



Licenciatura em Engenharia Electrotécnica e de  
Computadores

Trabalho Final de Curso

# E-Mail

Inês Oliveira

N.º 39846

Orientador: Prof. Doutor Alberto Silva

*Relatório Final*

*Julho 2000*

Orientador: Professor Doutor Alberto Silva  
Departamento de Engenharia Informática  
Instituto Superior Técnico

## **AGRADECIMENTOS**

Quero expressar os meus sinceros agradecimentos ao Prof. Doutor Alberto Silva por todo o apoio prestado no desenvolvimento do trabalho.

Quero também agradecer a todos os que me apoiarem na concretização deste trabalho especialmente ao Luís, à minha família e a todos os meus amigos.

Lisboa, 1 de Julho 2000

Inês

## **SUMÁRIO**

Este trabalho apresenta um centro comercial electrónico ou virtual mall. A aplicação consiste num centro comercial electrónico com lojas fáceis de gerir o que permite aos lojistas gerirem as suas lojas de forma eficiente e com um agente mediador que tem como finalidade oferecer serviços para satisfazer de forma simples e produtiva os possíveis clientes do centro comercial. Uma vantagem da aplicação desenvolvida é o facto de todas as interacções com os vários intervenientes serem executadas num *Web* browser suportando vários tipos de plataformas na parte Cliente da arquitectura.

No campo das tecnologias baseadas em agentes estudou-se de forma aprofundada o sistema de agentes AgentSpace ou “espaço de agentes” desenvolvido pelo Prof. Dr. Alberto Silva. Pretende-se observar as vantagens e desvantagens de fazer uma aproximação baseada em agentes a um modelo de negócio electrónico como é o virtual mall, por contra ponto com outros paradigmas e tecnologias mais comuns como o CGI, as JAVA Servlets, etc. ...

A integração dos três níveis é realizada de uma forma quase perfeita. São utilizados documentos HTML com applets embebidas para gerar a GUI da aplicação Cliente, no nível da Apresentação, com um resultado bastante eficiente pela sua fácil compreensão e pelo facto de ser possível interagir com os agentes residentes no nível da Lógica do Negócio de forma directa. Neste nível residem os agentes que implementam toda a funcionalidade do sistema. A utilização do AgentSpace é aqui de uma importância fulcral pois permite que os agentes estejam inseridos num ambiente que suporta o desenvolvimento de aplicações distribuídas, tendo como principais características a escalabilidade, o fácil desenvolvimento e manuseamento, a capacidade de extensão da funcionalidade e de integração de novas tecnologias. No nível de Serviços de Base de Dados o acesso ao Sistema de Bases de Dados foi implementado utilizando JDBC tendo-se verificado uma boa performance na execução das queries SQL na Base de Dados.

## **Palavras-chave**

Agentes de Software, Comércio Electrónico, Shopping Mall, Java

## Índice

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>1</b>
1.1	MOTIVAÇÃO .....	1
1.2	OBJECTIVOS .....	4
1.3	ORGANIZAÇÃO DO DOCUMENTO .....	6
<b>2</b>	<b>CONTEXTO .....</b>	<b>7</b>
2.1	CONCEITOS BÁSICOS DE TECNOLOGIA .....	7
	<i>JAVA</i> .....	7
	<i>Applet</i> .....	8
	<i>CORBA</i> .....	8
	<i>Voyager</i> .....	9
	<i>Agentes</i> .....	10
	<i>AgentSpace</i> .....	16
	<i>JDBC DRIVER</i> .....	19
2.2	CONCEITOS LIGADOS AO E-BUSINESS .....	21
	<i>Electronic Business</i> .....	21
	<i>Electronic Publishing</i> .....	21
	<i>Electronic Services Delivery</i> .....	22
	<i>Deliberative Purchasing</i> .....	22
	<i>Spontaneous Purchasing</i> .....	22
	<i>Electronic Commerce</i> .....	22
	<i>Classificação dos itens transaccionados</i> .....	23
	<i>Bens e Serviços</i> .....	24
	<i>Itens transaccionados fisicamente e digitalmente</i> .....	24
	<i>O grau de standardização dos itens transaccionados</i> .....	24
	<i>Standard Products</i> .....	24
	<i>Commodities</i> .....	24
	<i>Custom-Built Products</i> .....	24
	<i>Customised Products</i> .....	25
2.3	MODELOS DE NEGÓCIO .....	25
<b>3</b>	<b>REQUISITOS .....</b>	<b>27</b>
3.1	REQUISITOS ESTRUTURAIS.....	27

3.2	REQUISITOS FUNCIONAIS .....	27
3.3	ABORDAGEM BASEADA EM AGENTES .....	29
<b>4</b>	<b>ARQUITECTURA .....</b>	<b>31</b>
4.1	DESCRIÇÃO GERAL .....	31
	<i>Nível de Apresentação.....</i>	<i>32</i>
	<i>Nível da Lógica da aplicação.....</i>	<i>34</i>
	<i>Nível de serviços de base de dados .....</i>	<i>36</i>
4.2	MODELO DE DADOS .....	36
	<i>Entidades e Conceitos definidos no modelo de dados .....</i>	<i>37</i>
	<i>Classes utilizadas para troca de Informação.....</i>	<i>38</i>
	<i>Operações básicas.....</i>	<i>42</i>
4.3	BASES DE DADOS .....	45
	<i>Bases de Dados manager .....</i>	<i>45</i>
	<i>Base de Dados broker .....</i>	<i>48</i>
	<i>Base de Dados shop .....</i>	<i>52</i>
4.4	INTERACÇÃO ENTRE AGENTES .....	56
4.5	INTERACÇÃO ENTRE AGENTES E APPLETS .....	58
<b>5</b>	<b>AVALIAÇÃO DO TRABALHO .....</b>	<b>59</b>
5.1	TRABALHO FUTURO .....	60
<b>APÊNDICE A – MODELOS DE NEGÓCIO.....</b>		<b>63</b>
	<i>Brokerage .....</i>	<i>64</i>
	<i>Advertising .....</i>	<i>66</i>
	<i>Infomediary Model .....</i>	<i>67</i>
	<i>Recommender Systems .....</i>	<i>68</i>
	<i>Registration Model.....</i>	<i>68</i>
	<i>Merchant Model.....</i>	<i>68</i>
	<i>Manufacturer Model .....</i>	<i>69</i>
	<i>Affiliate Model.....</i>	<i>69</i>
	<i>Community Model .....</i>	<i>69</i>
	<i>Subscription Model .....</i>	<i>70</i>
	<i>Utility Model .....</i>	<i>70</i>
<b>APÊNDICE B - MANUAL DO UTILIZADOR .....</b>		<b>71</b>

PROCESSO DE INSTALAÇÃO .....	71
INICIAR A APLICAÇÃO.....	71
GERIR O CENTRO COMERCIAL.....	72
MECANISMO DE PROCURA.....	74
CRIAÇÃO DAS NOVAS LOJAS .....	75
GESTÃO DA LOJA E DA SUA INFORMAÇÃO.....	77
REGISTO DE UM CLIENTE.....	78
GESTÃO DO CESTO E DA INFORMAÇÃO PESSOAL DOS CLIENTES .....	78
INSERÇÃO DE UM PRODUTO OU SERVIÇO NO CESTO DO CLIENTE .....	80
<b>APÊNDICE C - SCRIPTS PARA CRIAR BASES DE DADOS .....</b>	<b>82</b>
SCRIPT DAS TABELAS DO GESTOR (MANAGER):.....	82
SCRIPT PARA AS TABELAS DAS LOJAS: .....	83
SCRIPT PARA AS TABELAS DO BROKER: .....	83
<b>APÊNDICE D - LISTAGENS DO CÓDIGO .....</b>	<b>85</b>
<b>7 REFERÊNCIAS .....</b>	<b>195</b>

## Índice de Figuras

Figura 1 - E-Business=EC+CRM+SCM+BI+KM+CT , [6] .....	1
Figura 2 - Espectro de relações E-Business .....	3
Figura 3 – Taxonomia possível para classificar agentes .....	16
Figura 4 – Visão geral da infraestrutura do Agentspace .....	17
Figura 5 – Modelo de objectos geral do AgentSpace – diagrama de classes .....	18
Figura 6 – Diagrama de solução do IDS Server JDBC .....	20
Figura 7 – Representação esquemática do comércio electrónico em 3 aneis .....	23
Figura 8 – Mapa do site que obedece aos requisitos .....	28
Figura 10 – Arquitectura genérica da aplicação.....	32
Figura 11 – Mapa navegacional do site construído.....	33
Figura 12 – Ilustração do modelo de gestão de locais de execução e de agentes .....	35
Figura 13 - Diagrama de relações da base de dados do agente gestor do centro comercial .....	45
Figura 14 – Diagrama de relações da base de dados do agente broker .....	48
Figura 15 – Evolução do modelo de negócio E-Mall.....	61
Figura 16 - Interfaces gráficas existentes para realizar a gestão do centro comercial.....	72
Figura 17 – applet que permite procurar uma loja por tipo de loja .....	74
Figura 18 – applet utilizada para criar novas lojas.....	75
Figura 19 – Janela de confirmação de criação de loja nova .....	76
Figura 20 - Interfaces gráficas utilizadas para realizar a gestão de produtos e serviços comercializados numa loja e também da sua informação pessoal .....	77
Figura 21 - Applet que permite registar um cliente no E-mall.....	78
Figura 22 - Interfaces gráficas para realizar gestão de informação pessoal e do cesto de compras de um cliente .....	79
Figura 23 - applet utilizada para realizar as compras .....	80
Figura 24 – Interface utilizada para adicionar itens ao cesto .....	80

## Índice de Tabelas

Tabela 1 – Atributos característicos de agentes .....	15
Tabela 2 – Customizações possíveis num produto .....	25
Tabela 3 – Descrição das classes utilizadas na troca de mensagens para passar informação .....	39
Tabela 4 – Descrição das classes utilizadas na troca de mensagens para passar informação (Continuação da Tabela anterior) .....	41
Tabela 5 – Descrição das interacções entre agentes na criação de uma loja .....	42
Tabela 6 – Descrição das interacções entre agentes na criação de um cliente .....	43
Tabela 7 – Descrição das interacções entre agentes na inserção de um produto ou serviço .....	43
Tabela 8 - Descrição das interacções entre agentes na inserção de um produto ou serviço .....	44
Tabela 9 – Descrição das interacções entre agentes na aquisição do conteúdo de um cesto .....	44
Tabela 10 – Descrição da tabela manager .....	46
Tabela 11 – Descrição da tabela lojista na base de dados manager .....	46
Tabela 12 – Descrição da tabela tipoLoja na base de dados manager .....	47
Tabela 13 – Descrição da tabela loja na base de dados manager .....	47
Tabela 14 – Descrição da tabela tipoLoja na base de dados broker .....	49
Tabela 15 – Descrição da tabela metodoPagamento na base de dados broker .....	49
Tabela 16 – Descrição da tabela metodoEntrega na base de dados broker .....	49
Tabela 17 – Descrição da tabela broker .....	50
Tabela 18 – Descrição da tabela loja na base de dados broker .....	50
Tabela 19 - Descrição da tabela cliente na base de dados broker .....	50
Tabela 20 – Descrição da tabela basket na base de dados broker .....	51
Tabela 21 – Descrição da tabela produto na base de dados broker .....	51
Tabela 22 – Descrição da tabela servico na base de dados broker .....	52
Tabela 23 – Descrição da tabela servicoLoja na base de dados broker .....	52
Tabela 24 – Descrição da tabela produtoLoja na base de dados broker .....	52
Tabela 25 – Descrição da tabela metaLoja na base de dados shop .....	53
Tabela 26 – Descrição da tabela cliente[SIGLAloja] na base de dados shop .....	54
Tabela 27 - Descrição da tabela produto[SIGLAloja] na base de dados shop .....	54
Tabela 28 - Descrição da tabela servico[SIGLAloja] na base de dados shop .....	54
Tabela 29 - Descrição da tabela transacciao[SIGLAloja] na base de dados shop .....	55
Tabela 30 - Descrição da tabela transacciaoServico[SIGLAloja] na base de dados shop .....	55
Tabela 31 - Descrição da tabela transacciaoProduto[SIGLAloja] na base de dados shop .....	56
Tabela 32 - Esquema de novos modelos de negócio aplicáveis ao comércio electrónico .....	63

# 1 Introdução

## 1.1 Motivação

Nos últimos anos o *e-business* tem sido um chavão utilizado para descrever a integração de estratégias de negócio, processos e tecnologias que focam a anulação das barreiras entre companhias.

Este é no entanto apenas um dos últimos de uma série de esforços e conceitos que focam os problemas de realizar negócios no mundo digital com entidades fora da empresa.

As aplicações *e-business* são aquelas que permitem e gerem relações entre empresas, as suas funções e processos dos seus clientes, fornecedores, cadeia de valores, comunidade e indústria [5].

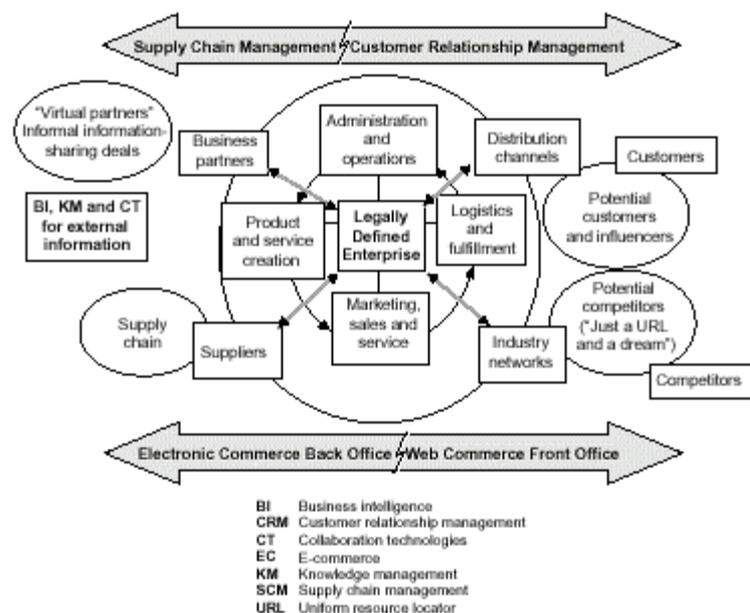


Figura 1 - E-Business=EC+CRM+SCM+BI+KM+CT , [6]

*E-bussines* é a visão de como um conjunto de **estratégias específicas** para funções, processos ou indústria podem ser integradas num conjunto de tecnologias e aplicações, estando principalmente orientadas para a partilha de informação e integração na *Web/Internet*.

A análise do crescimento do fenómeno do *e-business* leva a considerar os seguintes aspectos.

O que levará à integração das estratégias de negócio, processos e funções necessárias para implementar o *e-business*?

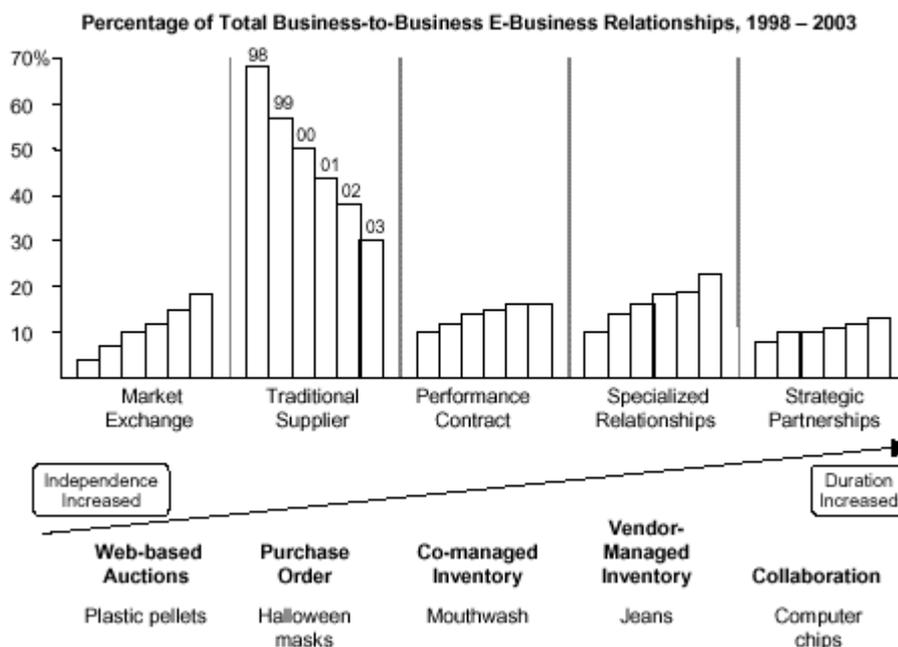
Como é que o facto de se realizarem negócios na *Internet* vai influenciar os modelos de negócio e as estratégias de gestão?

Como é que as ferramentas, aplicações e serviços de e-business vão ser produzidas, porque segmentos da comunidade de vendedores e que implicações é que isto vai trazer à estratégia da empresa?

O conceito de corporação virtual apesar de breve, levou ao aparecimento de componentes de e-business como customer relationship management (CRM) e supply chain management (SCM).

A troca de e-mails entre empresas e as transacções EDI (electronic data interchange) foram os precursores de *e-business*. Programas específicos para certas áreas da indústria ligaram vendedores, distribuidores e *consumer packaged-good companies*. No entanto estes programas apesar de serem bem sucedidos apresentavam muitas limitações pois as transacções eram efectuadas em modo batch, duravam uma noite, etc...

À medida que o *e-business* evolui, o processo tradicional factura/recibo está a ser substituído por um processo *de gestão de factura/recibo* mais diverso e que requer uma postura de colaboração entre parceiros de negócio.



**Figura 2 - Espectro de relações E-Business**

Os mercados electrónicos, catálogos electrónicos, sistemas de licitação electrónicos e os agentes de procura na Internet estão a criar um ambiente open-source. A aceleração das actividades de colaboração entre parceiros é visível na substituição dos antigos modelos de negócio centrados na empresa por modelos de negócio centrados no cliente. Uma competição global crescente e uma rede global logística estão a forçar as companhias a juntarem forças para poderem competir de forma eficaz. As exigências crescentes dos clientes em relação a serviços melhores, mais rápidos e mais baratos levam à necessidade de aumentar a velocidade de processamento de bens e informação na cadeia de fornecimento.

Os agentes de software ganharam muita popularidade nos últimos anos. Foram utilizados de forma muito eficaz para filtrar informação, encontrar pessoas com interesses semelhantes e automatizar comportamento repetitivo [11].

Mais recentemente as capacidades dos agentes têm sido aplicadas ao comércio electrónico prometendo revolucionar a forma como se conduzem as transacções business-to-business, business-to-consumer ou consumer-to-consumer.

Os agentes de software permitem automatizar um variedade de tarefas incluindo a compra e venda de produtos na Internet.

O comércio electrónico abarca uma variedade de assuntos como a segurança, a confiança entre intervenientes, mecanismos de pagamentos, anúncios, catálogos on-line, intermediários e gestão de back-office entre outros.

Questões como : Quanto tempo e dinheiro seriam poupados se um certo processo fosse automatizado (por exemplo, comparação de produtos)? Qual a facilidade de indicar preferências a um agente para permitir a compra de um produto ? Quais os riscos de deixar um agente tomar decisões transaccionais em nome do seu utilizador? Quais as consequências sofridas devido ao não aproveitamento de oportunidades (não monitorar efectivamente o aparecimento de novos anúncios de emprego)?

Em princípio quanto mais dinheiro e tempo poupados através da automatização de tarefas , mais fácil é exprimir preferências e menores são os riscos de tomar decisões transaccionais inadequadas.

Os agentes de software podem desempenhar uma variedade de papeis como mediadores no comércio electrónico para qualquer uma destas categorias de transacção: business-to-business, business-to-consumer and consumer-to-consumer.

As primeiras gerações de sistemas de agentes para o comércio electrónico estão a abrir portas para o aparecimento de novos mercados e a diminuir os custos transaccionais numa variedade de negócios. As indústrias mais afectadas por esta “revolução” são as que negociam produtos de curta duração (bilhetes, etc..) ou comodidades como o gás, a água, livros, cds, etc...

No entanto, ainda há um longo caminho a percorrer antes dos agentes de software se tornarem percursos de negócios.

A maioria das tecnologias que suportam os sistemas mediados por agentes de comércio electrónico derivam da investigação da inteligência artificial. Desde extrair significados de páginas Web ambíguas a planear viagens, a aprender as preferências musicais dos seus donos, etc, as tecnologias de IA continuarão a produzir software com crescente know-how para mediar eficazmente as transacções electrónicas [12].

## **1.2 Objectivos**

Neste trabalho pretende-se implementar um centro comercial electrónico com tecnologia de agentes. O sistema de suporte de agentes utilizado será o AgentSpace [1] ou “espaço de

agentes” desenvolvido pelo Prof. Alberto Silva. Pretende-se observar as vantagens e desvantagens de fazer uma aproximação baseada em agentes a um modelo de negócio electrónico como é o virtual mall. A avaliação do resultado obtido será em função não só das várias características da aplicação final mas também de todo o processo de desenvolvimento.

Inicialmente realiza-se um estudo sobre os vários modelos de negócio existentes em comércio electrónico já que os modelos tradicionais de comércio deixaram de ser válidos neste contexto. Posteriormente definem-se os vários requisitos a que a aplicação deverá obedecer com base no modelo escolhido e na tecnologia a implementar.

O modelo de dados resulta naturalmente das interacções existentes entre as várias entidades que participam em todo o processo levando à definição do modelo de bases de dados a utilizar.

Este projecto tem como objectivos a criação de um centro comercial electrónico baseado em agentes, com uma arquitectura de três níveis: Apresentação, Lógica do Negócio e Serviços de Bases de Dados.

**O Nível de Apresentação** consiste na interface com o cliente. Pretende-se criar uma aplicação que se execute num *Browser*, tendo uma funcionalidade semelhante a um *WebSite*. Desta maneira é possível suportar diversos tipos de utilizadores como os lojistas e os clientes. A aplicação opera como um centro comercial electrónico normal.

**Ao Nível da Lógica de Negócio** o objectivo é implementar um modelo de negócio apoiado por agentes para todo o tipo de plataformas, utilizando o AgentSpace e a linguagem de programação Java [2,3,4], e para todo o tipo de Sistemas de Bases de Dados utilizando a JDBC API. Os detalhes da funcionalidade e da interacção entre os vários agentes e o SBD encontram-se encapsulados em cada agente específico. Esta modularidade e abstracção da complexidade vai possibilitar a reutilização do código de forma simples e eficaz. A característica transaccional das operações é também um objectivo essencial de forma a garantir que o estado da aplicação esteja sempre correcto e coerente.

**O Nível dos Serviços de Bases de Dados** é constituído pelo SBD onde residirá toda a informação e dados da aplicação. A SBD tem como objectivo guardar o estado dos vários agentes e permitir a gestão de lojas e clientes.

