

**UNIVERSIDADE LUTERANA DO BRASIL**  
**FACULDADE DE INFORMÁTICA**  
**BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**  
**CAMPUS GRAVATAÍ**



**IMPLEMENTAÇÃO DE E-BUSINESS BASEADO  
EM AGENTES**

Eduardo Menegazzi

Monografia desenvolvida durante a disciplina de Trabalho de Conclusão de Curso em Informática II e apresentada à Faculdade de Informática da Universidade Luterana do Brasil, campus Gravataí, como pré-requisito para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Leandro Bento Pompermaier

Gravataí, julho de 2001.

**Universidade Luterana do Brasil – ULBRA**  
**Faculdade de Informática**  
**Curso de Bacharelado em Ciência da Computação – Campus Gravataí**

*Reitor:*

**Pastor Ruben Eugen Becker**

*Vice-Reitor:*

**Eng. Leandro Eugênio Becker**

*Diretor do Campus Gravataí:*

**Prof. Felício Korb**

*Diretor da Faculdade de Informática:*

**Prof. Miguel Rodrigues Fornari**

*Coordenador de Curso (Campus Gravataí):*

**Prof. Marco Antônio da Rocha**

*Coordenador das Disciplinas de Trabalho de Conclusão de Curso (Campus Gravataí):*

**Prof. Roland Teodorowitsch**

*Banca Avaliadora composta por:*

*Data da defesa: 11/07/2001.*

**Prof. Leandro Bento Pompermaier (Orientador)**

**Prof.<sup>a</sup> Patrícia Nogueira Hübler**

**Prof. Vinícius Gadis Ribeiro**

## **CIP – Catalogação na Publicação**

Menegazzi, Eduardo

Implementação de E-business Baseado em Agentes / Eduardo Menegazzi. –  
Gravataí: Faculdade de Informática da ULBRA, 2001.  
xii, 44p.

Trabalho de Conclusão de Curso (Graduação) – Universidade Luterana do  
Brasil, Faculdade de Informática, Curso de Bacharelado em Ciência da  
Computação, Gravataí, 2001. Orientador: Prof. Leandro Bento Pompermaier.

1. Agentes. 2. E-business. 3. E-commerce. I. Trabalho de Conclusão de  
Curso. II. Menegazzi, Eduardo. III. Título.

*Endereço:*

Universidade Luterana do Brasil – Campus Gravataí  
Estrada Itacolomi, 3.600 – Bairro São Vicente  
CEP 94170-240 Gravataí-RS – Brasil

À minha esposa, Cristiane, aos meus pais, Fernando e Marlise, e à minha irmã, Fernanda, com todo o carinho, pelo incentivo, pelo apoio e pela compreensão.

## **AGRADECIMENTOS**

Agradeço ao meu professor orientador, Leandro, por conduzir minhas idéias neste trabalho final.

Agradeço aos professores Patrícia e Vinicius, por aceitarem fazer parte de minha banca avaliadora.

Agradeço também aos meus colegas, pelo companheirismo e amizade.

Por fim, gostaria também de agradecer à todos que de alguma forma tenham contribuído para a realização deste trabalho.

## SUMÁRIO

<b>LISTA DE FIGURAS .....</b>	<b>vii</b>
<b>LISTA DE QUADROS .....</b>	<b>viii</b>
<b>LISTA DE ABREVIATURAS E SIGLAS.....</b>	<b>ix</b>
<b>RESUMO .....</b>	<b>xi</b>
<b>ABSTRACT .....</b>	<b>xii</b>
<b>1 INTRODUÇÃO.....</b>	<b>1</b>
1.1 MOTIVAÇÃO.....	1
1.2 OBJETIVOS.....	2
1.3 ORGANIZAÇÃO DO TEXTO .....	2
<b>2 COMÉRCIO ELETRÔNICO.....</b>	<b>4</b>
2.1 <i>E-COMMERCE</i> E <i>E-BUSINESS</i> .....	4
2.2 <i>BUSINESS-TO-BUSINESS</i> .....	5
2.3 <i>BUSINESS-TO-CONSUMER</i> .....	5
2.3.1 Produtos Tangíveis.....	5
2.3.2 Produtos Intangíveis .....	5
2.4 AS NOVAS TENDÊNCIAS E O FUTURO DO COMÉRCIO ELETRÔNICO .....	7
2.5 CONCLUSÃO .....	9
<b>3 SEGURANÇA EM COMÉRCIO ELETRÔNICO .....</b>	<b>10</b>
3.1 ASPECTOS DE SEGURANÇA QUE DEVEM SER CONSIDERADOS .....	10
3.2 SET.....	11
3.3 SSL.....	12
3.4 CONCLUSÃO .....	13
<b>4 AGENTES INTELIGENTES .....</b>	<b>14</b>
4.1 O QUE É UM AGENTE .....	14
4.2 CARACTERÍSTICAS DE AGENTES .....	15
4.3 TIPOS DE AGENTES.....	16
4.4 ABORDAGENS PARA A CONSTRUÇÃO DE AGENTES .....	17
4.5 AGÊNCIA E INTELIGÊNCIA .....	18
4.6 APLICAÇÕES DE AGENTES INTELIGENTES.....	18
4.7 ACESSO À INFORMAÇÃO NA INTERNET.....	20
4.8 AGENTES DE INFORMAÇÃO NA PRÁTICA .....	21
<b>5 DESENVOLVENDO UM <i>SITE</i> PARA INTERNET .....</b>	<b>22</b>
5.1 HTML.....	22

5.2	CGI.....	23
5.3	JAVA .....	23
5.3.1	Características da Linguagem Java .....	24
5.4	JAVASCRIPT.....	26
5.4.1	Diferença entre Java e JavaScript .....	26
5.5	ASP .....	26
5.5.1	Vantagens das Páginas ASP .....	27
5.5.2	Requisitos para Utilizar ASP .....	27
5.5.3	ASP e Outras Tecnologias .....	28
5.6	JSP.....	28
5.6.1	Diferenças entre JSP e ASP.....	29
5.7	CONCLUSÃO .....	30
<b>6</b>	<b>AGÊNCIA DE VIAGENS <i>ON-LINE</i>.....</b>	<b>31</b>
6.1	CARACTERÍSTICAS .....	31
6.2	ESTADO DA ARTE.....	31
<b>7</b>	<b>PROJETO .....</b>	<b>35</b>
7.1	FUNCIONALIDADES .....	35
7.2	MODELO DE INFORMAÇÕES SEMÂNTICAS .....	36
7.3	DIAGRAMA DE SEQUÊNCIA.....	36
7.4	TECNOLOGIAS ENVOLVIDAS.....	37
7.4.1	Webbot BDDBot .....	37
7.5	PROTÓTIPO .....	39
<b>8</b>	<b>CONCLUSÃO.....</b>	<b>41</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>42</b>
	<b>OBRAS CONSULTADAS.....</b>	<b>44</b>

## LISTA DE FIGURAS

Figura 1 – Modelo do Sistema Ecash .....	6
Figura 2 – Diagrama de Estados de uma Transação Simples .....	12
Figura 3 – Características de Agentes.....	16
Figura 4 – Funcionamento de Páginas JSP .....	29
Figura 5 – Página da Agência Virtual Priceline .....	32
Figura 6 – Página da Agência Virtual Decolar.....	32
Figura 7 – Página da Agência Virtual Descontur.....	33
Figura 8 – Página da Agência Virtual Viajo .....	33
Figura 9 – <i>Use Case</i> do Projeto de Agência Virtual.....	35
Figura 10 – Modelo Semântico do Projeto de Agência Virtual .....	36
Figura 11 – Diagrama de Seqüência – Compra de Pacote Turístico .....	37
Figura 12 – Diagrama de Interação de Classes do BDDBot.....	38
Figura 13 – Página Principal da Agência Virtual.....	39
Figura 14 – Página de Confirmação do Pedido da Agência Virtual.....	40
Figura 15 – Página com <i>Links</i> de Informações Turísticas .....	40

## **LISTA DE QUADROS**

Quadro 1 – Tipos de Criptografia em SSL.....	13
Quadro 2 – Diferenças entre JSP e ASP .....	30



## **LISTA DE ABREVIATURAS E SIGLAS**

ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
ASP	Active Server Pages
B2B	Business-to-Business
B2C	Business-to-Consumer
CGI	Common Gateway Interface
CORBA	Common Object Request Broker Architecture
EDI	Electronic Data Interchange
EJB	Enterprise Java Beans
ERP	Enterprise Resources Planning
FTP	File Transfer Protocol
HTML	Hyper Text Mark-up Language
HTTP	Hyper Text Transfer Protocol
HTTPS	Secure Hyper Text Transfer Protocol
IA	Inteligência Artificial
IAD	Inteligência Artificial Distribuída
IIS	Internet Information Server
JDBC	Java DataBase Connectivity
JSP	JavaServer Pages
ODBC	Open Data Base Connectivity
OO	Orientação a Objetos
PDAs	Portable Data Assistents
PWS	Peer Web Services
PWS	Personal Web Server
RAM	Random Access Memory
SET	Secure Electronic Transaction

SETco	Secure Electronic Transaction Consortium
SSL	Secure Sockets Layer
TCP/IP	Transmission Control Protocol/Internet Protocol
UML	Unified Modeling Language
URL	Uniform Resource Locator
WAIS	Wide Area Information Service
WWW	World Wide Web

## RESUMO

Este trabalho visa demonstrar a utilização de agentes inteligentes como facilitadores no comércio eletrônico, mais especificamente um *site* de agência de turismo *on-line*. É apresentado um estudo feito sobre as tecnologias que envolvem a construção do *site*: conceitos, linguagens e segurança. As principais tecnologias estudadas e implementadas são a utilização de agentes inteligentes, o comércio eletrônico propriamente dito e uma linguagem poderosa para a Internet: Java. Para tal foi desenvolvido um protótipo de agência de viagens virtual, onde o cliente escolhe o pacote de viagem conforme sua necessidade. Aqui os agentes inteligentes buscam na Internet, de maneira automática, informações turísticas sobre o local destino da viagem e trazem ao cliente uma listagem de *links* com as informações encontradas. A implementação deste protótipo é feita com a linguagem Java para desenvolvimento das regras de negócio e implementação dos agentes. Para apresentação da aplicação é utilizado JavaServer Pages (JSP). A soma do comércio eletrônico com o uso de agentes inteligentes, torna-se uma poderosa ferramenta de acesso e processamento de informações na Internet. No caso da Agência Virtual, os agentes inteligentes proporcionam um diferencial em relação às outras agências de turismo virtuais existentes, pois é necessário oferecer algo mais do que o concorrente, para ganhar a preferência do cliente.

## **ABSTRACT**

This work aims at to demonstrate the use of intelligent agents as facilitators in the electronic commerce, more specifically a site of travel agency on-line. A study made on the technologies is presented that involve the construction of the site: concepts, languages and security. The main studied and implemented technologies are the use of intelligent agents, the electronic commerce properly said and a powerful language for the Internet: Java. For such an archetype of virtual travel agency was developed, where the customer chooses the package of in agreement trip its necessity. Here the intelligent agents search in the Internet, of automatic way, tourist information on the local destination of the trip and bring to the customer a listing of links with the joined information. The implementation of this archetype is made with the Java language for development of the business rules and implementation of the agents. For presentation of the application JavaServer Pages is used (JSP). The addition of the electronic commerce with the use of intelligent agents, becomes a powerful tool of access and processing of information in the Internet. In the case of the Virtual Agency, the intelligent agents provide a differential in relation to the other existing virtual travel agencies, therefore it is necessary to more offer something the one that the competitor, to gain the preference of the customer.

# 1 INTRODUÇÃO

O comércio eletrônico é uma área de crescimento acelerado no Brasil e no resto do mundo, sendo alimentado pela popularização da Internet. As pessoas estão interessadas nas vantagens que este meio de comércio proporciona: facilidade de uso, conforto, muita informação, economia de despesas e tempo, entre outras.

Mas, em meio a tanta informação, é necessário um mecanismo para se ter a informação adequada. Este mecanismo é implementado através de agentes, que trabalham incansavelmente, dentro das suas limitações, para ofertar ao usuário a maior comodidade possível. São eles os agentes inteligentes.

Estes agentes inteligentes podem auxiliar, e muito, no comércio eletrônico.

## 1.1 MOTIVAÇÃO

Hoje em dia, é imprescindível rapidez para que as empresas possam obter informações e efetivar negócios, visando maior lucro e uma melhor colocação em comparação a seus concorrentes. E é por este motivo, entre outros, que as empresas e corporações estão utilizando a Internet como meio para concretizar seus negócios. Além de oferecer maior agilidade nas transações, a Internet proporciona um nível muito inferior de burocratização do que os processos tradicionais de negociação.

A partir disto, vem se desenvolvendo uma nova modalidade de comércio, o *e-commerce*. Mas a logística do *site* de comércio eletrônico precisa ser estruturada, não podendo ser a mesma utilizada para a forma tradicional de negócios da empresa, pois a velocidade na Internet é outra. Para a automatização desta logística por trás do *site* de *e-commerce*, é utilizada a própria Internet, e desta forma tem-se o *e-business*.

Uma das vantagens do *e-business*, já citada acima, é a velocidade com que o negócio em si é realizado. Outro ponto muito importante é a redução de custos devido ao uso do *e-business* nas empresas. Estima-se algo em torno de US\$ 1,25 trilhões em 2002 o tamanho desta economia (Krause, 2000).

Com o avanço tecnológico, principalmente com relação aos estudos de Inteligência Artificial (IA), deve-se cada vez mais desenvolver aplicações que a utilizem, explorando as suas potencialidades. A IA no desenvolvimento de aplicações distribuídas já está sendo amplamente utilizada (IA Distribuída). Os Agentes Inteligentes tendem a ser uma ferramenta poderosa, que se agregará nos sistemas de informação, auxiliando, entre outras coisas, a busca e filtragem de informações na Internet.

A utilização de análise e projeto Orientado a Objetos no desenvolvimento de aplicações para a Internet também são motivações para este trabalho. E da mesma forma, a utilização de Inteligência Artificial Distribuída (IAD), com a implementação de agentes inteligentes em Java.

O estudo e utilização de tantos conceitos, técnicas, metodologias e ferramentas no decorrer deste trabalho, são mais um fator encorajador para a possibilidade de escrita e submissão de um artigo a respeito do assunto proposto, sendo mais um item motivador.

## 1.2 OBJETIVOS

O objetivo geral deste trabalho é o estudo, projeto e implementação de um protótipo de uma aplicação de *e-business* para uma Agência de Turismo (Viagens) *on-line*.

A idéia é permitir que o usuário/turista selecione um dos pacotes de viagem existentes na aplicação. A partir disto, o sistema busca as informações na Internet sobre os locais turísticos incluídos no pacote (agentes inteligentes).

São objetivos específicos:

- estudar e trabalhar com conceitos de *e-Business* (*business-to-business* (B2B), *business-to-consumer* (B2C));
- estudar e trabalhar com ferramentas, linguagens e técnicas voltados para a Internet, como Java, Hyper Text Mark-up Language (HTML), segurança na Internet e agentes inteligentes;
- estudar e trabalhar com notação de projeto de sistemas, Unified Modeling Language (UML);
- definir o estado da arte em Agência de Viagens/Turismo na Internet;
- projetar e implementar um protótipo de sistema de *e-Business* baseado em agentes.

## 1.3 ORGANIZAÇÃO DO TEXTO

Este trabalho está organizado da seguinte forma: no capítulo 2, são apresentados conceitos de comércio eletrônico, bem como as características destes conceitos. Também são traçadas tendências futuras a respeito do comércio eletrônico.

No capítulo 3, são apresentadas algumas técnicas e aspectos sobre segurança em comércio eletrônico.

No capítulo 4, são abordados os agentes inteligentes, suas aplicações e como funcionam na prática. Também é abordado o acesso às informações na Internet.

No capítulo 5, são apresentadas algumas linguagens, ferramentas e tecnologias para desenvolvimento de aplicações para a Internet. São elas: HTML, Common Gateway Interface (CGI), Java, JavaScript, Active Server Pages (ASP) e JSP.

No capítulo 6, são apresentadas algumas características e estados da arte de agências de turismo *on-line*.

No capítulo 7, é apresentado o projeto do protótipo de uma agência de turismo virtual, que utiliza agentes inteligentes (agentes autônomos) para busca de informações turísticas na Internet.

