Essentials, O'Reiley, 2006

[65] Manish, B., Shahram, M., RFID Field Guide: Deploying Radio Frequency Identification Systems, Prentice Hall, 2005

[66] Sweeney II, P. J., RFID for Dummies, Wiley Publishing, 2005

[67] Finkenzeller, K., RFID Handbook, Second Edition, John Wiley & Sons, 2003

[68] Myerson, J. M., RFID in the Supply Chain, Auerbach Publications, 2007

[69] Thorton, F. et al, RFID Security, Syngress, 2006

[70] Lahiri, S., RFID Sourcebook, Prentice Hall, 2005

[71] Sohraby, K. et al, Wireless Sensor Networks, John Wiley & Sons, 2007

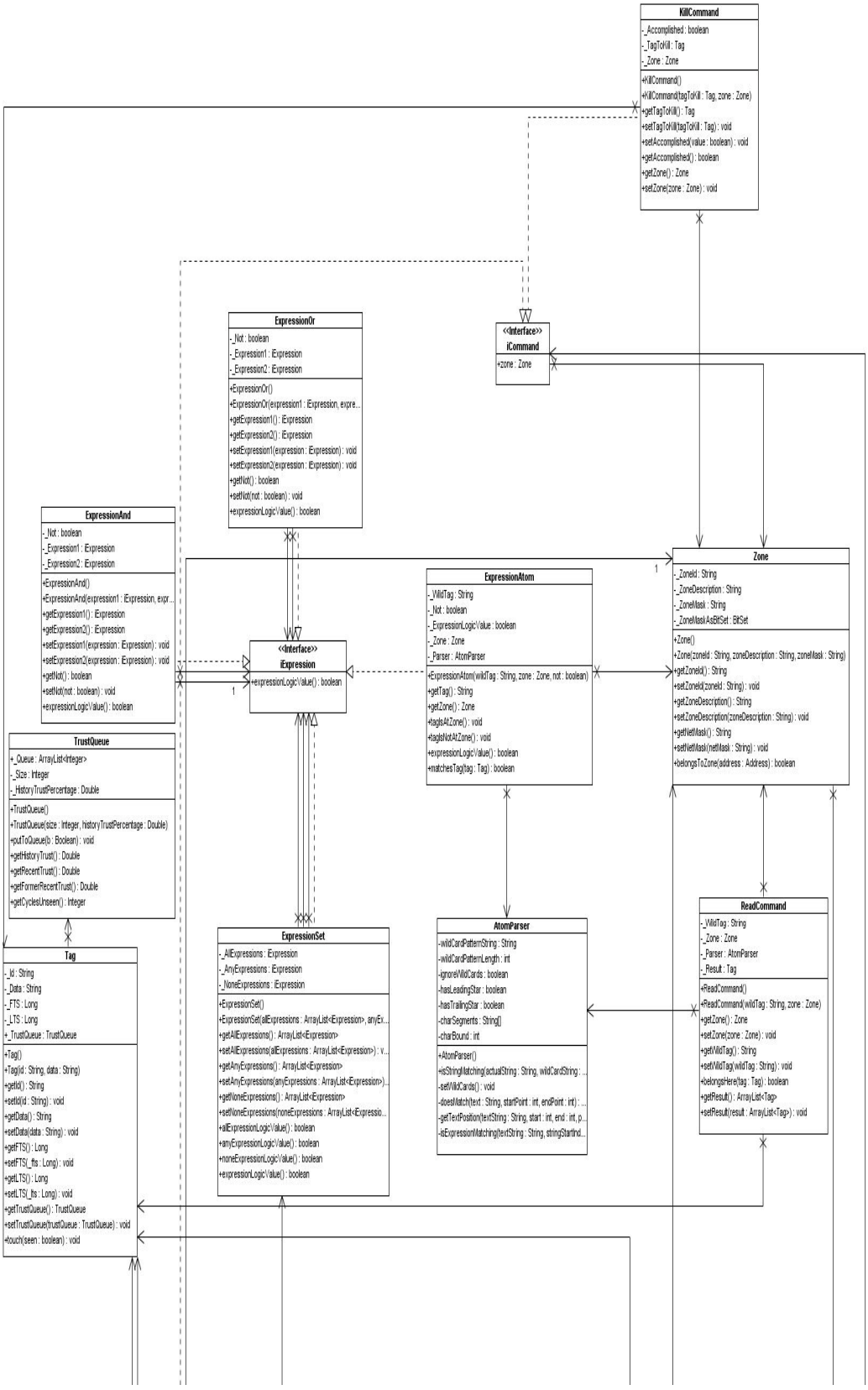
[72] Frischbier, S., Existing RFID Infrastructures Comparison and Evaluation, Parte de RFID Seminar, Databases and Distributed Group, Dept. of Computer Science, TU Darmstadt, Germany, 2006