### **1.3 Organização do Documento**

Este documento encontra-se dividido em várias secções:

Nesta secção faz-se uma breve introdução ao trabalho e aos objectivos que se pretendem alcançar.

Na segunda secção tecem-se considerações sobre a tecnologia a utilizar na implementação do trabalho. Analisam-se também, os vários conceitos de comércio electrónico actualmente vigentes e as suas implicações na forma de realizar transacções presentemente.

Na secção seguinte apresentam-se os requisitos do trabalho a nível funcional e estrutural.

A arquitectura do sistema e desenvolvimento será exposta de forma detalhada na secção 4 e cada agente será descrito individualmente na secção 5, incidindo especialmente esta descrição nas interfaces gráficas associadas aos vários agentes. Na secção 6 faz-se uma avaliação do trabalho realizado e quais as possíveis direcções a seguir no futuro, enquanto que na secção 7 se tiram as conclusões finais sobre o trabalho realizado. No apêndice A apresenta-se uma exposição mais prolongada sobre os vários modelos de negócio aplicáveis ao comércio electrónico. Apresenta-se no apêndice B um pequeno manual de utilização e no apêndice C os vários scripts em SQL necessários à criação das bases de dados da aplicação.

## **2 Contexto**

### **2.1 Conceitos Básicos de Tecnologia**

#### **JAVA**

É uma linguagem de programação expressamente desenhada para ser aplicada em ambientes distribuídos na Internet. Foi concebida para ter o "look and feel" da linguagem C++ conseguindo ser mais simples de utilizar e reforçando também o conceito de programação orientada por objectos. O Java pode ser utilizado para criar aplicações completas que podem correr numa só máquina ou distribuídas entre servidores e clientes numa rede. Também é utilizado para construir pequenos módulos aplicativos ou applets que podem estar embebidos numa página Web. As applets tornam possível a interacção entre o utilizador da página e a própria página.

As principais características do Java são:

Os programas criados são portáveis na rede. O programa pode ser compilado em bytecode Java que pode ser executado em qualquer máquina cliente ou servidor que tenha a Java Virtual Machine instalada. A Java Virtual Machine interpreta o bytecode em código específico para aquela plataforma de hardware. Isto significa que diferenças entre as várias plataformas, como por exemplo o comprimento das instruções, são reconhecidas e resolvidas localmente durante a execução. As versões específicas para plataformas não fazem sentido em Java.

O código é robusto, o que significa que ao contrário do que acontece em outras linguagens como o C++, os objectos Java não podem conter referências para dados externos a si próprios ou a outros objectos conhecidos. Isto impossibilita a hipótese de uma instrução ter acesso a um endereço de armazenamento de dados de outra aplicação ou do próprio sistema operativo, o que levaria a um crash da aplicação ou mesmo do próprio sistema operativo. A Java Virtual Machine realiza um conjunto de testes em cada objecto para poder garantir a sua integridade.

Java é orientada por objectos o que significa que entre outras características objectos similares podem tirar vantagens de pertencerem à mesma classe e herdarem código comum. Os métodos de cada objecto correspondem aos comportamentos e capacidades que o objecto pode ter.

O Java foi introduzido pela Sun MicroSystem em 1995 e levou a antever as imensas possibilidades interactivas da rede. Os dois Web browsers com maior quota de mercado incluem uma Java Virtual machine. Quase todos os fornecedores de sistemas operativos como por exemplo a IBM e a Microsoft integraram compiladores de Java nos seus produtos.

### **Applet**

Para muitos o Java é conhecido principalmente como sendo uma ferramenta para criar applets para a world wide web. Uma "Applet" é uma mini-aplicação em Java que corre dentro de um browser e que pode realizar tarefas e interagir com o utilizador do browser sem utilizar recursos do servidor Web após ter sido descarregada para o computador do cliente. As applets podem no entanto interagir com os servidores se para tal tiverem sido desenhadas.

### **CORBA**

O CORBA é uma arquitectura e especificação para criar, distribuir e gerir objectos distribuídos numa rede. Permite que programas em locais diferentes e desenvolvidos por diferentes empresas comuniquem numa rede através de um interface broker. O CORBA foi desenvolvido por um consórcio de empresas, o OMG (Object Management Group) que actualmente inclui como membros mais de 500 companhias.

A ISO e a X/OPEN aceitaram o CORBA como a arquitectura standard para objectos distribuídos (também conhecidos por componentes).

Todo o conceito CORBA baseia-se no ORB (Object Request Broker). O facto de uma rede de clientes e servidores suportar ORB significa que um programa cliente pode requisitar serviços de um servidor ou objecto sem ter de lidar com aspectos como a localização ou a interface do mesmo. Para realizar pedidos ou respostas entre ORBs os programas utilizam o General Inter-ORB Protocol (GIOP) e para a Internet o Internet Inter-ORB Protocol (IIOP). O IIOP mapeia os pedidos e respostas GIOP para o Transmission Control Protocol (TCP) layer de cada máquina.

A Microsoft mantém-se à parte deste consórcio e apresentou a sua própria arquitectura de objectos distribuídos designada por Distributed Component Object Model (DCOM). No

entanto, a CORBA e a Microsoft acordaram numa aproximação via gateway de modo a que um objecto cliente desenvolvido em DCOM possa aceder a um servidor CORBA e vice-versa.

## **Voyager**

Um ORB CORBA baseado em Java permite conjugar as características universais do Java com as vantagens da arquitectura CORBA. O Voyager da ObjectSpace revelou-se como sendo uma implementação fácil de programar. O modelo geral destas classes de aplicações revela uma aplicação tipo applet executada na máquina cliente. A applet oferece, para além das funcionalidades normais, a possibilidade de interagir com outras aplicações Java distribuídas no espaço gerido pela infra-estrutura comum. O servidor Web pode ou não ser uma aplicação Java e facilita a comunicação entre processos. Todo este processo é gerido e suportado através do ORB (<http://www.objectspace.com>).

O Voyager oferece as seguintes funcionalidades:

- Criação de objectos que podem ser invocados remotamente
- Obtenção de referências para objectos remotos
- Envio de mensagens e invocação dinâmica de métodos em objectos remotos
- Persistência – os objectos podem sobreviver aos programas que os criam
- Reciclagem automática e distribuída de objectos (distributed garbage collection)
- Um mecanismo de comunicação flexível baseado em mensageiros (messenger) inteligentes e com uma variedade de métodos de comunicação entre objectos (síncrona, assíncrona, diferida, assíncrona num único sentido, e num único sentido com difusão selectiva)
- Serviço distribuído de nomes
- Mobilidade de objectos – de modo que estes possam ser trocados entre diferentes programas Java
- Suporte a eventos distribuídos
- Mecanismo de publicação/subscrição de mensagens (publish/subscribe)
- Comunicação em grupo
- Integração adequada com a Web – possibilitando a interoperação de applets com objectos remotos
- Interoperação com objectos CORBA no servidor

A empresa ObjectSpace chega a afirmar que o Voyager também é um sistema de agentes móveis ao nível do Odyssey, Aglets e Concordia, mas na realidade o Voyager está ao nível do RMI da Sun, do Visibroker da Inprise ou do OrbixWeb da Iona.

Os objectos do Voyager não correspondem na realidade a agentes pois não conseguem, manter informação específica ao seu local de execução e ao seu utilizador, por exemplo, o que é condição necessária ao conceito de agente.

## **Agentes**

### **2.1.1.1 Conceito**

A palavra agente está actualmente em voga na imprensa especializada e nas comunidades da inteligência artificial. O conceito de agente é em si intuitivo para todos os utilizadores. Considere por exemplo um agente imobiliário e um agente de viagens. Ambos os agentes agem em representação de um cliente, na venda e compra de um bem imobiliário ou na reserva de bilhetes de avião e de estadias. A primeira propriedade fundamental de um agente pode ser definida como o facto de agir em representação de outrem. A segunda característica fundamental dos agentes reflecte-se no facto de todos terem um certo grau de autonomia. Um terceiro aspecto importante no comportamento dos agentes é o seu grau de proactividade e reactividade. Por exemplo um agente que põe um sinal de “Para Venda” numa propriedade e espera que os compradores apareçam tem um comportamento reactivo enquanto que um agente que anuncie a intenção de venda nos jornais locais é proactivo. Deve-se notar no entanto que a reactividade e a proactividade não são comportamentos mutuamente exclusivos e um agente pode apresentar ambas as características em alturas diversas [9].

Outros atributos importantes num agente são as suas capacidades de aprendizagem, cooperação e mobilidade.

As características acima mencionadas são válidas para agentes computacionais. O conceito agente surgiu no trabalho desenvolvido na inteligência artificial na altura em que os investigadores tentavam desenvolver entidades artificiais que imitavam certas características humanas. O termo agente pode ser aplicado a um largo espectro de entidades que vai desde sistemas de software a robots autónomos e organismos biológicos.

O conceito de agente foi sintetizado na seguinte definição:

Um agente é uma entidade computacional que:

- Age em função de outras entidades de forma autónoma
- Realiza acções com um certo grau de proactividade e/ou reactividade
- Exibe um certo nível de atributos como a aprendizagem, co-operação e mobilidade

Os agentes de software são sistemas de software que obedecem às definições e podem ser basicamente descritos como sistemas que povoam computadores e redes e ajudam os utilizadores em tarefas “electrónicas”. Porque é que no fim do século 20 a necessidade para utilizar agentes de software é tão premente. Os computadores actualmente apenas respondem ao que os designers chamam de manipulação directa. O computador é meramente uma entidade passiva à espera para executar instruções. Os investigadores e as companhias de software esperam conseguir que estes agentes de software conheçam os interesses dos seus utilizadores e consigam agir de forma autónoma em sua representação.

Os agentes de software são necessários porque:

- Cada vez mais tarefas diárias são realizadas por agentes
- O mundo informático é um caos de informação dinâmica e não estruturada
- A existência de cada vez mais utilizadores sem qualquer tipo de formação leva à necessidade de existirem agentes que tornam todo o processo tecnológico mais simples

A tecnologia de agentes está a ser utilizada em muitos domínios como a gestão de workflow, gestão de rede, controlo de tráfego aéreo, data mining, comércio electrónico, etc. ...

A investigação relacionada com agentes inteligentes está focada em 3 áreas de investigação:

- Intelligent Users Interfaces
- Distributed Agent Technology
- Mobile Agent Technology.

Estas três áreas correspondem na realidade a três dos atributos principais de um agente: aprendizagem, cooperação e mobilidade.

As *Intelligent User Interfaces* (IUI) foram desenvolvidas recentemente utilizando a aproximação orientada ao agente mas já havia IUIs desenvolvidos anteriormente com outras aproximações. O objectivo de uma IUI é o desenvolvimento de uma interface adaptativa que permite maximizar a interacção do utilizador com o sistema.

*Inteligência artificial distribuída* é uma das áreas de investigação da inteligência artificial que lida com sociedades ou agentes que interagem para solucionar um problema comum. Estas sociedades podem ser constituídas por pessoas, computadores, sensores e robots, entre outros e são denominadas de Multi-Agent System (MAS). Um MAS é uma rede de agentes que resolvem problemas trabalhando em conjunto. O interesse crescente na investigação em MAS é o facto deste tipo de sociedade apresentarem vantagens significativas inerentes ao próprio sistema incluindo a possibilidade de resolver problemas demasiados complexos para um agente por si só, aumentam a velocidade etc. ...

A resolução de problemas efectuada por um MAS é conhecida por Distributed Problem Solving e envolve investigação em áreas como a coordenação, negociação e comunicação. Para os agentes conseguirem resolver problemas de forma coerente num MAS, os agentes têm de comunicar entre si, coordenar as suas actividades e negociar em caso de conflito. Os conflitos podem ter origem no facto de existirem recursos limitados ou devido a discrepâncias entre os próprios agentes na forma de resolver determinado aspecto do problema. A coordenação é necessária para determinar a estrutura organizacional entre grupos de agentes e para gerir a distribuição de tarefas. A negociação é utilizada para detectar e resolver possíveis conflitos entre agentes.

Os MAS são actualmente alvos de grande esforço na área da investigação devido às possibilidades imensas que oferecem como:

- Resolução de problemas demasiado complexos para apenas um agente só devido a limitações nos recursos
- Interligação a sistemas já existentes como sistemas de suporte à decisão, etc.
- Soluções para problemas distribuídos como o controlo de tráfego aéreo
- Soluções que conseguiam extrair informação de fontes distribuídas
- Soluções com expertise distribuída como um sistema de saúde distribuído
- Aumentar a velocidade minimizando a comunicação, aumentar a confiança ou seja, a capacidade de recuperar de falhas e aumentar a tolerância à informação vaga
- Oferecem clareza na concepção e simplicidade no design

Os MAS são classificados em Cooperative MAS (CMAS) ou em Self-Interested MAS (SMAS) dependendo esta classificação do grau de cooperação entre os agentes.

Um *agente móvel* é uma entidade de software que existe num ambiente de software e que apresenta todas as características definidas anteriormente. Tem de conter os seguintes modelos: modelo de agente, modelo de ciclo de vida, modelo computacional, modelo de segurança, modelo de comunicação e modelo de navegação.

O ambiente dos agentes móveis (*mobile agent environment*) é o ambiente no qual o agente existe. Corresponde a um sistema de software distribuído por várias máquinas. Este ambiente implementa a maioria dos modelos mencionados na definição anterior podendo também oferecer serviços extra aos agentes, etc. ....

Passo a citar uma parte da definição de um agente enunciada na dissertação sobre o AgentSpace realizada pelo Prof. Dr. Alberto Silva [1], pag 66.

“

*A noção de agente adoptada no âmbito desta tese é definida segundo diferentes perspectivas, nomeadamente, segundo:*

- *Perspectiva genérica: Um agente é uma entidade de software com uma entidade, estado e comportamentos bem definidos, e representa, de alguma forma, o seu utilizador nas tarefas que realiza. Um agente deverá apresentar pelo menos as características de autonomia, sociabilidade, persistência, pró-actividade e reactividade.*
- *Perspectiva técnica: Um agente é um objecto activo de média granularidade. Isto significa que é uma instância de uma determinada classe, com o seu próprio grupo de actividades (threads), códigos e dados, e representado por um identificador único e global. Visto como um novo paradigma o agente providencia o conceito de objecto activo e autónomo que pode ser adequado à concepção de aplicações dinâmicas, distribuídas e/ou complexas.*
- *Perspectiva de utilização: Para o utilizador, um agente pode ser visto como um novo paradigma de interacção homem-máquina, baseado no modelo de delegação, ou de gestão indirecta, por oposição ao modelo tradicional da interacção directa. O modelo da*

*delegação é especialmente adequado às classes de aplicações emergentes na Internet, em actividades várias, caracterizadas por serem complexas, tediosas e/ou rotineiras, tais como: a pesquisa de informação em espaço vastos e pouco estruturados; gestão de correio electrónico; ou comércio electrónico.*

“

O autor refere ainda que atributos como a inteligência e a mobilidade, não são fundamentais na definição de um agente como entidade de software.

Do ponto de vista funcional a utilização dos agentes oferece muitas vantagens como a delegação, ou seja, o utilizador deixa de interagir directamente com as máquinas e delega tarefas no agente o que por seu lado levanta dificuldades pois os utilizadores têm dificuldade em aceitar delegar decisões aos seus agentes [10].

Uma outra vantagem é a personalização que uma solução com agentes pode oferecer. Cada utilizador pode associar o seu perfil a um agente de forma a que toda a informação que lhe seja oferecida se coadune com os seus interesses. As lojas virtuais passam a acompanhar cada utilizador de forma pessoal

Outras vantagens que a utilização de agentes de software proporcionam são a automatização, a notificação, a resolução de tarefas complexas e a mobilidade.

A automatização faz especialmente sentido para tarefas repetitivas, sistemáticas e por vezes complexas. O melhor exemplo disto são os wizards utilizados para configurar hardware.

Um agente de notificação toma uma cadeia de acções e/ou notifica o seu utilizador quando um determinado evento se realiza. É especialmente útil quando o utilizador deseja por exemplo ser informado do momento em que as acções  $x$  atingiram o valor  $Y$ .

As dificuldades de adoptar esta solução não são apenas a desconfiança que o utilizador sente em relação ao seu agente, mas também o facto de os novos modelos de negócio ainda não serem adequados a esta tecnologia. Para além disso o próprio conceito de agente de software ainda não gera consenso.

### 2.1.1.2 Classificação de agentes

Existem muitos esquemas possíveis para classificar agentes. Os agentes podem ser classificados de acordo com os atributos que possuem, com as tarefas que executam ou de acordo com a sua arquitectura de controlo. Também podem ser classificados de acordo com a sua sensibilidade reactiva, pela sua eficácia, etc. ...

É possível classificar os agentes de acordo com as características e atributos que possuem.

Os agentes podem ser classificados de acordo com os seus atributos sendo os 4 primeiros comuns a todos os tipos de agentes.

<i>Atributos</i>	<i>Sinónimos</i>	<i>Significado</i>
<i>Reactivo</i>		Reage às mudanças no seu ambiente
<i>Autónomo</i>		Controla as suas próprias acções
<i>Orientado por objectivo</i>	Pró-activo	Não reage apenas ao seu ambiente
<i>Contínuo</i>		Processo contínuo
<i>Comunicativo</i>		Comunica com outros agentes ou utilizadores
<i>Móvel</i>		Consegue-se mover de máquina em máquina
<i>Flexível</i>		As acções não estão codificadas
<i>Capacidade de aprendizagem</i>	adaptativo	Muda o seu comportamento baseado em experiências anteriores

**Tabela 1 – Atributos característicos de agentes**

Apresenta-se uma outra classificação possível para agentes baseada no método de classificação utilizado pelos seres humanos para classificar o seu meio ambiente. Todas as culturas humanas distinguem os organismos abstractos, dos artefactos e de outros conceitos abstractos. Os agentes podem ser classificados à partida como biológicos, robóticos ou agentes computacionais. Os agentes computacionais podem ser divididos em agentes de software e em agentes de vida artificial. Os agentes de software ainda podem ser divididos em agentes específicos para realizar certas tarefas, agentes “recreativos” e em vírus.

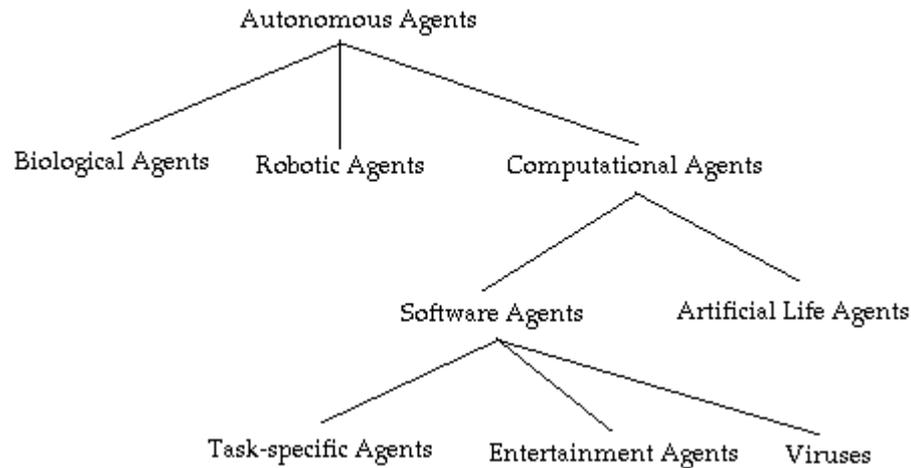


Figura 3 – Taxonomia possível para classificar agentes

## **AgentSpace**

O AgentSpace é um “Sistema de Suporte a Agentes”, ou seja um “SAS” que permite o suporte, o desenvolvimento e a gestão de aplicações dinâmicas e distribuídas baseadas em agentes.

### **2.1.1.3 Arquitectura**

O ambiente de execução dos agentes é a máquina virtual Java versão JDK 1.1. pois esta revela-se a tecnologia adequada para realizar a integração com a tecnologia da Internet/Web. O próprio SSA é realizado em Java e os agentes são executados exclusivamente como actividades no contexto computacional do próprio servidor.

A infra-estrutura comunicacional de suporte utilizada foi o ORB Voyager que pode ser descrito como um ORB tradicional com um conjunto de especificadas adicionais que já foram referidas anteriormente.

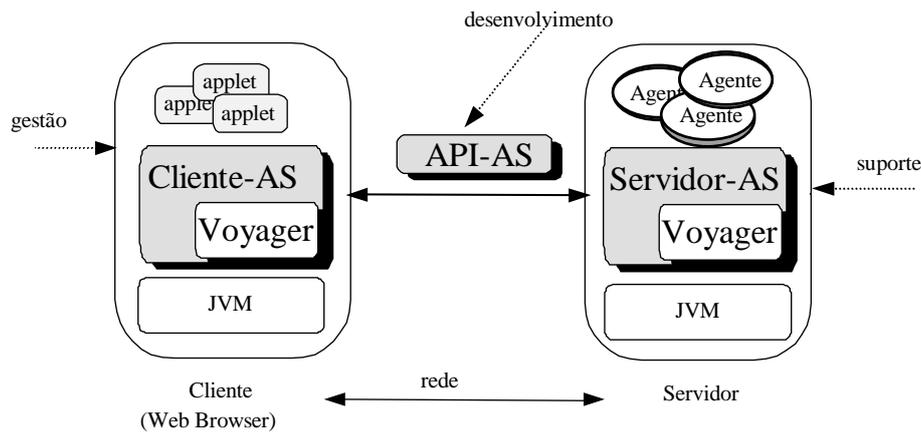


Figura 4 – Visão geral da infraestrutura do AgentSpace

A arquitectura do AgentSpace divide-se em três componentes integradas conforme é visível neste gráfico:

- **Servidor AgentSpace**

Consiste no processo Java no qual os agentes são executados. O servidor permite realizar a criação de agentes e locais de execução, execução de agentes, persistência, controlo de acessos, suporte à mobilidade e comunicação de agentes, geração de identificadores e interface simples (shell) de gestão e monitorização .

- **API AgentSpace**

É uma biblioteca de classes e interfaces em java que suporta a construção de agentes e applets para posterior utilização no Servidor-AS e como interface gráfica dos utilizadores respectivamente.

- **Cliente AgentSpace**

Também é uma aplicação em Java desenvolvida para permitir a gestão e monitorização de agentes e outros recursos de forma remota em um ou mais Servidores-AS.

Tanto o cliente como o servidor são executados sobre a MJV (máquina virtual Java) e utilizam algumas das funcionalidades do Voyager. Os agentes interagem com os utilizadores através de applets sendo no entanto sempre executados no contexto do servidor.

#### 2.1.1.4 Modelo de Objectos

Os agentes são apenas um objecto num conjunto vasto gerido pelo AgentSpace. A compreensão correcta do modelo de objectos facilita a compreensão de todo o conceito de agente e como não podia deixar de ser da própria aplicação em si. O modelo de objectos é

constituído por objectos como o contexto, local, controlo de acessos, grupo de utilizadores, utilizadores e agentes entre outros.