Finalmente, no capítulo 8, são apresentadas as conclusões referentes ao trabalho.

## 2 COMÉRCIO ELETRÔNICO

A Internet é o meio de comunicação que mais cresce no mundo atual. De um momento para outro, todos os comerciais de TV começaram a mostrar o endereço *Web* do anunciante, isso porque as empresas acreditam que a Internet tenha um grande potencial de *marketing* (Wladawsky, 1998).

A questão do Comércio Eletrônico tem sido extensivamente tratada na mídia nacional e internacional nos últimos anos. Este novo canal de comercialização tem sido apresentado como elemento impulsionador de grandes transformações nos padrões de concorrência, estratégias de *marketing* e nas próprias estruturas organizacionais das empresas.

O estudo do Comércio Eletrônico e a compreensão do perfil dos clientes vêm favorecendo a ampliação deste canal de vendas. As pessoas consomem cada vez mais produtos e serviços pela Internet, o que faz com que a rede deixe de ser somente um grande canal de informação, para se tornar também um efetivo meio de negócios, onde cliente e empresa podem se beneficiar.

O Comércio Eletrônico é uma forma de comércio onde o produto é conhecido, demonstrado e vendido por meios eletrônicos (Wladawsky, 1998). Pode ser definido, também, como a capacidade de realizar transações envolvendo a troca de bens ou serviços entre duas ou mais partes, utilizando ferramentas eletrônicas e tecnologias emergentes (Wladawsky, 1998). Atualmente, o meio mais popular de Comércio Eletrônico é a Internet. A localização geográfica é irrelevante, contribuindo sensivelmente para a globalização do comércio mundial.

### 2.1 *E-COMMERCE E E-BUSINESS*

Existem diferenças entre *e-commerce* e *e-business*.

O *e-commerce* é baseado na compra e venda (na Internet) e em todos os processos que suportem essas compras e vendas, como a propaganda, o *marketing*, o atendimento ao cliente, as atividades com cartões de crédito, etc. (Wladawsky, 1998).

O *e-business*, adicionalmente, engloba o *e-commerce* e inclui muitas outras aplicações para garantir que os negócios ocorram de um modo mais eficiente. Também inclui mais aplicações internas, conectando funcionários para a realização de seus trabalhos de um modo mais produtivo. O *e-business* também envolve a publicação e o acesso a informações (Wladawsky, 1998).

Portanto, pode-se definir que o *e-business* é a grande estratégia e o *e-commerce* é uma atividade extremamente importante do *e-business*.



## 2.2 **BUSINESS-TO-BUSINESS**

São *business-to-business* transações de compra e venda de produtos, serviços e até informações que ocorrem entre corporações (empresas).

Como exemplos de B2B, citam-se:

- publicidade na Internet;
- intranet e transações integradas com *software* de gestão;
- gerenciamento eletrônico de documentos;
- *cross-selling* (retenção e fidelização);
- operações de crédito e débito automática;
- *electronic data interchange* (EDI) .

## 2.3 **BUSINESS-TO-CONSUMER**

São *business-to-consumer* transações de venda de produtos e serviços entre empresas e seus clientes. Existem duas categorias para B2C: uma de produtos tangíveis e outra de produtos intangíveis.

### 2.3.1 **Produtos Tangíveis**

Os produtos tangíveis são produtos físicos (livros, CDs, flores, entre outros) que envolvem manipulação, transporte, seguro, estocagem, embalagem, e assim por diante (Sperb et al., 1999). O custo marginal de cada transação é elevado, mas o preço do produto pode absorver este custo. O ponto crítico é a logística do varejista, e o tipo de pagamento para as transações normalmente é por cartão de crédito ou cobrança bancária (macropagamentos).

### 2.3.2 **Produtos Intangíveis**

Os produtos intangíveis (*software*, artigos, notícias, imagens, músicas, vídeos, textos em geral) trazem consigo a problemática do custo operacional dos modelos de pagamento utilizados atualmente, e do tempo de processamento que estes modelos tradicionais implicam (Sperb et al., 1999). Para a comercialização *on-line* de informações trabalha-se com larga escala e baixo preço (micropagamentos) – podendo ser até mesmo na ordem de centavos. É necessário, portanto, que mecanismos eficientes sejam adotados, para que o tempo de espera do consumidor *on-line* não seja um fator desestimulante no ato de compra, e para que o custo operacional da venda não torne inviável a operação para o vendedor. É nesse cenário que se enquadram os modelos de microtransações.

De uma forma geral, as propostas de sistemas de micropagamento têm três tipos de participantes (Sperb et al., 1999):

- cliente (usuário, comprador), é o usuário da Internet que navega entre diversos *sites*, encontra informações de seu interesse e tem a iniciativa da compra;
- vendedor, é o *site* na Internet que disponibiliza alguma informação, colocando-a a venda;
- *broker* (banco, corretor), é uma entidade centralizadora que é responsável pelo controle e gerenciamento dos créditos e débitos, validando as transações entre clientes e vendedores. Em um sistema escalável e que se proponha ser de grande alcance, é interessante que haja mais de um *broker*.

Para que as transações entre estes participantes seja possível, é importante que haja uma interação entre estes três tipos de entidades, e que todas conversem entre si usando-se do mesmo protocolo. Os clientes devem entrar em contato com os *brokers* para obterem crédito e os vendedores devem entrar em contato para se cadastrarem para recebimento dos pagamentos. Além disso, existe a comunicação entre os clientes e os vendedores, que é quando ocorrem as transações de compra e venda.

A interação entre estes três tipos de entidades é visualizado na Figura 1.

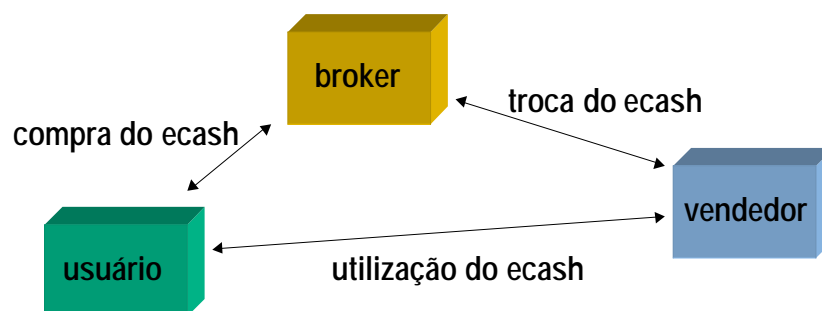
Um fator de grande importância é a confiabilidade dos pagamentos. Existem várias formas de pagamento. Pode-se citar: ligações 0800 para informar os dados necessários; *fax* ou *e-mail* com as devidas informações para os comerciantes; preencher um formulário no *site* com as devidas informações, utilizando um servidor seguro (que dispõe de recursos de criptografia para enviar os dados) ou um servidor inseguro (que não utiliza nenhum recurso de segurança para a transmissão dos dados).

Existem ainda os esquemas de pagamento digital, como o Ecash, o DigiCash, o CyberCash e o PayWord.

Ecash (Sperb et al., 1999): é o nome dado ao modelo criado pela empresa DigiCash. O Ecash é totalmente anônimo, e baseado nas técnicas de *blind signatures*. O Ecash é implementado através do uso de algoritmos de chave assimétrica RSA. Cada usuário do sistema tem seu par de chaves públicas e secretas. As operações feitas pelos compradores são executadas por um *software* denominado *cyberwallet*.

O sistema Ecash tem como seus pontos fortes a segurança e a manutenção do anonimato dos compradores. As compras com Ecash são impossíveis de serem auditadas. Através do uso de protocolos robustos e seguros baseados no sistema de criptografia assimétrica RSA, o Ecash se previne quanto ao roubo de moedas em transmissão, falsificação das transações e espionagem.

Entretanto, uma grande desvantagem e um gargalo do sistema é o armazenamento de todos os números de série de moedas que já foram usados, e sua verificação *on-line*. Se um grande número de pessoas começar a usar esse esquema, o tamanho deste banco de dados se tornará enorme e de difícil gerenciamento. Manter um banco de dados com os números de série de todas as moedas que já foram usadas alguma vez não é uma solução escalável.



**Figura 1 – Modelo do Sistema Ecash**

DigiCash (Sperb et al., 1999): é um produto da empresa DigiCash de origem holandesa, é um sistema de dinheiro digital que trabalha similarmente ao sistema de cartão telefônico.

Com este sistema, o usuário adquire CyberBucks de um banco que suporta o sistema DigiCash. Os CyberBucks podem ser comprados também por cartão de crédito. O *software* que suporta DigiCash previne que o CyberBucks seja forjado ou usado mais de uma vez. Assim como dinheiro real, o DigiCash pode ser gasto de forma anônima. Você não precisa se autenticar a fim de receber DigiCash, e seu uso não deixa registros. Além disso, o DigiCash pode ser usado para transmitir dinheiro entre pessoas, permitindo que pessoas comuns vendam mercadorias e prestem serviços através da Internet, sem envolver entidades centralizadoras, como por exemplo algum tipo de sistema bancário.

A principal desvantagem do DigiCash é que ele requer *software* específico para ser instalado nos computadores clientes e servidores.

CyberCash (Sperb et al., 1999): a idéia básica do CyberCash é ser um *gateway* que liga instituições financeiras à Internet. O CyberCash portanto não vende dinheiro digital nem possui contas dos clientes, só administra a interface entre os bancos do mundo real e as pessoas que comprem na Internet. Cada usuário do sistema tem uma *wallet* que carrega com dinheiro pego da sua conta no banco e usa, debitando a cada compra.

CyberCash oferece os serviços CyberCoins e PayNow. O PayNow é um esquema que funciona como cheques. As informações de conta corrente, agência e banco ficam na *wallet* do comprador.

PayWord (Sperb et al., 1999): o PayWord é uma proposta que tem por objetivo diminuir o custo computacional dos protocolos de micropagamentos. Seu principal objetivo é minimizar o número de operações de cifragem e decifragem com chaves assimétricas em cada pagamento, para que o impacto no tempo de resposta ao cliente seja o menor possível, assim como o custo de processamento das transações.

Esta proposta é fortemente baseada em funções *hash*. A grosso modo, as funções *hash* são 100 vezes mais rápidas que uma verificação de assinatura RSA, e 10.000 vezes mais rápidas que a geração de uma assinatura (Sperb et al., 1999). Mas outra grande característica do modelo é a possibilidade de se efetuar uma transação *off-line*. No momento do pagamento não é necessária a intervenção do banco no processo.

A questão de pagamento é indicada como responsável pela não aceitação em massa do comércio eletrônico, pois muitas pessoas ainda temem colocar seu número do cartão de crédito na rede. Por isso, várias empresas estão investindo em uma solução que possa garantir segurança nas transações.

## 2.4 AS NOVAS TENDÊNCIAS E O FUTURO DO COMÉRCIO ELETRÔNICO

Um dos maiores incentivos para a entrada do Comércio Eletrônico na Internet é o baixo custo de manutenção e a possibilidade de alcançar milhares de clientes via rede. Além das vantagens de se entrar na Internet para fins de *marketing*, propaganda e vendas, as empresas também podem se beneficiar da utilização desta para aprimorar as suas próprias redes corporativas. O resultado disso é que algumas empresas já estão reduzindo custos, alugando escritórios menores, pois muitos empregados podem trabalhar em campo ou em casa.

O usuário poderá (e já pode em alguns casos) gerenciar sua identidade financeira diante do seu computador e transferir dinheiro entre diversas contas bancárias, comprar ou vender ações da bolsa, pagar as suas contas e impostos.

Em vista disso, o mundo dos negócios vem percebendo rapidamente o imenso potencial do Comércio Eletrônico. Em função de sua utilização, uma série de conceitos estão sendo revistos e modificados. Alguns dos benefícios que vêm sendo obtidos em função da utilização das novas soluções são os seguintes (Sperb et al., 1999):

- *Marketing One-to-One*: os sistemas de Comércio Eletrônico passam a incorporar regras de negócio voltadas para a determinação do perfil dos clientes e oferecimento de promoções e produtos complementares. Através das técnicas de *Marketing One-to-One* pode-se personalizar totalmente as sessões de consulta de clientes a *sites* de Comércio Eletrônico, maximizando as possibilidades de venda e oferecendo um tratamento totalmente personalizado;
- produção “*Build to order*” e “*Mass Customization*”: os produtos oferecidos em *sites* de Comércio Eletrônico passam a ser produzidos seguindo exatamente a especificação do cliente. Através da utilização de regras de negócio voltadas para a configuração de produtos, os sistemas podem guiar o usuário durante todo o processo de configuração, possibilitando a criação de produtos totalmente personalizados;
- *Customer Care*: cuidar bem do cliente, antecipando-se com relação as suas necessidades também é um dos desafios dos sistemas de Comércio Eletrônico. Regras de negócio que automatizam a condução do relacionamento com o cliente através da emissão inteligente de *e-mails*;
- integração da cadeia de fornecimento: a integração entre os elementos de toda a cadeia de fornecimento (cliente, *sites* de Comércio Eletrônico, fornecedores, terceiros) passa a ser muito maior em relação aos métodos tradicionais. Sistemas que integram toda a cadeia de fornecimento consistem em uma nova filosofia de negócios;
- fabricação “*Just-in-time*”: os produtos passam a poder ser produzidos somente em função de pedidos específicos. Com isso, se consegue uma diminuição nos níveis de estoque. Em indústrias, a tendência é que sistemas de Comércio Eletrônico sejam conectados a *software* de Enterprise Resources Planning (ERP) para que se possa viabilizar a fabricação *just-in-time* em função de pedidos vindos da Internet;
- agentes inteligentes e regras de negócio: as regras de negócio contidas nos sistemas de Comércio Eletrônico devem poder ser rapidamente modificadas, uma vez que novos produtos e promoções sempre estarão sendo dinamicamente incluídos. Para tal, é extremamente desejável que as regras de negócio estejam separadas da lógica da aplicação. Dessa forma, a manutenção e atualização das mesmas pode ser facilmente realizada. As regras de negócio são encapsuladas em agentes inteligentes programados em Java, responsáveis por atividades como determinar o perfil do cliente, auxiliá-lo na configuração de produtos, sugerir promoções e produtos complementares, etc.;
- servidores *Web* com conteúdo dinâmico: em função do grande volume de informações que devem ser constantemente atualizadas, um dos requisitos dos sistemas de Comércio Eletrônico é que o conteúdo das páginas HTML seja modificado dinamicamente;
- sistemas de *middleware* e tecnologia *Common Object Request Broker Architecture* (CORBA): para tornar mais simples a integração entre os sistemas já existentes na organização, residentes em diferentes plataformas, programas de *middleware* que funcionam com o conceito de subscrição/publicação disparadas por eventos, são extremamente úteis na implementação dos novos sistemas de Comércio Eletrônico. Suporte à tecnologia CORBA para integração entre as

plataformas é uma tendência que se concretiza no que se refere à integração entre os ambientes.

## 2.5 CONCLUSÃO

A tecnologia de hoje já suporta as necessidades para o desenvolvimento de um *site* de Comércio Eletrônico. Muitas empresas estão desenvolvendo e outras já comercializando *software* especialmente para o Comércio Eletrônico. São programas que se preocupam com os aspectos da loja virtual, dentre os principais: a segurança e seu desempenho na efetivação da compra e na própria elaboração do *site*, facilitando e agilizando o desenvolvimento e manutenção das páginas HTML.

É impossível ficar indiferente às mudanças de cenário causadas pelo Comércio Eletrônico. Novas alternativas demandam uma nova filosofia de trabalho. Agregar tecnologia ao negócio de forma eficiente, tendo o foco ajustado na operação e no mercado é a chave do sucesso neste novo canal. No entanto, o Comércio Eletrônico não tem retorno imediato, é preciso investir e aguardar para ter os lucros desejados. Mas ignorá-lo é definitivamente parar no tempo.

### 3 SEGURANÇA EM COMÉRCIO ELETRÔNICO

O papel desempenhado pelos sistemas de pagamento nas aplicações de comércio eletrônico é de fundamental importância para a plena aceitação desta forma de relação entre empresas e consumidores. O conjunto de requisitos exigidos destes sistemas compreende, entre outros, o estabelecimento de padrões abertos para o processamento de transações de pagamento via rede, a transmissão confidencial, a autenticação das partes envolvidas e a integridade das instruções de pagamentos.