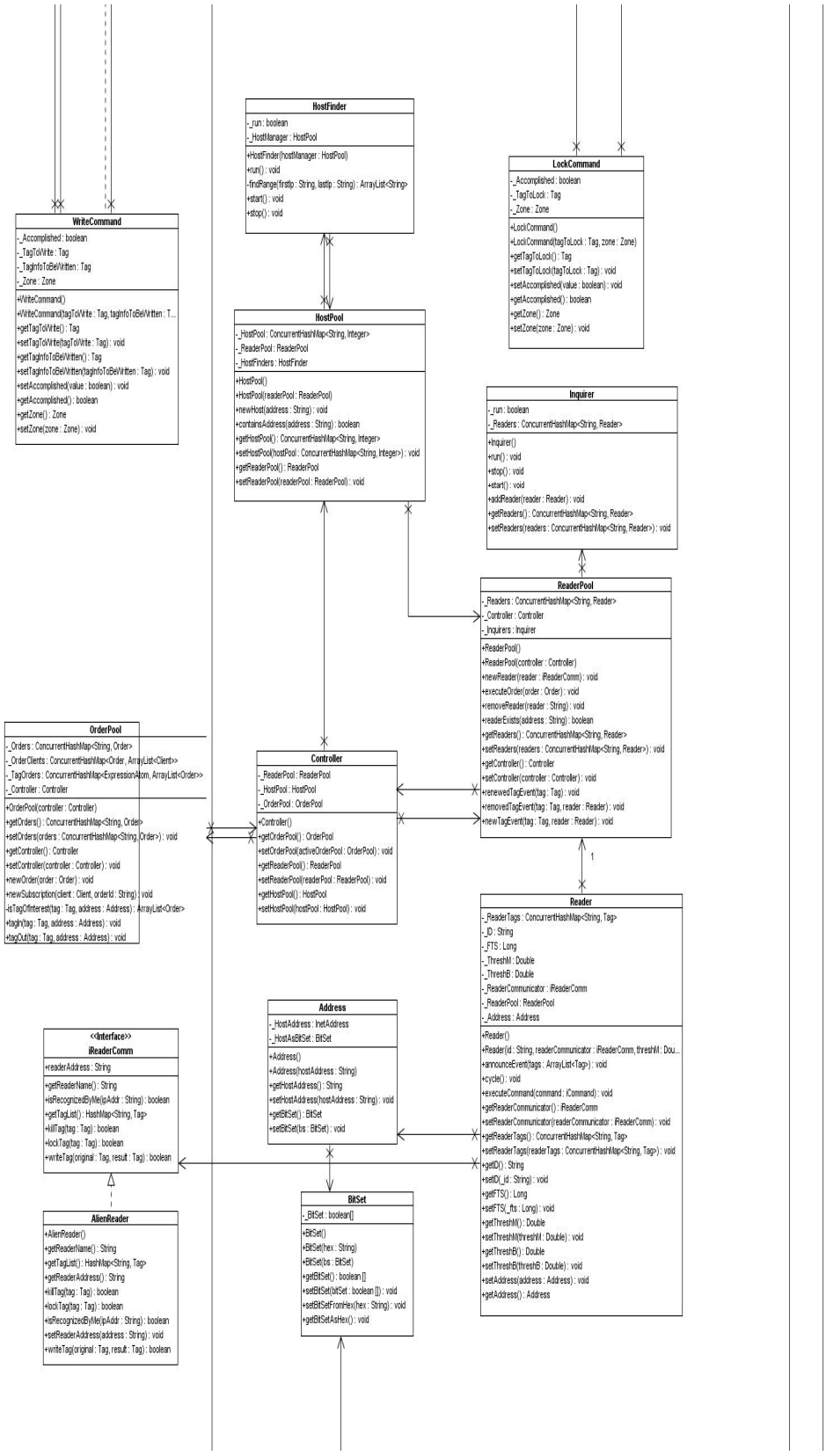
8 Anexos

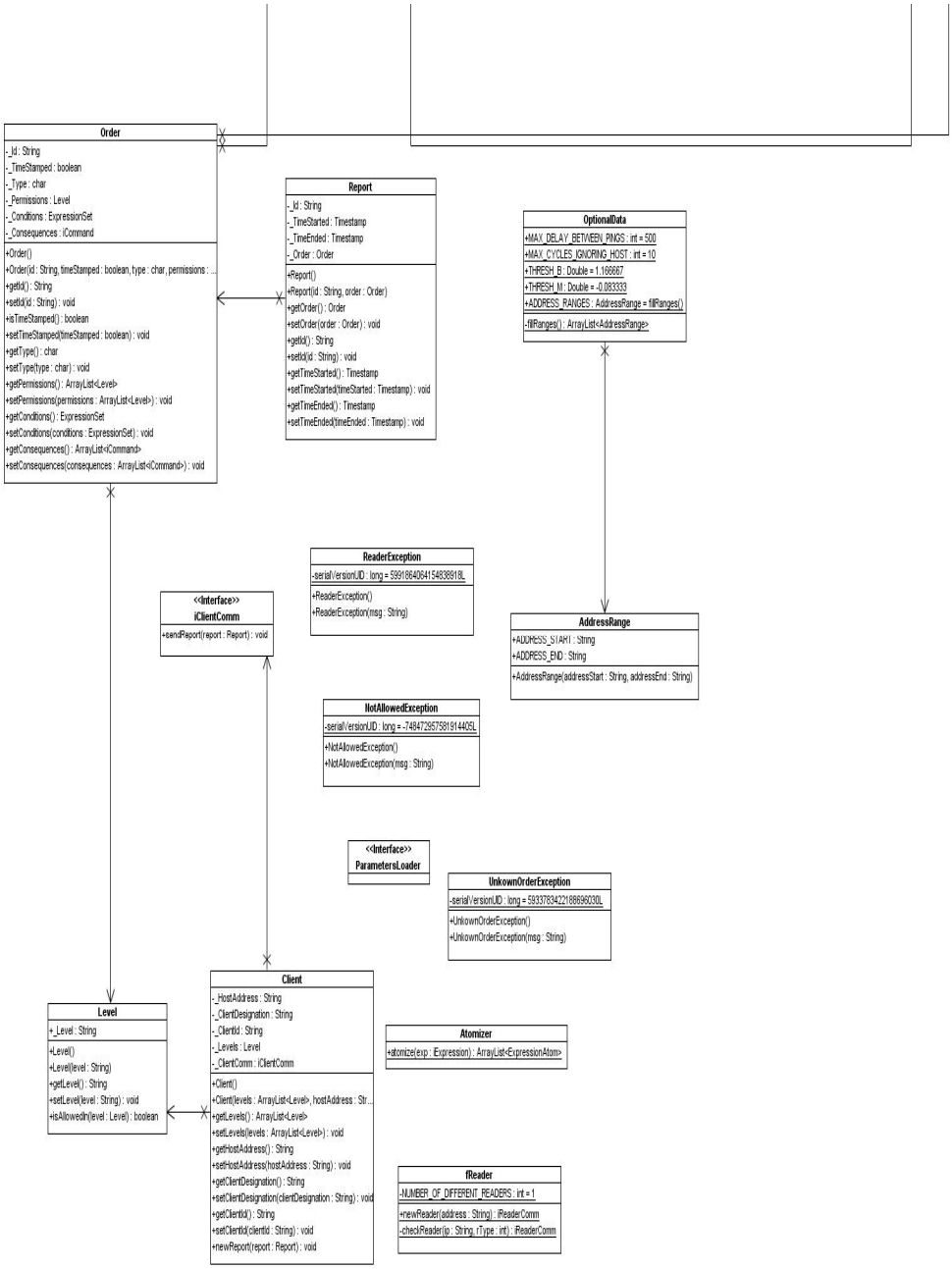
8.1 Diagrama de Classes

Os seguintes diagramas, dada a dimensão da solução, são apresentados de forma cortada. As ligações entre classes cortadas prosseguem na imagem seguinte. No final é apresentado o diagrama sem os cortes, de forma a ter uma perspectiva geral da solução.