O diagrama UML aqui apresentado permite clarificar as relações entre objectos de uma forma intuitiva e clara.

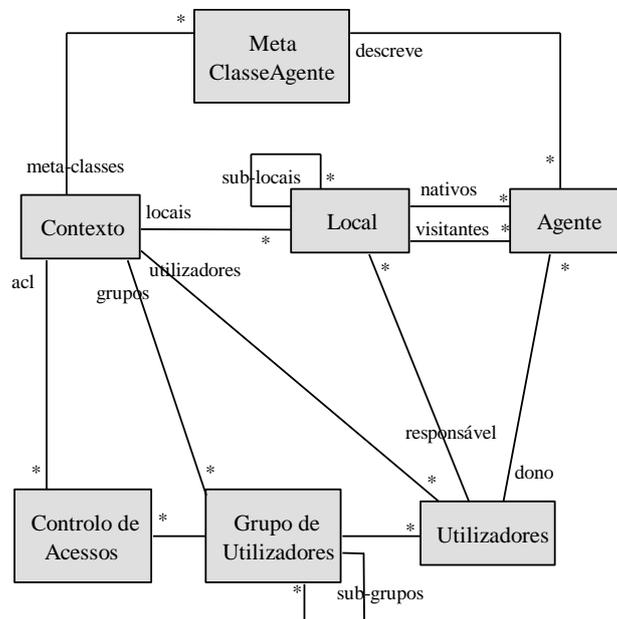


Figura 5 – Modelo de objectos geral do AgentSpace – diagrama de classes

O objecto que se evidencia é o contexto pois representa o Servidor-AS e reúne as principais estruturas como locais, utilizadores, grupos de utilizadores e listas de controlo de acessos. O objecto contexto permite segundo o seu autor:

- providenciar uma metáfora de programação onde os agentes sejam executados e se encontrem com outros agentes
- providenciar um mecanismo consistente de definição e gestão de controlos de acesso e de gestão de recursos

O contexto pode englobar vários locais de execução que podem ou não estar organizados hierarquicamente. A cada local de execução é atribuído um identificador único e global, sendo a identificação do seu gestor responsável e a caracterização informal (do local) através de uma lista de palavras-chaves, parte da sua identificação.

Cada local mantém uma lista de agentes nativos, que foram originalmente criados no próprio local, e outra com agentes visitantes, que se encontram no local de execução mas não foram lá criados.

As expressões local nativo e local de visita correspondem à mesma situação vista da perspectiva dos agentes. É possível definir para cada local o número máximo de agentes nativos e de agentes visitantes permitido, característica esta que também é usada neste trabalho.

Os agentes podem navegar entre vários locais de execução desde que tenham as permissões necessárias. Segundo o autor, cada classe de agente é descrita por uma estrutura designada por “MetaClasseAgente” que contém informação sobre o seu autor, versão, ontologia e outras propriedades. Tal como é visível no modelo de objectos, o Servidor-AS mantém listas de utilizadores e grupos de utilizadores.

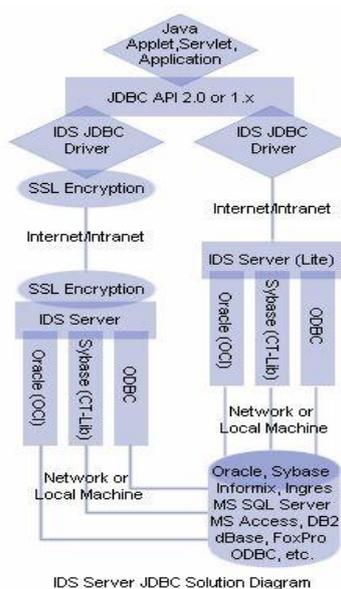
À imagem do que acontece para os locais, cada agente tem também um identificador único e global. Segundo o autor, cada agente representa uma unidade básica de computação providenciada pelo AgentSpace, sendo cada agente um objecto activo executado num conjunto de actividades do contexto computacional do Servidor-AS.

Cada agente pertence a um utilizador que por sua vez pertence a um grupo de utilizadores. Os outros utilizadores existentes podem interagir com o agente desde que tenham permissões para tal.

### **JDBC DRIVER**

O aparecimento das tecnologias Java permitiu à indústria da Internet criar conteúdos dinâmicos, plataformas independentes e aplicações de Internet distribuídas. Utilizando o Java e a Java Database Connectivity API (JDBC) é possível construir aplicações internet poderosas com acesso a bases de dados e uma interface gráfica complexa. O JDBC obedece ao conceito “write once, run everywhere” de modo a que o código criado seja independente da plataforma em que corre.

As aplicações Java podem aceder a qualquer base de dados através do IDS Server. O JDBC driver é feito em “100% Java” independente de qualquer plataforma e apropriado para Java applets e programas Java. É compatível com todos os browsers que suportam java, todas as versões do JDK, com o Microsoft SDK e outras ferramentas de desenvolvimento de Java.



**Figura 6 – Diagrama de solução do IDS Server JDBC**

O driver utilizado é um driver de tipo 3 ou seja um *Type 3-Net Protocoll All-java Driver*. Este tipo de drivers comunicam via um protocolo genérico para um middleware genérico. Neste caso o IDS Server faz parte do middleware.

O IDS Server é um servidor de acesso às bases de dados da Internet. Este servidor permite que programadores de HTML e Java possam criar páginas Web interactivas, Java applets, etc.

O IDS Server suporta todos os sistemas que suportam ODBC, para além das seguintes bases de dados: Oracle nativo, Sybase e Informix através das suas respectivas APIs clientes. Estas APIs clientes são a Oracle Call Interface, Sybase CT-Lib and Informix Client-SDK. O IDS Server suporta as plataformas Windows 98/95 e Windows NT em Intel e Alpha, Solaris no Sun Sparc, e Linux no Intel.

A versão utilizada foi a IDS Server Versão 3.2 Lite mais apropriada para applets pois é um driver muito compacto e de alta performance. As classes Java só ocupam 88 Kbytes (Versao Lite).

Muitos gateways de bases de dados são implementados utilizando CGI, ISAPI e NSAPI. A grande desvantagem deste tipo de design é o facto de estas aplicações se apoiarem no funcionamento do servidor web, o que implica que para além de consumirem muito tempo do

CPU e outros recursos, também necessitam que o servidor Web processe informação de e para os browsers clientes.

Como estes gateways correm normalmente na mesma máquina as suas actividades nas bases de dados descem a performance do sistema. No caso das implementações de ISAPI e NSAPI um erro no gateway pode levar ao crash do próprio servidor Web.

O IDS Server elimina estes problemas pois não foi construído sobre estas 3 APIs. O IDS Server é uma aplicação stand-alone que corre sobre um IP e um porto determinado. Os browsers clientes e as aplicações ligam-se directamente ao IDS Server para realizar o acesso às bases de dados, aliviando o servidor web destas actividades.

Note-se que o driver da IDS é apenas um dos muitos drivers disponíveis no mercado sendo a aplicação compatível com qualquer um dos drivers existentes. A instalação deste trabalho na máquina cosmos foi realizada utilizando as bases de dados postgres e o seus drivers.

## **2.2 Conceitos ligados ao E-business**

Antes de estabelecer os requisitos a que este trabalho deveria obedecer foi necessário realizar um estudo sobre o comércio electrónico e os modelos de negócio actualmente vigentes na Web.

### ***Electronic Business***

O Electronic Business é um termo geral utilizado para descrever a condução de negócio com a assistência das telecomunicações e as suas ferramentas.

Esta definição compreende muitos segmentos que se sobrepõem entre si como o comércio electrónico, o electronic publishing, o electronic services delivery, deliberative purchasing e spontaneous purchasing. [8]

### ***Electronic Publishing***

Electronic Publishing é o comércio electrónico em bens e serviços digitais que são para ser consumidos pelos sentidos humanos. Esta definição inclui um conjunto de formatos como texto, imagens, som, etc. ...

### **Electronic Services Delivery**

Electronic Services Delivery é o fornecimento de serviços com a assistência das telecomunicações e das suas ferramentas. O ESD lida apenas com actividades que possam ser completamente realizadas de forma electrónica. Exemplos do ESD são actividades como o netbanking e outros serviços financeiros, reservas para viagens e eventos culturais, etc...

### **Deliberative Purchasing**

Deliberative Purchasing refere-se ao processo praticado pelas corporações quando estas realizam aquisições grandes. Caracteriza-se este processo por apresentar 6 fases distintas: A fase pre-contratual, a contratual, a fase de pedido, a fase logística, a fase de pagamento e a fase pós-processamento.

### **Spontaneous Purchasing**

As transacções que envolvem pequenas quantidades de dinheiro são geralmente feitas de forma espontânea.

Este conceito refere-se ao tipo de *procurement* onde o comprador realiza o mínimo de decisões racionais, ou seja, a um processo de 3 fases : SEE, PAY, GET. Este modelo aplica-se a situações diversas como o supermercado e as trocas de acções on-line.

### **Electronic Commerce**

Electronic Commerce ou e-commerce é normalmente descrito como a condução de negócio em bens e serviços com a assistência das telecomunicações e suas ferramentas.

Os termos *electronic trading*, *electronic procurement*, *electronic purchasing* ou *electronic marketing* apresentam o mesmo significado[16].

O próprio comércio electrónico inclui vários segmentos com o segmento de catálogos electrónicos que permitem aos vendedores exporem os seus produtos a potenciais compradores. Outro segmento é o Electronic Data Interchange (EDI) [17], que é um standard utilizado para expressar dados estruturados que representam transacções EC.

Os leilões electrónicos são uma outra vertente atribuir preços a bens de forma dinâmica.

do comércio electrónico sendo estes um conjunto de mecanismos utilizados para A figura seguinte esquematiza um conjunto de conceitos que aparecem constantemente associados ao comércio electrónico. O círculo interior representa a infraestrutura tanto a nível de hardware como de software necessária ao comércio electrónico.

O anel intermédio representa a camada correspondente às aplicações e ferramentas utilizadas. O anel exterior corresponde às aplicações e funções que definem o web site. Neste gráfico vê-se apenas algumas das aplicações possíveis.

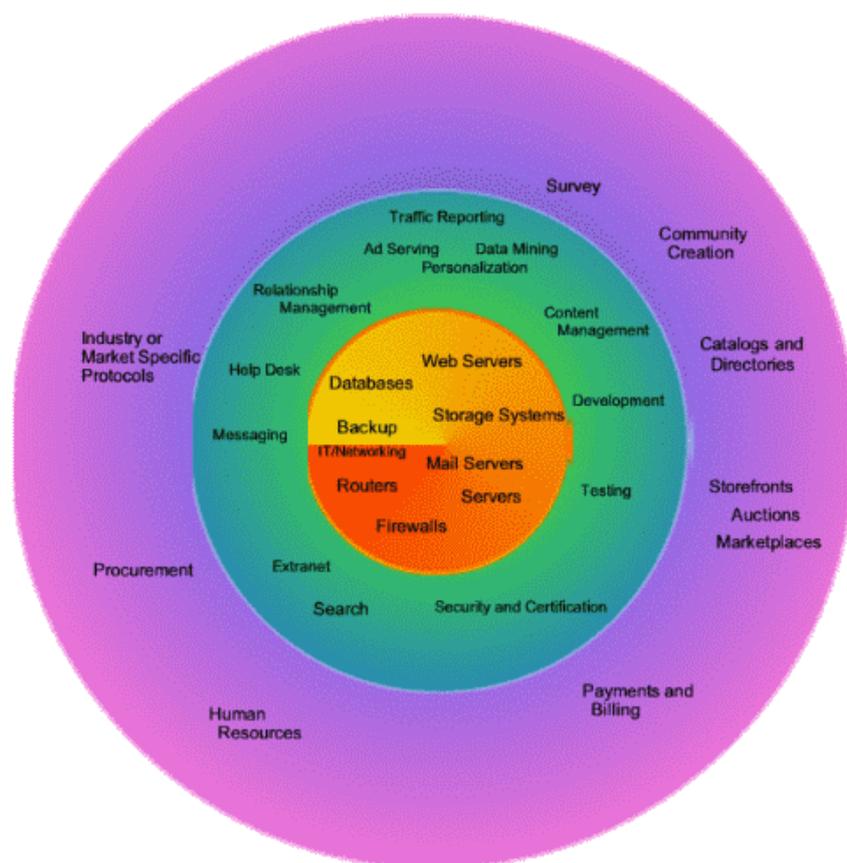


Figura 7 – Representação esquemática do comércio electrónico em 3 aneis

### **Classificação dos itens transaccionados**

As características dos itens transaccionados são um determinante no mecanismo de transacção. É possível caracterizar estes itens segundo várias dimensões.

## **Bens e Serviços**

Distinguem-se dois tipos de itens, os bens e os serviços. Define-se um *bem* como uma entidade física que se pode entregar enquanto um *serviço* é uma acção que é prestada.

## **Itens transaccionados fisicamente e digitalmente**

Também é importante distinguir bens físicos de bens digitais. Um item digital é um item que pode ser entregue ou executado através de uma rede de telecomunicações. A entrega ou performance de um item físico envolve actividades logísticas como o transporte de bens ou da pessoa (às instalações) onde o serviço será executado.

Esta dimensão é separada da anterior pois tanto bens como serviços podem ser físicos ou digitais. Um audio CD é um bem físico, enquanto um ficheiro audio que se pode descarregar da Internet é um bem digital. Exemplos de um serviço físico é um corte de cabelo e de um serviço digital é a resposta a dúvidas de programação numa lista de dúvidas.

Neste trabalho classificaram-se todos os itens transaccionados em serviços e bens sem fazer a distinção se são electrónicos ou físicos.

## **O grau de standardização dos itens transaccionados**

Os itens transaccionados podem ser classificados de acordo com um de quatro níveis de standardização.

### **Standard Products**

Um item standard é um produtos ou serviço existente em quantidade num determinado fornecedor e que tem uma identidade única permitindo aos consumidores fazerem pedidos desse produto ou serviço especificamente através do catálogo do fornecedor.

### **Commodities**

Esta é uma classe particular de produtos que existe debaixo de uma forma identificável em quantidades consideráveis e que estão disponíveis numa variedade de fontes. Exemplos deste tipo de produtos são as acções e instrumentos financeiros derivados, moeda estrangeira e produtos primários como crude e café.

### **Custom-Built Products**

Este tipo de produtos é desenhado especificamente para cada cliente de modo a satisfazer um caso específico. Fábricas de embarcações são um exemplo de produtos custom-built e a consultadoria de

negócio é um exemplo de um serviço custom-built. Esta classe de produtos ou serviços são normalmente sujeitos a Request for Information (RFI), Propostas (RFP) e Oferta (RFO).

### **Customised Products**

As três classes anteriores não conseguem abarcar todas as formas de negócio. Muitas circunstâncias existem em que os produtos base existentes são modificados de forma a satisfazerem as necessidades do cliente. Este tipo de situação é particularmente recorrente onde existe uma combinação de produtos e serviços.

Como customização entende-se:

<i>Tipo de Customização</i>	<i>Exemplos</i>
<i>Opções de produto</i>	cor e tamanho
<i>Extras opcionais</i>	Ex: ar condicionado no automóvel
<i>Parametrização</i>	
<i>Especialização</i>	inserção do nome da entidade no produto
<i>Extensões customizadas</i>	adição de elementos extra
<i>Modificações customizadas</i>	
<i>Serviços suplementares</i>	

**Tabela 2 – Customizações possíveis num produto**

As lojas deste centro comercial estão configuradas para comercializarem bens e serviços standard. Uma evolução do modelo das lojas pode permitir criar produtos com um grau de liberdade superior.

### **2.3 Modelos de Negócio**

Não havendo presentemente uma taxonomia única e compreensiva dos modelos de negócio existentes na Web adoptou-se a taxonomia apresentada no Apêndice A [20].

As formas genéricas de modelos de negócio electrónico que se encontram actualmente na Web são: Brokerage, Advertising, Infomediary, Merchant, Manufacturer, Affiliate, Community, Subscription e Utility.

Os Brokers (vide Brokerage) juntam compradores e vendedores e facilitam as transacções entre estes. Estas transacções podem ser business-to-business, business-to-consumer ou consumer-to-consumer. Um broker faz dinheiro cobrando taxas por cada transacção realizada. Os modelos de brokerage podem tomar várias formas sendo de destacar o Virtual Mall. O Virtual Mall é um site que alberga muitos mercadores online. O modelo cobra a instalação das lojas e normalmente por transacção. Exemplo deste modelo são as lojas no Yahoo. Este modelo de shopping virtual torna-se mais eficiente quando combinado com um portal geral. Os malls mais sofisticados oferecem serviços transaccionais automatizados e oportunidades de marketing relationship.

Este foi o modelo de negócio escolhido na implementação deste trabalho. A comparação com os outros modelos de negócio existentes (vide Apêndice B) deixa antever as vantagens e desvantagens deste modelo. Este modelo permite uma evolução contínua dos agentes broker, ou seja, uma evolução na complexidade e riqueza de serviços que estes agentes podem oferecer aos clientes. Uma clara vantagem na utilização de uma tecnologia baseada em agentes advém das suas características inerentes, ou seja, no facto do próprio conceito de agente personificar um utilizador dentro do sistema, o que leva a uma capacidade de personalização superior a de outras tecnologias. O modelo implementado neste trabalho revela-se como um excelente modelo básico mas necessita de apresentar algumas características ou funcionalidades extra para atrair e fidelizar o cliente, ou seja, para fazer a diferença. Um portal especializado com conteúdos específicos e ricos e uma gama de serviços de procura, monitorização e acompanhamento do cliente poderão levar à mencionada diferença.

### **3 Requisitos**

Este centro comercial deverá ser de fácil gestão, atraente e a sua interface deverá ser compatível com qualquer browser que suporte Java. O facto de ser implementado em tecnologia de agentes deverá ser completamente transparente aos utilizadores. Também deverá ser possível acrescentar de forma modular novos serviços ao centro comercial sem ter de reescrever toda a lógica de negócio. O aparecimento de novos lojistas e utilizadores deverá ser suportado de forma dinâmica. O aparecimentos de produtos e serviços novos deverá estar reflectido nas ferramentas de procura do centro comercial. O aparecimento de um grande número de lojistas e utilizadores deverá ser imperceptível, ou seja, a escalabilidade deverá ser suportada.

#### **3.1 Requisitos estruturais**

Os requisitos estruturais deste centro comercial electrónico resumem-se ao seguinte:

O centro comercial deverá ser construído com base na tecnologia de agentes, neste caso, ligada ao AgentSpace. Deverá apresentar uma interface gráfica de fácil navegação e que corra em qualquer browser cliente que suporte Java. Ficou ainda claramente explicitado que as páginas HTML têm de ser estáticas sendo todas as ligações entre os utilizadores e os agentes realizadas através de applets. O facto das páginas não serem geradas dinamicamente traz algumas desvantagens à apresentação das lojas do centro comercial. O sistema de catálogo das lojas tem de ser estático e todas as acções ligadas ao processo de compra, têm de ser realizadas através de applets.

O acesso às bases de dados deverá ser por JDBC de modo a que a portabilidade da aplicação desenvolvida não fique comprometida.

#### **3.2 Requisitos funcionais**

Os requisitos funcionais foram definidos com base no modelo de negócio já anteriormente mencionado. O modelo de centro comercial virtual prevê a existência de um conjunto de lojas agregadas por um mediador. Prevê também a obrigatoriedade dos clientes se registarem no centro comercial. O mediador deverá oferecer serviços de pesquisa aos clientes entre as várias lojas do centro comercial. O cliente deverá ainda poder usufruir da possibilidade de utilizar

um cesto de compras e de facilidades como não ter de reinserir os seus dados sempre que desejar efectuar uma aquisição.

Passa-se à especificação mais detalhada do modelo de negócio “Virtual Mall” tendo em conta que se está a especificar o centro comercial para uma aplicação que utiliza tecnologia de agentes. A descrição completa do modelo implementado será no entanto apenas realizada no capítulo seguinte.

### 3.2.1.1 Requisitos do centro comercial

O centro comercial electrónico que se vai implementar vai ser um site com um conjunto de lojas cada uma representada por pelo menos uma página HTML.

Este centro comercial deverá ter uma página inicial com links para os vários serviços e funcionalidades que o centro comercial proporciona, sendo o principal, um serviço de procura por produtos e/ou serviços e por nome de loja. A procura deve ser realizada por mais do que um parâmetro. Deverá também existir uma lista com os nomes de todas as lojas e com links directos para elas.

Haverá também uma interface gráfica que possibilite aos visitantes interessados candidatarem-se para terem a sua própria loja, processo este que terá de ser aprovado pelo gestor do centro comercial.

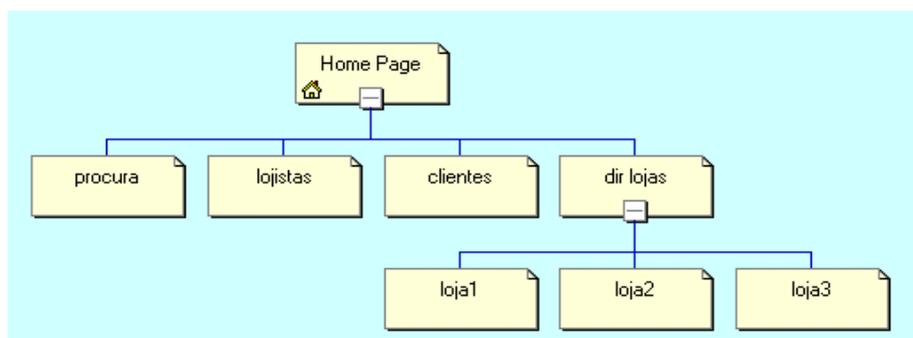


Figura 8 – Mapa do site que obedece aos requisitos

O cesto de cada cliente será gerido pelo centro comercial e não pelas diversas lojas. Isto implica que a informação do cliente será mantida pelo centro comercial e que haverá uma interface para os utilizadores se registarem e gerirem a sua informação pessoal.

### 3.2.1.2 Requisitos do Gestor

O centro comercial deverá ter uma interface gráfica que permita criar e gerir as lojas, os lojistas e os seus clientes. O gestor não deverá ter acesso directo aos produtos e serviços

criados oferecidos por cada loja. No entanto tem a possibilidade de eliminar qualquer loja existente no centro comercial tal como de aceder a toda a informação sobre a mesma. O aparecimento de uma loja nova tem de ser sempre autorizado pelo gestor.

### **3.2.1.3 Requisitos da Loja**

Cada loja deverá apresentar uma lista dos produtos e/ou serviços que oferece.

A facilidade de gestão das lojas com uma interface gráfica simples e acessível e compatível com qualquer browser é mais um ponto a favor deste centro comercial e de atrair os possíveis lojistas. Os lojistas deverão ter uma forma de gerir os seus produtos por interface gráfica.