Dentre os padrões propostos mais recentemente, se destacam o Secure Electronic Transaction (SET), desenvolvido por um consórcio de empresas interessadas no desenvolvimento do Comércio Eletrônico na *Web*, e o Secure Sockets Layer (SSL).

#### 3.1 ASPECTOS DE SEGURANÇA QUE DEVEM SER CONSIDERADOS

Ao se implementar segurança para o comércio eletrônico, devem ser considerados os seguintes aspectos:

- abuso de crédito: isso pode ocorrer no caso de uma conta ser usada para fazer pagamentos sucessivos sem que a pessoa tenha a intenção de realmente pagar. Este tipo de cuidado deve ser tomado no momento do cadastramento dos clientes do *broker*;
- falsificação: a falsificação de ordens de pagamento, ou então a venda de informações falsas deve ser combatida. O primeiro caso, a falsificação de uma ordem de pagamento, pode ser facilmente resolvida através de técnicas existentes hoje de criptografia simétrica e/ou assimétrica. O segundo caso é mais complicado de se lidar. Por venda de informações falsas deve-se tratar o caso de entregar ao cliente algo diferente do que ele comprou;
- saque não autorizado: este problema está relacionado com o fato do *broker* (banco), ou um dos seus funcionários, fazer um saque não autorizado da conta de um dos clientes;
- modificação da ordem de compra: esse é o cenário onde um cliente faz uma ordem de compra com a intenção de comprar determinado produto mas a ordem é interceptada e modificada por um terceiro qualquer;
- falha na operação de débito/crédito: isso pode ocorrer se, devido a alguma falha no sistema, o *broker* fizer um débito na conta do cliente mas falhar em creditar na conta do vendedor, ou então debitar da pessoa errada ou creditar para a pessoa errada. Estes problemas são bastante sérios;
- falha na entrega: um vendedor pode receber o pagamento por determinado produto mas falhar na entrega da mercadoria vendida, seja por intenção, seja por

falha nos seus sistemas. Nesse caso deve-se ter mecanismos que registrem a atividade do sistema, de forma que se possa descobrir as verdadeiras causas do problema.

### 3.2 SET

O SET foi desenvolvido por um consórcio de empresas (Visa e Mastercard, além de GTE, IBM, Microsoft, Netscape, RSA, SAIC, Terisa Systems e VeriSign) criado em 1996, tendo como premissa a criação de uma especificação aberta de métodos para trazer segurança às transações de pagamento via cartão de crédito sobre redes abertas como a Internet (Sperb et al., 1999). Hoje a especificação do SET é coordenada pelo Secure Electronic Transaction Consortium, mais comumente conhecido como SETco.

A criação e o suporte do SET como um padrão aberto inclui em seus objetivos oferecer informações detalhadas sobre o protocolo de forma que os vários sistemas possam interoperar em nível global, utilizando sempre que possível os padrões já existentes. Outro objetivo é o de assegurar a compatibilidade com quaisquer combinações de *hardware* e *software*. O SET visa conseguir a aceitação global através da facilidade de implementação, da integração com as aplicações clientes já existentes, da minimização nas mudanças no relacionamento entre compradores, comerciantes e administradoras de cartão, e por último pela eficiência do protocolo para as instituições financeiras.

O SET apresenta sete grandes requisitos de negócio (Sperb et al., 1999):

1. oferecer confidencialidade sobre as informações relacionadas a pagamentos e pedidos de compra;
2. assegurar a integridade das informações trafegadas;
3. oferecer a autenticação de que o proprietário de cartão é um usuário legítimo de uma conta em uma administradora de cartões;
4. oferecer a autenticação de que o comerciante está habilitado para aceitar pagamentos da administradora de cartões em questão, através do seu relacionamento com uma instituição financeira;
5. assegurar o uso das melhores práticas de segurança e técnicas de projeto de sistemas, a fim de proteger todas as partes envolvidas na transação de compra eletrônica;
6. criar um protocolo que não dependa de mecanismos de segurança no nível de transporte e não previna seu uso;
7. facilitar e encorajar a interoperabilidade entre provedores de *software* e serviços de rede.

As entidades definidas na especificação do protocolo são: o proprietário do cartão (*cardholder*), a instituição financeira ligada ao proprietário do cartão (*issuer*), o comerciante (*merchant*), a instituição financeira ligada ao comerciante (*acquirer*), o gateway de pagamento (dispositivo operado por um *acquirer* ou terceiro, que processa as mensagens de pagamento dos comerciantes) e a administradora de cartões (*brand*).

As operações executadas por estas entidades determinam o estado da transação. As principais operações realizadas no SET são: a requisição da compra pelo proprietário do cartão (*Purchase Request*), a captura desta requisição pelo comerciante e a posterior submissão da requisição à instituição financeira (*Authorization Request*). O Comerciante pode optar por efetivar a autorização imediatamente, ou então armazená-la localmente para posterior submissão (*Capture Request*). Uma vez realizada a validação dos dados pela

instituição financeira, esta envia a confirmação de autorização para o comerciante (*Authorization Response*), que então confirma a compra para o proprietário do cartão (*Purchase Response*). O diagrama de estados de uma transação simples é exibido na Figura 2 (Sperb et al., 1999).

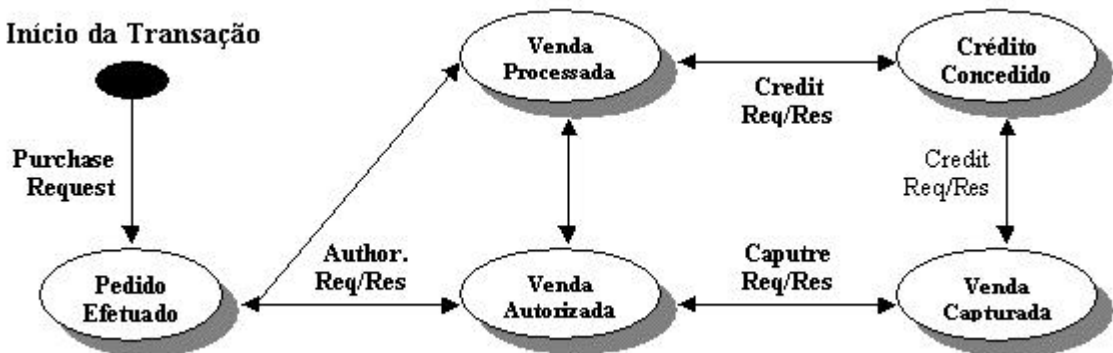


Figura 2 – Diagrama de Estados de uma Transação Simples

Algumas empresas como a Netscape e Microsoft já estão desenvolvendo sistemas para utilizar o SET. Aliado ao apoio dos bancos e das administradoras de cartões, espera-se que esse padrão universal seja aceito pela grande maioria de consumidores e comerciantes da Web, e com isso fazendo decolar a área de comércio eletrônico pela Internet.

### 3.3 SSL

A tecnologia atualmente utilizada na maioria das transações comerciais na Internet é o SSL, ou *Secure Socket Layer*. O SSL é um protocolo de segurança criado pela Netscape para permitir transações comerciais pela Web. Ele oferece três serviços de segurança (Sperb et al., 1999):

- criptografia – as informações entre as partes são todas criptografadas;
- autenticação – permite saber se as partes realmente são quem dizem que são;
- integridade – verifica se os dados foram recebidos exatamente como enviados.

Para usar o SSL, ambas (ou somente uma das partes) precisam ter certificados de identidade digital. Estes podem ser conseguidos de empresas certificadoras, como a Verisign, Certisign, Thawte, Belsign, e outras. Após as partes trocarem os seus certificados, elas escolhem o tipo de criptografia que usarão e a partir daí toda informação que entre elas trafegar irá criptografada de acordo com o combinado. É fácil saber quando se está usando SSL: de acordo com convenções, qualquer página que o utilize começará com "HTTPS:" (Secure Hyper Text Transfer Protocol (HTTPS)) em vez de "HTTP:" (Hyper Text Transfer Protocol (HTTP)). Além disso, o *browser* costuma mostrar um cadeado iluminado ou similar. O Internet Explorer e o Netscape suportam SSL. O *browser* permite até saber o nome exato da empresa ou pessoa que possui o certificado, quando o adquiriu e quando este vai expirar. Assim, o cliente tem a segurança de que estará pagando a uma empresa de verdade, não a algum criminoso que diz ser ela.

O SSL oferece proteção contra os seguintes atos (Sperb et al., 1999):

- *snooping* – trata-se de ouvir a conversa entre as duas partes. Uma vez que a conversa está criptografada, é necessário para o *hacker* saber a chave para



entender a conversa. O SSL até oferece um modo de operação sem criptografia, o que é obviamente inútil para quem deseja privacidade;

- modificações – uma vez que ninguém consegue entender a conversa entre as partes, ninguém consegue alterá-la. Alguns protocolos até permitem ao *hacker* mudar os dados, mesmo sem entendê-los. O SSL detecta mudanças nos dados até no modo, sem uso de criptografia;
- mudança de identidade – trata-se de entrar no meio da conversa dizendo ser uma das partes. Uma vez que as duas possuem uma chave que as autentica, torna-se extremamente difícil esta forma de ataque.

Porém, o SSL é ineficaz contra as seguintes formas de ataque (Sperb et al., 1999):

- análise de tráfego – a partir da quantidade e dos tipos dos dados sendo transmitidos, um hacker consegue imaginar que tipo de conversa está acontecendo. SSL não protege contra este tipo de ataque;
- negação de serviço ("*denial of service*") – um *hacker* realmente mal-intencionado pode sobrecarregar o servidor da *webstore* (loja virtual) mandando lixo para este, ou pode corromper os dados transmitidos entre as partes forçando-as a constantes retransmissões e conexões perdidas.

Deve-se observar o tipo de criptografia que será usada. Para se ter uma idéia, podemos observar o Quadro 1 (Sperb et al., 1999).

**Quadro 1 – Tipos de Criptografia em SSL**

Tamanho da Chave	Número de chaves possíveis	Tempo necessário para quebrá-la
40 bits (RC2 & RC4)	1 trilhão	3 horas e ½
56 bits (DES)	72 quatrilhões	2 meses
128 bits (RC2 & RC4)	340 decilhões	1.6 trilhões de anos

### 3.4 CONCLUSÃO

Para o desenvolvimento do protótipo da aplicação proposto neste trabalho, não foi utilizado especificamente nenhum dos dois padrões de sistema de pagamento de aplicações de comércio eletrônico apresentados neste capítulo, SET e SSL, por não ser o foco do trabalho o desenvolvimento de uma aplicação de *e-commerce* completa. Contudo, utilizando-se de um *Webserver* com tecnologia Apache, estes oferecem suporte ao padrão SSL, que é o mais utilizado atualmente como padrão de segurança nas transações eletrônicas.

## **4 AGENTES INTELIGENTES**

Com a popularização da Internet – principalmente através da World Wide Web (WWW) – e a evolução das tecnologias de comunicação, o volume e a complexidade da informação colocada a disposição pública vêm apresentando índices continuados de crescimento, trazendo a necessidade de ferramentas avançadas para coletar, filtrar, produzir e processar informação, atendendo às preferências ou especificações particulares de um usuário ou grupos de usuários. Faz-se necessário, por exemplo, o emprego de arquiteturas orientadas a agentes nessa tarefa, em função de suas características.

A pesquisa sobre agentes inteligentes já é bastante popular, objeto de estudos não somente da ciência da computação como também da psicologia, sociologia e filosofia. Grande parte deste estudo, entretanto, concentra-se na disciplina da IA, mais precisamente na IAD, onde foram realizados estudos pioneiros sobre o comportamento de agentes inteligentes, tanto tomados individualmente quanto em populações ou sociedades multi-agentes.

### **4.1 O QUE É UM AGENTE**

Segundo Wood (1994), agente é uma entidade cognitiva, ativa e autônoma, ou seja, que possui um sistema interno de tomada de decisões. Agindo sobre o mundo e sobre os outros agentes que o rodeiam, e, por fim, que é capaz de funcionar sem necessitar de algo ou de alguém para o guiar (tem mecanismos próprios de percepção do exterior).

A partir da analogia feita com agentes no mundo real, um agente é conceituado como uma entidade ativa, sempre ao lado do usuário e que possui conhecimento específico sobre um determinado domínio. Devem ser capazes de reconhecer situações em que devam se ativar de forma transparente ao usuário, utilizando suas bases de conhecimento e de seus mecanismos de raciocínio.

Deste modo, se os agentes forem implementados conforme o esperado, podem trazer grandes melhoras ao campo de interface homem-máquina, uma vez que eles podem executar cada vez mais tarefas de auxílio aos usuários.

Segundo Lingnau & Drobnik (1995), um agente é composto por:

- código: o programa que define o agente;
- atributos: descrevem as informações do agente, tais como sua origem e proprietário, requerimentos de recursos, chaves de autenticação, tarefas já executadas, etc.;
- estado: variáveis internas ao agente, que lhe permite continuar suas atividades após se mover para outro computador.

Luck et al. (1997) propõem uma hierarquia de quarta ordem, que envolve os seguintes componentes:

- entidades: a idéia básica é de que todos os componentes do mundo são entidades. Estas seriam um mecanismo de abstração muito útil, que permite diferenciá-las das demais entidades do ambiente;
- objetos: é uma entidade que possui atributos e métodos;
- agentes: é uma instância de um objeto que possui um propósito ou um conjunto de propósitos;
- agentes autônomos: conforme Davidsson (1995), podem ser conceituados como sistemas dotados da capacidade de interação independente e efetiva com o ambiente onde se encontra, regida pelo objetivo de concluir alguma tarefa.

Andrew Wood (1994) sugere inclusive algumas representações para utilização no campo de agentes inteligentes para as seguintes entidades ou processos:

- agente humano: qualquer pessoa que interage direta ou indiretamente com um sistema de computador (geralmente chamada de usuário ou usuário final);
- agente inteligente: uma entidade que comporta-se como um agente;
- ferramenta de *software* ou aplicação: qualquer *software* que não seja um agente. Inclui o software normalmente utilizado no processamento de informações, como editores de texto, planilhas e agendas eletrônicas, programas de desenho e editoração, etc. (De Carlo & Santos, 1995);
- comunicação ou interação: passagem de controle ou informações que podem ser transmitidas entre as entidades;
- observação: quando uma entidade está obtendo informações sobre o estado de outra entidade;
- interface: o ponto pelo qual duas entidades comunicam-se ou interagem. Com relação à interação homem-máquina, esta é tradicionalmente conhecida como a *interface* do usuário.

## 4.2 CARACTERÍSTICAS DE AGENTES

As características encontradas em agentes agrupam-se em torno de duas noções, respectivamente denominadas fraca e forte (Wooldridge & Jennings, 1995).

Os agentes denominados fracos, possuem uma ou mais das seguintes características:

- autonomia: os agentes funcionam sem a intervenção direta de operadores de qualquer tipo e possuem algum tipo de controle sobre suas ações e seu estado interno;
- habilidade social: os agentes interagem com outros agentes e, possivelmente, com seres humanos por meio de algum tipo de linguagem de comunicação;
- reatividade: os agentes percebem seu ambiente (que pode ser o mundo físico, um usuário através de uma interface gráfica, uma coleção de outros agentes, a Internet ou talvez tudo isto combinado) e respondem aos estímulos dele recebidos;
- iniciativa: os agentes não somente reagem ao seu ambiente, mas também devem exibir um comportamento orientado à satisfação de seus objetivos;
- continuidade temporal: os agentes são processos em execução contínua, que tanto podem estar ativos, em *foreground*, quanto adormecidos, em *background*;
- orientação a objetivos (OO): um agente deve ser capaz de lidar com tarefas complexas em alto nível. A decisão de como a tarefa deve ser segmentada em

sub-tarefas menores e em que ordem ou de que forma estas sub-tarefas devem ser executadas deve ser tomada pelo próprio agente.

Os agentes que se enquadram na noção mais forte possuem uma ou mais das seguintes propriedades:

- mobilidade: é a habilidade que um agente possui de movimentar-se em uma rede, ocupando diferentes nodos e recursos ao longo do tempo;
- benevolência: é a idéia de que o agente não possui objetivos conflitantes e que cada agente irá sempre tentar fazer o que lhe for pedido;
- racionalidade: é a hipótese de que os agentes irão agir de forma a atingir seus objetivos e não contra eles, pelo menos dentro do alcance de suas crenças;
- adaptabilidade: um agente deve ser capaz de adaptar-se aos hábitos, métodos de trabalho e preferências de seus usuários;
- colaboração: um agente não deve aceitar (e executar) instruções impensadamente. Deve levar em conta que seres humanos cometem erros. Assim deve também ser capaz de recusar ordens que, por exemplo, produzissem uma sobrecarga inaceitável na rede ou que ocasionassem danos a outros usuários.