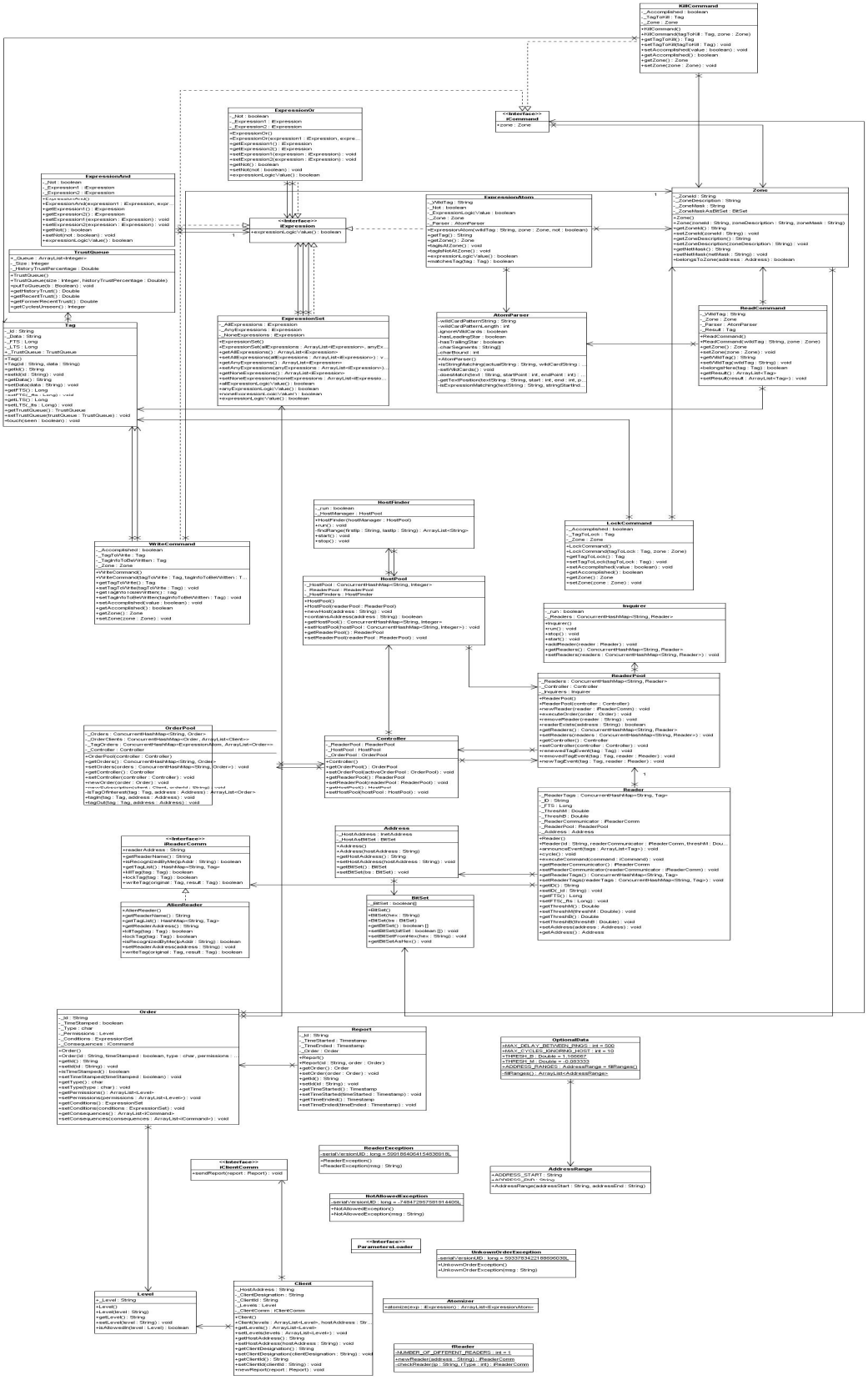
8.1.1 Diagrama Cortado


KillCommand
-_Accomplished : boolean
-_TagToKill : Tag
-_Zone : Zone
+KillCommand()
+KillCommand(tagToKill : Tag, zone : Zone)
+getTagToKill() : Tag
+setTagToKill(tagToKill : Tag) : void
+setAccomplished(value : boolean) : void
+getAccomplished() : boolean
+getZone() : Zone
+setZone(zone : Zone) : void
ExpressionOr
-_Not : boolean
-_Expression1 : Expression
-_Expression2 : Expression
+ExpressionOr()
+ExpressionOr(expression1 : Expression, expre...
+getExpression1() : Expression
+getExpression2() : Expression
+setExpression1(expression : Expression) : void
+setExpression2(expression : Expression) : void
+getNot() : boolean
+setNot(not : boolean) : void
+expressionLogicValue() : boolean
<<Interface>>
iCommand
+zone : Zone
ExpressionAnd
-_Not : boolean
-_Expression1 : Expression
-_Expression2 : Expression
+ExpressionAnd()
+ExpressionAnd(expression1 : Expression, expr...
+getExpression1() : Expression
+getExpression2() : Expression
+setExpression1(expression : Expression) : void
+setExpression2(expression : Expression) : void
+getNot() : boolean
+setNot(not : boolean) : void
+expressionLogicValue() : boolean
<<Interface>>
iExpression
+expressionLogicValue() : boolean
1
1
Zone
-_ZoneId : String
-_ZoneDescription : String
-_ZoneMask : String
-_ZoneMaskAsBitSet : BitSet
+Zone()
+Zone(zoneId : String, zoneDescription : String, zoneMask : String)
+getZoneId() : String
+setZoneId(zoneId : String) : void
+getZoneDescription() : String
+setZoneDescription(zoneDescription : String) : void
+getNetMask() : String
+setNetMask(netMask : String) : void
+belongsToZone(address : Address) : boolean
ExpressionAtom
-_WildTag : String
-_Not : boolean
-_ExpressionLogicValue : boolean
-_Zone : Zone
-_Parser : AtomParser
+ExpressionAtom(wildTag : String, zone : Zone, not : boolean)
+getTag() : String
+getZone() : Zone
+tagIsAtZone() : void
+tagIsNotAtZone() : void
+expressionLogicValue() : boolean
+matchesTag(tag : Tag) : boolean
TrustQueue
+_Queue : ArrayList<Integer>
-_Size : Integer
-_HistoryTrustPercentage : Double
+TrustQueue()
+TrustQueue(size : Integer, historyTrustPercentage : Double)
+putToQueue(b : Boolean) : void
+getHistoryTrust() : Double
+getRecentTrust() : Double
+getFormerRecentTrust() : Double
+getCyclesUnseen() : Integer
Tag
-_Id : String
-_Data : String
-_FTS : Long
-_LTS : Long
+_TrustQueue : TrustQueue
+Tag()
+Tag(id : String, data : String)
+getId() : String
+setId(id : String) : void
+getData() : String
+setData(data : String) : void
+getFTS() : Long
+setFTS(_fts : Long) : void
+getLTS() : Long
+setLTS(_lts : Long) : void
+getTrustQueue() : TrustQueue
+setTrustQueue(trustQueue : TrustQueue) : void
+touch(seen : boolean) : void
ExpressionSet
-_AllExpressions : Expression
-_AnyExpressions : Expression
-_NoneExpressions : Expression
+ExpressionSet()
+ExpressionSet(allExpressions : ArrayList<Expression>, anyEx...
+getAllExpressions() : ArrayList<Expression>
+setAllExpressions(allExpressions : ArrayList<Expression>) : v...
+getAnyExpressions() : ArrayList<Expression>
+setAnyExpressions(anyExpressions : ArrayList<Expression>)...
+getNoneExpressions() : ArrayList<Expression>
+setNoneExpressions(noneExpressions : ArrayList<Expressio...
+allExpressionLogicValue() : boolean
+anyExpressionLogicValue() : boolean
+noneExpressionLogicValue() : boolean
+expressionLogicValue() : boolean
AtomParser
-wildCardPatternString : String
-wildCardPatternLength : int
-ignoreWildCards : boolean
-hasLeadingStar : boolean
-hasTrailingStar : boolean
-charSegments : String[]
-charBound : int
+AtomParser()
+isStringMatching(actualString : String, wildCardString : ...
-setWildCards() : void
-doesMatch(text : String, startPoint : int, endPoint : int) : ...
-getTextPosition(textString : String, start : int, end : int, p...
-isExpressionMatching(textString : String, stringStartInd...
ReadCommand
-_WildTag : String
-_Zone : Zone
-_Parser : AtomParser
-_Result : Tag
+ReadCommand()
+ReadCommand(wildTag : String, zone : Zone)
+getZone() : Zone
+setZone(zone : Zone) : void
+getWildTag() : String
+setWildTag(wildTag : String) : void
+belongsHere(tag : Tag) : boolean
+getResult() : ArrayList<Tag>
+setResult(result : ArrayList<Tag>) : void