O site inicial da loja deverá ser gerado dinamicamente a partir das informações inseridas pelo candidato a lojista.

Cada loja deverá também oferecer ao cliente a possibilidade de ter um cesto de compras ou possibilitar o acesso a um cesto de compras onde o utilizador possa pôr os artigos em que esteja interessado. O processo de compra deve ser simples e intuitivo para os clientes.

### **3.2.1.4 Requisitos do Cliente**

O cliente tem de se registar no centro comercial para poder realizar qualquer compra no centro comercial. A sua informação pessoal deve ser gerida pelo próprio centro comercial e partilhada por todas as lojas. O cliente deverá ter um cesto onde pode pôr todos os produtos ou serviços que deseja adquirir. Este cesto vai ser persistente ou seja quando o cliente voltar ao centro comercial os artigos no seu cesto de compras ainda lá estão.

## **3.3 Abordagem Baseada em Agentes**

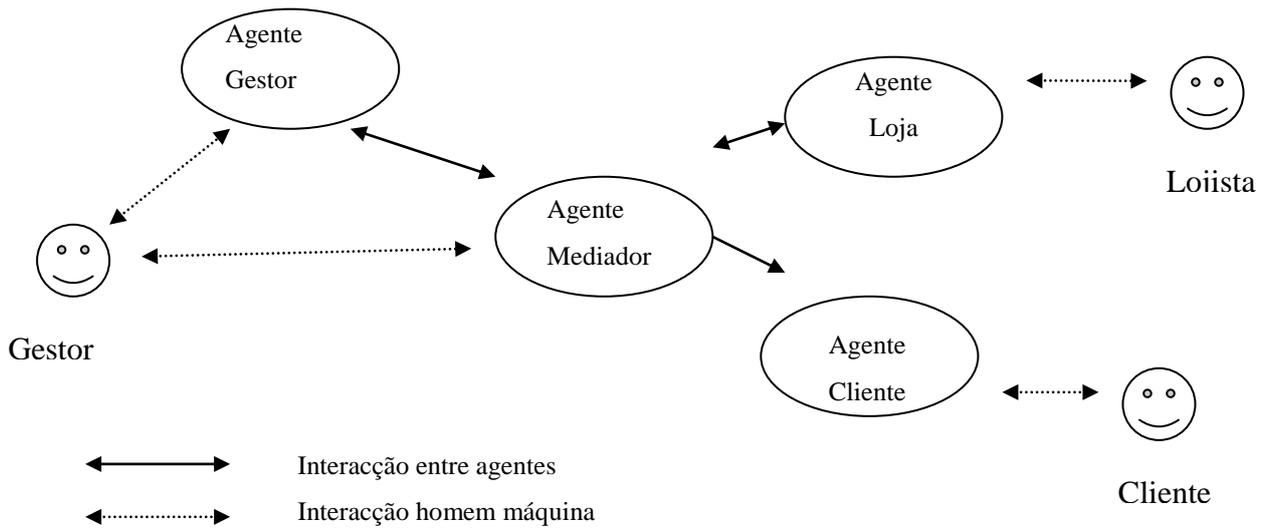
A abordagem baseada em agentes apresenta, tal como todas as outras, vantagens e desvantagens. As suas desvantagens estão principalmente ligadas ao facto da tecnologia de agentes ser emergente e de ainda não ter conquistado a confiança dos utilizadores nem o consenso da comunidade científica.

Os utilizadores receiam delegar decisões e tarefas em agentes e a comunidade científica ainda não desenvolveu standards para esta tecnologia.

As principais vantagens desta solução são o seu dinamismo, pois os agentes entram e saem da comunidade em qualquer altura, o facto de ser descentralizada e escalável, pois o agente mediador é o ponto de congestão e podem existir vários agentes deste tipo a funcionar em

simultâneo. A sua autonomia e flexibilidade também é superior a qualquer outro tipo de abordagem pois cada interveniente mantém e constrói os seus próprios agentes.

A definição dos requisitos do centro comercial permite facilmente realizar uma abordagem baseada em agentes. Uma abordagem baseada em agentes implica a existência de um ou mais agentes intermediários. A atribuição de um agente a cada entidade interveniente no processo leva a um esquema do género:



**Figura 9 - Aproximação ao modelo de centro comercial baseada em agentes**

Cada utilizador interage com o seu agente especializado definindo os objectivos que o agente tem de cumprir e dando-lhe um maior ou menor grau de autonomia. O agente mediador neste caso pode ou não ser o gestor do próprio centro comercial. Para a mediação ser eficaz e mesmo possível entre agentes é necessário que os agentes loja se registem no mediador de modo a que este possa fornecer serviços de procura adequados.

No caso deste centro comercial pode-se considerar que o gestor e o mediador do centro comercial são a mesma pessoa/entidade.

Este tipo de abordagem em comparação com a aproximação centralizada e a aproximação descentralizada revela-se mais autónoma e flexível pois permite aos potenciais compradores encontrarem rapidamente e sem esforço os produtos e serviços que desejam [14].

## **4 Arquitectura**

Neste capítulo descreve-se a arquitectura de um centro comercial virtual construído sobre o AgentSpace. A aplicação apresenta uma arquitectura lógica de 3 níveis baseada em objectos Java distribuídos que são sustentados pelo AgentSpace e o Voyager.

Na próxima secção é descrita de uma forma geral a arquitectura escolhida. Ao longo das restantes secções são abordados de forma mais pormenorizada os pontos fundamentais desta arquitectura.

### **4.1 Descrição Geral**

Os objectivos que nortearam a concepção deste trabalho podem ser resumidos no seguinte: construção e implementação de uma aplicação que tem como base uma arquitectura de três níveis constituída por vários agentes (objectos java) ao nível da Lógica de Negócio que suportem operações transaccionais. Os utilizadores da aplicação interagem com esta através de applets podendo realizar uma vasta gama de tarefas dependendo estas do seu papel (do utilizador) em todo o sistema. O sistema de base de dados aqui apresentado é gerido por vários agentes e de forma transaccional de modo a que haja sempre consistência nos dados armazenados.

Este tipo de arquitectura permite alcançar os seguintes objectivos:

#### **Extensão da funcionalidade do sistema**

O sistema deve ser constituído por um conjunto de agentes, deve ser expansível de forma modular, deve oferecer uma interface uniforme e intuitiva e deve permitir a ligação a todos os tipos de bases de dados.

#### **Reusabilidade e facilidade da manutenção dos agentes**

Os detalhes de toda a funcionalidade encontram-se encapsulados em cada agente específico. Esta abstracção da complexidade permite que os agentes possam ser reutilizados nouro tipo de aplicação.

#### **Suporte a múltiplos tipos de agentes**

O sistema deve suportar qualquer tipo de cliente independentemente da sua localização. Esta característica é garantida uma vez que a localização vai ser executada num Browser.

## Operações Transaccionais

Quando um cliente efectua um pedido a um serviço disponibilizado pelo sistema, ele deve ser realizado de forma transaccional. A possibilidade de recuperação de erros torna-se assim muito mais viável. Desta maneira toda a informação que existe no sistema pode ser considerada válida podendo ser manipulada com segurança.

## Escalabilidade da Aplicação

Para proporcionar aos utilizadores uma boa performance na sua interacção com o sistema é necessário que os recursos deste não sejam gravemente afectados pelo aumento dos utilizadores.

Esta é uma arquitectura baseada nas mais recentes linhas de orientação para a construção de aplicações cliente-servidor, e que assentam numa arquitectura a três níveis:

- Nível de Apresentação
- Nível da Lógica da Aplicação
- Nível dos Serviços de Base de dados

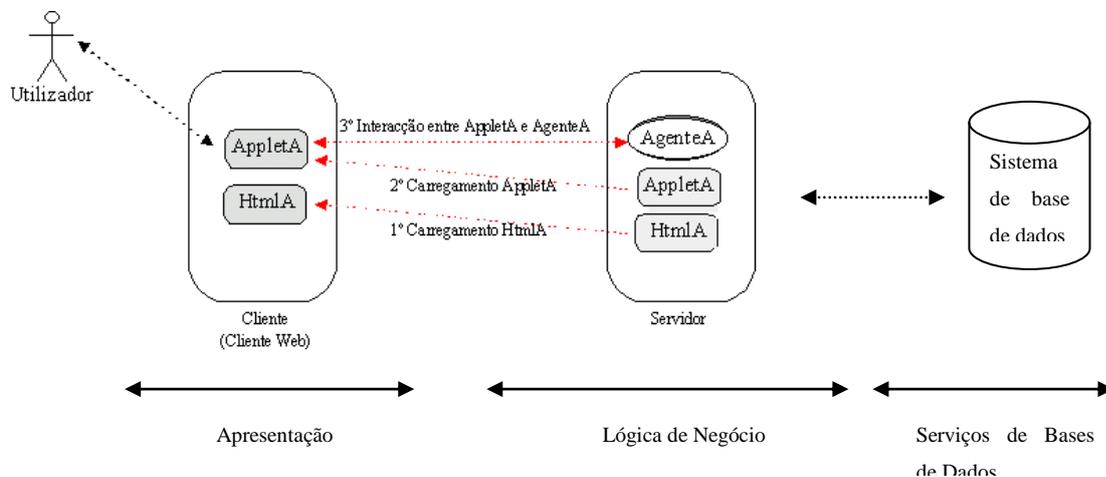


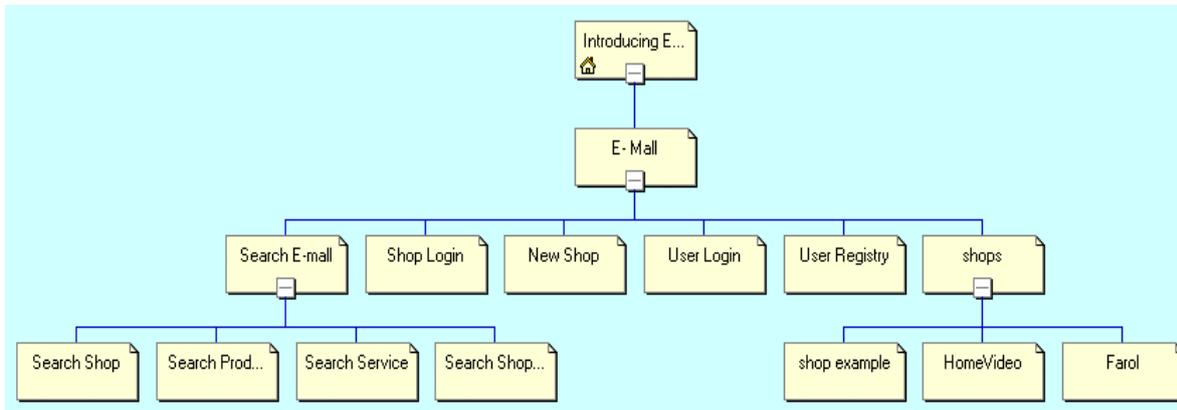
Figura 10 – Arquitectura genérica da aplicação

## Nível de Apresentação

A este nível construiu-se uma estrutura de páginas HTML que representam o centro comercial electrónico.

Note-se que todo o aspecto gráfico desta aplicação é independente do funcionamento da aplicação, o que implica que o aspecto gráfico de todo o centro comercial pode ser facilmente alterado ao gosto de cada gestor.

No entanto, como já foi mencionado, a única forma que as páginas HTML têm de interagir com a aplicação é através de applets.



**Figura 11 – Mapa navegacional do site construído**

As primeiras páginas servem para introduzir o centro comercial virtual, contendo a primeira uma explicação técnica do trabalho e a segunda um índice de links para todas as outras.

### **Search Email**

Esta página apresenta as quatro formas possíveis de efectuar procuras no centro comercial electrónico e tem ligações para as quatro páginas que permitem realizar procuras por lojas, produtos, serviços ou tipos de lojas. Cada uma destas quatro páginas contém uma applet que permite realizar uma procura. Não se incluíram todas as applets numa applet ou mesmo as quatro applets numa página por razões de performance.

### **Shop Login**

Esta página inclui uma applet que permite ao lojista gerir os produtos e serviços que a sua loja oferece.

### **New Shop**

Aqui um possível interessado em ser um lojista introduz os seus dados e a informação é enviada para ser posteriormente processada pelo agente gestor do centro comercial.

**User Login**

Esta página serve para os utilizadores do centro comercial gerirem os seus cestos de compras e a sua informação pessoal.

**User Registry**

Este centro comercial só permite a realização de compras se o utilizador for registado. Esta página serve para os utilizadores se registarem.

**Shops**

Nesta página existe um índice para todas as lojas existentes no centro comercial.

O aparecimento de uma loja nova implica a geração automática de uma página HTML correspondente à loja e também a inserção de um novo link para a loja nova nesta página.

***Nível da Lógica da aplicação***

Este nível, também conhecido por middle tier ou middleware, serve para processar o fluxo de informação entre a interface e um sistema de bases de dados. O software utilizado para realizar este processamento é um software que pode incluir lógica de negócio. Sem este nível intermédio, a aplicação ter-se-ia de ligar directamente ao servidor de bases de dados o que implicaria que qualquer mudança tanto na aplicação como no servidor de bases de dados poderia levar ao mau funcionamento da aplicação.

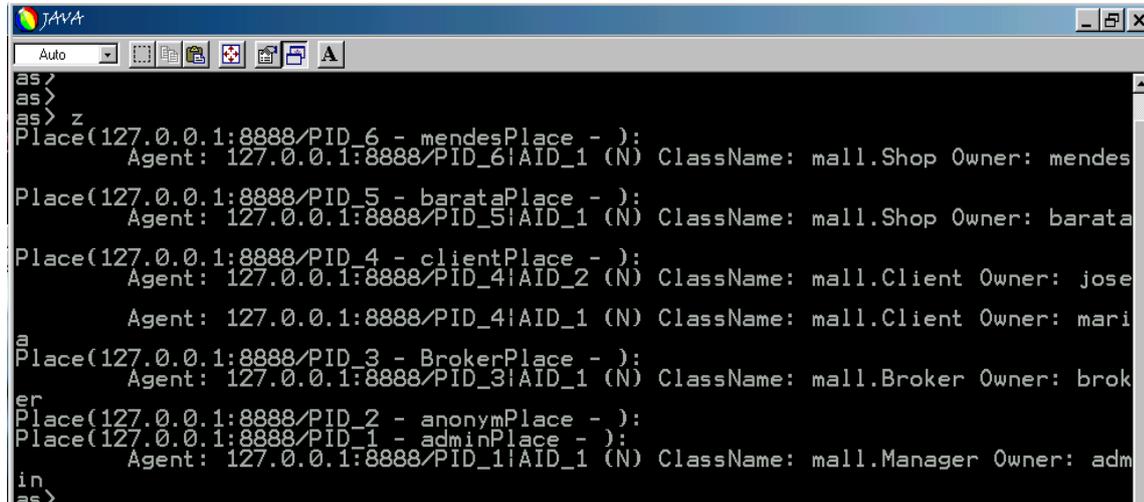
A lógica de negócio é uma abstracção de tarefas low-level para tarefas high-level como o processamento de uma encomenda ou a gestão do cesto de cada cliente.

O Middleware permite melhorar o desempenho do sistema porque passa a ser possível realizar a distribuição de carga por vários servidores de back-end. Este nível consegue utilizar a largura de banda de forma mais eficiente pois a comunicação passa a ser realizada em duas etapas distintas: o cliente diz ao middleware o que quer e o middleware comunica com o servidor de bases de dados de forma mais rápida.

São os próprios agentes que implementam a funcionalidade do sistema e portanto a lógica de negócio. Cada agente foi concebido e desenhado para cumprir um conjunto específico de tarefas.

## Definição dos Modelo de Gestão de Locais de Execução, Agentes e Utilizadores

A definição do modelo de gestão dos vários objectos descritos na secção anterior revela--se importante para definir todo o funcionamento da aplicação.



```
as >
as >
as > z
Place(127.0.0.1:8888/PID_6 - mendesPlace - ):
  Agent: 127.0.0.1:8888/PID_6|AID_1 (N) ClassName: mall.Shop Owner: mendes
Place(127.0.0.1:8888/PID_5 - barataPlace - ):
  Agent: 127.0.0.1:8888/PID_5|AID_1 (N) ClassName: mall.Shop Owner: barata
Place(127.0.0.1:8888/PID_4 - clientPlace - ):
  Agent: 127.0.0.1:8888/PID_4|AID_2 (N) ClassName: mall.Client Owner: jose
  Agent: 127.0.0.1:8888/PID_4|AID_1 (N) ClassName: mall.Client Owner: mari
a
Place(127.0.0.1:8888/PID_3 - BrokerPlace - ):
  Agent: 127.0.0.1:8888/PID_3|AID_1 (N) ClassName: mall.Broker Owner: brok
er
Place(127.0.0.1:8888/PID_2 - anonymPlace - ):
Place(127.0.0.1:8888/PID_1 - adminPlace - ):
  Agent: 127.0.0.1:8888/PID_1|AID_1 (N) ClassName: mall.Manager Owner: adm
in
as >
```

Figura 12 – Ilustração do modelo de gestão de locais de execução e de agentes

### Manager

Ao contrário de todos os outros agentes nesta aplicação, o agente manager é iniciado manualmente pelo utilizador, ou seja, pelo gestor do centro comercial. O facto de não ser iniciado automaticamente implica que o seu local de execução e o seu dono não podem ser controlados à partida pela aplicação.

### Broker

O agente broker é criado pelo agente manager quando este se inicializa.

O manager começa por criar um utilizador que faz parte do grupo de utilizadores “adminGr” e um local de execução chamado “BrokerPlace”. Posteriormente o manager cria o broker associando-o ao BrokerPlace e ao utilizador interno broker.

### Client

O broker por sua vez cria o clientPlace que vai ser o local de execução nativo para todos os agentes dos clientes. Considerou-se à partida que um local de execução para todos os agentes clientes é suficiente. Associado a cada agente cliente está um utilizador interno definido pelo broker e criado a partir dos dados fornecidos pelos utilizadores.

**Shop**

É o agente manager que inicializa o agente loja. Tal como no caso do agente Client é criado um utilizador para cada agente Shop com base na informação preenchida pelo lojista. No entanto, ao contrário do que se passa com o agente Client é criado um local de execução especificamente para cada agente Shop.

***Nível de serviços de base de dados***

Aqui é realizado o armazenamento de dados do sistema. A ligação utilizada é por JDBC que permite uma portabilidade única a todo o sistema. Utilizam-se três bases de dados distintas:

1. Manager – informação do gestor sobre todas as lojas existentes no centro comercial. Esta base de dados é manipulada exclusivamente pelo agente “manager”.
2. Broker – informação sobre todos os clientes do centro comercial, todos os produtos, serviços, tipo de lojas e lojas mantida pelo agente broker. Como o agente dos clientes não mantém nenhuma ligação a bases de dados é o agente broker que faz isso por ele.
3. Shop – informação detalhada sobre todos os produtos de cada loja, transacções efectuadas, etc... Todos os agentes “shop” partilham a mesma base de dados.

**4.2 Modelo de Dados**

A concepção do modelo de dados levou à definição de todas as bases de dados e classes utilizadas para encapsular a informação neste sistema. Começa-se por abordar brevemente as várias classes utilizadas para transmitir informação, passando-se posteriormente à descrição detalhada de algumas das operações principais suportadas pelo sistema e por fim, à descrição das várias bases de dados.

O modelo de dados foi definido em função das tarefas e funcionalidades que cada agente do sistema tem de realizar e portanto, de toda a informação que tem de ser partilhada e trocada entre eles.

O fluxo de dados é controlado pelos vários agentes havendo um pequeno conjunto de dados mantido dinamicamente pelos agentes durante a sua execução e armazenados posteriormente nas várias bases de dados. A troca de informação entre os vários agentes e as várias formas de interface existentes é feita através do mecanismo de mensagens assíncronas e síncronas. Para

permitir o fluxo de dados entre os vários agentes e entre as interfaces definiram-se várias classes que encapsulam a informação de forma adequada às várias operações a efectuar.

### **Entidades e Conceitos definidos no modelo de dados**

O próprio conceito de centro comercial implica a definição de várias entidades como lojista, loja, cliente e de alguns conceitos como tipo de loja, cesto de compras, etc. ...

#### **Entidades:**

- **Lojista:** A definição de lojista neste centro comercial inclui apenas um identificador único, um nome e o email do lojista. Note-se que o lojista pode possuir mais do que uma loja e a sua interacção com o centro comercial é sempre realizada através da interface das lojas.
- **Loja:** Uma loja caracteriza-se por um conjunto de parâmetros que a tornam única. Considerou-se que os parâmetros necessários para descrever as lojas eram os seguintes: nome, sigla, lojista, tipo de loja, nif, email, login, password e o número máximo de agentes nativos e de agentes visitantes no local de execução do agente loja correspondente. A cada loja é atribuído um identificador único, um agente e um local de execução.
- **Cliente:** Um cliente caracteriza-se também por um conjunto de atributos relacionados com a sua habilidade para realizar aquisições. Decidiu-se que os seguintes atributos eram absolutamente indispensáveis para a definição do cliente: nome, e-mail, número de cartão de crédito e validade do cartão de crédito.

#### **Conceitos:**

- **Tipo de Loja:** O tipo de loja é utilizado para classificar as lojas criadas. O tipo de loja é definido por um identificador único, um nome e uma descrição. Um tipo de loja válido é por exemplo farmácia.

- **Produto/Serviço:** As lojas comercializam bens e serviços que se consideram definidos pelo seguinte conjunto de características: nome, descrição, preço unitário, taxa de IVA a aplicar e unidades disponíveis em stock caso se trate de um produto. Note-se que tanto aos produtos como aos serviços é atribuído um identificador único no sistema.
- **Cesto de compras:** O cesto de compras é utilizado para armazenar todos os produtos ou serviços que os utilizadores desejem adquirir numa fase posterior. O cesto de compras está agregado de forma indissolúvel ao cliente. A informação contida num cesto de compras referencia a informação privada do cliente, a loja para a qual o cesto de compras é válido e o conjunto de produtos e/ou serviços a adquirir.

### ***Classes utilizadas para troca de Informação***

Estas são as classes utilizadas na troca de mensagens entre os vários agentes e as várias interfaces gráficas. Estas classes permitem encapsular a informação de uma forma simples e adequada. O facto de se usarem tantas classes e não uma classe geral relaciona-se com questões de desenho do modelo de dados e de eficácia. Garante-se através da utilização destas classes nas várias situações específicas que não se trocam estruturas de informação desnecessariamente grandes.