Já, segundo Davidsson (1995), de um agente de *software* são esperadas pelo menos três características, conforme mostra a Figura 3:

- adaptação: os agentes se adaptam aos seus ambientes e usuários, e aprendem com suas experiências;
- cooperação: agentes usam linguagens e protocolo padrão para cooperar e colaborar para alcançar metas comuns;
- autonomia: agentes agem autonomamente para seguir sua agenda.

A mobilidade, embora não seja rigorosamente necessária, constitui-se em uma característica também importante para um agente.

### Características de Agentes

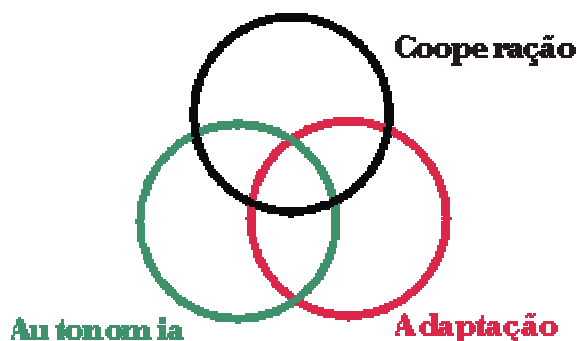


Figura 3 – Características de Agentes

## 4.3 TIPOS DE AGENTES

Os agentes inteligentes podem ser classificados em várias categorias, de acordo com suas características. A seguinte classificação é apresentada por (Wood, 1994):

- conselheiro: oferece ajuda e treinamento. Ensina os passos iniciais para usar um determinado sistema. Fornece suporte contínuo, observando todas as ações do usuário, as quais ele pode interceptar e pedir confirmação. Pode ser consultado

para mostrar como executar uma atividade particular, ou então, sugerir métodos alternativos e mais rápidos para executá-la;

- **guia:** ajuda a navegação em bancos de dados e hipermídia. Classifica, recupera e filtra grandes quantidades de informações, apresentando somente os dados relevantes e importantes aos usuários, no formato personalizado. Fornece caminhos apropriados para o usuário navegar pelo banco de dados, e auxiliá-lo caso se sinta "perdido";
- **empregado:** executa as atividades tediosas ou repetitivas. Atividades são executadas imediatamente, e algum tipo de *feedback* pode ser fornecido tanto pelo usuário como pelo próprio agente;
- **representante:** trabalha na ausência do usuário. De certa forma, seria parecido ao agente Empregado citado anteriormente, exceto pelo fato de que as atividades não precisam ser imediatamente executadas ou então, são executadas somente após eventos específicos. Por exemplo, pode fazer *backups* de arquivos de madrugada ou fazer pedidos de compras, caso algum produto atinja o limite mínimo no estoque;
- **comunicador:** trabalha com outros usuários e seus agentes, para assim, conseguir executar as atividades às quais foi designado. Pode, por exemplo, organizar reuniões de recursos e pessoas. Ou então, pode reunir um grupo de agentes para que assim possam executar uma atividade mais complexa.

#### 4.4 ABORDAGENS PARA A CONSTRUÇÃO DE AGENTES

Dois problemas principais podem ser identificados na relação entre agentes autônomos e usuários (Aparicio et al., 1995):

- **competência:** como fazer um agente adquirir conhecimento suficiente para decidir quando agir e qual o tipo de ação ele deve realizar;
- **confiança:** como pode-se conseguir com que o usuário sinta-se confortável, ao delegar atividades a um agente.

Estes problemas devem ser tratados como aspectos eminentes na construção de agentes. Devem ser consideradas duas abordagens básicas para a construção de agentes (Aparicio et al., 1995):

- a primeira abordagem consiste em tornar o próprio programa do usuário em um agente inteligente. Um exemplo pode ser um sistema constituído por agentes semi-autônomos, os quais utilizam uma coleção de regras programadas pelo usuário para processar as informações relacionadas a uma atividade particular. O grande problema dessa abordagem é que ela não trata do critério de competência, de um modo satisfatório. A abordagem requer muita perspicácia, compreensão e esforço por parte do usuário. É ele quem tem de reconhecer a oportunidade para empregar um agente, tomar a iniciativa de criá-lo, fornecer ao agente um conhecimento explícito, e manter constantemente atualizadas todas as regras do agente, na medida em que seus hábitos de trabalho ou interesses possam mudar. Nesta abordagem praticamente não há o problema da confiança no agente, uma vez que o usuário (normalmente) confia em suas próprias habilidades de programação.
- a segunda abordagem ou "abordagem baseada no conhecimento", consiste em dotar um agente com um extensivo conhecimento específico a um determinado domínio, sobre a aplicação e sobre o usuário (chamado de modelo do domínio e

modelo do usuário, respectivamente). Em tempo de execução, o agente usa seu conhecimento para reconhecer os planos do usuário e encontrar oportunidades para contribuir com ele. Porém, ambos os critérios de competência e confiança, constituem-se como problemas nesta abordagem. A competência pode ser atingida, porém com altos custos, pois a construção de agentes baseados em conhecimento necessita de uma grande quantidade de trabalho e esforço por parte do engenheiro do conhecimento, para conseguir extrair conhecimento suficiente sobre a aplicação e o domínio. Por outro lado, pouco deste conhecimento ou mesmo da arquitetura de controle do agente podem ser usados na construção de agentes específicos a outras aplicações. O segundo problema é que o conhecimento do agente é fixo, ou seja, não pode ser customizado de acordo com os hábitos e preferências de usuários individuais. Outro problema é o de confiança. Provavelmente, não é uma boa idéia fornecer ao usuário um agente que é muito sofisticado, qualificado e autônomo desde o início do seu uso. Isto pode acarretar no usuário um sentimento de perda de controle e de compreensão da atividade a ser realizada. Uma vez que o agente foi programado por alguma outra pessoa, o usuário pode não compreender as limitações do agente e o modo como ele trabalha, entre outras coisas.

#### 4.5 AGÊNCIA E INTELIGÊNCIA

O grau de autonomia e autoridade com que um agente é investido é denominado a sua agência. Este grau pode ser medido, ao menos qualitativamente, pela natureza da interação entre o agente e outras entidades no sistema em que ele opera. O grau de agência é reforçado se um agente representa um usuário de alguma forma. Agentes mais avançados conseguiriam interagir com outras entidades, tais como dados, aplicações ou serviços e ainda colaborar e negociar com outros agentes.

Pode-se entretanto considerar a inteligência de um agente como sendo o seu grau de raciocínio e a capacidade de aprender padrões de comportamento, aceitar as declarações do usuário e executar as tarefas que lhe são delegadas (Aparicio et al., 1995). Níveis mais altos de inteligência incluiriam um modelo do usuário ou alguma outra forma de entendimento e raciocínio sobre o que o usuário deseja que seja feito e o planejamento dos meios necessários para atingir tais objetivos. Mais além na escala da inteligência estariam os agentes que aprendem e se adaptam a seu ambiente, seja em termos dos objetivos do usuário, seja em função dos recursos que encontram disponíveis. Tais agentes poderiam descobrir sozinhos novos relacionamentos, conexões ou conceitos e explorá-los na antecipação e satisfação das necessidades de seus usuários.

#### 4.6 APLICAÇÕES DE AGENTES INTELIGENTES

Segundo Hermans (1996), alguns pesquisadores têm focalizado aplicações bastante elementares, como por exemplo:

- agentes que parcial ou totalmente automatizam caixas de correio eletrônico;
- agentes que filtram ou pesquisam listas de artigos do *News* em busca de informação que possa ser interessante para seus usuários;
- agentes que agendam, registram e executam o *follow-up* de reuniões ou conferências *on-line*.

São identificadas oito principais áreas onde se concentra, ou deverá se concentrar, num futuro próximo, o desenvolvimento de tecnologias de agentes (Aparicio et al., 1995):

1. gerenciamento de sistemas e redes: esta foi uma das primeiras áreas de aplicações a empregar a tecnologia de agentes inteligentes. O uso crescente de arquiteturas cliente/servidor elevou a complexidade dos sistemas em operação, principalmente em redes locais. As arquiteturas de agentes empregadas são em sua maioria não-inteligentes, entretanto sistemas inteligentes encontrariam muita aplicação em níveis mais altos de abstração, por exemplo, aprendendo a reagir a determinados padrões no comportamento dos sistemas. Além disso poderiam ser também empregados no gerenciamento dinâmico de grandes configurações;
2. acesso e gerenciamento móvel: à medida em que a computação vai se tornando cada vez mais distribuída e difusa, surge a necessidade dos usuários empregarem tecnologias móveis, tais como comunicações sem fio. Os usuários desejam não apenas conectar-se a partir de qualquer lugar e também não sofrer as restrições de largura de faixa por vezes impostas pelas telecomunicações. A manipulação inteligente da transmissão da informação é uma área promissora para o emprego de agentes;
3. correio eletrônico e troca de mensagens: agentes vem sendo empregados nesta área já há algum tempo, priorizando mensagens e automaticamente organizando o correio eletrônico de seus usuários. Os agentes inteligentes podem facilitar todas essas funções; por exemplo, por meio de regras – que poderiam ser inclusive deduzidas a partir de padrões de comportamento observados em seus usuários;
4. acesso e gerenciamento da informação: esta é uma área de grande atividade, tendo em vista a rápida popularização da Internet e a explosão da informação disponível a seus usuários. Aqui agentes inteligentes podem ser empregados não apenas na pesquisa e filtragem de informação, mas também na categorização, priorização, disseminação seletiva, anotação e no compartilhamento cooperativo de documentos e informações;
5. colaboração: é uma área em rápido crescimento onde os usuários trabalham juntos em documentos compartilhados na rede. Aqui é necessário não apenas uma infraestrutura que permitam compartilhamento robusto e escalável de dados e outros recursos, mas também funções que permitam gerenciar equipes e o produto de seu trabalho. O exemplo mais conhecido de aplicações deste tipo é o Lotus Notes;
6. gerenciamento administrativo: inclui o gerenciamento de fluxos de trabalho e também áreas de suporte, como a integração entre computadores e serviços de telefonia, por exemplo, onde processos são definidos e então automatizados. Nestas áreas, os usuários necessitam não somente tornar os processos mais eficientes, como também reduzir o custo dos agentes humanos. Agentes inteligentes poderiam ser empregados aqui para identificar e automatizar processos de possível interesse do usuário;
7. comércio eletrônico: esta é uma área em acelerado crescimento, que é alimentada pela popularização da Internet. Os consumidores, em busca de produtos e serviços necessitam de informações sobre o que estão comprando (inclusive especificações técnicas, configurações viáveis, etc.). Os comerciantes necessitam localizar (e atrair) clientes, oferecer suporte especializado sobre seus produtos, controlar e realizar o *follow-up* de suas vendas, etc. Tanto os consumidores quanto os comerciantes necessitam automatizar sua participação neste "mercado eletrônico". Agentes inteligentes poderiam ser empregados aqui de diversos modos. Por exemplo, poderiam "ir às compras" para seus usuários, coletar especificações de

um determinado produto e retornar com sugestões de compras que atendessem descrições recebidas. Também poderiam atuar como assistentes de vendas, fornecendo aconselhamento sobre os produtos e tentando solucionar possíveis problemas e dificuldades do usuário;

8. interfaces inteligentes: apesar da disseminação de interfaces gráficas, para muitas pessoas os computadores continuam difíceis de usar. Por outro lado, à medida em que a população de usuários cresce e se diversifica, as interfaces se tornam mais e mais complexas para acomodar hábitos e preferências variadas. Agentes de interface inteligentes poderiam, por exemplo monitorar as ações do usuário para desenvolver um modelo com suas habilidades e automaticamente ajudá-lo quando os problemas surgirem.

#### **4.7 ACESSO À INFORMAÇÃO NA INTERNET**

O armazenamento e recuperação de dados por pessoas e organizações passou por modificações substanciais ao longo dos últimos anos até a recente popularização da Internet, que transformou a visão clássica de base de dados, de uma coleção centralizada de dados homogêneos em coleções de informação distribuídas e heterogêneas. Um dado não é mais uma entidade estruturada e pode ser quase qualquer tipo de objeto representável, por exemplo um desenho, um som, um texto, um registro, etc. A disseminação das redes de computadores tornou informações desse tipo disponíveis a qualquer um e qualquer um pode representar simultaneamente o papel de produtor ou consumidor dessa informação (Brown et al., 1995).

Em consequência, um número muito grande de repositórios de informações se encontra à disposição de todos, ainda que apenas uma quantidade restrita dessa informação seja de interesse para um determinado usuário. Além disso, o mercado da informação anteriormente dirigido por um grupo relativamente pequeno e facilmente identificável de fornecedores cresceu de tal forma que se torna muito difícil formar um quadro razoavelmente claro de quem são os atuais fornecedores e que informação é oferecida sob quais condições. Além disso, a informação passa a representar um papel cada vez maior na vida humana, na medida em que esta se desloca rumo à sociedade da informação, onde este recurso se torna um instrumento, ou ferramenta capaz de solucionar um grande número de problemas.

A disponibilidade de informações na Internet, que à primeira vista aparenta ser o seu maior apelo, é ao mesmo tempo o seu maior ponto fraco: a quantidade de informação disponível é tamanha que, muitas vezes, a tarefa de localizar uma informação específica se torna inexecutável. A informação procurada (muito provavelmente) se encontra disponível em algum lugar, mas pode ser recuperada apenas em parte, ou a fonte não é confiável, ou até mesmo algumas vezes nada pode ser diretamente obtido. Os métodos de pesquisa convencionais não parecem capazes de lidar com tais problemas. A questão é que tais métodos partem do princípio de que se conhece de antemão qual informação está disponível (e qual não está) e onde exatamente esta informação pode ser encontrada. Para tornar isto possível, índices gigantescos são construídos sobre bases de dados alimentando grandes sistemas de informações. Por meio de tais índices, é possível saber a qualquer momento que informação está disponível e onde tal informação se encontra. Esta estratégia, entretanto, falha por completo ao lidar com a complexidade da Internet.

Isto se deve (Hermans, 1996)):

- à natureza dinâmica da Internet, cujo crescimento e desenvolvimento não obedece a qualquer supervisão;



- à natureza dinâmica da informação na Internet, onde o que não está disponível hoje pode vir a sê-lo amanhã ou, ao contrário;
- à heterogeneidade da informação e dos serviços de informação disponíveis na Internet, oferecidos nos mais diversos formatos e de muitas diferentes maneiras.

Como alternativa aos métodos de pesquisa convencionais, utilizam-se programas auxiliares inteligentes, chamados agentes autônomos. São mais sofisticados do que os programas semi-inteligentes, pois eles percorrem a Internet com alguma missão específica. Por exemplo, um agente autônomo poderá "caçar" todas as noites na Internet, para capturar todas as novidades que satisfaçam seu perfil de interesse, e montar automaticamente uma página com as principais notícias e endereços, que poderão ser lidos no dia seguinte. Outro exemplo, é o agente realizar uma busca na Internet por um determinado produto, fazer uma lista de todos os que achou e de seus respectivos preços, e decidir automaticamente qual é o mais barato deles.

Os programas que utilizam agentes autônomos já existem, estão operacionais na Internet e são chamados de *softbots* (*software* robô), ou *web robots*, ou simplesmente *webbots*. Como exemplo, o índice de busca do Google utiliza essa tecnologia para, todos os dias, construir automaticamente um índice de muitos *links* existentes na WWW.

## 4.8 AGENTES DE INFORMAÇÃO NA PRÁTICA

Deve-se notar que os agentes de informação podem apresentar características variadas, sendo por exemplo móveis ou estáticos, não-cooperativos ou sociais, capazes ou não de aprender, etc. Não há, portanto, um padrão no seu modo de operação. Os agentes de informação na Internet podem ser móveis – isto é, podem percorrer a rede, atravessando diferentes plataformas, coletando informação e relatando o resultado obtido ao seu local de origem, ou (mais provavelmente) estáticos, controlando todo o processo de pesquisa a partir de um único local.