HostFinder
- _run : boolean
- _HostManager : HostPool
+HostFinder(hostManager : HostPool)
+run() : void
-findRange(firstIp : String, lastIp : String) : ArrayList<String>
+start() : void
+stop() : void
LockCommand
- _Accomplished : boolean
- _TagToLock : Tag
- _Zone : Zone
+LockCommand()
+LockCommand(tagToLock : Tag, zone : Zone)
+getTagToLock() : Tag
+setTagToLock(tagToLock : Tag) : void
+setAccomplished(value : boolean) : void
+getAccomplished() : boolean
+getZone() : Zone
+setZone(zone : Zone) : void
WriteCommand
- _Accomplished : boolean
- _TagToWrite : Tag
- _TagInfoToBeWritten : Tag
- _Zone : Zone
+WriteCommand()
+WriteCommand(tagToWrite : Tag, tagInfoToBeWritten : T...
+getTagToWrite() : Tag
+setTagToWrite(tagToWrite : Tag) : void
+getTagInfoToBeWritten() : Tag
+setTagInfoToBeWritten(tagInfoToBeWritten : Tag) : void
+setAccomplished(value : boolean) : void
+getAccomplished() : boolean
+getZone() : Zone
+setZone(zone : Zone) : void
HostPool
- _HostPool : ConcurrentHashMap<String, Integer>
- _ReaderPool : ReaderPool
- _HostFinders : HostFinder
+HostPool()
+HostPool(readerPool : ReaderPool)
+newHost(address : String) : void
+containsAddress(address : String) : boolean
+getHostPool() : ConcurrentHashMap<String, Integer>
+setHostPool(hostPool : ConcurrentHashMap<String, Integer>) : void
+getReaderPool() : ReaderPool
+setReaderPool(readerPool : ReaderPool) : void
Inquirer
- _run : boolean
- _Readers : ConcurrentHashMap<String, Reader>
+Inquirer()
+run() : void
+stop() : void
+start() : void
+addReader(reader : Reader) : void
+getReaders() : ConcurrentHashMap<String, Reader>
+setReaders(readers : ConcurrentHashMap<String, Reader>) : void
ReaderPool
- _Readers : ConcurrentHashMap<String, Reader>
- _Controller : Controller
- _Inquirers : Inquirer
+ReaderPool()
+ReaderPool(controller : Controller)
+newReader(reader : IReaderComm) : void
+executeOrder(order : Order) : void
+removeReader(reader : String) : void
+readerExists(address : String) : boolean
+getReaders() : ConcurrentHashMap<String, Reader>
+setReaders(readers : ConcurrentHashMap<String, Reader>) : void
+getController() : Controller
+setController(controller : Controller) : void
+renewedTagEvent(tag : Tag) : void
+removedTagEvent(tag : Tag, reader : Reader) : void
+newTagEvent(tag : Tag, reader : Reader) : void
OrderPool
- _Orders : ConcurrentHashMap<String, Order>
- _OrderClients : ConcurrentHashMap<Order, ArrayList<Client>>
- _TagOrders : ConcurrentHashMap<ExpressionAtom, ArrayList<Order>>
- _Controller : Controller
+OrderPool(controller : Controller)
+getOrders() : ConcurrentHashMap<String, Order>
+setOrders(orders : ConcurrentHashMap<String, Order>) : void
+getController() : Controller
+setController(controller : Controller) : void
+newOrder(order : Order) : void
+newSubscription(client : Client, orderId : String) : void
+isTagOfInterest(tag : Tag, address : Address) : ArrayList<Order>
+tagIn(tag : Tag, address : Address) : void
+tagOut(tag : Tag, address : Address) : void
Controller
- _ReaderPool : ReaderPool
- _HostPool : HostPool
- _OrderPool : OrderPool
+Controller()
+getOrderPool() : OrderPool
+setOrderPool(activeOrderPool : OrderPool) : void
+getReaderPool() : ReaderPool
+setReaderPool(readerPool : ReaderPool) : void
+getHostPool() : HostPool
+setHostPool(hostPool : HostPool) : void
1
Reader
- _ReaderTags : ConcurrentHashMap<String, Tag>
- _ID : String
- _FTS : Long
- _ThreshM : Double
- _ThreshB : Double
- _ReaderCommunicator : IReaderComm
- _ReaderPool : ReaderPool
- _Address : Address
+Reader()
+Reader(id : String, readerCommunicator : IReaderComm, threshM : Dou...
+announceEvent(tags : ArrayList<Tag>) : void
+cycle() : void
+executeCommand(command : ICommand) : void
+getReaderCommunicator() : IReaderComm
+setReaderCommunicator(readerCommunicator : IReaderComm) : void
+getReaderTags() : ConcurrentHashMap<String, Tag>
+setReaderTags(readerTags : ConcurrentHashMap<String, Tag>) : void
+getID() : String
+setID(_id : String) : void
+getFTS() : Long
+setFTS(_fts : Long) : void
+getThreshM() : Double
+setThreshM(threshM : Double) : void
+getThreshB() : Double
+setThreshB(threshB : Double) : void
+setAddress(address : Address) : void
+getAddress() : Address
Address
- _HostAddress : InetAddress
- _HostAsBitSet : BitSet
+Address()
+Address(hostAddress : String)
+getHostAddress() : String
+setHostAddress(hostAddress : String) : void
+getBitSet() : BitSet
+setBitSet(bs : BitSet) : void
<<Interface>>
IReaderComm
+readerAddress : String
+getReaderName() : String
+isRecognizedByMe(ipAddr : String) : boolean
+getTagList() : HashMap<String, Tag>
+killTag(tag : Tag) : boolean
+lockTag(tag : Tag) : boolean
+writeTag(original : Tag, result : Tag) : boolean
AlienReader
+AlienReader()
+getReaderName() : String
+getTagList() : HashMap<String, Tag>
+getReaderAddress() : String
+killTag(tag : Tag) : boolean
+lockTag(tag : Tag) : boolean
+isRecognizedByMe(ipAddr : String) : boolean
+setReaderAddress(address : String) : void
+writeTag(original : Tag, result : Tag) : boolean
BitSet
- _BitSet : boolean[]
+BitSet()
+BitSet(hex : String)
+BitSet(bs : BitSet)
+getBitSet() : boolean []
+setBitSet(bitSet : boolean []) : void
+setBitSetFromHex(hex : String) : void
+getBitSetAsHex() : void