<i>Classe</i>	<i>atributos</i>	<i>Tipo</i>	<i>Descrição</i>
<b><i>lojistaInfo</i></b>	abreviatura	String	<i>Esta classe é utilizada para troca de informação entre a interface gráfica do agente manager e o agente manager nas operações que envolvem a adição e modificação do lojista.</i>
	nome	String	
<b><i>tipoLojaInfo</i></b>	abreviatura	String	<i>Esta classe é utilizada para troca de informação entre a interface do agente manager, o agente manager, e o agente broker nas operações que envolvem a adição e modificação do tipo de loja .</i>
	nome	String	
<b><i>configInfo</i></b>	pagHTML	String	<i>Esta classe é utilizada para troca de informação entre a interface do agente manager e o agente manager nas operações que envolvem a modificação de parâmetros do sistema.</i>
	pagHTMLVirtual	String	
<b><i>pwdInfo</i></b>	oldPwd	String	<i>Esta classe é utilizada quando um cliente ou um lojista trocam de password. No caso do agente cliente a informação é trocada entre as interfaces gráficas do agente cliente, o agente cliente e o agente broker Para o agente loja é tudo análogo excepto o facto da troca de informação se dar com o agente manager e não com o broker.</i>
	newPwd	String	
<b><i>clientInfo</i></b>	login	String	<i>Esta classe é utilizada para criar um agente cliente novo e é passada pela applet que recebe os dados para o agente broker. Esta classe também é utilizada pelo broker para passar a informação ao agente novo. O agente cliente mantém uma instância desta classe enquanto está activo e é essencial ao seu correcto funcionamento.</i>
	password	String	
	email	String	
	nCartaoCredito	String	
	validade	String	
	clientId	Integer	
	broker	AgentVie w	
	lojaSigla	String	

Tabela 3 – Descrição das classes utilizadas na troca de mensagens para passar informação

<i>Classe</i>	<i>atributos</i>	<i>tipo</i>	<i>Descrição</i>	
<b>shopInfo</b>				
	name	String		
	email	String		
	shopName	String		
	shopSigla	String		
	shopNif	String	<i>Esta classe segue o padrão da classe anterior servindo para criar lojas, para modificar a sua informação de configuração e para passar a informação.</i>	
	shopEmail	String		
	shopType	String		
	shopLogin	String		
	shopPassword	String		
	shopPlaceAlias	String		
	maxNativeAgents	Integer		
	maxVisitAgents	Integer		
	shopPath	String		
	shopAgentPlace	AgentView		
	broker	AgentView		
	shopPlaceView	PlaceView		
	shopURL	URL		
<b>ProdServInfo</b>				
	tipoVenda	String		<i>Esta classe é utilizada para realizar a gestão de produtos e serviços das lojas servindo entre outras coisas para adicionar produtos e serviços a uma determinada loja.</i>
	abreviatura	String		
	descrição	String		
	imagem	String		
	precoUni	Integer		
	taxaIva	Integer		
	unidades	Integer		
	clienteId	Integer		
<b>item</b>				
	abreviatura	String	<i>Esta classe relaciona o agente loja e o agente cliente com um produto ou serviço que o cliente tenciona adquirir ou pelo menos pôr no seu cesto de compras</i>	
	descricao	String		
	tipoVenda	String		
	tipoFacturacao	String		
	shopSigla	String		
	clienteLogin	String		
	clientePassword	String		
	eProduto	boolean		
	clienteAv	AgentView		
	shopAv	AgentView		

subtotal	Integer	
subtotalIVA	Integer	
total	Integer	
precoUnitario	Integer	
taxaIVA	Integer	
clienteId	Integer	
unidades	Integer	
<b>buyInfo</b>		
itens	Vector	
pagamento	String	<i>Esta classe é utilizada para descrever uma transacção. Basicamente descreve o conteúdo do cesto , os dados do cliente e os métodos de pagamento e entrega segundo os quais a transacção dever ser processada. Esta informação é passada do agente broker para o agente loja respectivo.</i>
entrega	String	
clienteNome	String	
clienteLogin	String	
clientePassword	String	
clienteEmail	String	
clienteValidade	String	
nCartaoCredito	Integer	
cliente	clientInfo	

**Tabela 4 – Descrição das classes utilizadas na troca de mensagens para passar informação (Continuação da Tabela anterior)**

## Operações básicas

O modelo de dados foi concebido após se ter considerado quais as operações características num centro comercial e qual é o fluxo de dados que é necessário processar.

Apresentam-se apenas algumas das operações mais significativas.

### 4.2.1.1 Nova Loja

<i>Acção</i>	<i>Agente</i>	<i>Tarefas</i>
<i>Nova Loja</i>	Manager	<i>interface do manager envia shopInfo para manager manager verifica se lojista existe, se não existe cria novo verifica a existência de loja se já existir faz return cria loja e actualiza a tabela loja envia à loja a mensagem init com a shopInfo devolve mensagem de sucesso inicializa a ligação às bases de dados</i>
	Loja	<i>usa o shop info recebido para inicializar variáveis inicializa ligação às bases de dados cria as tabelas da loja cria a página html da loja abre o shops.htm e acrescenta link desta loja envia mensagem de registo da loja ao broker com shopInfo</i>
	Broker	<i>recebe shopInfo (equivale ao registo da loja) insere na tabela loja a informação desta loja</i>

**Tabela 5 – Descrição das interacções entre agentes na criação de uma loja**

Este quadro descreve as várias tarefas que os vários agentes cumprem quando se dá a criação de uma loja. Na troca de mensagens efectuada utiliza-se a classe shopInfo que foi descrita anteriormente.

Basicamente o que acontece é que o manager recebe a informação através de uma das interfaces disponíveis, verifica a não existência da loja e cria um agente novo com a respectiva introdução da informação na sua base de dados. Após a sua criação o agente loja inicializa-se e regista-se no broker.

#### 4.2.1.2 Novo cliente

No quadro seguinte descrevem-se as tarefas executadas para criar um agente cliente. Neste caso, a classe utilizada na troca de informação entre os vários intervenientes foi a classe `clientInfo`. Esta classe tem toda a informação necessária para inicializar um agente `Client`.

<i>Acção</i>	<i>Agente</i>	<i>Tarefas</i>
<i>Novo Cliente</i>	Broker	<i>recebe clientInfo enviado pela applet clientApplet</i> <i>verifica se cliente existe</i> <i>se não existe cria agente e envia-lhe informação de inicialização</i> <i>insere cliente na tabela cliente</i>
	Client	<i>recebe clientInfo do broker</i> <i>mostra interface (classe clientInterface)</i>

Tabela 6 – Descrição das interações entre agentes na criação de um cliente

#### 4.2.1.3 Inserção de um produto ou serviço numa loja

A inserção de um produto ou serviço numa loja pode ser realizada através de duas interfaces distintas, que se apresentaram no capítulo anterior. O produto ou serviço é inserido na base de dados da loja. Para além disso, o produto ou serviço tem de ser registado no broker. Não basta no entanto apenas registar o produto ou serviço, é necessário registar a associação entre o produto/serviço e a loja. O mesmo produto pode ser comercializado por várias lojas.

<i>Acção</i>	<i>Agente</i>	<i>Tarefas</i>
<i>Inserção de um produto ou serviço novo</i>	Shop	<i>recebe ProdServInfo enviado por interface ou applet</i> <i>determina se é produto ou serviço</i> <i>insere informação na tabela adequada</i> <i>envia ao broker a ProdServInfo</i>
	Broker	<i>Recebe ProdServInfo</i> <i>Determina se é produto ou serviço</i> <i>Verifica se produto ou serviço já existe e se não não existe insere nas tabelas de produtos ou serviços</i> <i>Insere associação entre produto/serviço e loja</i>

Tabela 7 – Descrição das interações entre agentes na inserção de um produto ou serviço

#### 4.2.1.4 Inserção de um produto ou serviço num cesto de um cliente

Todo este processo começa na applet da loja, `AppletBuy`. O cliente tem de se registar na loja quando deseja efectuar uma compra através da inserção do seu login e password. Depois

escolhe o produto ou serviço que deseja adquirir, o número de unidades, sendo esta informação passada para o cesto.

<i>Acções</i>	<i>Agente</i>	<i>Tarefas</i>
<i>Inserção de um produto ou serviço no cesto do cliente</i>	Cliente	<i>Recebe item e envia-o ao broker</i>
	Broker	<i>Recebe item e insere a informação na tabela basket</i>

**Tabela 8 - Descrição das interações entre agentes na inserção de um produto ou serviço**

Esta acção aparentemente tão complexa acaba por ser uma das mais simples, pois a informação relacionada com os cestos dos clientes está toda concentrada numa só tabela do broker. Note-se que o agente cliente não mantém qualquer ligação às bases de dados.

#### 4.2.1.5 Aquisição do conteúdo de um cesto

A aquisição do conteúdo do cesto acaba por ser também um processo relativamente fácil de implementar.

<i>Acções</i>	<i>Agente</i>	<i>Tarefas</i>
<i>Cliente faz a aquisição de todos os produtos e serviços dentro do cesto para uma determinada loja</i>	Broker	<i>Recebe buyInfo de uma das interfaces gráficas Passa a buyInfo à loja com a qual vai ser realizada a aquisição</i>
	Shop	<i>Recebe buyInfo Se é cliente novo regista-o na base de dados da loja Insere transacção nas tabelas adequadas Ordena os vários serviços e produtos adquiridos por linha de factura</i>
	Broker	<i>Esvazio o cesto do cliente para aquela loja</i>

**Tabela 9 – Descrição das interações entre agentes na aquisição do conteúdo de um cesto**

Toda a informação relativa aos cestos de cada agente cliente é mantida pelo agente broker. No entanto a informação relativa às transacções efectuadas é armazenada no correspondente agente shop. Quando o utilizador adquire o conteúdo do seu cesto para uma determinada loja o que acontece é que o conteúdo do seu cesto e a sua informação pessoal são enviados para a loja onde esta, posteriormente, faz a inserção da informação nas tabelas correspondentes às aquisições.

### 4.3 Bases de Dados

Neste trabalho utilizaram-se bases de dados relacionais, a linguagem SQL (Structured Query Language) para manipular as bases de dados e a JDBC API (Java Database Connectivity API).

A JDBC API permite uma ligação a qualquer tipo de base de dados. Note-se que a JDBC API pode ser utilizada para aceder a muitos tipos de bases de dados incluindo ODBC data sources, Oracle e Sybase. Tal como já foi descrito anteriormente escolheu-se o JDBC Driver da IDS Software.

Utilizam-se três bases de dados nesta aplicação que são geridas da seguinte forma:

#### Bases de Dados manager

Esta base de dados foi denominada de manager e é exclusivamente gerida pelo agente manager. É constituída por apenas quatro tabelas que contém toda a informação de gestão relativa às lojas, aos tipos de loja, aos lojistas e alguns parâmetros necessários para a gestão do centro comercial em si.

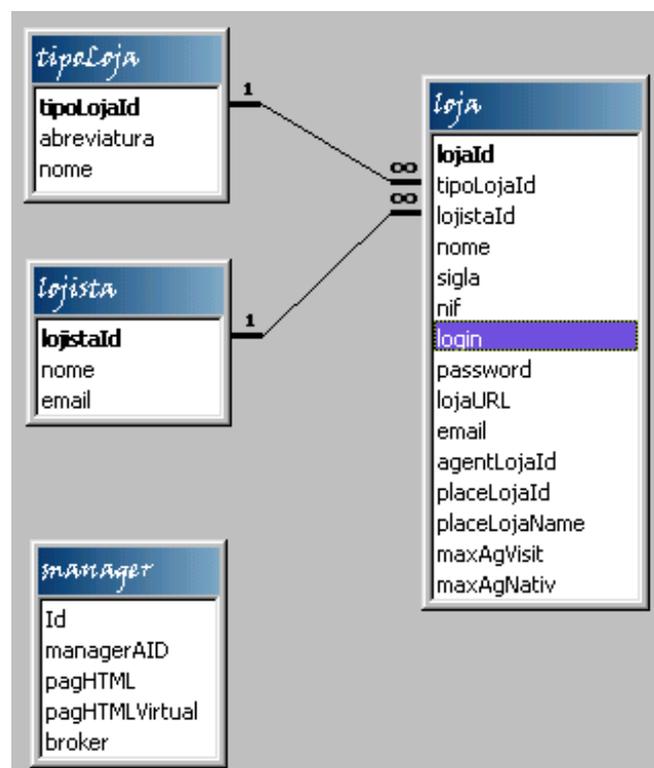


Figura 13 - Diagrama de relações da base de dados do agente gestor do centro comercial

Faz-se seguidamente a descrição pormenorizada das várias tabelas pertencentes a esta base de dados e do contexto em que são utilizadas.

#### 4.3.1.1 Tabela manager

Esta tabela tem como função armazenar informação de configuração importante ao correcto funcionamento do sistema. A informação sobre as directorias virtuais e físicas onde as páginas das lojas devem ser armazenadas é utilizada durante a criação de uma loja nova pelo agente manager. Esta informação é configurável através da interface gráfica do agente .

Toda a informação presente nesta tabela serve para permitir a persistência do agente manager sem afectar o correcto funcionamento da aplicação.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>ManagerAID</i>	Char	Identificador do agente manager
<i>PagHTML</i>	Char	Directoria onde páginas HTML das lojas estão armazenadas
<i>pagHTMLVirtual</i>	Char	Directoria virtual das lojas
<i>Broker</i>	Char	Identificador do agente broker

**Tabela 10 – Descrição da tabela manager**

#### 4.3.1.2 Tabela lojista

Esta tabela contém toda a informação sobre o lojista, informação esta que pode ser directamente manipulada na interface do agente. Tal como já foi referido, cada loja tem um lojista associado podendo pertencer várias lojas ao mesmo lojista. A informação sobre os lojistas é exclusiva do agente manager.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>LojistaId</i>	Int	Identificador único de cada lojista
<i>Nome</i>	Char	Nome do lojista
<i>Email</i>	Char	Email do lojista

**Tabela 11 – Descrição da tabela lojista na base de dados manager**

#### 4.3.1.3 Tabela tipoLoja

Esta tabela tem uma função semelhante à tabela anterior pois permite realizar a gestão do tipo de lojas existentes no centro comercial. Note-se que ao contrário do que se passa com a tabela anterior, também o broker mantém uma tabela semelhante e portanto têm ambas de estar perfeitamente sincronizadas.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>TipoLojaId</i>	Int	Identificador único de cada tipo de loja existente
<i>Abreviatura</i>	Char	Abreviatura do tipo de loja
<i>Nome</i>	Char	Nome do tipo de loja

Tabela 12 – Descrição da tabela tipoLoja na base de dados manager

#### 4.3.1.4 Tabela loja

Esta tabela mantém a informação sobre todas as lojas existentes na aplicação. A gestão das lojas é realizada através da interface gráfica do agente manager.

Os campos login e password ficam exclusivamente na posse do agente manager por questões de privacidade, os outros são passados ao broker.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>LojaId</i>	Int	Identificador único de cada loja
<i>TipoLojaId</i>	Int	Identificador do tipo de loja a que pertence a loja
<i>LojistaId</i>	Int	Identificador do lojista a quem pertence a loja
<i>Nome</i>	Char	Nome da loja
<i>Sigla</i>	Char	Sigla da loja
<i>Nif</i>	Char	Número de identificação bancária
<i>Login</i>	Char	Login para entrar no backoffice da loja
<i>Password</i>	Char	Password para entrar no backoffice da loja
<i>LojaURL</i>	Char	URL da página HTML da loja
<i>Email</i>	Char	Email da loja
<i>AgentLojaId</i>	Char	Identificador único do agente loja
<i>PlaceLojaId</i>	Char	Identificador único do sítio da loja
<i>placeLojaName</i>	Char	Nome do sítio da loja
<i>MaxAgVisit</i>	Int	Nº máximo de agentes visitantes permitidos no sítio da loja
<i>MaxAgNativ</i>	Int	Nº máximo de agentes nativos permitidos no sítio da loja

Tabela 13 – Descrição da tabela loja na base de dados manager

Os campos tipoLojaId e lojistaId referenciam os campos com os mesmos nomes nas tabelas tipoLoja e lojista. Como é visível na Figura 13 trata-se de relações de 1 para muitos e de muitos para um. Isto significa, que um lojista pode ter várias lojas, mas só há um lojista por loja. Verifica-se a mesma relação entre o tipo de loja e as lojas.

## Base de Dados broker

Esta base de dados é utilizada exclusivamente pelo agente broker e serve para este gerir informação sobre os vários clientes existentes no centro comercial e os seus cestos. É o agente broker que gere também os métodos de pagamento e de entrega.

O facto do agente broker permitir executar procuras no sistema por nome de loja, tipo de loja e produtos ou serviços levou à necessidade de haver referências a todas as lojas, tipos de loja, produtos e serviços existentes. Para além disso, é necessária a referência às ligações entre produto/serviço e lojas.

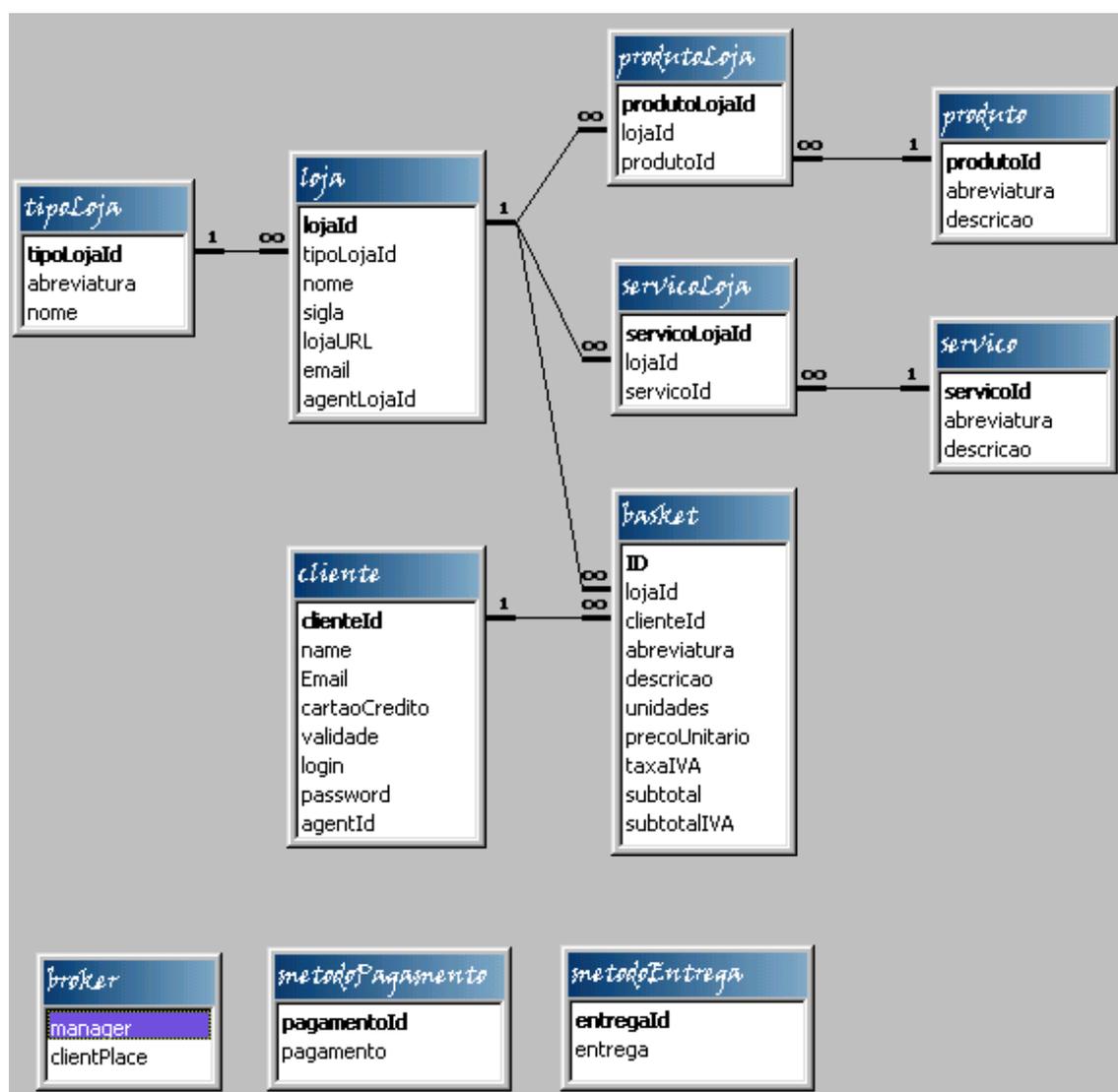


Figura 14 – Diagrama de relações da base de dados do agente broker

#### 4.3.1.5 Tabela tipoLoja

A tabela tipoLoja é idêntica à tabela tipoLoja descrita para a base de dados manager. Esta tabela é utilizada pela applet `AppletSearchShopType` para realizar procuras por tipo de loja em conjunção com a tabela loja. Também a relação entre ambas as tabelas é idêntica.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>tipoLojaId</i>	Int	Identificador único de cada tipo de loja existente
<i>abreviatura</i>	Char	Abreviatura do tipo de loja
<i>Nome</i>	Char	Nome do tipo de loja

Tabela 14 – Descrição da tabela tipoLoja na base de dados broker

#### 4.3.1.6 Tabela metodoPagamento

Esta tabela enumera todos os métodos de pagamento suportados pelo sistema. O método de pagamento é escolhido pelos utilizadores na altura da aquisição dos produtos e/ou serviços que se encontram no cesto.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>pagamentoId</i>	Int	Identificador único de cada método de pagamento existente
<i>Pagamento</i>	Char	Nome do método de pagamento

Tabela 15 – Descrição da tabela metodoPagamento na base de dados broker

#### 4.3.1.7 Tabela metodoEntrega

Esta tabela enumera todos os métodos de entrega suportados pelo sistema. Tal como no caso da tabela anterior, esta informação é utilizada durante o processo de aquisição dos produtos ou serviços que se encontram no cesto do cliente.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>entregaId</i>	Int	Identificador único de cada método de entrega existente
<i>entrega</i>	Char	Nome do método de entrega

Tabela 16 – Descrição da tabela metodoEntrega na base de dados broker

#### 4.3.1.8 Tabela broker

A tabela broker mantém os identificadores únicos do agente manager e do local de execução dos agentes Client. Esta tabela, tal como a tabela manager para o agente manager, existe apenas para permitir a persistência dos vários agentes sem afectar a sua performance na aplicação (vide capítulo 5).

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>Manager</i>	Char	Identificador do agente manager
<i>ClientPlace</i>	Char	Identificador do sítio onde os agentes clientes se encontram

Tabela 17 – Descrição da tabela broker

#### 4.3.1.9 Tabela loja

Esta tabela equivale a uma enumeração de todas as lojas existentes no centro comercial e contém, entre outros campos, o identificador único que descreve a loja no centro comercial e o identificador único do agente Shop correspondente.