Um agente de informações, pode ser associado a algum tipo de ferramenta de indexação, tal como um *spider*. Um *spider* é um indexador capaz de percorrer a WWW e armazenar sua topologia em um banco de dados e o índice dos *Uniform Resource Locators* (URLs) percorridos em *Wide Area Information Service* (WAIS). O agente de informações estático, ao ser requisitado a coletar informações sobre algum assunto, emite diversas consultas a vários mecanismos de pesquisa na *Web*. Parte da pesquisa, entretanto pode ser feita localmente por meio de uma *cache*. A informação coletada é posteriormente encaminhada ao usuário.

## 5 DESENVOLVENDO UM *SITE* PARA INTERNET

A Internet, nos dias de hoje, está além de uma simples diversão. Ela pode ser utilizada para que as empresas divulguem seus produtos e/ou a utilizem como base operacional dos seus sistemas de informação. A principal vantagem é que o aplicativo pode ser acessado não só dentro da rede da empresa, como em qualquer parte do mundo.

Para desenvolver estes sistemas é necessária a adoção de algumas linguagens, ferramentas e tecnologias. São várias as opções que existem no mercado de desenvolvimento. A diferença de escolha depende principalmente do objetivo final do negócio e da infraestrutura atual/projetada.

### 5.1 HTML

Apesar de uma aparente sofisticação, as páginas WWW não passam de documentos de texto simples. Podem ser produzidas com qualquer editor de texto. A diferença é que as páginas WWW contêm algumas marcas especiais para determinar o papel de cada elemento dentro do texto. Alguns elementos são marcados como títulos, outros como parágrafos. As marcações são usadas também para indicar os *links* que levam a outros documentos na rede. Essas marcas são chamadas de *tags* e são especificações da linguagem HTML.

O primeiro conceito que deve-se ter em mente ao projetar páginas WWW com HTML, é que esta linguagem não foi criada para controlar a aparência dos documentos, ao contrário dos processadores de texto e programas de *layout* de página. Os *tags* de HTML apenas informam ao navegador o que são os elementos que estão na página. Eles dizem, por exemplo, que um determinado trecho é o título principal do documento e outro é um item de lista. A formatação do trecho é deixada para o navegador. Cada navegador mostra a página de uma forma um pouco diferente, o que dificulta o trabalho de programação visual na WWW. Quando um cliente não entende uma determinada marcação, ele simplesmente a ignora. Ao criar um documento, é importante testá-lo com vários clientes. Para complicar ainda mais, cada usuário pode modificar a configuração padrão de seu navegador para que o seu programa mostre o texto na fonte (tipo de caractere) que quiser.

Em compensação, é muito simples criar uma página para colocar na Internet, ainda mais com os vários editores HTML disponíveis para auxiliar na sua confecção.

As características gerais do HTML são:

- documentos HTML são arquivos escritos em American Standard Code for Information Interchange (ASCII) (texto);
- podem ser criados em qualquer editor de texto (vi, emacs, edit, notepad);
- existem editores específicos para várias plataformas;

- existem conversores de vários formatos, por exemplo, doc para html;
- a unidade mínima de informação é a “página”.

## 5.2 CGI

CGI é um padrão, definindo a associação entre aplicações e servidores *Web*. Através dele, um *site* pode oferecer conteúdo definido de acordo com o usuário. Um exemplo de CGI é um *search engine*, onde você preenche os dados da sua procura, e o CGI, após ler e processar os dados, retorna os resultados da busca ao usuário.

Programas ou *scripts* em CGI permitem adicionar à uma página de WWW real capacidade de interagir com o usuário. A possibilidade de geração de páginas *on-line* e de procura em banco de dados são apenas duas de muitas possibilidades.

Uma página HTML, inicialmente, possui informações de como chamar um CGI, que pode ser por um *hyperlink* direto, ou por meio de um formulário, que é uma seção do documento HTML onde definem-se campos de entrada, e um destino para estes campos, que é o endereço do programa CGI. Após interpretado pelo *browser*, o *hyperlink* ou o formulário aparecem ao usuário. No último caso, além de requisitá-lo, o usuário ainda pode definir valores para os campos. Então, o *browser* chama o servidor *Web*, através do CGI, enviando os valores dos campos. O servidor *Web*, de posse desses dados, chama o programa CGI e transmite os dados. O programa CGI interpreta os dados, processando-os, e retornando-os da seguinte forma: primeiro, ele retorna o *status* do pedido, com o tipo de arquivo que será enviado, códigos de erro, ou redirecionando para outra página. Depois disso, ele retorna os dados propriamente dito. O servidor *Web* recebe a saída do CGI e retorna ao usuário, terminando a transação.

## 5.3 JAVA

Java é uma poderosa linguagem de programação Orientada a Objetos. Foi projetada para resolver os problemas da área de programação cliente servidor. A linguagem Java foi criada de forma a poder ser executada nas diversas plataformas de trabalho diferentes disponíveis. Diferente de outras linguagens de programação, Java, quando interpretada por *browsers*, não tem permissão de gravar dados no disco rígido, sendo portanto uma linguagem segura (isto é, os *applets* escritos em Java podem ser executados sem riscos para as informações contidas no computador) (Macinta, 1997).

Java foi criada como parte de um grande projeto da Sun Microsystems, cuja missão era desenvolver aplicativos complexos e avançados para aplicação em pequenos dispositivos eletrônicos. Esses dispositivos são sistemas portáteis, distribuídos, confiáveis e incorporados em tempo real, dirigidos aos consumidores em geral. As primeiras tentativas de criação de tais dispositivos receberam nome de Portable Data Assistents (PDAs).

A primeira tentativa da Sun com esses aplicativos deveria ter sido implementada em C++. Mas em consequência de má compilações com compiladores e uma lista cada vez maior de problemas com o C++ – principalmente estouro de memória e vários problemas de herança – a Sun desistiu do C++ e implementou uma nova linguagem, chamada Java. Desde o seu início, surgiram muitas outras aplicações para ela. Java tem a capacidade de criar programas auxiliares, que podem ser usados para montar documentos HTML particularmente interessantes.

A arquitetura da linguagem Java não é nem radical nem especialmente nova. Resumindo, os aplicativos em Java são compilados em *bytecodes* independente de arquitetura. Esses *bytecodes* podem então serem executados em qualquer plataforma que suporte um interpretador Java. Java requer somente um código fonte e um *bytecode* e, mesmo assim, é capaz de funcionar em diversas plataformas.

### 5.3.1 Características da Linguagem Java

A linguagem Java possui as seguintes características (Macinta, 1997):

- Java é orientada a objetos: a linguagem Java é um membro do paradigma orientado a objetos das linguagens de programação. As linguagens que aceitam este paradigma, como Java e C++, seguem a mesma filosofia básica, mas diferem em sintaxe e estilo. As linguagens orientadas a objetos oferecem muitas vantagens sobre as linguagens procedurais tradicionais. Como os objetos encapsulam dados e funções relacionados em unidades coesas, é fácil localizar dependências de dados, isolar efeitos de alterações e realizar outras atividades de manutenção, e talvez o mais importante: as linguagens OO facilitam a reutilização.
- Java é distribuída: distribuição de informações para compartilhamento e trabalho conjunto, com a distribuição de carga de trabalho do processamento, é uma característica essencial dos aplicativos cliente/servidor. Felizmente para os programadores Java, há uma biblioteca de procedimentos Transmission Control Protocol/Internet Protocol (TCP/IP) incluída nos códigos-fonte e de distribuição binária do Java. Isso facilita aos programadores o acesso remoto às informações, usando protocolos como HTTP e File Transfer Protocol (FTP).
- Java é uma linguagem ao mesmo tempo compilada e interpretada: os programas em Java são compilados em formato binário de *bytecodes*, que então são interpretados por um ambiente de execução do Java específico da plataforma em questão.
- Java é independente de arquitetura: a neutralidade desta linguagem em relação à arquitetura é fascinante, mas não se trata de um conceito novo. Derivado da natureza distribuída de cliente/servidor, um importante recurso de projeto do Java é o suporte a cliente e servidores em configurações heterogêneas de rede. O método escolhido para atingir esse objetivo foi uma representação binária de arquitetura neutra para os programas em Java.
- Java é portátil: a característica de neutralidade da arquitetura Java é o grande motivo pelo qual os programas em Java são portáteis. Outro aspecto da portabilidade envolve a estrutura ou os tipos de dados inerentes da linguagem, como inteiro, string e ponto flutuante. O compilador Java foi escrito com o próprio Java, enquanto seu ambiente de tempo de execução foi escrito em *American National Standards Institute* (ANSI) C, e tem uma interface de portabilidade bem definida e concisa.
- Java é multitarefa: os objetos binários de *bytecodes* do Java são formados por seqüências de execução múltiplas e simultâneas. Essas seqüências são conhecidas como contextos de execução ou processos leves. As linguagens C e C++ são membros de um paradigma de execução em seqüência única, por não oferecerem suporte a seqüências no nível de linguagem. Java, no entanto, oferece suporte no nível de linguagem para multitarefa, resultando em uma abordagem de programação mais poderosa e de múltiplas facetas.

- Java é dinâmica: o projeto dinâmico permite que os programas Java se adaptem aos ambientes computacionais mutantes. Por exemplo, a maior parte dos desenvolvimentos típicos em C++ se baseia em bibliotecas de classe, que podem ser de propriedade e desenvolvida por terceiros. Muitas bibliotecas de terceiros – como as distribuídas com sistemas operacionais ou sistemas de janelas – são *linkeditadas* dinamicamente e vendidas ou distribuídas separadamente dos aplicativos que delas dependam. Quando essas bibliotecas são atualizadas, os aplicativos que dependerem dela poderão apresentar problemas, até que sejam recompilados e redistribuídos. Isso adiciona mais um custo à manutenção do *software*. Java evita esse problema atrasando a união dos módulos, pois compila separadamente e integra na execução. Isso permite que os programadores tirem total proveito da OO. É possível adicionar novos métodos e variáveis de instâncias em classes de bibliotecas, sem causar problemas aos programas, aplicativos ou clientes já existentes.
- Java é robusta: quanto mais robusto um aplicativo, mais confiável ele será. Isso é desejável tanto para os desenvolvedores de *software* quanto aos consumidores. A maioria das linguagens OO, como C++ e Java, possuem tipos bastante fortes. Isso significa que a maior parte da verificação de tipos de dados é realizada em tempo de compilação, e não em tempo de execução. Isso evita muitos erros e condições aleatórias nos aplicativos. Java, ao contrário de C++, exige declarações explícitas de métodos, o que aumenta a confiabilidade dos aplicativos.
- Java é segura: como Java foi criada para ambientes de rede, os recursos de segurança receberam muita atenção. Por exemplo, se for executado um binário transferido por *download* da rede, o mesmo poderá estar infectado por vírus. Os aplicativos Java apresentam garantia de resistência contra vírus e de que não são infectados por vírus, pois não são capazes de acessar *heaps*, *stacks* ou memória do sistema. Na linguagem Java, a autenticação do usuário é implementada com um método de chave pública de criptografia. Isso impede de maneira eficaz que *hackers* e *crakers* examinem informações protegidas como nomes e senhas de contas.
- Java é simples: um dos principais objetivos do projeto desta linguagem foi criar uma linguagem o mais próxima possível do C++, para garantir sua rápida aceitação no mundo do desenvolvimento OO. Outro objetivo do seu projeto foi eliminar os recursos obscuros e danosos do C++, que fugiam à compreensão e aumentavam a confusão que poderia ocorrer durante as fases de desenvolvimento, implementação e manutenção do *software*. Java é simples porque é pequena. O interpretador básico de Java ocupa aproximadamente 40k de Random Access Memory (RAM), excluindo-se o suporte a multitarefa e as bibliotecas padrão, que ocupam outros 175k. Mesmo a memória combinada de todos esses elementos é insignificante, se comparada a outras linguagens e ambientes de programação.
- Java oferece alto desempenho: há muitas situações em que a interpretação de objetos de *bytecodes* proporciona desempenho aceitável. Mas outras circunstâncias exigem desempenhos mais altos. O Java concilia tudo isso oferecendo a tradução dos *bytecodes* para o código de máquina nativo em tempo de execução. O alto desempenho permite a implementação de seus aplicativos *Web* em Java, na forma de programas pequenos e rápidos, que podem ampliar significativamente os recursos tanto do cliente quanto do servidor.

## 5.4 JAVASCRIPT

JavaScript é uma linguagem de propriedade da Netscape que é executada juntamente com HTML visando incrementar a criação de *home pages* tornando-as mais interativas. Com JavaScript é possível, por exemplo, responder muito facilmente a eventos iniciados pelo usuário. Alguns dos efeitos que agora podem ser feitos com JavaScript, há algum tempo atrás só eram possíveis com CGI.

JavaScript permite injetar lógica em páginas escritas em HTML. Os parágrafos de lógica do JavaScript podem estar em qualquer parte do documento ou atrelados a ocorrência de eventos. Os parágrafos atrelados a eventos são executados apenas quando o evento ocorre, os demais são executados na sequência em que aparecem na página (documento).

É uma adaptação da linguagem Java, de forma a ser interpretada pelo *browser*, sem necessidade de ser compilada. A linguagem JavaScript é mais simples que a Java, permitindo que as pessoas com menos experiência em programação tenham a possibilidade de criar páginas interativas.

### 5.4.1 Diferença entre Java e JavaScript

Ainda que os nomes sejam similares, Java não é a mesma coisa que JavaScript. Essas são duas técnicas diferentes de programação na Internet. Java é uma linguagem de programação. JavaScript é uma linguagem de *scripting* (tal como diz o nome). A diferença é que se pode criar programas reais com Java. Em alguns casos, porém, o que se quer é apenas criar um efeito chamativo, sem se importar com qualquer programa real. Assim, JavaScript foi pensado como algo fácil de se compreender e de se usar. Os autores de JavaScript não têm que se importar com programação. Poderia-se até dizer que JavaScript é uma extensão do HTML, ao invés de uma linguagem de computador separada. Naturalmente essa não é uma definição oficial, porém torna mais compreensível a diferença entre Java e JavaScript.

## 5.5 ASP

ASP são páginas *Web* que possuem conteúdo dinâmico, interativo e de alto desempenho. Estas páginas consistem em combinações de *scripts* e *tags* HTML. Os *scripts* são executados no servidor e, raramente, no cliente. É o próprio servidor que transforma os *scripts* em HTML padrão, fazendo com que qualquer *browser* do mercado seja capaz de acessar um *site* que usa ASP.

ASP veio para complementar a linguagem HTML. Ela traz para a linguagem HTML, através dos *scripts*, todo o poder do acesso a banco de dados, do acesso a arquivos texto, da captação de informações de formulário, da captação de informações sobre o visitante e sobre o servidor, do uso de variáveis, *loops*, etc.

Entre os recursos que podem ser implementados via ASP, citam-se:

- programação em VBScript ou JScript;
- acesso a banco de dados;
- sessões (persistência de informações no servidor).

As páginas ASP são, na verdade, uma junção de programação em VBScript e Objetos ActiveX. VBScript é uma linguagem criada a partir do Visual Basic, mas com algumas limitações, por motivos de segurança. Objetos ActiveX são funções prontas, já existentes no servidor. Essas funções captam os parâmetros de entrada dos dados, manipula-os de acordo

com a sua função e então envia-os para a saída. Um exemplo de ActiveX é o objeto ASPMail, o qual capta os dados de entrada (nome, *e-mail*, corpo da mensagem, etc.), cria um *e-mail* com esses dados e depois envia o *e-mail*.

ASP surgiu juntamente com o lançamento do Internet Information Server 3.0. Esta é uma solução Microsoft, que exige que o seu servidor execute um sistema operacional da Microsoft (Windows 9x ou NT).

Os seguintes servidores suportam o uso de páginas ASP:

- Microsoft Internet Information Server versão 3.0 no Windows NT Server;
- Microsoft Peer Web Services versão 3.0 no Windows NT Workstation;
- Microsoft Personal Web Server no Windows 95 ou Windows 98.

A grande vantagem, porém, é que existe esta exigência apenas do lado do servidor. No lado do cliente, pode-se utilizar qualquer *browser*, mesmo os que não suportam VBScript.