Order
-_Id : String
-_TimeStamped : boolean
-_Type : char
-_Permissions : Level
-_Conditions : ExpressionSet
-_Consequences : ICommand
+Order()
+Order(id : String, timeStamped : boolean, type : char, permissions : ...
+getId() : String
+setId(id : String) : void
+isTimeStamped() : boolean
+setTimeStamped(timeStamped : boolean) : void
+getType() : char
+setType(type : char) : void
+getPermissions() : ArrayList<Level>
+setPermissions(permissions : ArrayList<Level>) : void
+getConditions() : ExpressionSet
+setConditions(conditions : ExpressionSet) : void
+getConsequences() : ArrayList<ICommand>
+setConsequences(consequences : ArrayList<ICommand>) : void
Report
-_Id : String
-_TimeStarted : Timestamp
-_TimeEnded : Timestamp
-_Order : Order
+Report()
+Report(id : String, order : Order)
+getOrder() : Order
+setOrder(order : Order) : void
+getId() : String
+setId(id : String) : void
+getTimeStarted() : Timestamp
+setTimeStarted(timeStarted : Timestamp) : void
+getTimeEnded() : Timestamp
+setTimeEnded(timeEnded : Timestamp) : void
OptionalData
+MAX_DELAY_BETWEEN_RNGS : int = 500
+MAX_CYCLES_IGNORING_HOST : int = 10
+THRESH_B : Double = 1.166667
+THRESH_M : Double = -0.083333
+ADDRESS_RANGES : AddressRange = fillRanges()
-fillRanges() : ArrayList<AddressRange>
ReaderException
-serialVersionUID : long = 5991864064154838918L
+ReaderException()
+ReaderException(msg : String)
<<Interface>>
iClientComm
+sendReport(report : Report) : void
AddressRange
+ADDRESS_START : String
+ADDRESS_END : String
+AddressRange(addressStart : String, addressEnd : String)
NotAllowedException
-serialVersionUID : long = -7484729575819144095L
+NotAllowedException()
+NotAllowedException(msg : String)
<<Interface>>
ParametersLoader
UnkownOrderException
-serialVersionUID : long = 5933783422188696030L
+UnkownOrderException()
+UnkownOrderException(msg : String)
Client
-_HostAddress : String
-_ClientDesignation : String
-_ClientId : String
-_Levels : Level
-_ClientComm : iClientComm
+Client()
+Client(levels : ArrayList<Level>, hostAddress : Str...
+getLevels() : ArrayList<Level>
+setLevels(levels : ArrayList<Level>) : void
+getHostAddress() : String
+setHostAddress(hostAddress : String) : void
+getClientDesignation() : String
+setClientDesignation(clientDesignation : String) : void
+getClientId() : String
+setClientId(clientId : String) : void
+newReport(report : Report) : void
Level
+_Level : String
+Level()
+Level(level : String)
+getLevel() : String
+setLevel(level : String) : void
+isAllowedIn(level : Level) : boolean
Atomizer
+atomize(exp : Expression) : ArrayList<ExpressionAtom>
fReader
-NUMBER_OF_DIFFERENT_READERS : int = 1
+newReader(address : String) : iReaderComm
-checkReader(ip : String, rType : int) : iReaderComm

8.1.2 Diagrama Completo



8.2 Normas e Regulamentação

Como foi abordado em diversas alturas neste trabalho, normas e regulamentos começam a existir, com o intuito de, de alguma forma, uniformizar esta tecnologia.

Sendo esta uma tecnologia que, apesar de ter visto o seu aparecimento há já algumas décadas, apenas recentemente teve os seus maiores desenvolvimentos, é natural que as normas e regulamentos se encontrem ainda, em alguns dos seus pontos basilares, num formato embrionário.

Neste anexo vamos então fazer referência às normas e regulamentos mais comummente associados a esta tecnologia, assim como às instituições e organizações a estes associados.

Não se pretende aqui, tão pouco, fazer uma dissecagem exaustiva destes documentos, mas sim, e com especial ênfase no código EPC, apresentar um breve resumo destes.