A URL da loja é necessária para o broker poder redireccionar os clientes para as páginas das lojas.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>LojaId</i>	Int	Identificador único de cada loja
<i>tipoLojaId</i>	Int	Identificador do tipo de loja a que pertence a loja
<i>Nome</i>	Char	Nome da loja
<i>Sigla</i>	Char	Nome da sigla da loja
<i>LojaURL</i>	Char	URL da loja
<i>Email</i>	Char	Email da loja
<i>agentLojaId</i>	Char	Identificador do agente da loja

Tabela 18 – Descrição da tabela loja na base de dados broker

#### 4.3.1.10 Tabela cliente

A tabela cliente enumera todos os clientes existentes no centro comercial. Os agentes cliente não mantêm uma ligação a nenhuma base de dados sendo a gestão da sua informação realizada pelo agente broker. A informação presente nesta tabela corresponde à definição da entidade cliente previamente apresentada neste capítulo.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>ClienteId</i>	Int	Identificador único de cada cliente
<i>Name</i>	Char	Nome do cliente
<i>Email</i>	Char	Email do cliente
<i>CartaoCredito</i>	Char	Nº Cartão Crédito
<i>Validade</i>	Char	Validade do Cartão de Crédito
<i>Login</i>	Char	Login do cliente
<i>password</i>	Char	Password do cliente
<i>AgentId</i>	Char	Identificador único do agente do cliente

Tabela 19 - Descrição da tabela cliente na base de dados broker

#### 4.3.1.11 Tabela basket

É nesta tabela que o agente broker mantém o conteúdo dos cestos de todos os clientes do centro comercial.

Cada linha corresponde à aquisição de um produto/serviço por parte de um cliente a uma determinada loja. Cada uma destas linhas corresponde a uma linha possível de uma factura caso o cliente decida realizar a aquisição dos produtos/serviços que estão no cesto.

Caso o cliente decida adquirir os produtos/serviços que estão no seu cesto relacionados com uma determinada loja, todas as linhas da tabela com o identificador de loja e cliente correspondente são processadas de modo a realizar a aquisição.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>ID</i>	Int	Identificador único de cada associação de artigos de uma loja com um cliente
<i>LojaId</i>	Int	Identificador único da loja
<i>clienteId</i>	Int	Identificador único do cliente
<i>abreviatura</i>	Char	Abreviatura ou sigla do artigo
<i>descricao</i>	Char	Descrição do artigo
<i>unidades</i>	Char	Nº de unidades a adquirir
<i>precoUnitario</i>	Char	Preço unitário do produto
<i>TaxaIVA</i>	Char	Taxa de IVA sobre o produto
<i>Subtotal</i>	Char	Subtotal calculado com base no nº itens a adquirir e o preço unitário destes
<i>subtotalIVA</i>	Char	Valor de IVA calculado com base no nº itens a adquirir, o preço unitário destes e a taxa IVA a aplicar

Tabela 20 – Descrição da tabela basket na base de dados broker

#### 4.3.1.12 Tabela produto

Esta tabela contém uma referência para todos os produtos comercializados no centro comercial.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>produtoId</i>	Int	Identificador do produto
<i>abreviatura</i>	Char	Abreviatura do produto
<i>descricao</i>	Char	Descrição do produto

Tabela 21 – Descrição da tabela produto na base de dados broker

#### 4.3.1.13 Tabela servico

Esta tabela contém uma referência para todos os serviços comercializados no centro comercial.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>servicoId</i>	Int	Identificador do serviço
<i>abreviatura</i>	Char	Abreviatura do serviço
<i>descricao</i>	Char	Descrição do serviço

Tabela 22 – Descrição da tabela servico na base de dados broker

#### 4.3.1.14 Tabela servicoLoja

Esta tabela associa os serviços existentes no centro comercial às lojas que os comercializam.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>servicoLojaId</i>	Int	Identificador único deste serviço nesta loja
<i>LojaId</i>	Char	Identificador da loja
<i>servicoId</i>	Char	Identificador do serviço

Tabela 23 – Descrição da tabela servicoLoja na base de dados broker

#### 4.3.1.15 Tabela produtoLoja

Esta tabela associa os produtos existentes no centro comercial às lojas que os comercializam.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>produtoLojaId</i>	Int	Identificador único deste produto nesta loja
<i>LojaId</i>	Char	Identificador da loja
<i>produtoId</i>	Char	Identificador do produto

Tabela 24 – Descrição da tabela produtoLoja na base de dados broker

No caso de uma loja ser eliminada do centro comercial as relações existente entre essa loja e os serviços/produtos que comercializa são apagadas das tabelas produtoLoja e servicoLoja, mas os produtos e serviços continuam registados nas tabelas produto e servico do broker.

### **Base de Dados shop**

Cada agente shop mantém uma ligação à base de dados shop. Existe apenas uma base de dados para todas as lojas e a tabela metaLoja é a única tabela inicial cuja existência é

“estática”. Todas as outras tabelas existentes nesta base de dados são criadas de forma dinâmica durante a inicialização dos agentes shop e apagadas com a eliminação das lojas.

#### 4.3.1.16 Tabela metaLoja

A tabela metaLoja é na realidade uma forma de “mapear” as várias tabelas relacionadas com o agente loja. Esta tabela contém ainda a informação característica à entidade loja.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>LojaId</i>	Int	O identificador único e universal da loja
<i>nome</i>	Char	O nome da loja
<i>sigla</i>	Char	A sigla da loja
<i>nif</i>	Char	O valor do seu número de identificação financeira
<i>lojaURL</i>	Char	A URL da loja
<i>email</i>	Char	O email da loja
<i>produto</i>	Char	Nome da tabela de produtos da loja nesta BD
<i>servico</i>	Char	Nome da tabela de servicos da loja na BD
<i>agentId</i>	Char	Identificador único do agente loja a
<i>cliente</i>	Char	Nome da tabela de clientes da loja na BD
<i>Transacciao</i>	Char	Nome da tabela de transacções da loja na BD
<i>TransacciaoProduto</i>	Char	Nome da tabela que regista todas as linhas de todas as facturas relacionadas com produtos da loja
<i>TransacciaoServico</i>	Char	Nome da tabela que regista todas as linhas de todas as facturas relacionadas com serviços da loja

**Tabela 25 – Descrição da tabela metaLoja na base de dados shop**

As restantes tabelas da base de dados loja são tabelas criadas de forma dinâmica por cada agente Shop. O nome destas tabelas é construído em runtime através da concatenação do nome geral das tabelas com a sigla da loja.

#### 4.3.1.17 Tabela Cliente[SIGLAloja]

Esta tabela serve para registar na loja todos os seus clientes. Na realidade todos os clientes do centro comercial já estão registados no agente broker, mas considerou-se que deveria existir uma lista individual de clientes para cada loja.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>ClienteId</i>	Int	<i>Identificador do cliente desta loja</i>
<i>Nome</i>	Char	<i>Nome do cliente</i>
<i>Email</i>	Char	<i>Email do cliente</i>
<i>Login</i>	Char	<i>Login do cliente</i>
<i>Password</i>	Char	<i>Password do cliente</i>
<i>NcartaoCredito</i>	Char	<i>Nº do CartaoCredito do cliente</i>
<i>Validade</i>	Char	<i>Validade do cliente</i>

Tabela 26 – Descrição da tabela cliente[SIGLAloja] na base de dados shop

#### 4.3.1.18 Tabela Produto[SIGLAloja]

Esta tabela serve para apoiar e permitir a gestão dos produtos comercializados pela loja, tendo sido concebida para salvaguardar toda a informação e atributos característicos dos produtos.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>ProdutoId</i>	Int	<i>Identificador do produto desta loja</i>
<i>Abreviatura</i>	Char	<i>Abreviatura do produto</i>
<i>Descricao</i>	Char	<i>Descrição do produto</i>
<i>PrecoUnitario</i>	Int	<i>Preço unitário do produto</i>
<i>Iva</i>	Int	<i>Taxa de IVA aplicável ao Produto</i>
<i>Unidades</i>	Int	<i>Nº de unidades do produto em stock</i>

Tabela 27 - Descrição da tabela produto[SIGLAloja] na base de dados shop

#### 4.3.1.19 Tabela servico[SIGLAloja]

Esta tabela serve para apoiar e permitir a gestão dos serviços comercializados pela loja, tendo sido concebida para salvaguardar toda a informação e atributos característicos dos serviços.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>ServicoId</i>	Int	<i>Identificador do serviço desta loja</i>
<i>Abreviatura</i>	Char	<i>Abreviatura do serviço</i>
<i>Descricao</i>	Char	<i>Descrição do serviço</i>
<i>PrecoUnitario</i>	Int	<i>Preço unitário do serviço</i>
<i>Iva</i>	Int	<i>Taxa de IVA aplicável ao serviço</i>

Tabela 28 - Descrição da tabela servico[SIGLAloja] na base de dados shop

#### 4.3.1.20 Tabela Transaccão[SIGLAloja]

Nesta tabela é armazenada toda a informação relacionada com as aquisições realizadas na loja. Nesta tabela relacionam-se os clientes com os métodos de pagamento, envio e data da transacção .

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>TransaccãoId</i>	Int	<i>Identificador único da transacção realizada Equivale ao nº da factura</i>
<i>Data</i>	Char	<i>Data em que foi realizada a transacção</i>
<i>ClienteId</i>	Int	<i>Identificador do cliente desta loja</i>
<i>totalLiquido</i>	Int	<i>Total liquido a facturar</i>
<i>TotalIVA</i>	Int	<i>Total IVA a facturar</i>
<i>FormaPAG</i>	Char	<i>Forma de pagamento</i>
<i>FormaENT</i>	Char	<i>Forma de entrega</i>

Tabela 29 - Descrição da tabela transaccão[SIGLAloja] na base de dados shop

As duas tabelas que se seguem servem para especificar os vários produtos e/ou serviços transaccionados. A transaccãoId identifica a transacção a que se refere a linha da tabela e o transaccãoServicoId ou transaccãoProdutoId identificam a linha dentro de cada transacção. O conceito de linha torna-se mais claro se se considerar que cada transacção é descrita por uma factura.

#### 4.3.1.21 Tabela transaccãoServico[SIGLAloja]

Esta tabela especifica as linhas das transacções correspondentes à comercialização de serviços.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>transaccãoServicoId</i>	Int	<i>Identificador único</i>
<i>TransaccãoId</i>	Int	<i>Identificador único da transacção realizada Linha da factura equivale à ordem dos artigos comercializados</i>
<i>LinhaFactura</i>	Int	
<i>ServicoId</i>	Int	<i>Identificador único do serviço</i>
<i>Subtotal</i>	Int	<i>Subtotal equivalente à comercialização deste serviço</i>
<i>SubtotalIVA</i>	Int	<i>Subtotal em IVA equivalente à comercialização deste serviço</i>

Tabela 30 - Descrição da tabela transaccãoServico[SIGLAloja] na base de dados shop

#### 4.3.1.22 Tabela transacaoProduto[SIGLAloja]

Esta tabela específica as linhas das transacções correspondentes à comercialização de produtos.

<i>Campo</i>	<i>Tipo</i>	<i>Significado</i>
<i>transacaoProdutoId</i>	Int	<i>Identificador único</i>
<i>TransacaoId</i>	Int	<i>Identificador único da transacção realizada</i>
<i>LinhaFactura</i>	Int	<i>Linha da factura equivale à ordem dos artigos comercializados</i>
<i>ProdutoId</i>	Int	<i>Identificador único do produto</i>
<i>Unidades</i>	Int	<i>Nº unidades transaccionadas</i>
<i>Subtotal</i>	Int	<i>Subtotal equivalente à comercialização deste serviço</i>
<i>SubtotalIVA</i>	Int	<i>Subtotal em IVA equivalente à comercialização deste serviço</i>

Tabela 31 - Descrição da tabela transacaoProduto[SIGLAloja] na base de dados shop

## 4.4 Interacção entre agentes

As interacções entre agentes ficaram definidas a partir do momento em que se definiram os primeiros conceitos e entidades intervenientes no sistema. Cada funcionalidade ou acção extra a implementar no sistema contribuiu para a definição das interacções entre os agentes.

O modelo da Figura 9 sintetiza as interacções entre agentes mostrando que o agente broker interage com todos os agentes de forma centralizada não havendo interacções directas entre agentes shop, client e manager. Existe no entanto uma excepção a este modelo centralizado que são as interacções entre manager e agentes shop (vide explicação mais à frente).

De uma forma mais generalizada pode-se fazer as seguintes afirmações sobre as interacções entre agente:

As interacções entre manager e broker são maioritariamente de carácter configurativo o que significa que envolvem a troca de informação sobre parâmetros de configuração, etc. ...

As interacções excepcionais entre manager e loja existem só no momento de criação do agente shop por parte do manager e no momento em que agente shop interage com o agente manager para mudar de password.

As interações entre broker e cliente e broker e loja vão no sentido de facilitar e mesmo possibilitar possíveis transações entre ambos os agentes.

Todas as interações entre agentes são implementadas através da troca de mensagens síncronas ou assíncronas.

A troca de mensagens síncronas implica que o agente que envia a mensagem fica à espera de uma resposta por parte do agente que a recebe. Este tipo de troca de mensagens síncronas é utilizado para trocar dados e informação entre agentes.

Um exemplo de uma mensagem síncrona utilizada na aplicação entre o agente broker e o agente loja é a mensagem “clear basket” enviada ao broker. Esta mensagem contém uma instância da classe `clientInfo` que o broker utiliza para limpar o cesto do cliente. Após execução da limpeza do cesto devolve uma `String` com informação sobre o status da operação efectuada.

Os métodos utilizados para enviar e receber mensagens síncronas são os métodos *doOperation* e *handleOperation*.

A troca de mensagens assíncronas é utilizada apenas para situações em que um agente necessita de informar outro da ocorrência de um evento, etc. ...

Um exemplo de uma mensagem assíncrona é a mensagem de “save all” enviada pelo manager ao broker. Esta mensagem indica ao broker que deve salvar o seu contexto de execução e que para além disso deve passar esta mensagem a todos os agentes cliente e lojas da aplicação.

Os métodos utilizados para enviar e receber mensagens assíncronas são os métodos *sendMessage* e *handleMessage*.

## 4.5 Interação entre agentes e applets

A maioria das applets desenvolvidas interagem directamente com o agente broker. A applet de interface do manager estabelece inicialmente a ligação com o agente manager através do agente broker solicitando-lhe que envie o AID do agente manager.

Note-se que a applet que permite realizar a gestão do e-mall não necessita de login e password porque se partiu do princípio que a página de back-office não seria de acesso público.

Toda a comunicação entre a applet AppletManager e o agente manager se processa de forma quase idêntica à entre o agente manager e a sua interface gráfica.

A applet de cada loja estabelece a ligação com o seu agente correspondente através do seu AID. Desta forma consegue-se eliminar a necessidade de inserir qualquer tipo de informação extra. A applet estabelece a ligação através do AID do agente.

A applet AppletNovaLoja interage apenas com o agente broker. Ambas as applets clientApplet e shopApplet utilizam o agente broker para acederem ao agente cliente ou loja alvo após a inserção de um login e password adequados.

O agente broker foi utilizado como o agente de “acesso” para todas as applets da aplicação com excepção das AppletBuy.

Apresenta-se aqui um exemplo dos métodos de init e destroy da AppletNovaLoja.

O método init aqui apresentado inicia a ligação ao Voyager e posteriormente obtém uma referência para o agente broker.

```
public void init() {
    try {
        jbInit();
        Voyager.setExtendedStackTrace(true);
        Voyager.startup(this);

        asAddress=Voyager.getServerAddress();
        cv=AgentSpace.getContextView(asAddress,"broker","brokerpwd");
        for(Enumeration elements=cv.myAgents();elements.hasMoreElements();){
            broker=(AgentView)elements.nextElement();
        }
        ...
    }
}
```

É essencial implementar este método pois fecha as ligações ao Voyager disponibilizando os recursos para outras applets.

```
public void destroy() {
    try {
        Voyager.shutdown();
    }
    catch(Exception e) {
        e.printStackTrace();
    }
}
```

## 5 Avaliação do Trabalho

Este trabalho foi desenvolvido com base numa arquitectura lógica de três níveis. A especificação desta arquitectura demonstrou ser a mais correcta para o desenvolvimento de aplicações Web, como era o caso da aplicação deste projecto.

A integração dos três níveis foi realizada de uma forma quase perfeita. A utilização de documentos HTML com applets embebidas usados para gerar a GUI da aplicação Cliente, no nível da Apresentação, foi bastante eficiente pela sua fácil compreensão e pelo facto de ser possível interagir com os vários agentes do sistema. O acesso ao Sistema de Bases de Dados, que foi implementado utilizando JDBC, não trouxe dificuldades acrescidas, tendo-se verificado uma boa performance na execução das *queries* SQL na Base de Dados.

Uma das dificuldades encontradas relaciona-se com a integração entre o back-office das lojas e o seu storefront, ou seja, a introdução de um produto novo na loja não significa que esse produto passe a estar visível nas páginas da loja. Na realidade neste momento ainda tem de haver uma edição manual da interface gráfica da loja.

Verificou-se uma grande escalabilidade e eficiência da aplicação que se deve ao ambiente de desenvolvimento distribuído proporcionado pelo AgentSpace.

Visto ser uma aplicação *Web*, a necessidade de ser um sistema partilhado permitindo o acesso e execução a múltiplos utilizadores ao mesmo tempo garantindo a integrabilidade e fiabilidade do sistema é proporcionado automaticamente pelo AgentSpace.

O facto de a aplicação ser executada num *Browser* é uma mais valia face às aplicações Cliente tradicionais. Não é necessário qualquer instalação da aplicação, estando sempre disponível e actualizada, e suporta múltiplos tipos de clientes. Apesar de utilizar como mecanismo de comunicação o protocolo HTTP a sua performance consegue manter-se a níveis bastantes razoáveis.

O modelo de negócio criado pelo conjunto de agentes que implementam a Lógica do Negócio foi dividido em quatro, que corresponde aos quatro tipo de agentes existentes. A extensão da funcionalidade do sistema sem prejudicar a estrutura já existente, é garantida devido em grande parte à sua divisão modular, mas também à abstracção da complexidade e encapsulamento funcional realizado pelos componentes.

## **5.1 Trabalho Futuro**

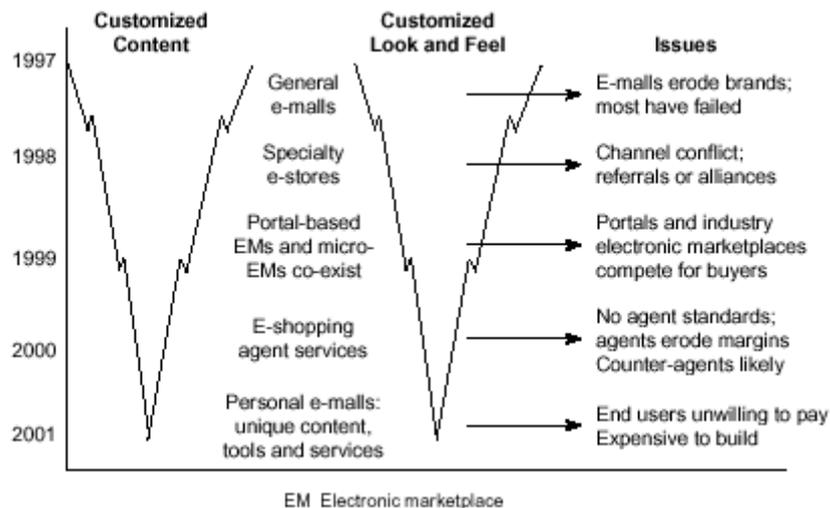
O trabalho desenvolvido afigura-se como uma boa aproximação ao modelo de centro comercial tradicional. A aproximação baseada em agentes adaptou-se perfeitamente ao modelo e permitiu fazer um desenvolvimento rápido e modular. No entanto existem ainda muitas funcionalidades possíveis de adicionar a todo o sistema. O modelo CBB já previamente definido, vem revelar quais as funcionalidades e tarefas extra que os vários agentes desta aplicação poderiam suportar.

A monitorização da ocorrência de um certo evento, a comparação de produtos ou vendedores são apenas algumas das funcionalidades extra que se podem implementar num trabalho futuro para enriquecer a aplicação.

## 6 Conclusões

Os centros comerciais electrónicos (e-malls) não conseguiram gerar a receita esperada nem para os seus gestores, nem para os lojistas que os integravam. Falharam porque apesar de apresentarem uma larga gama de produtos não ofereciam aos consumidores qualquer tipo de serviço apto a resolver problemas específicos ou comparação de preços, etc ....

No entanto prevê-se que este segmento evolua até 2001 no sentido destes se transformarem em micromercados regionais e especializados.



**Figura 15 – Evolução do modelo de negócio E-Mall**

Um micromercado é um tipo de mercado que ajuda compradores dentro de um segmento de indústria ou grupo de interesses a tomar decisões referentes à aquisição de produtos através da agregação de conteúdos, fornecimento de serviços de valor acrescentado e oferta de alternativas de vários vendedores.

Existem actualmente milhares de industry-focused Web sites patrocinados por uma associação de comércio. A maioria dos sites apresentam referências aos membros destas associações e alguns catálogos com capacidade de processamento de transacções. Os micromercados emergentes irão explorar as falhas de informação presentes em nichos de mercados.

Os micromercados diferenciam-se dos sites de comércio electrónico dos produtores e distribuidores porque a maioria destes apresentam apenas produtos de um produtor. Os distribuidores baseados em catálogos regionais estão especialmente em risco porque os micromercados conseguem construir uma massa crítica de produtos de produtores numa escala global. Os micromercados vão ser em muitos casos neutrais em relação aos produtores

e representantes de todos os produtos e serviços existentes na sua categoria. Na realidade a sua lealdade vai para com o consumidor e não para com o vendedor. Os micromercados providenciarão possivelmente editoriais imparciais e comparações de produtos ilustrativas. Esta neutralidade irá proporcionar credibilidade aos micromercados e conquistar a confiança do consumidor. Esta credibilidade no entanto tem um preço. Será muito difícil conseguir subsídios ou apoios das indústrias produtoras que lideram o e-business actualmente. A maioria dos micromercados diferenciar-se-ão através das seguintes características:

- Personalização
- Especialização de conteúdo
- Funções adequadas às tarefas específicas de indústria dos seus utilizadores chave

Note-se que estes três factores se revelam adequados à utilização dos agentes e que traduzem a evolução do mercado num futuro muito próximo. A evolução da tecnologia torna os clientes cada vez mais exigentes e impacientes. Os agentes permitem a realização automática de comparações de produtos e comparações de vendedores, monitorizam o sistema e comunicam a ocorrência de certos eventos aos seus utilizadores.

Estes três factores permitirão aos micromercados atingir uma média de visitas e receitas acima do normal. Uma vez conquistada a confiança dos consumidores em relação ao preço e à qualidade dos produtos será difícil para as outras entidades competirem com estes minimercados. Alguns podem mesmo vir a ser gatekeepers para grupos de clientes específicos.