Aplicações ASP podem armazenar dados que são mantidos durante toda uma sessão. Desta forma, um usuário pode fornecer seu nome somente uma vez em uma página e as demais páginas podem obter este dado automaticamente. Este recurso é ideal para aplicações de venda pela Internet.

Uma aplicação ASP pode ser usada com qualquer banco de dados compatível com Open Data Base Connectivity (ODBC). Isto inclui dados do Access, Microsoft SQL Server, Oracle, Sybase, Informix, DB2, entre outros.

### 5.5.1 Vantagens das Páginas ASP

As páginas ASP apresentam as seguintes vantagens:

- independência do *browser*: ASP pode rodar páginas complexas no servidor e enviar somente os resultados para o cliente;
- acesso à Banco de Dados: permite visualizar, atualizar e adicionar informações nos servidores SQL;
- segurança do código fonte: como o servidor retorna somente o resultado HTML, o código fonte (lógica) fica preservado;
- linguagens: ASP pode utilizar comandos do VBScript, JavaScript e HTML, além de outras linguagens *script*.

### 5.5.2 Requisitos para Utilizar ASP

Para utilizar ASP em *sites*, é necessário atender aos requisitos abaixo:

- servidor que hospedará as páginas deve rodar Windows 95, Windows 98, Windows NT, ou então, Windows 2000;
- servidor deve possuir instalado o Microsoft Internet Information Server (IIS), o Peer Web Services (PWS) ou o Personal Web Server (PWS, também);
- para utilizar os recursos de acesso a bancos de dados, os dados devem estar acessíveis através de ODBC. É necessário que um *driver* de ODBC esteja instalado e funcionando no servidor;
- para utilizar os recursos de acesso a bancos de dados, é necessário, também, a existência de uma "Fonte de Dados" ODBC para este banco de dados. Isto é feito usando a opção "ODBC" do "Painel de Controle" do Windows. Para o SQL

Server, é necessário criar um usuário com direito para acessar este banco de dados;

- criar um diretório para guardar as páginas ASP no seu servidor. Incluir este diretório na lista do IIS/PWS, dando direito de "*Execute/Execução*". As páginas ASP só podem ser rodadas a partir de um diretório com o direito de "*Execução*". Não habilitar a opção de "*Read/Leitura*", para aumentar a segurança neste diretório.

### 5.5.3 ASP e Outras Tecnologias

Abaixo, comparação de ASP com outras tecnologias:

- ASP x CGI: ASP traz todos os recursos de aplicações CGI de uma forma mais fácil e mais robusta. Com ASP, é bem mais fácil criar conexões entre o *browser* e os dados em formatos normalmente incompatíveis com HTML, como bancos de dados. ASP é mais robusto por não criar um processo no servidor para cada pedido do usuário, como acontece com o CGI. Usando ASP ao invés de CGI, um servidor pode atender a um grande número de pedidos de usuários de forma mais rápida e usando menos memória. Além disso, criar páginas ASP é em geral muito mais fácil do que criar aplicações CGI;
- ASP x Perl: Perl é apenas uma linguagem *script* e não uma ferramenta de desenvolvimento. Usando ASP, tem-se objetos predefinidos para criar aplicações complexas, como os que permitem o acesso a bancos de dados ou o uso de sessões. Além do mais, ASP pode utilizar Perl como linguagem *script*, bastando para isso, usar *plug-ins* ActiveX de terceiros.

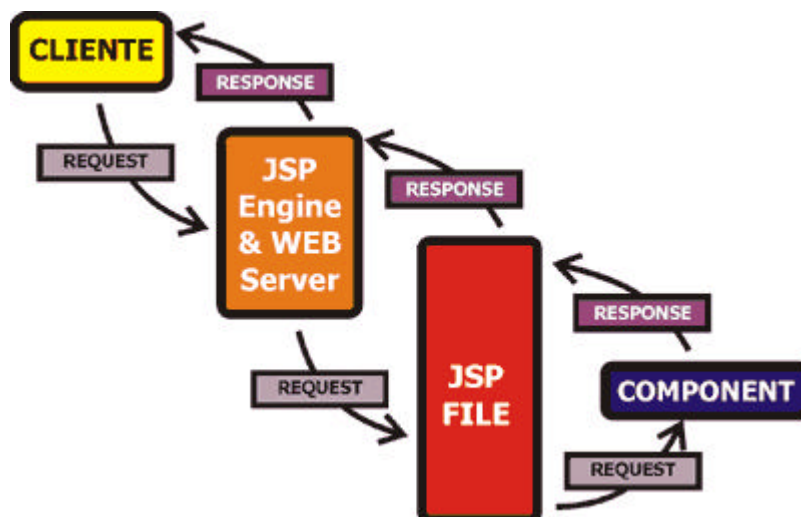
## 5.6 JSP

JSP é uma tecnologia para desenvolvimento de aplicações *Web* semelhante ao ASP, porém tem a vantagem da portabilidade de plataforma podendo ser executado em outros Sistemas Operacionais, além dos da Microsoft. Ela permite que ao desenvolvedor de *sites* produzir aplicações que permitam o acesso a banco de dados, o acesso a arquivos texto, a captação de informações a partir de formulários, a captação de informações sobre o visitante e sobre o servidor, o uso de variáveis e loops entre outras coisas (Field & Kolb, 2000).

Comparando *servlets* com JSP, este não oferece nada que não se possa conseguir com os *servlets* puros. O JSP, entretanto, oferece a vantagem de ser facilmente codificado, facilitando assim a elaboração e manutenção de uma aplicação. Além disso, essa tecnologia permite separar a programação lógica (parte dinâmica) da programação visual (parte estática), facilitando o desenvolvimento de aplicações mais robustas, onde programador e *designer* podem trabalhar no mesmo projeto, mas de forma independente. Outra característica do JSP é produzir conteúdos dinâmicos que possam ser reutilizados.

Quando uma página JSP é requisitada pelo cliente através de um *browser*, esta página é executada pelo servidor, e a partir daí será gerada uma página HTML que será enviada de volta ao *browser* do cliente. A Figura 4 ilustra esse funcionamento (Field & Kolb, 2000).





**Figura 4 – Funcionamento de Páginas JSP**

Quando o cliente faz a solicitação de um arquivo JSP, é enviado um *object request* para a JSP engine. A JSP engine envia a solicitação de qualquer componente (podendo ser um *JavaBeans component*, *servlet* ou *enterprise Bean*) especificado no arquivo. O componente controla a requisição possibilitando a recuperação de arquivos em banco de dados ou outro dado armazenado, em seguida, passa o objeto *response* de volta para a JSP engine. A JSP engine e o Web server enviam a página JSP revisada de volta para o cliente, onde o usuário pode visualizar os resultados através do *browser*. O protocolo de comunicação usado entre o cliente e o servidor pode ser HTTP ou outro protocolo.

Por definição, JSP usa Java como sua linguagem de *scripts*. Por esse motivo, o JSP se apresenta mais flexível e mais robusto do que outras plataformas baseadas simplesmente em JavaScripts e VBScripts.

Para tirar um bom proveito do JSP é preciso entender a linguagem Java. Isso porque, a parte lógica do JSP envolve *Java Beans*, Objetos Java DataBase Connectivity (JDBC), Enterprise Java Beans (EJB) entre outros componentes que interagem com a plataforma Java. O ideal é que se conheça um pouco de HTML. É comum em grandes aplicações que o programador JSP e o *designer* sejam pessoas diferentes.

### 5.6.1 Diferenças entre JSP e ASP

No Quadro 2 podemos ver as diferenças que existem entre JSP e ASP.

**Quadro 2 – Diferenças entre JSP e ASP**

	<b>JSP</b>	<b>ASP</b>
<i>Servidores web disponíveis</i>	Internet Information Server Apache Netscape	Internet Information Server Personal Web Server
<i>Plataformas HW</i>	Disponível em todas as plataformas que existe Java	Plataformas compatíveis com Win32
<i>Uso de scripts</i>	Uso do JavaScript	Uso do VBScript e Jscript
<i>Segurança</i>	Modelo de segurança do Java	Modelo de segurança baseado na arquitetura do NT
<i>Acesso a bases de dados</i>	Através do JDBC	Através do ADO (Active Data Object)
<i>Personalização de tags</i>	Se pode ampliar através do uso de bibliotecas	Não pode ser ampliado

## 5.7 CONCLUSÃO

A finalidade das diversas tecnologias para construção de *sites* e aplicações para a Internet é praticamente a mesma. O que difere umas das outras, são características como: facilidade de uso, desempenho e segurança.

## 6 AGÊNCIA DE VIAGENS *ON-LINE*

Este capítulo apresenta algumas características de agências de viagens *on-line* existentes na Internet. Também são apresentados os Estados da Arte em matéria de agência de viagem virtual.

### 6.1 CARACTERÍSTICAS

Algumas características apresentadas em agências de viagens virtuais nacionais e internacionais:

- disponibilidade/reserva de hotéis, pousadas e cabanas;
- disponibilidade/reserva vôos e pacotes;
- busca de informações turísticas;
- aluguel de veículos;
- venda de veículos;
- financiamento de moradias;
- cruzeiros;
- viagens de trem;
- cartões de incentivo/desconto (gasolina, compras, etc.);
- cursos no exterior;
- informações sobre companhias aéreas/vôos;
- informações sobre tarifas;
- conversão de moedas;
- informações sobre clima;
- mapas;
- reportagens sobre lugares turísticos/exóticos;
- dicas sobre: diversão, arrumação de bagagens, fotografias, ... ;
- promoções/ofertas.

### 6.2 ESTADO DA ARTE

Existem várias agências de viagens virtual em todo o mundo. Porém, destacam-se, entre outras, pela qualidade dos serviços, produtos e informações prestadas, as seguintes agências:

- **Priceline** ([www.priceline.com](http://www.priceline.com)): Figura 5;

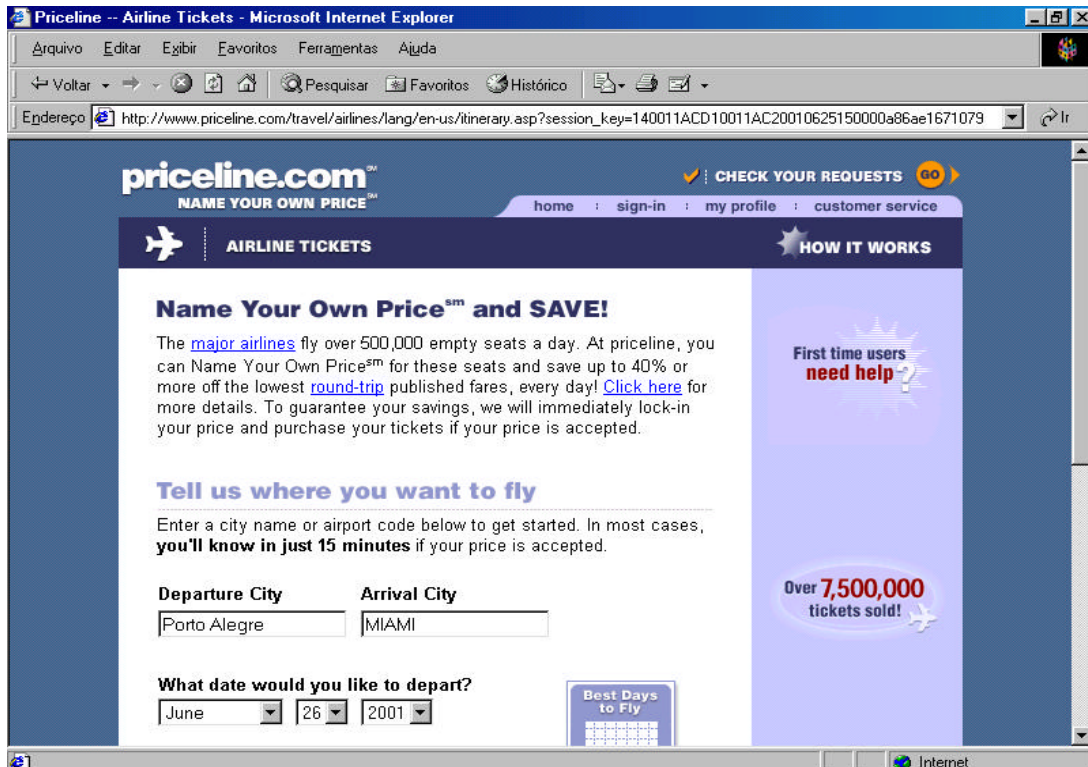


Figura 5 – Página da Agência Virtual Priceline

- Decolar ([www.decolar.com.br](http://www.decolar.com.br)): Figura 6;

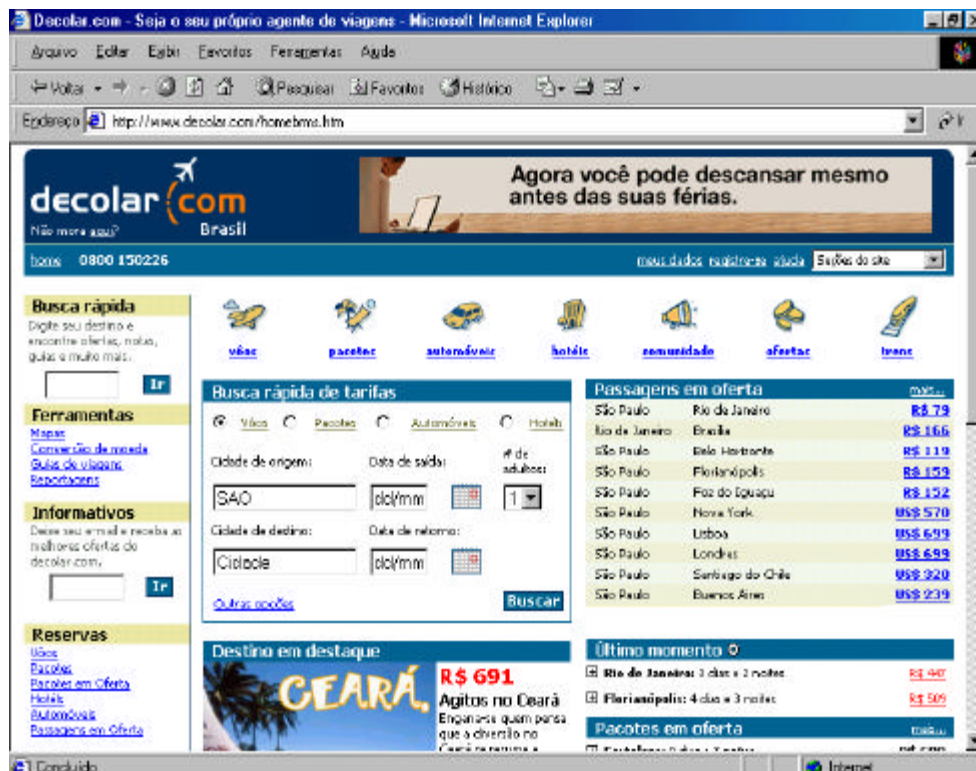


Figura 6 – Página da Agência Virtual Decolar

- Descontur (www.descontur.com.br): Figura 7;



Figura 7 – Página da Agência Virtual Descontur

- Viajo (www.viajo.com): Figura 8.