8.2.1 A EPC Global

A EPC-Global é uma entidade, criada pela GS1, com o intuito primário de estabelecer normas mundiais que uniformizem o desenho, implementação e concepção do código EPC e do EPCIS. Esta surgiu em 2003, dando seguimento ao trabalho até então desenvolvido no MIT, pelo *Auto-ID Center*. Desde então, várias normas e directivas têm sido lançadas por esta, enquanto o *Auto-ID Center* ficou apenas com a responsabilidade de realizar trabalho de investigação nesta área.

8.2.2 O Código EPC

Uma das principais lutas da EPC-Global prende-se com a implementação e alastramento do código EPC.

O EPC é, no fundo, um esquema de códigos identificativos de objectos, em certos aspectos semelhante ao comum código de barras. No entanto, e em oposição a estes, o código EPC tem como objectivo ser único por unidade de objecto, enquanto o código de barras era comummente utilizado para identificar uma classe de produtos.

Apesar de tal ainda, nem de longe nem de perto, se verificar, o grande objectivo da EPC-Global é fazer com que este código passe a ser de utilização geral nas etiquetas RFID.

O esquema utilizado para este código pode, naturalmente, tomar diversos tamanhos e formatos.

O tamanho admitido, comum a todos os formatos, tal como definido na norma, é o de 96 bits. No entanto é dada, também nesta norma, a possibilidade de utilizar diferentes tamanhos para cada formato individualmente, entre eles: 198 bits, para o formato SGTIN, 195 bits, para o formato SGLN, 170 bits, para o formato GRAI e, finalmente, 202 bits para o formato GIAI.

Um aspecto importante é o facto de, grande parte dos códigos EPC, terem um campo de nome “partição” e todos eles terem um campo de nome “filtro”.

O campo “partição” existe pois alguns campos de algumas codificações são de tamanho variável. Este campo indica, então, o tamanho atribuído a estes outros campos.

Já o campo filtro serve, por sua vez, para fazer uma espécie de classificação destas codificações, possibilitando uma maior especificidade destas.

8.2.2.1 SGTIN-96

A codificação SGTIN foi criada com o intuito de dar ao código EPC um equivalente ao já existente código GTIN. No entanto, e como o código GTIN é apenas aplicável a conjuntos de produtos, não à unidade, foi-lhe acrescentado o número de série.

Tal como anteriormente mencionado, este código é indicado para utilizar em artigos vendáveis.

Na Tabela A.3 podemos encontrar o formato desta codificação.

	Cabeçalho	Filtro	Partição	Companhia	Item	N/S
Nº de bits	8	3	3	20 até 40	44 - Companhia	38
Valor Máximo	0011 0000* (binário)	-*	-*	999.999 até 999.999.999.999 (decimal)	9 até 9.999.999 (decimal)	274.877.906.943 (decimal)

*Valores fixos identificativos.

Tabela A.3 - Formato da codificação SGTIN-96

8.2.2.2 SSCC-96

A codificação SSCC é indicada para utilizações em contentores de entregas, por exemplo: caixas, paletes, etc.

Na Tabela A.4 podemos encontrar o formato desta codificação.

	Cabeçalho	Filtro	Partição	Companhia	S/N	Livres
Nº de bits	8	3	3	20 até 40	58 - Companhia	24
Valor Máximo	0011 0001* (binário)	-*	-*	999.999 até 999.999.999.999 (decimal)	99.999 até 99.999.999.999 (decimal)	Não Utilizados

*Valores Fixos Identificativos

Tabela A.4 - Formato da codificação SSCC-96

8.2.2.3 SGLN-96

Esta codificação é utilizada para identificar pontos no espaço. Estes pontos podem ser objectos singulares (um item), objectos agregados (um edifício) ou mesmo entidades (uma empresa).

Na Tabela A.5 podemos encontrar o formato desta codificação.

	Cabeçalho	Filtro	Partição	Companhia	Localização	Extensão
Nº de bits	8	3	3	20 até 40	41 - Companhia	41
Valor Máximo	0011 0010* (binário)	-*	-*	999.999 até 999.999.999.999 (decimal)	0 até 999.999 (decimal)	999.999.999.999 (decimal)

*Valores Fixos Identificativos

Tabela A.5 - Formato da codificação SGLN-96

8.2.2.4 GRAI-96

A codificação GRAI foi concebida para identificar objectos retornáveis, ou seja, cuja devolução seja possível ou mesmo obrigatória.

Na Tabela A.6 podemos encontrar o formato desta codificação.

	Cabeçalho	Filtro	Partição	Companhia	Item	S/N
Nº de bits	8	3	3	20 até 40	44 - Companhia	38
Valor Máximo	0011 0011* (binário)	-*	-*	999.999 até 999.999.999.999 (decimal)	0 até 999.999 (decimal)	274.877.906.943 (decimal

*Valores Fixos Identificativos

Tabela A.6 - Formato da codificação GRAI-96

8.2.2.5 GIAI-96

Esta codificação é utilizada para identificar objectos em geral. Na Tabela A.7 podemos encontrar o formato desta codificação.

	Cabeçalho	Filtro	Partição	Companhia	N/S
Nº de bits	8	3	3	20 até 40	82 - Companhia
Valor Máximo	0011 0100* (binário)	-*	-*	999.999 até 999.999.999.999 (decimal)	4.398.046.511.103 até 4.611.686.018.427.387.903 (decimal)

*Valores Fixos Identificativos

Tabela A.7 - Formato da codificação GIAI-96

8.2.2.6 DoD-96

Esta codificação foi criada especificamente para objectos cujo destino é o DoD e tenham origem em empresas a quem tenha já sido atribuído, pelo governo dos EUA, um código CAGE.

Na Tabela A.8 podemos encontrar o formato desta codificação.

	Cabeçalho	Filtro	Identificador do Governo dos EUA	N/S
Nº de bits	8	3	48	36
Valor Máximo	0010 1111* (binário)	-*	Codificado com o CAGE em ASCII de 8 bits (ASCII)	68.719.476.735 (decimal)

*Valores Fixos Identificativos

Tabela A.8 - Formato da codificação DoD-96

8.2.2.7 GID-96

Esta codificação é a codificação geral. Normalmente esta é atribuída pelo fabricante ou gestor da tecnologia, ficando posteriormente a cargo da entidade que as utiliza, para realmente identificar objectos, codifica-las correctamente. Esta é também utilizada para codificações de teste.
Na Tabela A.9 podemos encontrar o formato desta codificação.