A junção ou parceria com os micromercados irá ser uma opção viável para muitas empresas de e-business . Os micromercados serão uma boa forma de escoar produtos excedentes ou realizar leilões anonimamente. Para os consumidores, os mercados agregam opções, providenciam escolha e consomem transacções. No entanto, para atrair consumidores, os micromercados necessitam de assistência das empresas. A maioria dos micromercados não possuirão pelo menos inicialmente as mesmas competências de uma empresa estabelecida e necessitam de apresentar produtos de marcas com sucesso. Os compradores online tendem a confiar pelo menos inicialmente em marcas conhecidas e a orientar a sua decisão de compra nesse sentido. Para além disso os micromercados irão ser lentos na aquisição de expertise funcional nas áreas de logística e fullfilment. Por esta razão é que os micromercados irão necessitar de empresas estabelecidas e como incentivo deverão proporcionar a estas acesso a mercados novos.

## Apêndice A – Modelos de Negócio

Descrição breve dos vários modelos de negócio actualmente utilizados no comércio electrónico que estão esquematizados na tabela 3.

<i>Modelo de Negócios</i>	<i>Variantes do Modelo de Negócios</i>
<b><i>Brokerage</i></b>	Buy/Sell Fullfilment Market Exchange Business Trading Community Buyer Aggregator Distributor Virtual Mall Metamediary Auction Broker ReverseAuction Classifieds Search Agent
<b><i>Merchant</i></b>	Virtual Merchant Catalog Merchant Surf-and-Turf Bit Vendor
<b><i>Community</i></b>	Voluntary Contributor Model Knowledge Networks
<b><i>Advertising</i></b>	Generalized Portal Personalized Portal Specialized Portal Attention/Incentive Marketing Free Model Bargain Discounter
<b><i>Infomediary</i></b>	Recommender System Registration Model
<b><i>Manufacturer</i></b>	
<b><i>Subscription</i></b>	
<b><i>Utility</i></b>	

Tabela 32 - Esquema de novos modelos de negócio aplicáveis ao comércio electrónico

## **Brokerage**

Os Brokers juntam compradores e vendedores e facilitam as transacções entre estes. Estas transacções podem ser business-to-business, business-to-consumer ou consumer-to-consumer. Um broker faz dinheiro cobrando taxas por cada transacção realizada. Os modelos de brokerage podem tomar as seguintes formas:

### **Buy/Sell Fullfilment**

A esta categoria pertencem os agentes de viagens e os sites de online financial brokerage, onde os utilizadores efectuam as suas ordens de compra e venda de instrumentos financeiros. O broker cobra ao consumidor e/ou vendedor uma taxa por cada transacção efectuada.

### **Market Exchange**

Um modelo de mercados B2B cada vez mais comum. Bons exemplos são a MetalSite ou ChemConnect's World Chemical Exchange. Neste modelo, o broker cobra ao vendedor uma taxa por transacção baseada no valor transaccionado. Os mecanismos de atribuição de preços podem ser tão simples como as aproximações offer/buy, offer/negotiated buy ou auction offer/bid.

### **Business Trading Community**

Ou comunidade vertical é um conceito implementado pela VerticalNet. É um site que apresenta toda a informação essencial e compreensiva a um certo segmento vertical do mercado.

### **Buyer Aggregator**

Este modelo foi inicialmente implementado pela Accompany, e descreve a agregação de compradores como o acto de juntar compradores individuais de modo a que estes possam beneficiar de preços normalmente apenas praticados com grupos. Os vendedores pagam uma pequena percentagem da venda por transacção.

### **Distributor**

Um distribuidor é um operador com catálogo que junta um largo número de produtores e compradores de retalho. Os brokers facilitam as transacções entre distribuidores franchised e os seus parceiros de negócio. O DigitalMarket descreve os benefícios deste modelo em duas

vertentes. Para os compradores, permite um “faster time to market” e permite reduzir custos. O facto de providenciar aos compradores formas de retrieving quotes dos distribuidores preferidos torna as transacções mais eficientes. Para os distribuidores diminui o custo das vendas pois executa atribuição de preços, processamento de encomendas, acompanhamento da evolução das encomendas, etc. ...

### **Virtual Mall**

Um site que alberga muitos mercadores online. O modelo cobra a instalação das lojas e normalmente por transacção. Exemplo deste modelo são as lojas no Yahoo. Este modelo de shopping virtual torna-se mais eficiente quando combinado com um portal geral. Os malls mais sofisticados oferecem serviços transaccionais automatizados e oportunidades de marketing relationship.

### **Metamediary**

É um centro comercial virtual que processa as transacções, track orders, e providencia serviços de billing e collection. O metamediary protege os vendedores porque garante a satisfação dos consumidores através da prestação de serviços de valor acrescentado.

### **Auction Broker**

Site que conduz leilões. O Broker cobra uma taxa aos vendedores, cujo valor depende do valor da transacção. Os vendedores aceitam a licitação máxima acima do mínimo. Os leilões podem variar de acordo com as regras de oferta e de licitação.

### **Reverse Auction**

Este modelo pioneiro foi patenteado pela Pricekine. Os compradores efectuem uma licitação final e com compromisso para um determinado bem ou serviço. O broker realiza o concretização do negócio e a taxa cobrada resulta do spread entre a licitação e o fullfilment price e talvez uma taxa de processamento.

### **Classifieds**

Uma lista de itens para venda ou procurados para compra, tipicamente gerida por um local news provider. O preços destes itens pode ser especificado ou não. As taxas por utilização do serviço são cobradas independentemente da realização ou não das transacções.

**Search Agent**

Um agente (i.e. um agente inteligente de software ou “robot”) utilizados para procurar o melhor preço para um dado bem ou serviço previamente definido pelo comprador. Também são utilizados para procurar informação sobre um determinado assunto. Uma agência de emprego pode actuar como um agente de procura encontrando emprego para desempregados ou encontrando pessoas para preencher as listas fornecidas pelos empregadores. [Career Central]

**Advertising**

Este modelo é uma extensão do modelo de media broadcasting tradicional. O broadcaster, neste caso um Web Site, providencia conteúdo (pode ser gratuito ou não) e serviços como email, chats, forums, etc. ... As mensagens de anúncio são apresentadas sob a forma de banner ads e podem ser ou não a única fonte de rendimento do site. O broadcaster pode ser um criador de conteúdo ou apenas um distribuidor de conteúdo gerado noutra sítio. Este modelo apenas funciona quando o volume do tráfego é grande ou altamente especializado.

**Portal Geral**

Caracteriza-se pelo seu tráfego intenso, tipicamente na ordem das dezenas de milhões de visitas por mês. Motores de pesquisa e directorias como o Excite, AltaVista e Yahoo são exemplos conhecidos deste modelo. O volume elevado de tráfego torna os anúncios rentáveis e permite a oferta de conteúdo livre e serviços como o e-mail, portfolio de acções, chats, news, etc. ...

**Portal Personalizado**

A natureza genérica de um portal geral não estimula a lealdade dos consumidores. Isto levou à criação de portais que permitem a customização da interface e do conteúdo. Isto aumenta a lealdade do consumidor devido ao tempo que este investiu na personalização do portal. A rentabilidade deste portal é baseada no volume e possivelmente no valor da informação derivada das escolhas do utilizador. A personalização pode suportar o modelo de portal especializado.

**Portal Especializado**

Também conhecido como Vortal. Neste caso o volume é menos importante do que uma base de utilizadores bem definida. Por exemplo, um site que apenas atraia golfistas ou aquaríofistas é com certeza uma fonte de receitas para um anunciante disposto a pagar o necessário para atingir aquela audiência. Prevê-se a proliferação deste tipo de sites num futuro próximo.

**Attention/ Incentive Marketing**

Neste modelo “pay for attention” paga-se aos clientes para estes visionarem conteúdo e completarem formulários. Esta aproximação é utilizada por aquelas companhias que tem produtos com mensagens muito complexas.

**Free Model**

Oferece algo aos utilizadores: site hosting, web services, Internet acess, free hardware, electronic greeting cards. Este modelo gera um grande volume de visitas proporcionando óptimas oportunidades para anunciar produtos e serviços. A viabilidade económica deste modelo é excelente quando este está associado ao metamediary model.

**Bargain Discounter**

Este site vende os produtos abaixo ou a preço de custo atraindo muitos consumidores. Este site realiza receitas a partir dos anúncios expostos.

**Infomediary Model**

A informação acerca dos consumidores e os seus hábitos de consumo é extremamente valiosa. Especialmente se essa informação é analisada de uma forma cuidadosa e utilizada para realizar campanhas de marketing. Algumas empresas funcionam como intermediárias na colecção e venda de informação a outros negócios. Um infomediário pode oferecer aos utilizadores acesso grátis à Internet ou hardware grátis em troca de informação detalhada acerca dos hábitos de navegação e aquisições na Internet. Este modelo tem mais probabilidades de ter sucesso do que o modelo de advertising.

O infomediary model também pode disponibilizar informação ao cliente dos sites existentes num determinado segmento de mercado.

### ***Recommender Systems***

É um site onde utilizadores trocam informação entre si sobre a qualidade dos produtos e serviços ou dos vendedores a quem adquiriram os mesmos. Existe a possibilidade de levar este conceito ainda mais longe através da integração do recommender system com o browser. Este agente monitoriza os hábitos do utilizador conseguindo aumentar a relevância e importância das suas sugestões. Os Recommender Systems conseguem tirar vantagens dos modelos afiliados oferecidos pelos vendedores para aumentarem as receitas vindas das vendas da informação sobre os consumidores.

### ***Registration Model***

Normalmente são sites de conteúdo livre para consulta em que os utilizadores se tem de registar. O registos permite o tracking inter-session de cada cliente sendo este tipo de informação extremamente útil para as campanhas de marketing especializadas. Esta é a forma mais básica do infomediary model.

### ***Merchant Model***

São os retailers de bens e serviços que costumam integrar este modelo. As vendas podem ser realizadas com base numa lista de preços ou através de um leilão. Em alguns casos os bens e serviços podem ser exclusivos à Web, ou seja, não existe nenhuma loja física.

### ***Virtual Merchant***

Um negócio que funciona exclusivamente na Web e que oferece bens ou serviços tradicionais ou específicos da Web. O método de venda utilizado pode ser uma lista de preços ou o leilão.

### ***Catalog Merchant***

Corresponde na realidade a uma migração de um negócio baseado no correio electrónico para um negócio baseado na Web.

### ***Bit Vendor***

Um vendedor que lida estritamente com produtos digitais e serviços realizando as vendas e distribuição na web.

### **Manufacturer Model**

Este modelo permite aos produtores interagir directamente com os compradores eliminando quaisquer intermediários. Este modelo de produtor pode ser baseado na eficiência (redução de custos que pode ou não ser visível ao consumidor), serviço ao consumidor melhorado e uma melhor compreensão das preferências do consumidor. Os produtos perecíveis que beneficiam de uma distribuição rápida como as flores frescas, serão largamente compensados pela eliminação de intermediários.

### **Affiliate Model**

Em contraste com o modelo do portal geral que procura gerar grande volume de tráfego para um site, o modelo afiliado providencia oportunidades de venda pois os sites interessados montam uma rede de sites afiliados com ligação ao site originário. Este é um modelo pay-for-performance que só paga aos sites afiliados que geram negócio. Existem variações a este modelo que incluem pay-per-click, banner exchange e programas de partilha de receitas.

### **Community Model**

A viabilidade de um modelo comunitário é baseado na lealdade de um utilizador (oposto ao tráfego elevado). Os utilizadores investem tanto em tempo como em emoção no site. Em alguns casos, os utilizadores são contribuintes regulares em conteúdo e/ou em dinheiro. Este modelo pode também exigir aos seus utilizadores uma taxa de subscrição para os serviços que proporciona.

### **Voluntary Contributor Model**

Às vezes conhecido por modelo NPR (NOT FOR PROFIT). Este modelo está sujeito à existência de uma comunidade de utilizadores que suportem o site através de doações voluntárias. As organizações Not-for-profit podem também procurar obter fundos de fundações ou empresas de forma a suportar a sua missão.

### **Knowledge Networks**

Knowledge Networks também conhecido por expert sites, são sites que providenciam uma fonte de informações baseada no conhecimento profissional ou na experiência de outros utilizadores. Estes sites são normalmente apresentados como um fórum onde as pessoas à procura de informação possam pôr questões e receber respostas.

### ***Subscription Model***

Os utilizadores pagam para aceder ao site em que o conteúdo seja de valor acrescentado. Um estudo de 1999 efectuada pela Jupiter Communications concluiu que 46 por cento dos utilizadores da Internet se recusam a pagar para ter acesso a informação. Alguns negócios combinaram o conteúdo livre com o conteúdo e serviços exclusivos a subscritores.

### ***Utility Model***

Este modelo é uma aproximação do género paga à medida que navega. O seu sucesso depende principalmente da habilidade de cobrar ao byte incluindo também a possibilidade de executar os micropagamentos, ou seja, os valores demasiado baixos para pagar com cartão de crédito devido às taxas de processamento.

## Apêndice B - Manual do Utilizador

*Serve este manual para clarificar os vários procedimentos de instalação e gestão do centro comercial. Descrevem-se neste manual as várias interfaces do centro comercial e as operações que possibilitam.*

### **Processo de instalação**

O processo de instalação de toda a aplicação é muito simples e faz-se aqui a descrição do processo de instalação numa máquina com WINDOWS.

1. Instalar Voyager seguindo as instruções próprias
2. Instalar AgentSpace seguindo as instruções próprias
3. Instalar classes do exemplo em agentspace/ag\_natives/mall
4. Instalar site correspondente e actualizar os parâmetros de configuração do manager
5. Instalar bases de dados
  - a. *Ambiente windows*: Registrar base de dados por ODBC ou instalar IDS Server
  - b. *Ambiente unix*: Instalar IDS Server ou os drivers para o postgres

O código foi desenvolvido para os 3 casos descritos.

Para ver uma explicação mais detalhada do processo de instalação por favor ver <http://cosmos.inesc.pt/~ioliveira/> .

### **Iniciar a aplicação**

1. Para iniciar a aplicação basta correr o AgentSpace (executar jas)
2. Registrar a classe de manager: **amac manager m**
3. Criar Agente manager: **ca localhost:8888/PID\_1 manager m**.

Neste momento iniciou-se o agente manager que também criou o agente broker e o clientPlace

4. Para interagir com a aplicação tem de chamar a interface da aplicação: **ia localhost:8888/PID\_1|AID\_1** .

## Gerir o centro comercial

O gestor do centro comercial pode realizar a gestão do centro comercial através da interface do agente manager (imagem da direita) ou da applet AppletManager (imagem da esquerda) inserida em <http://cosmos.inesc.pt/~ioliveira/manager.htm>.

The image shows two versions of a configuration interface. The left version is a standard web form with the following sections:

- Configuration Parameters:** Two text input fields for 'Virtual Directory of the HTML Pages' and 'Directory of the HTML Pages', followed by an 'Apply' button.
- Shop Type:** Fields for 'Initials' and 'Name', with 'Find', 'Add', 'Change', and 'Del' buttons.
- Shop Owner:** Fields for 'Name' and 'E-Mail', with 'Find', 'Add', 'Change', and 'Del' buttons.
- Shop:** Fields for 'Initials', 'Shop Owner' (dropdown), 'Shop Type' (dropdown), 'Name', 'NIF', 'E-Mail', 'Max Native Agents' (50), 'Max Visit Agents' (50), 'Login', and 'Password'. It includes 'Find', 'Add', 'Change', 'Del', 'Save All', and 'Close' buttons.

The right version is a Java applet window titled 'MANAGER' with the same layout, but with pre-filled values:

- Configuration Parameters:** 'Virtual Directory of the HTML Pages' is 'http://fera/Email/' and 'Directory of the HTML Pages' is 'C:/inetpub/wwwroot/Email/'.
- Shop Type:** 'Name' is empty.
- Shop Owner:** 'Name' is empty.
- Shop:** 'Shop Owner' is 'sofia' and 'Shop Type' is 'livraria'.

Figura 16 - Interfaces gráficas existentes para realizar a gestão do centro comercial

O gestor do centro comercial pode realizar o seguinte conjunto de acções:

### Editar parâmetros de configuração

Por parâmetros de configuração entendem-se neste caso as directorias virtuais e físicas onde as páginas HTML das lojas vão ficar armazenadas e podem ser acedidas.

**Save all**

O save all permite realizar a salvaguarda de todo o contexto da aplicação incluindo os vários agentes presentes no sistema. Para fechar o agentspace deverá digitar “q” na shell do AgentSpace.

*Para reinicializar a aplicação correctamente deverá chamar a interface do agente manager ou seja:*

**ia fera:8888/PID\_1|AID\_1**

**Criar, apagar, modificar e procurar lojista**

Todas as lojas existentes pertencem a um lojista. A informação aqui gerida é exclusiva ao agente manager. Um lojista é definido por um nome e um e-mail e cada lojista pode ter uma ou mais lojas.

**Criar, apagar, modificar e procurar tipos de lojas**

O tipo de lojas é mais uma característica que define uma loja. Os tipos de lojas são descritos por uma abreviatura e uma descrição sendo exemplos de um tipo de loja uma farmácia ou uma sapataria.

**Criar, apagar, modificar e procurar lojas**

Esta é de longe a função mais importante desta interface pois permite ao gestor do centro comercial fazer a gestão das lojas existentes no sistema podendo criar, encontrar, modificar e apagar lojas. *Importante: Note-se que as siglas das lojas deverão ser definidas com uma palavra apenas.*

Este não é no entanto o único processo para criar lojas nesta aplicação.

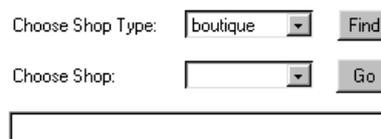
## Mecanismo de Procura

Todo o mecanismo de procura integrado nesta aplicação é realizado através de applets e do agente broker. O processamento dos pedidos de procura pode ser efectuado por loja, tipo de loja, produto e serviço correspondendo cada uma destas funcionalidades à existência de uma applet.

A interface gráfica destas applets é muito simples e como se pode ver na imagem apresentada tem um funcionamento muito intuitivo!

Basta o utilizador escolher um tipo de loja e pressionar “Find”. Todas as lojas que correspondem àquele tipo de loja aparecem na choice inferior. O utilizador escolhe a loja pretendida e pressiona “Go” sendo imediatamente o seu browser redireccionado para a loja.

Para comodidade dos utilizadores todo este processo está descrito nas páginas HTML em que as applets estão inseridas.



The image shows a graphical user interface for a search applet. It consists of two rows of controls. The first row is labeled 'Choose Shop Type:' and contains a dropdown menu with 'boutique' selected, a 'Find' button, and a 'Go' button. The second row is labeled 'Choose Shop:' and contains a dropdown menu that is currently empty, a 'Go' button, and a large empty text input field below it.

Figura 17 – applet que permite procurar uma loja por tipo de loja

A applet que se encontra em [http://cosmos.inesc.pt/~ioliveira/serach\\_shopType.htm](http://cosmos.inesc.pt/~ioliveira/serach_shopType.htm) inicia a ligação ao AgentSpace através da API oferecida pelo AgentSpace e realizando o login com os dados do Internal User do agente broker.

Após a obtenção de uma referência para o agente broker, a applet envia mensagem ao broker a pedir um vector com todos os tipos de loja existentes. Por vector entende-se a classe Vector que faz parte do JDK. O broker vai ler a informação à tabela tipoLoja e devolve a informação pretendida que é exposta na choice superior.

Quando o utilizador pressiona o “Find” é novamente enviada uma mensagem ao broker a pedir todas as lojas daquele tipo. O broker consulta a tabela loja e devolve o vector com as siglas das lojas que obedecem à condição especificada. A applet expõe os resultados da procura na choice inferior. Se o utilizador decidir escolher uma das lojas e pressionar o “Go”, a applet envia uma mensagem ao broker a pedir a URL que depois é utilizada para redireccionar o browser.

## Criação das novas lojas

A criação de novas lojas nesta aplicação pode-se realizar de duas formas completamente distintas. A primeira é por inserção directa das nova lojas através da interface do agente gestor ou da applet AppletManager descrita anteriormente e é realizada directamente pelo gestor do sistema.

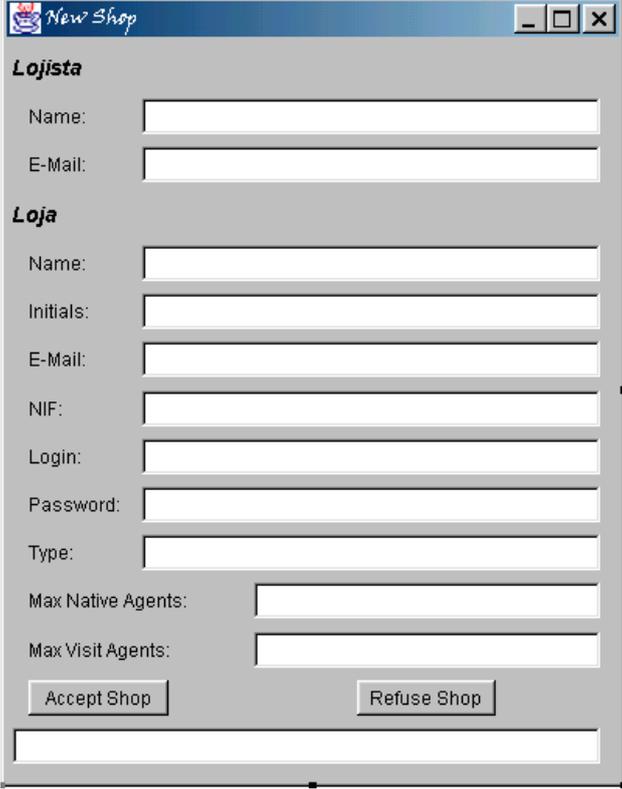
The image shows a graphical user interface for creating new shops. It is divided into two sections: 'Lojista' and 'Loja'. The 'Lojista' section has two text input fields: 'Name:' and 'E-Mail:'. The 'Loja' section has six text input fields: 'Name:', 'Initials:', 'NIF:', 'E-Mail:', 'Login:', and 'Type:'. The 'Type:' field is a dropdown menu. Below the 'Loja' section is a 'New Shop' button and a long empty text input field.

Figura 18 – applet utilizada para criar novas lojas

A segunda é realizada em duas etapas distintas:

Na primeira fase o candidato a lojista preenche o formulário da applet que está inserida em [http://cosmos.inesc.pt/~ioliveira/new\\_shop.htm](http://cosmos.inesc.pt/~ioliveira/new_shop.htm).

Esta applet está ligada ao agente broker que passa a informação para o agente manager. Esta informação fica armazenada no agente manager até ao momento em que o gestor do sistema volta a aceder à interface do agente ou à applet AppletManager e aparece uma janela com toda a informação preenchida pelos candidatos a lojistas.



The image shows a software window titled "New Shop" with a standard Windows-style title bar. The window is divided into two main sections: "Lojista" (Merchant) and "Loja" (Shop). The "Lojista" section contains two text input fields labeled "Name:" and "E-Mail:". The "Loja" section contains seven text input fields labeled "Name:", "Initials:", "E-Mail:", "NIF:", "Login:", "Password:", and "Type:". Below these fields are two more text input fields labeled "Max Native Agents:" and "Max Visit Agents:". At the bottom of the form, there are two buttons: "Accept Shop" on the left and "Refuse Shop" on the right. A vertical scrollbar is visible on the right side of the window.