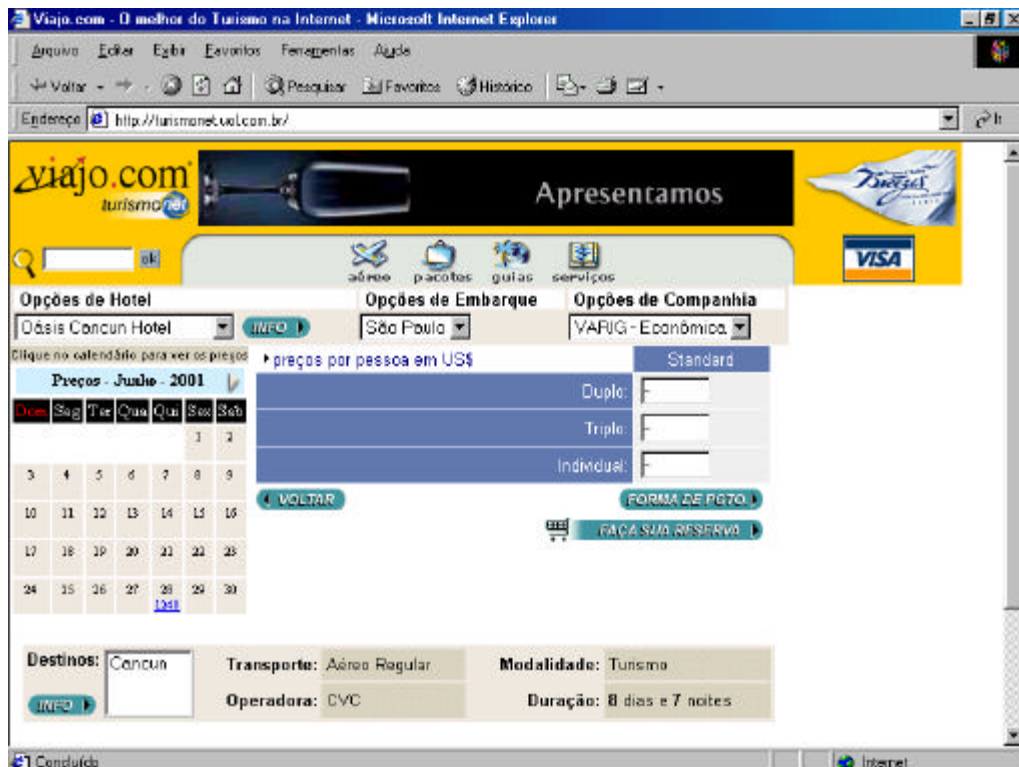


Figura 8 – Página da Agência Virtual Viajo

Estas agências possuem a maioria, senão todas as características apresentadas anteriormente. Suas interfaces são ricas em informação, porém são poluídas visualmente. Apresentam informações das mais variadas possíveis relacionadas ao turismo e viagens. A forma de pagamento em todas elas é basicamente por cartão de crédito. Uma característica apresentada pelas agências Decolar e Viajo é a possibilidade de escolha do idioma em que se quer acessar as páginas.

Porém, a oferta de informações turísticas a respeito do pacote de viagem adquirido, não ocorre de maneira automática. Algumas páginas possuem informações disponíveis, mas necessitam ser pesquisadas como numa página de busca. Esta é a característica proposta neste trabalho, ou seja, executar a busca de informações de determinada localidade a partir do momento de sua escolha, de maneira automática.

Nenhum dos *sites* pesquisados utiliza agentes móveis e/ou pesquisa automática de informações sobre os locais turísticos.

## 7 PROJETO

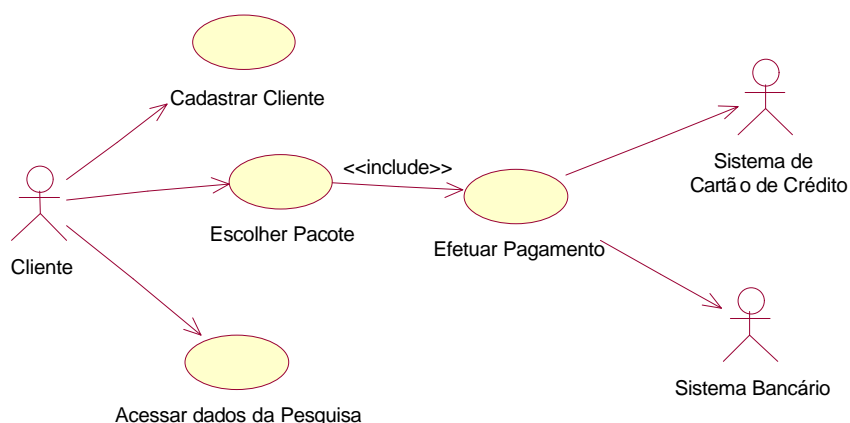
Este capítulo apresenta o projeto de uma agência de turismo virtual, que utiliza agentes inteligentes para busca de informações turísticas. São apresentados: as principais funcionalidades do protótipo, o modelo de informações semânticas a utilizado, o diagrama de seqüência e as tecnologias envolvidas no desenvolvimento do protótipo.

A proposta deste trabalho é a utilização de agentes no comércio eletrônico. A utilização de conceitos envolvendo IA e Engenharia de *Software* não é atual, porém o uso de IA em comércio eletrônico é novidade.

### 7.1 FUNCIONALIDADES

A funcionalidade proposta para a agência virtual não compreende a totalidade das características apresentadas no capítulo anterior. As principais funcionalidades do protótipo são (figura 9):

- Cadastrar Cliente: Neste módulo o sistema deverá, através de um formulário, requisitar as informações do cliente, para que sejam armazenadas na base de dados e futuramente utilizadas para analisar o perfil dos clientes da agência.
- Escolher Pacote: O cliente poderá escolher locais para visita, e o sistema buscará informações turísticas sobre essas localidades. Essa busca é feita através da implementação de agentes inteligentes.
- Acessar dados da Pesquisa: O cliente poderá acessar, através da tela de retorno populada pelo agente inteligente, as informações a respeito da localidade destino selecionada pelo cliente.



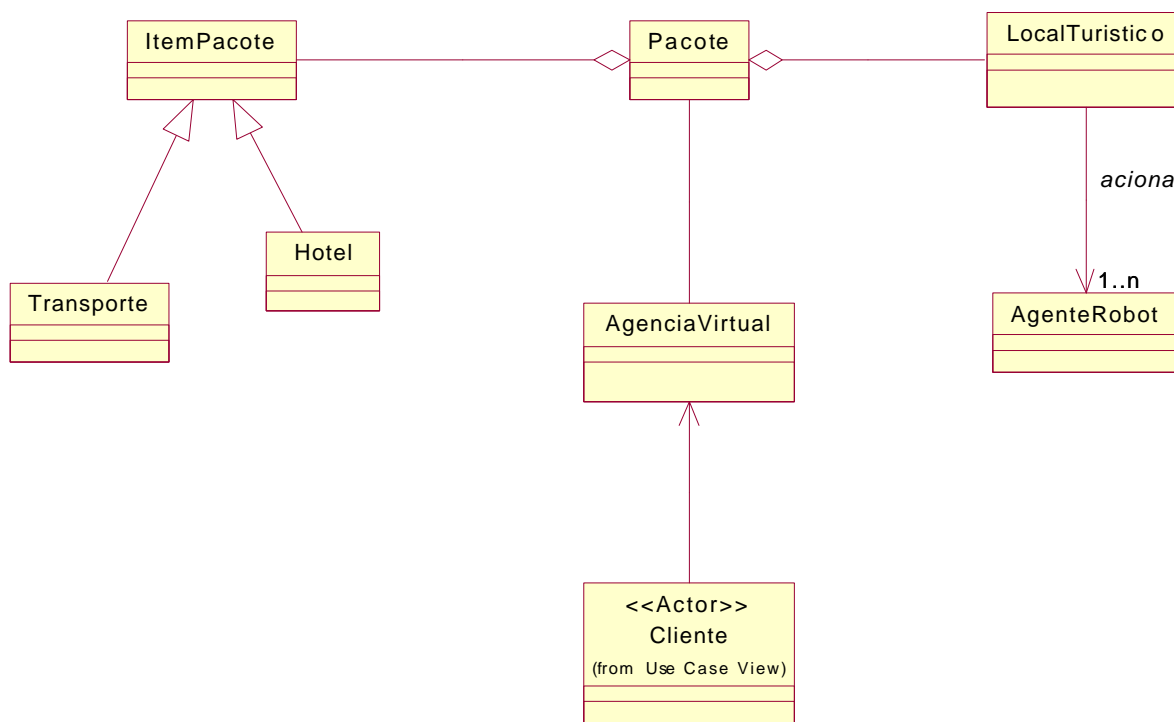
**Figura 9 – Use Case do Projeto de Agência Virtual**



Para desenvolvimento das regras de negócio e implementação dos agentes foi utilizado a linguagem Java, e para apresentação da aplicação foi utilizado JSP.

## 7.2 MODELO DE INFORMAÇÕES SEMÂNTICAS

Neste item é apresentado o modelo de informações semânticas. Nele pode-se verificar quais informações são importantes para a implementação do protótipo e apresentação dos resultados.



**Figura 10 – Modelo Semântico do Projeto de Agência Virtual**

O modelo semântico apresentado na Figura 10 foi desenvolvido utilizando a UML, por ser uma metodologia orientada a objetos e por permitir, através de suas ferramentas (diagramas), demonstrar o funcionamento estático e dinâmico do sistema.

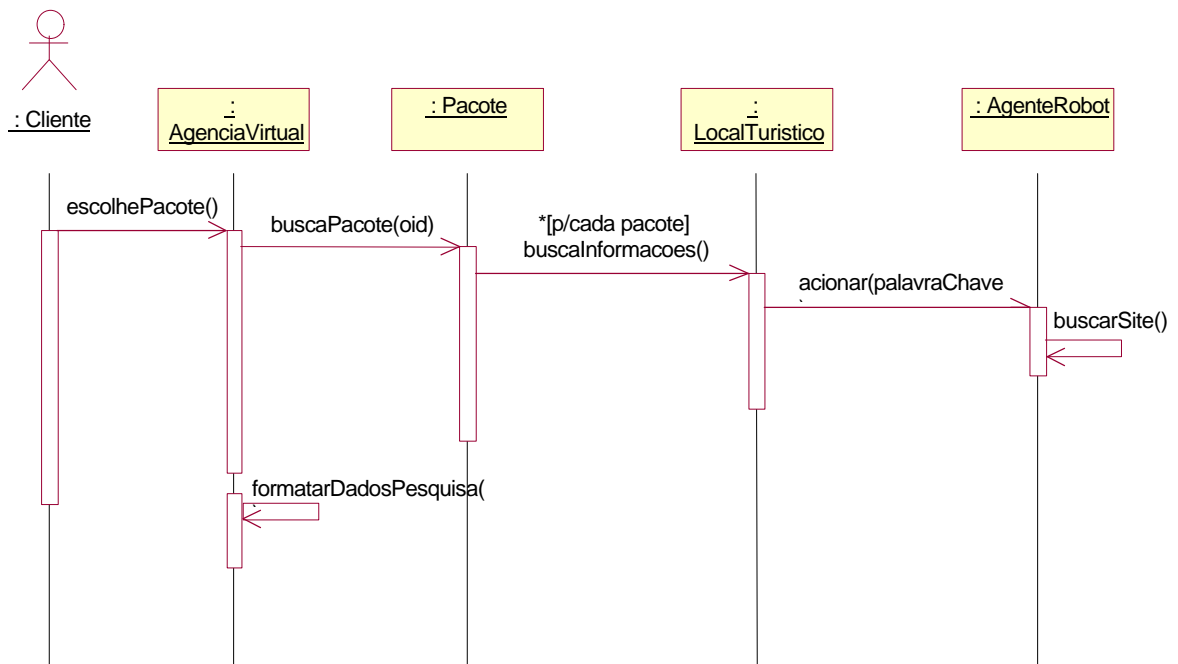
## 7.3 DIAGRAMA DE SEQUÊNCIA

O objetivo do trabalho é apresentar a utilização de agentes móveis em sistemas de comércio eletrônico. No protótipo proposto esta característica está presente na aquisição de um pacote turístico por um determinado cliente. O diagrama de sequência apresentado na Figura 11 demonstra o funcionamento desta característica.

Após o cliente selecionar um determinado pacote turístico, o sistema aciona os agentes para buscar informações sobre os locais turísticos envolvidos no referido pacote. Esses agentes buscam informações de *sites* contendo informações sobre os locais, e esses são apresentados ao usuário em forma de *links*.

Este diagrama demonstra o funcionamento dinâmico do protótipo da agência virtual.





**Figura 11 – Diagrama de Seqüência – Compra de Pacote Turístico**

## 7.4 TECNOLOGIAS ENVOLVIDAS

Para o desenvolvimento deste projeto foram utilizadas algumas das tecnologias descritas no capítulo 5. No que se refere ao desenvolvimento das regras de negócio e implementação de agentes, foi utilizada a linguagem Java. Para o desenvolvimento da apresentação do protótipo, foi utilizado JSP.

Na parte referente a agentes, foi utilizado um *webbot* chamado BDDBot que implementa agentes autônomos. A descrição deste webbot é apresentada na próxima subseção.

### 7.4.1 Webbot BDDBot

BDDBot é um *webbot*, um *search engine* e um *web server* implementado totalmente em Java (Macinta, 1997). Este mecanismo de busca apresenta as seguintes características/vantagens:

- simples e boa ferramenta de aprendizagem para mecanismos de busca;
- por ser simples, pode facilmente ser expandido;
- possui seu próprio *web server* implementado;
- é completamente livre/gratuito;
- implementado em Java, uma linguagem poderosa;
- pode funcionar em qualquer computador que tenha uma máquina virtual do Java;
- é orientado a objeto;
- é pequeno.

O BDDBot é formado basicamente por três *packages*. A *package search* é quem possui a classe *Monitor*, que é responsável por gerenciar o funcionamento do *webbot*. A

*package spider* possui as classes necessárias para a pesquisa (classe *Crawler*) e armazenamento (classe *Indexer*) dos *sites* especificados na configuração do BDDBot. A *package query* possui as classes *QueryWebServer*, *QueryConnection* e *DBQuery*, responsáveis pela busca de informações na base formada pela *package spider*. A classe *QueryWebServer* é o *web server* do BDDBot. A classe *QueryConnection* é encarregada por fazer a conexão do *web server* com a base de dados do *webbot*. A classe *DBQuery* é encarregada da busca na base de dados, através da consulta solicitada.

Esta estrutura pode ser visualizada na Figura 12 (Macinta, 1997).

Através do Monitor se tem o controle sobre o BDDBot. Ao inicializar o Monitor, também é inicializado o *web server* automaticamente. Os URLs indicados na configuração do BDDBot são pesquisados pelo *Crawler* para geração ou atualização de seus índices. No momento da pesquisa, a classe *DBQuery* faz a busca na base gerada pelo *Crawler*.

Este *webbot* não está projetado para realizar a atualização automática de sua base de índices. Para a atualização do arquivo de índices é necessário fazer uma adaptação no *webbot* BDDBot, porém, como o foco deste trabalho é a demonstração da tecnologia de agentes inteligentes no comércio eletrônico, esta modificação no *webbot* não foi realizada, podendo ser implementada em trabalhos futuros.