	Cabeçalho	Gestor	Classe	N/S
Nº de bits	8	28	24	36
Valor Máximo	0011 0101* (binário)	268.435.455 (decimal)	16.777.215 (decimal)	68.719.476.735 (decimal)

*Valor Fixo Identificativo

Tabela A.9 - Formato da codificação GID-96

8.2.3 Outras Normas

Existem muitas outras normas, de maior ou menor importância, tanto da EPC-Global como de outras organizações, que pretendem introduzir alguma regulamentação nesta tecnologia. Dada a recente explosão desta, é natural que muitas delas se encontrem ainda em fase de discussão e desenvolvimento. No entanto, é importante nesta fase referir alguma delas, de maior importância e maturidade.

8.2.3.1 A norma ALE da EPC-Global

Esta norma especifica uma interface de comunicação através da qual, qualquer software que pretende informação, o consiga através de uma interacção simples e transparente, com diversos tipos de fonte.

Apesar das vantagens que semelhante norma poderia representar para este trabalho, na verdade, esta encontra-se demasiado voltada para a EPC-Global e as suas normas. Uma vez que o pretendido será fazer um middleware genérico, que não se prenda a uma tecnologia, aplicação ou norma em concreto, esta não será aplicada.

8.2.3.2 A norma LLRP da EPC-Global

A norma LLRP pretende que, no futuro, a interface de comunicação entre um leitor e o software, neste caso o middleware, seja normalizada para todos os modelos e marcas de leitor. Caso esta norma consiga a aceitação pretendida,

deixará de ser necessário ao nível do software, para cada leitor diferente, uma diferente implementação, uma vez que todos terão a mesma interface de acesso.

Esta norma, apesar de todas as vantagens que dela se poderia retirar, traria, em termos tanto económicos como funcionais, desvantagens. Uma vez que todos os leitores teriam a mesma interface de acesso, dificilmente, e mesmo sendo este um protocolo de baixo nível, se utilizariam novas funcionalidades que fossem sendo acrescentadas aos leitores.

O facto de este ser, como já mencionado, um protocolo de baixo nível, também para os utilizadores dos leitores se tornaria mais complexa a integração dos leitores, uma vez que diversos parâmetros, mesmo físicos, teriam de ser levados em consideração.

De qualquer das formas, e como o pretendido neste trabalho não é descer ao nível da comunicação com o leitor, mas sim, e como veremos adiante, criar para este uma interface de comunicação com o middleware, este não é um protocolo de grande relevância para este trabalho.

8.3 Privacidade

Uma das grandes preocupações humanas, relacionadas com esta tecnologia, é os problemas associados à facilidade com que se poderá obter informações acerca de um objecto.

É preciso ter em atenção que, quando aqui é mencionado o objecto, este pode também ser uma pessoa. Podemos imaginar um cenário em que, não só a pessoa está identificada através de uma etiqueta, como grande parte dos objectos que essa pessoa possui. Com a tecnologia RFID, esta identificação pode ser obtida facilmente, mesmo sem a permissão, ou sequer o conhecimento, da pessoa em questão. Num cenário ainda mais futurista, podemos conceber um mundo onde esta tecnologia esteja completamente disseminada, e a existência de leitores de alta potencia seja comum, mesmo nas ruas, com os mais variados propósitos. Num cenário semelhante a este, poderíamos cair no ponto em que fosse, senão impossível, extremamente difícil, manter o anonimato enquanto caminhávamos pela rua. Em qualquer momento poderíamos saber onde se encontrava determinada pessoa ou objecto.

Naturalmente que quando o assunto em causa é, como no cenário anteriormente descrito, a privacidade das pessoas, vozes se levantam e preocupações e entraves surgem.

Como tal, algumas soluções começam a surgir no sentido de, de alguma forma, assegurar que, caso seja a nossa vontade, temos a possibilidade de permanecer anónimos.

Vamos então, de seguida, abordar algumas das soluções existentes para este problema.

8.3.1 Desactivação de Etiquetas

Uma das soluções que garantem a privacidade dos objectos marcados com uma etiqueta RFID é a sua desactivação, imediatamente após a compra de um produto. Apesar de esta ser, provavelmente, a solução mais segura, esta trás imensas desvantagens, no que toca ao potencial que esta tecnologia oferece.

Basta imaginar uma situação em que seja necessário devolver um objecto numa loja, para imediatamente ficar claro o quanto seria útil que esta continuasse activa.

Num cenário mais futurista, podemos mesmo imaginar uma situação em que o nosso frigorifico de casa tenha a capacidade, através das etiquetas dos produtos que nele se encontram, de fazer o seu inventário e, caso haja necessidade, tomar medidas quando um produto estivesse perto de se esgotar. Também isto ficaria dificultado, caso esta técnica fosse empregue.

8.3.2 Etiqueta Bloqueadora

Outra solução para a preservação da privacidade, igualmente eficaz, é a de utilizar etiquetas bloqueadoras. Estas etiquetas têm a possibilidade de bloquear, totalmente ou parcialmente, um determinado espectro de códigos, baralhando o algoritmo anti-colisão dos leitores.

Apesar de esta ser uma solução bastante eficaz, tem, naturalmente, as suas desvantagens. A principal desvantagem seria que, não só seria possível utilizar esta tecnologia para preservar a privacidade como, noutras situações, seria também possível bloquear os leitores de determinado serviço.

Naturalmente que, fornecendo uma etiqueta com esta capacidade a todas as pessoas que o desejassem, certamente que, mesmo por pura distracção, muitas vezes teríamos situações de bloqueios de sistemas.

8.4 Outras Tecnologias de Identificação

É de extrema importância, para qualquer nova proposta, a existência de um comparativo com anteriores ou igualmente novas soluções.

Como tal, aqui se apresentam diversas tecnologias de identificação, concorrentes da tecnologia em estudo, a RFID, mantendo sempre uma análise crítica e comparativa com esta.

Como vem sendo mote deste trabalho, a necessidade leva a que a invenção e reinvenção de soluções surjam de forma natural. Pois a necessidade de identificação é já tão ou mais antiga que a Humanidade. Com uma tão longínqua ascensão, é natural que uma solução para uma necessidade se venha aprimorando e sofrendo evoluções constantes, muitas vezes se especializando e ramificando abrindo, algumas vezes, a porta para novas tecnologias.

Assim sendo, vamos passar agora a discutir um pouco das diversas soluções, para a problemática da identificação, às quais o tempo e o génio se encarregaram de dar grande desenvolvimento.