**Figura 19 – Janela de confirmação de criação de loja nova**

Após a aprovação do gestor é criado um agente loja novo e uma página html com a sigla da loja.

A implementar seria o envio de uma mensagem mail ao lojista a confirmar ou a rejeitar a criação da loja nova.

## Gestão da loja e da sua informação

Estas interfaces permitem realizar a gestão da informação sobre a loja e sobre os produtos e serviços comercializados. O lojista consegue inserir de uma forma fácil e rápida um produto ou serviço novo. Para além disso, o lojista também pode procurar, modificar, apagar e, no caso dos produtos, adicionar novas unidades.

Uma das limitações do centro comercial actualmente relaciona-se com o facto de não ser possível fazer uma gestão gráfica ou automatizada das páginas html ligadas à loja, o que implica que o próprio lojista tem de fazer a edição da(s) página(s) da sua loja manualmente e inserir os produtos manualmente.

The figure displays two versions of a web-based shop management interface. The left version is a standard web page layout, while the right version is a windowed application titled "Shop Interface". Both interfaces are divided into three main sections: "Shop Information", "Change Password", and "Management of Products and Services".

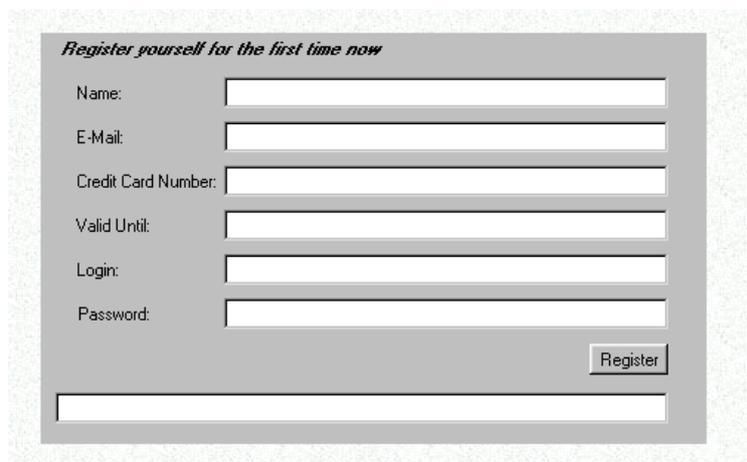
- Shop Information:** Contains input fields for Name (lisboa), E-Mail (lisboa.Email@mar.te.pt), NIF (524561246514), and Shop URL (http://fera/Email/lisboa.htm), followed by a "Change" button.
- Change Password:** Features three input fields for Old Password, New Password, and a second New Password, with a "Change Pwd" button.
- Management of Products and Services:** Includes radio buttons for "Products" (selected) and "Services", and input fields for Initials, Description, Price, I.V.A. %, and Units. Below these is a search bar with buttons for "Find", "Add", "Change", "Del", and "Add Units".

The windowed version on the right also includes a "Close" button at the bottom right.

Figura 20 - Interfaces gráficas utilizadas para realizar a gestão de produtos e serviços comercializados numa loja e também da sua informação pessoal

## **Registo de um Cliente**

O registo dos clientes no centro comercial também é processado através do agente broker. A applet que faz o registo envia uma mensagem ao broker com os dados preenchidos pelo utilizador. Utiliza-se uma instância da classe clientInfo para passar os dados. O broker verifica se cliente já existe, e se não existir cria um agente cliente e insere a informação na tabela cliente. Após a finalização de todo o processo é mostrada uma mensagem de confirmação ao utilizador.



The image shows a registration form with the following fields and a button:

- Name:
- E-Mail:
- Credit Card Number:
- Valid Until:
- Login:
- Password:
- Register button

**Figura 21 - Applet que permite registar um cliente no E-mail**

A applet encontra-se inserida na página <http://cosmos.inesc.pt/~ioliveira/registry.htm>.

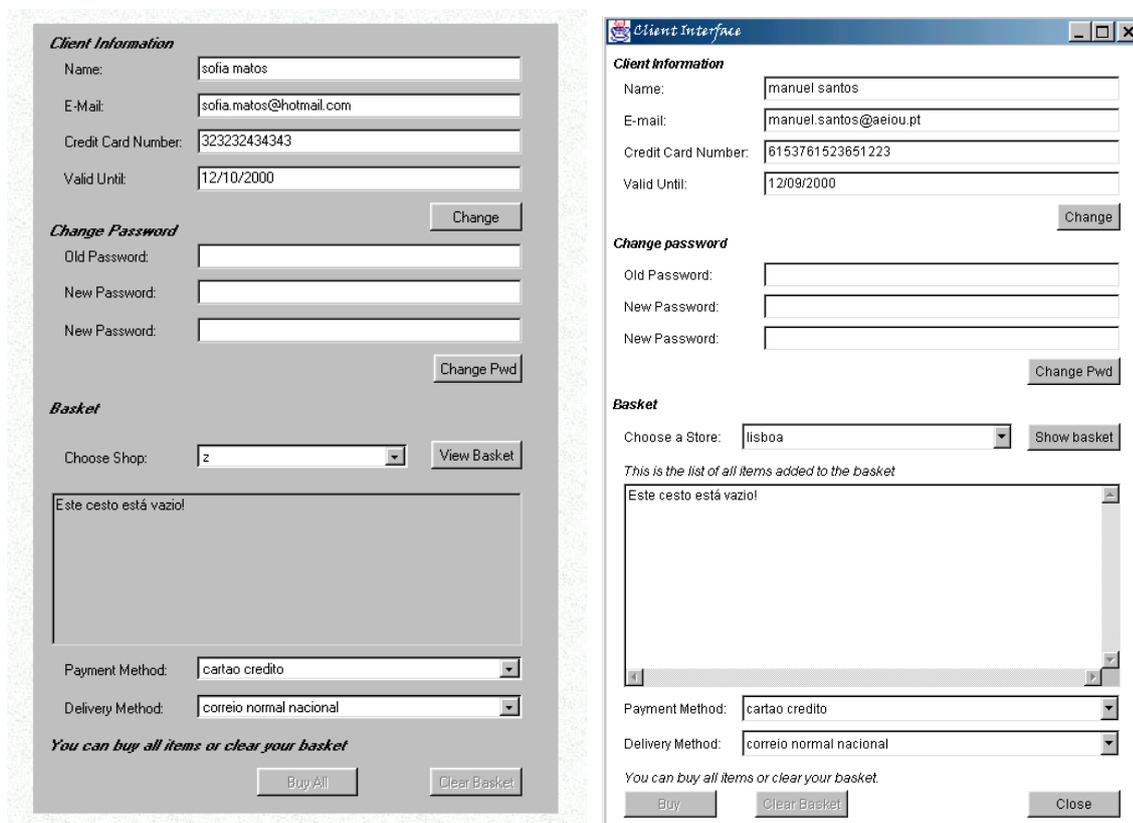
## **Gestão do Cesto e da Informação Pessoal dos Clientes**

A gestão do cesto e da informação pessoal do cliente pode ser feita de duas formas:

Através de uma applet no centro comercial ou através da interface dos agentes acessível pelo Agentspace. Ambas as interfaces permitem realizar exactamente as mesmas operações:

- Configuração da Informação dos Clientes
- Possibilidade de mudar a Password dos Clientes
- Gestão do cesto sendo o estado do cesto independente de loja para loja

A imagem à esquerda corresponde à imagem da applet que pode ser utilizada para fazer a gestão do cesto e da informação do cliente. O que se está a observar é o segundo painel da applet que só se torna visível após o utilizador ter feito o login na applet.



**Figura 22 - Interfaces gráficas para realizar gestão de informação pessoal e do cesto de compras de um cliente**

É possível nesta interface aceder a qualquer cesto do cliente em qualquer loja do centro comercial bastando para tal escolher a sigla da loja pretendida e pressionar “View Basket”.

A possibilidade de comprar ou apagar o conteúdo do cesto só está activa após a selecção de um cesto.

Os restantes aspectos da interface gráfica como as applets que permitem realizar procuras no sistema estão explicados no próprio site.

## Inserção de um produto ou serviço no cesto do cliente

A applet aqui apresentada encontra-se integrada em todas as lojas do centro comercial e é o único processo que os clientes têm de pôr um produto ou serviço no seu cesto. O funcionamento desta applet é muito simples:

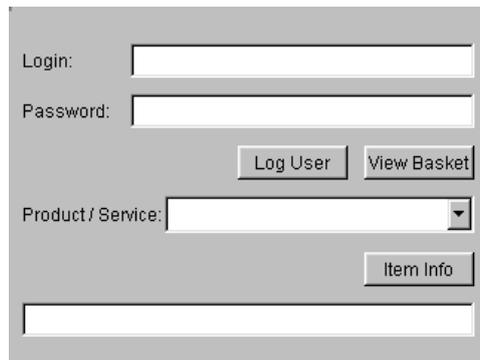
A screenshot of a web applet interface. It features a 'Login:' label with a text input field, a 'Password:' label with a text input field, and two buttons: 'Log User' and 'View Basket'. Below these is a 'Product / Service:' label with a dropdown menu and an 'Item Info' button. At the bottom, there is a wide, empty text input field.

Figura 23 - applet utilizada para realizar as compras

O cliente autentica-se na loja através do seu login e password, escolhe o produto ou serviço que deseja pôr no cesto e pressiona ItemInfo.

A janela que aparece permite visionar todos os detalhes do produto ou serviço escolhido. Caso seja produto, é necessário que o cliente insira o número de unidades que deseja adquirir e pressione em “Show Cost” de modo a ser calculado o novo valor do produto ou serviço que o cliente deseja adquirir.

Se cliente estiver interessado em adquirir o produto ou serviço basta pressionar “Add”.

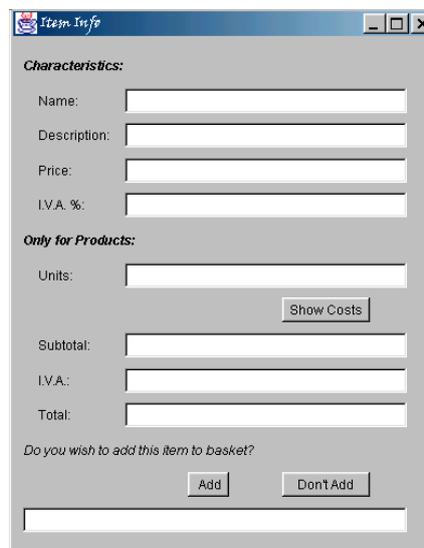
A screenshot of a window titled 'Item Info'. It contains a 'Characteristics:' section with fields for 'Name:', 'Description:', 'Price:', and 'I.V.A. %:'. Below this is an 'Only for Products:' section with a 'Units:' field and a 'Show Costs' button. At the bottom, there are fields for 'Subtotal:', 'I.V.A.:', and 'Total:'. A question 'Do you wish to add this item to basket?' is followed by 'Add' and 'Don't Add' buttons. A wide, empty text input field is at the very bottom.

Figura 24 – Interface utilizada para adicionar itens ao cesto

Para finalizar a compra o cliente tem de aceder ao seu cesto (ver interface cliente) e realizar a compra.

Neste trabalho são utilizados dois tipos de interface gráfica claramente diferentes. O primeiro tipo de interface gráfica são as applets , já anteriormente descritas e que podem ser executadas dentro de qualquer browser que suporte Java.

Ambas as interfaces são classes Java com as suas características próprias tendo-se implementado para os agentes client e os agentes shop ambos os tipos de interface.

## Apêndice C - Scripts para criar bases de dados

Apresenta-se aqui os scripts em SQL que permitem criar as tabelas necessárias ao funcionamento da aplicação desenvolvida.

Note-se que estes scripts se dirigem a três bases de dados diferentes: manager, broker e shop. Estes scripts foram testados numa base de dados PostgreSQL para Unix.

### ***Script das tabelas do gestor (Manager):***

```
CREATE TABLE manager
(managerId SERIAL PRIMARY KEY,
managerAID CHAR(50) NOT NULL,
pagHTML CHAR(50) NOT NULL,
pagHTMLVirtual CHAR(50) NOT NULL,
broker CHAR(50) NOT NULL);

CREATE TABLE tipoLoja
(tipoLojaId SERIAL PRIMARY KEY,
abreviatura CHAR(50) NOT NULL,
nome CHAR(50) NOT NULL);

CREATE TABLE lojista
(lojistaId SERIAL PRIMARY KEY,
nome CHAR(50) NOT NULL,
email CHAR(50) NOT NULL);

CREATE TABLE loja
(lojaId SERIAL PRIMARY KEY,
tipoLojaId INTEGER NOT NULL REFERENCES tipoLoja(tipoLojaId),
lojistaId INTEGER NOT NULL REFERENCES lojista(lojistaId),
nome CHAR(50) NOT NULL,
sigla CHAR(50) NOT NULL,
nif CHAR(50) NOT NULL,
login CHAR(50) NOT NULL,
password CHAR(50) NOT NULL,
lojaURL CHAR(50),
email CHAR(50) NOT NULL,
agentLojaId CHAR(50) NOT NULL,
placeLojaId CHAR(50),
placeLojaName CHAR(50),
maxAgVisit INTEGER,
maxAgNativ INTEGER);
```

**Script para as tabelas das lojas:**

```
CREATE TABLE metaLoja
(lojaId SERIAL PRIMARY KEY,
nome CHAR(50) NOT NULL,
sigla CHAR(50) NOT NULL,
nif CHAR(50) NOT NULL,
lojaURL CHAR(50),
email CHAR(50) NOT NULL,
produto CHAR(50) NOT NULL,
servico CHAR(50) NOT NULL,
agentId CHAR(50) NOT NULL,
cliente CHAR(50) NOT NULL,
transacao CHAR(50) NOT NULL,
transacaoProduto CHAR(50) NOT NULL,
transacaoServico CHAR(50) NOT NULL);
```

**Script para as tabelas do broker:**

```
CREATE TABLE produto
(produtoId SERIAL PRIMARY KEY,
abreviatura CHAR(50) NOT NULL,
descricao CHAR(50) NOT NULL);
```

```
CREATE TABLE servico
(servicoId SERIAL PRIMARY KEY,
abreviatura CHAR(50) NOT NULL,
descricao CHAR(50) NOT NULL);
```

```
CREATE TABLE tipoLoja
(tipoLojaId SERIAL PRIMARY KEY,
abreviatura CHAR(50) NOT NULL,
nome CHAR(50) NOT NULL);
```

```
CREATE TABLE loja
(lojaId SERIAL PRIMARY KEY,
tipoLojaId INTEGER NOT NULL REFERENCES tipoLoja(tipoLojaId),
nome CHAR(50) NOT NULL,
sigla CHAR(50) NOT NULL,
lojaURL CHAR(50) NOT NULL,
email CHAR(50) NOT NULL,
agentLojaId CHAR(50) NOT NULL);
```

```
CREATE TABLE produtoLoja
```

```
(produtoLojaId SERIAL PRIMARY KEY,  
lojaId INTEGER NOT NULL REFERENCES loja(lojaId),  
produtoId INTEGER NOT NULL REFERENCES produto(produtoId));
```

```
CREATE TABLE servicoLoja  
(servicoLojaId SERIAL PRIMARY KEY,  
lojaId INTEGER NOT NULL REFERENCES loja(lojaId),  
servicoId INTEGER NOT NULL REFERENCES servico(servicoId));
```

```
CREATE TABLE broker  
(brokerId SERIAL PRIMARY KEY,  
manager CHAR(50) NOT NULL,  
clientPlace CHAR(50) NOT NULL);
```

```
CREATE TABLE cliente  
(clienteId SERIAL PRIMARY KEY,  
name CHAR(50) NOT NULL,  
Email CHAR(50) NOT NULL,  
cartaoCredito CHAR(50) NOT NULL,  
validade CHAR(50) NOT NULL,  
login CHAR(50) NOT NULL,  
password CHAR(50) NOT NULL,  
agentId CHAR(50) NOT NULL);
```

```
CREATE TABLE basket  
(ID SERIAL PRIMARY KEY,  
lojaId INTEGER NOT NULL REFERENCES loja(lojaId),  
clienteId INTEGER NOT NULL REFERENCES cliente(clienteId),  
abreviatura CHAR(50) NOT NULL,  
descricao CHAR(50) NOT NULL,  
unidades INTEGER NOT NULL,  
precoUnitario INTEGER NOT NULL,  
taxaIVA INTEGER NOT NULL,  
subtotal INTEGER NOT NULL,  
subtotalIVA INTEGER NOT NULL);
```

```
CREATE TABLE metodoEntrega  
(entregaId SERIAL PRIMARY KEY,  
entrega CHAR(50) NOT NULL);
```

```
CREATE TABLE metodoPagamento  
(pagamentoId SERIAL PRIMARY KEY,  
pagamento CHAR(50) NOT NULL);
```

## Apêndice D - Listagens do Código

Todas as classes desenvolvidas pertencem à package mall e podem-se subdividir nos seguintes conjuntos de classes:

Applet	Painel	Agente	Suporte ao modelo de dados	Janela	Outros
<i>AppletBuy</i>	<i>BasketPanel</i>	<i>Broker</i>	<i>buyInfo</i>	<i>BrokerInterface</i>	<i>GrigBagCon</i>
<i>AppletCliente</i>	<i>loginPanel</i>	<i>Client</i>	<i>clientInfo</i>	<i>ClientInterface</i>	<i>strainsUtil</i>
<i>AppletManager</i>	<i>shopLoginPanel</i>	<i>Manager</i>	<i>configInfo</i>	<i>ClientValidation</i>	
<i>AppletNovaLoja</i>	<i>shopPanel</i>	<i>Shop</i>	<i>item</i>	<i>ManagerIn</i>	
<i>AppletSearchProduct</i>			<i>ItemInfo</i>	<i>ManagerShopChanges</i>	
<i>AppletSearchService</i>			<i>loginInfo</i>	<i>ShopIn</i>	
<i>AppletSearchShop</i>			<i>lojistaInfo</i>	<i>shopValidation</i>	
<i>AppletSearchShopType</i>			<i>ProdServInfo</i>		
<i>clientApplet</i>					
<i>shopApplet</i>					

As applets permitem fazer a interface com o E-mall através de um browser.

Os painéis são utilizados nas applets *clientApplet* e na *shopApplet*.

Os agentes existentes já foram descritos pormenorizadamente ao longo do relatório.

As classes de suporte ao modelo de dados servem para suportar o modelo de dados implementado.

As janelas implementam a interface gráfica dos vários agentes.

A listagem de código que se segue apresenta as classes da aplicação organizadas por ordem alfabética.

## 7 Referências

- [1] Silva, A.M.R., *Espaço de Agentes: Suporte, Desenvolvimento e Gestão de Aplicações Baseadas em Agentes, Dinâmicas e Distribuídas*, IST/UTL, Tese de Doutorado, Junho de 1998
- [2] Hunter, J. & Crawford, W., *Java Servlet Programming*, O'Reilly, Sebastopol, Outubro 1998
- [3] Campione, M. & Walrath, K., *The Java Tutorial*, Addison Wesley, Massachusetts, Maio 1995
- [4] Arnold, K. & Gosling, J., *The Java Programming Language*, Addison Wesley, 1996
- [5] CalKins, J.D. , Farello, M.J. & Shi, C. S., *From retailing to e-tailing*,  
[www.mckinseyquarterly.com](http://www.mckinseyquarterly.com)
- [6] Gartner Group, *Transforming Business Through E-Commerce*, Conference Presentation, 1999
- [7] Rozwell, C. , *Electronic Commerce & Extranet Applications*, Gartner Group, Janeiro 2000
- [8] Rozwell, C & Berg, T. , *How to Devise a Practical, Effective E-Business Strategy*, Strategic Analysis Report, Gartner Group, Setembro 1999
- [9] Franklin, S. & Graesser, A., *Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, Proceedings of the Agent Theories, Architectures, and Languages Workshop*, Berlin: Springer Verlag, 193-206
- [10] Lawton, G. Agents to roam the Internet, <http://www.sunworld.com/Março> 2000
- [11] Guttman, R.H. & Maes, P. , *Agent-mediated Integrative Negotiation for Retail Electronic Commerce*, <http://ecommerce.media.mit.edu>, Maio 1998
- [12] Guttman, R.H. & Maes, P., *Agent-mediated Electronic Commerce: A Survey*, Knowledge Engineering Review, Junho 1998
- [13] Guttman, R.H. & Maes, P. & Moukas, A., *Agent-mediated Electronic Commerce: An MIT Media Laboratory Perspective*, <http://ecommerce.media.mit.edu>, 1997
- [14] Guttman, R.H. & Maes, P., *Cooperative vs. Competitive Multi-Agent Negotiations in Retail Electronic Commerce*, <http://ecommerce.media.mit.edu>, 1998
- [15] Wooldridge, W. & Jennings, N.R., *Pitfalls of Agent-Oriented Development*, The Knowledge Engineering Review, 10(2):115-152, 1995
- [16] Clarke, R. , *Electronic Commerce Definitions*, <http://www.anu.edu.au/people/RogerClarke/EC/>, Australian National University, Fevereiro 1999
- [17] Clarke, R., *EDI is but One Element of Electronic Commerce*,  
<http://www.anu.edu.au/people/RogerClarke/EC/>, Australian National University, Fevereiro 1999
- [18] Truog, D. & Manning, H. & Goldman, H. , *The Forrester Report, Which Commerce Platform?*, Outubro 1999
- [19] Guttman, R.H. & Maes, P. & Moukas, A., *Agents that Buy and Sell: Transforming Commerce as we know it*, Communications of the ACM, Março 1999
- [20] Rappa, M. *Business Models on the Web*, NC State University, Março 2000
- [21] Franklin, S. , *Coordination without Communication*, University of Memphis, January 1997

**URL**

Apresenta-se ainda uma lista de endereços de sites utilizados na elaboração do trabalho.

1. <http://berlin.inesc.pt/>
2. <http://cosmos.inesc.pt>
3. <http://www.objectspace.com>
4. <http://www.agent.org>
5. <http://agents.umbc.edu/>
6. <http://beta.hotbot.com>
7. <http://www.gwnmagic.com/technology/>
8. <http://www.cs.nccu.tw/~jong/agent.html>
9. <http://www.infosys.tuwien.ac.at/Research/>
10. <http://www.ecom.cmu.edu-resources-elibrary.html>
11. <http://ecommerce.media.mit.edu/>
12. <http://www.sims.berkeley.edu/resources/infoecon/Commerce.html>
13. <http://www.commerce.net>
14. <http://ecommerce.internet.com>
15. <http://java.sun.com>
16. <http://www.deja.com>
17. <http://www.developer.com>
18. <http://www.sunworld.com>
19. <http://www.idsoftware.com>