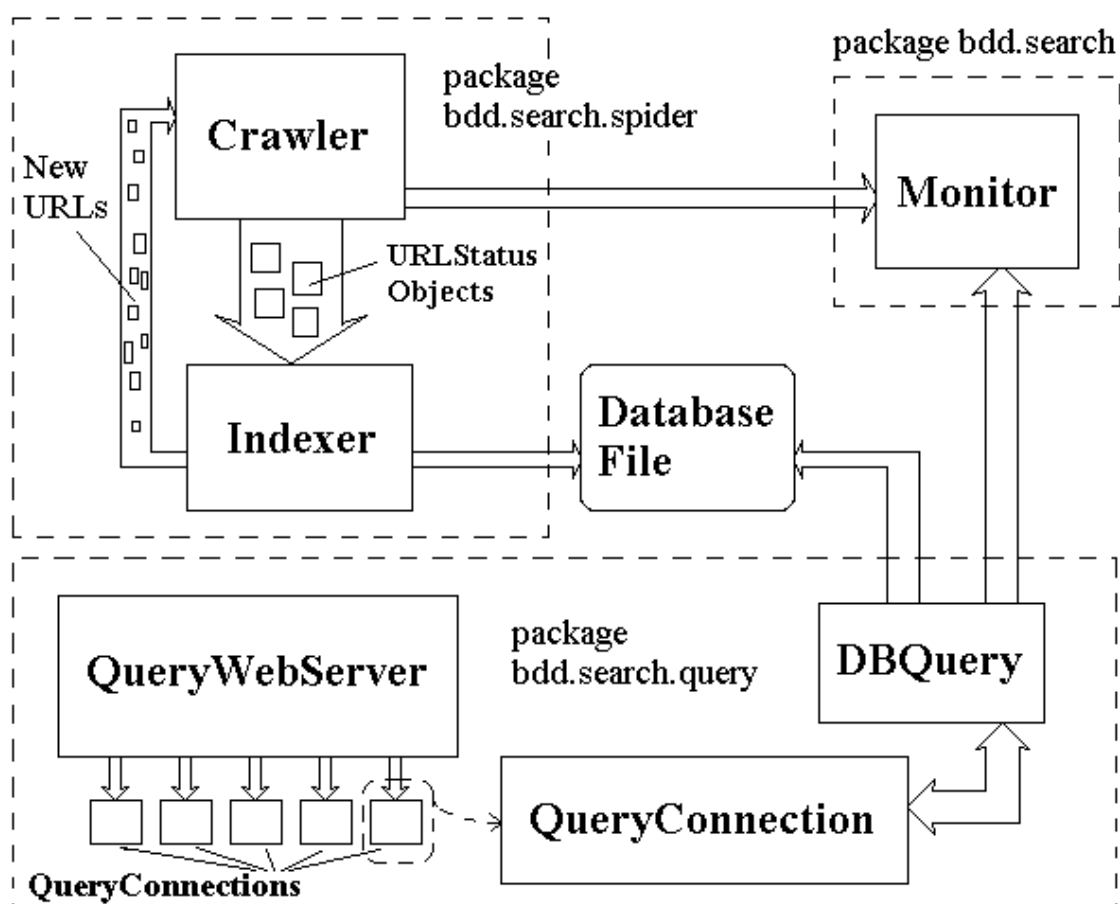


Figura 12 – Diagrama de Interação de Classes do BDDBot

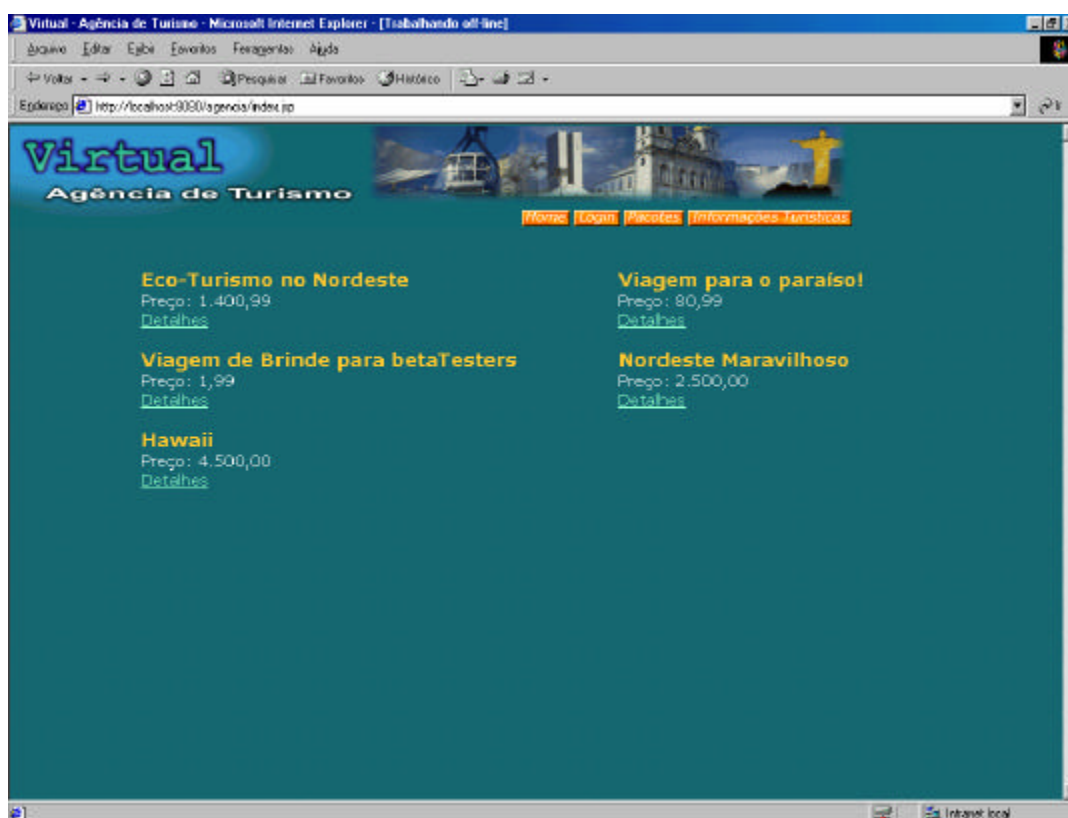
## 7.5 PROTÓTIPO

O desenvolvimento do protótipo foi realizado com o emprego de tecnologia totalmente *free*: linguagem Java, páginas de apresentação JSP, banco de dados MySQL, *web server* Resin e *webbot* BDDBot.

Por não ser o foco deste trabalho o desenvolvimento de uma solução completa de e-commerce, não é apresentado o modelo relacional da base de dados do protótipo. Porém, as informações referentes ao protótipo da aplicação, como: dados dos clientes, dados dos pacotes turísticos e dados dos pedidos; estão sendo armazenadas em uma base de dados relacional, utilizando o banco de dados MySQL. Portanto, em trabalhos futuros, pode-se fazer um tratamento sobre os dados armazenados nesta base de dados.

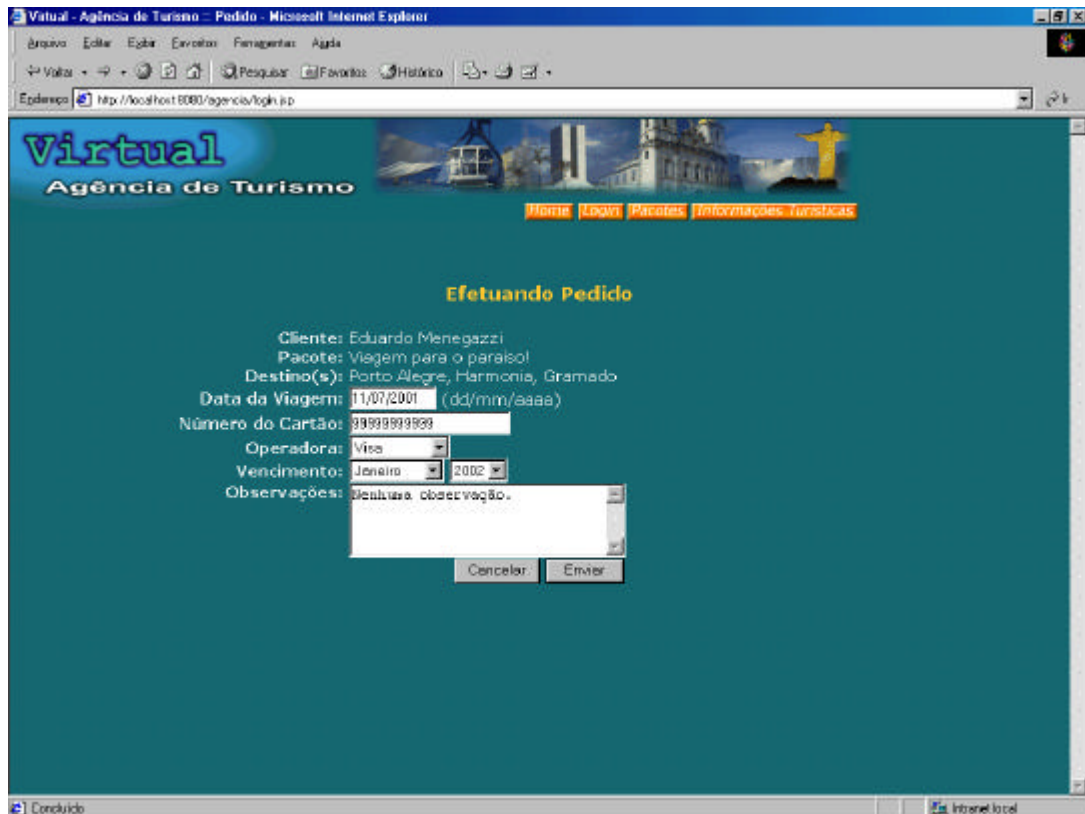
A seguir, são apresentadas as principais telas do protótipo desenvolvido:

- tela principal da agência virtual (Figura 13);



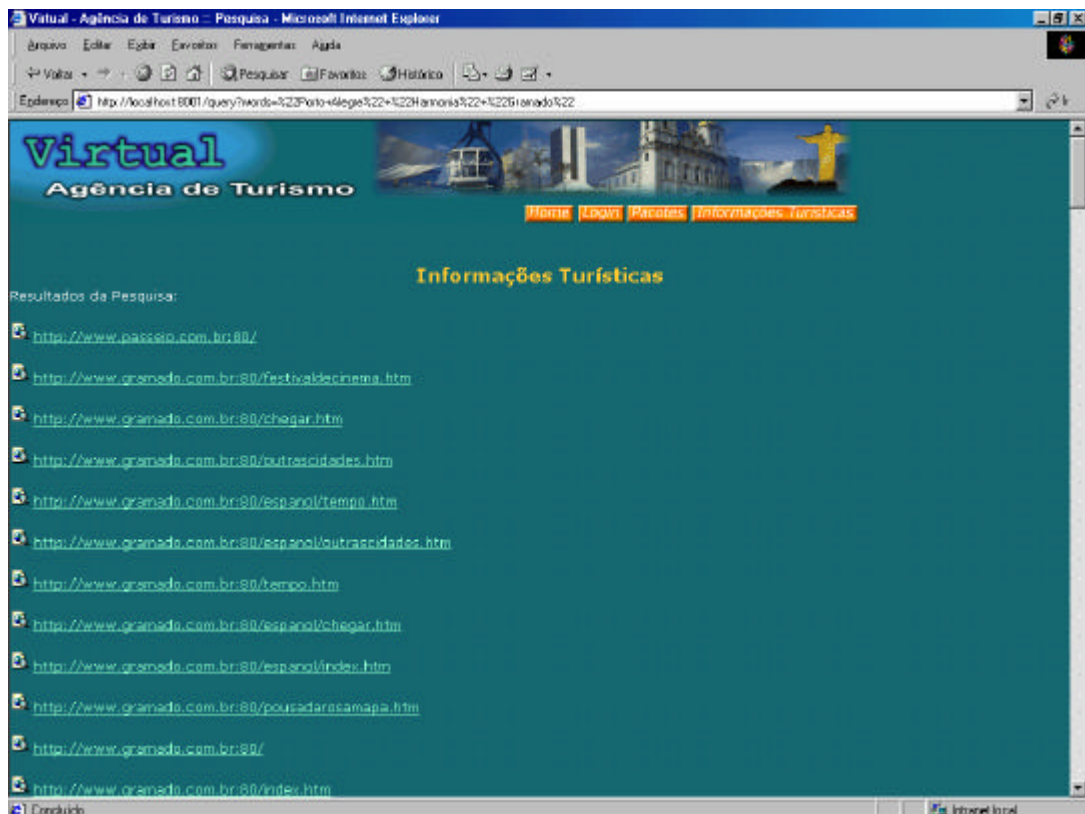
**Figura 13 – Página Principal da Agência Virtual**

- tela de confirmação de pedido (Figura 14);



**Figura 14 – Página de Confirmação do Pedido da Agência Virtual**

- tela com *links* de informações turísticas referente ao pedido efetuado (Figura 15).



**Figura 15 – Página com *Links* de Informações Turísticas**

## 8 CONCLUSÃO

O comércio eletrônico, através da Internet, tem grande importância para a economia, proporcionando grandes vantagens para clientes e empresas – principalmente, vantagens econômicas.

A Internet, por característica própria, também proporciona vantagens, assim como grande quantidade de informações. Contudo, perde-se muito tempo até que o usuário encontre a informação adequada para a sua necessidade. Para resolver este problema, podem ser utilizados agentes inteligentes, propiciando maior comodidade ao usuário.

Além de fornecer uma forma diferenciada para a comercialização de produtos e serviços, os agentes inteligentes podem ser úteis em tarefas como o controle de estoque, controle de logística, fornecimento personalizado de informações, divulgação, entre outras.

A soma do comércio eletrônico com tecnologias como as de agentes inteligentes, torna-se uma poderosa ferramenta de acesso e processamento de informações na Internet.

No comércio eletrônico da Agência Virtual, os agentes inteligentes proporcionam um diferencial em relação às outras agências de turismo virtuais existentes. Com a livre concorrência e as facilidades que a Internet apresenta, é necessário mais do que um *site* “bonitinho” para ganhar a preferência, e principalmente o dinheiro, dos clientes.

Neste trabalho, foi explorada a pesquisa automática de informações na Internet através de agentes. Porém, é possível explorar esta tecnologia de outras maneiras. Uma delas seria a personalização do *site* de acordo com as preferências do seu visitante. Outra seria a consulta de disponibilidade e efetivação da reserva em hotéis, locadoras de automóveis, aeroportos, restaurantes, etc. Também é possível a implementação de agentes para a verificação da veracidade dos dados pessoais fornecidos pelos clientes.

O conhecimento adquirido com o estudo e desenvolvimento deste trabalho, pode ser aplicado não somente no desenvolvimento de uma agência virtual completa, mas também em outros sistemas de comércio eletrônico ou até mesmo em outras ferramentas, como por exemplo, um sistema de busca.

Assim como a aplicação das tecnologias estudadas neste trabalho, principalmente a de agentes inteligentes, torna-se um diferencial na implementação e no negócio final exposto na Internet, pelos mesmos motivos torna-se um diferencial em termos de conhecimento e de mercado para o profissional que as detém.

## REFERÊNCIAS BIBLIOGRÁFICAS

- 1 APARICIO, G. et al. **The Role of Intelligent Agents in the Information Infrastructure** [online]. 1995. Disponível em: <http://activist.glp.ibm.com:81/WhitePaper/ptc2.htm> [acessado em 16 mar. 2000].
- 2 BROWN, C. et al. AI on the WWW: Supply on Demand Agents. **IEEE Expert**, v.10, n.4. August 1995, pp. 50-55.
- 3 DAVIDSSON, P. **On the Concept of Concept in the Context of Autonomous Agents**. In: Second World Conference on the Fundamentals of Artificial Intelligence, 1995.
- 4 DE CARLO, R., SANTOS, N. M. **Estudo e Aplicação de Agentes Inteligentes no Processo de Engenharia de Software**. Trabalho de Graduação, Universidade Estadual de Maringá, Paraná, Brasil, 1995.
- 5 FIELDS, Duane K., KOLB, Mark A. **Desenvolvendo na Web com JavaServer Pages**. Ciência Moderna, 2000. 559 p.
- 6 HERMANS, B. **Intelligent Software Agents on the Internet: an inventory of currently offered functionality in the information society & a prediction of future development**. PhD Thesis. Tilburg University, 1996.
- 7 KRAUSE, Walther. Evoluindo em Direção ao Conceito de Empresa Virtual. **Developers'**, Rio de Janeiro, n.42, p. 16-19, fev. 2000.
- 8 LINGNAU, A., DROBNIK, O. **An Infrastructure for Mobile Agents: Requirements and Architecture**, Proceedings 13th DIS Workshop, Orlando, Flórida, 1995.
- 9 LUCK, M., GRIFFITHS, N., D'INVERNO, M. **From Agent Theory to Agent Construction: A Case Study**. Springer-Verlag, 1997.
- 10 MACINTA, Tim. **BDDBot** [online]. 1997. Disponível em: <http://www.endware.com/bddbot> [acessado em 18 mai. 2001].
- 11 SCHILDT, Herbert, NAUGHTON, Patrick. **The Complete Reference Java2**. McGrawHill, 1999. 1108 p.
- 12 SPERB, Josué Klafke, FILHO, Eduardo Isaia, MÜLLER, Viviane. **E-commerce**. Santa Cruz do Sul: [s/e], 1999. 22p. Apostila da disciplina "Análise e Projeto de Sistemas II".
- 13 WLADAWSKY, Irving. **E-commerce e E-business**. IBM Corporation, Research Triangle Park, NC, USA, 1998.

- 14 WOOD, A. **Towards a Medium for Agent-Based Interaction**. School of Computer Science, The University of Birmingham, 1994.
- 15 WOOLDRIDGE, M., JENNINGS, N. R. **Intelligent Agentes**: Theory and Practice [online]. 1995. Disponível em: <http://doc.mmu.ac.uk/pwk/ii.htm> [acessado em 23 abr. 2000].

## **OBRAS CONSULTADAS**

- 1 BARRETO, Alexandre. **ASP – Active Server Pages – Guia de Referência**. 1998.
- 2 SILVA, Fabiana Maria. **Mini-Curso sobre Arquitetura de Agentes Inteligentes** [online]. [s/d]. Disponível em <http://di.ufpe.br/~fmmcs/agentes/resumo.html> [acessado em 03 mar. 2000].
- 3 TORRES, Anderson Barros. **JavaScript – Guia de Referência**. 1997.