8.4.1 Códigos de Barras

Os muito conhecidos e, de certa forma, ubíquos códigos de barras têm já, aproximadamente, o mesmo tempo de existência que a tecnologia RFID. No entanto, dadas as facilidades tecnológicas e económicas que esta proporcionava, teve uma expansão francamente superior à RFID.

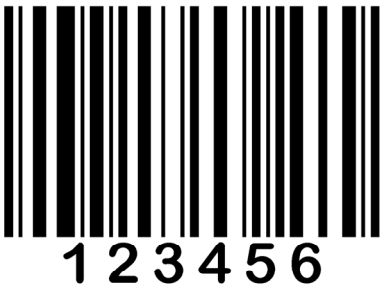


Figura A.33 - Exemplo de código de barras

O código de barras consiste na utilização de um código alfanumérico, representado por barras e falhas, tal como se pode ver na Figura A.33. Esta sequência, criada por barras escuras, de diferentes larguras, sobre um fundo claro ou reflector, são reconhecidas numérica ou alfanumericamente, por uma laser óptico, baseando-se esta leitura na reflexão ou não de luz que o código provoca.

O mais utilizado, e consequentemente mais conhecido, tipo de código de barras é o EAN, criado em 1976 para cumprir com a necessidade de identificação da indústria de então. Este código é composto por 13 caracteres, que identificam os seguintes campos: o país, a companhia, o nº de item ou classe e, finalmente, um caracter de controlo.

Comparativamente com a tecnologia RFID, esta em análise tem, naturalmente, vantagens e desvantagens.

Uma vantagem da tecnologia RFID sobre o código de barras é, sem dúvida, a capacidade que uma etiqueta RFID tem de ser reprogramada, ou seja, ter a possibilidade de, quando necessário, o seu código ser alterado.

Uma vantagem, esta apontada como uma das mais explicitas, é o facto de uma etiqueta RFID, ao contrario de uma etiqueta com código de barras, não necessitar de estar visível nem sequer próxima do seu leitor, tendo uma alcance de leitura superior mesmo quando existem obstáculos entre o leitor e a etiqueta.

Para a quantidade de objectos que, correntemente, pretendemos identificar univocamente, o comum código de barras não terá capacidade suficiente. No

entanto, com a tecnologia RFID, a capacidade de armazenamento de dados de uma etiqueta é, teoricamente, ilimitada.

Outra vantagem, também esta das mais citadas, é a capacidade que a tecnologia RFID tem, em oposição ao código de barras, de identificar simultaneamente um grande número de objectos. Utilizando códigos de barras, apenas nos é permitido identificar um objecto de cada vez.

No entanto, nem tudo é mais vantajoso para a RFID. Uma enorme vantagem que a tecnologia de códigos de barras tem sobre a RFID é, sem dúvida, o preço. Enquanto uma etiqueta de código de barras pode ser colocada num artigo por um preço quase nulo, com a tecnologia RFID tal não acontece, tendo esta os custos já discutidos.

Outra vantagem do código de barras, de menor importância e que se prevê venha a ser ultrapassada a curto prazo, é a correcção nas leituras. Nas evoluções mais avançadas desta tecnologia, as falhas podem atingir níveis tão remotos como 1 falha em cada 600 milhões de leituras, tornando-a praticamente infalível. Para isto muito contribui o facto de, contrariamente à RFID, o código de barras ser insensível ao material, podendo ser aplicado em qualquer objecto.

Para alem das vantagens, maioritariamente técnicas, até agora apresentadas, temos também as vantagens de cariz social e político. Com o código de barras dificilmente se tem o problema da privacidade, anteriormente discutido como problema de resolução fundamental para o sucesso da tecnologia RFID. Para alem deste, outro problema que, com o código de barras, não se levanta, são as restrições internacionais. Enquanto a tecnologia RFID, tem restrições, por exemplo, ao nível da atribuição de frequências, muitas vezes dependente de instituições nacionais, o mesmo não se passa com o código de barras.

8.4.2 Reconhecimento Biométrico

O reconhecimento biométrico encontra-se, actualmente, em fase de grande expansão. Apesar da sua também longa história de utilizações de sucesso, como

é o caso da identificação por impressão digital, muitas novas soluções, e possíveis aplicações destas, se encontram actualmente em estudo.

A biometria é definida pela utilização de identificadores corporais únicos, tanto de pessoas como de animais.

Estes identificadores podem passar pela já mencionada impressão digital, o reconhecimento de voz, a leitura da retina ou íris do olho, etc.

Apesar de esta ser, também, uma tecnologia de identificação, prende-se mais com a segurança e autenticação pessoal do que com a identificação em si, pelo que não será directamente comparável com a tecnologia RFID.

8.4.3 Smart Cards

Os *Smart Cards* são, na sua essência, dispositivos electrónicos com capacidade de armazenamento de dados, normalmente sem capacidade de processamento, que se encontram inseridos em cartões que têm usualmente o tamanho de cartões de crédito.

Uma grande desvantagem, de forma semelhante aos códigos de barras, desta tecnologia em relação à tecnologia RFID é o facto de apenas ser possível obter a identificação de um objecto de cada vez. Para além disto, cada uma destas identificações pode ser bastante demorada, uma vez que, para a correcta identificação, é exigido contacto directo com o leitor.

Figura A.34 - Exemplo de *Smart Card*

Para alem disto, os *Smart Cards* têm sérios problemas de degradação, uma vez que os contactos destes, sendo metálicos e estando expostos ao ambiente, rapidamente oxidam ou sofrem da degradação que resulta naturalmente do uso.

Um outro problema, agora de ordem económica, deste tipo de sistema, é o seu preço tanto de aquisição como de manutenção. Estes, comparativamente aos sistemas RFID, têm um custo muito elevado.

8.4.4 OCR

A tecnologia OCR permite o reconhecimento automático de caracteres, maioritariamente através de software específico para o efeito.

A primeira utilização desta tecnologia ocorreu em 1960, quando foi criado um tipo de letra específico e estilizado, com o objectivo de facilitar o seu reconhecimento.

Esta tecnologia, apesar de ser muitas vezes referida como tecnologia de identificação, tem a sua maior vantagem no reconhecimento de caracteres, possibilitando uma maior velocidade de inserção de texto.

Uma vez que é uma tecnologia com uma relação qualidade/custo pouco vantajosa, esta é apenas utilizada em situações que o justifiquem realmente, por exemplo, em bancos para reconhecimento automático de cheques, ou bibliotecas onde se pretende digitalizar grandes quantidades de texto.

Assim sendo, e tal como na biometria, esta tecnologia não pode ser directamente comparável com a RFID.