IT 210

IT-210

UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE MATEMÁTICA E INFORMÁTICA

TRABALHO DE LICENCIATURA

EXVENTSHOP - UM SISTEMA DISTRIBUIDO PARA SUPORTE DE COMÉRCIO ELECTRÓNICO

Kennedy Ismael

IT-210

TRABALHO DE LICENCIATURA

EventShop – Um Sistema Distribuído para Suporte do Comércio Electrónico

Autor: Kennedy Ismael
Supervisor: Prof. Dr. Adérito F. Marcos
Coo Supervisora: drª Generosa Cossa

Realizado em Portugal, Cooperação com o Centro de Computação Gráfica - CCG, Coimbra.

Fevereiro, 1999

10.388
5-10-2005
oferta
I-210

Declaração de honra

"Declaro que este trabalho é resultado das minhas próprias investigações e o mesmo foi realizado apenas para ser submetido como trabalho de Licenciatura em Informática na Universidade Eduardo Mondlane".

Maputo, 28 de Fevereiro de 1999

(Kennedy Mahomed Jussub Ismael)

Agradecimentos

Não podia deixar passar esta oportunidade para agradecer às pessoas, sem as quais este trabalho não seria possível.

Ao meu Supervisor, Prof. Dr. Adérito F. Marcos pela iniciativa e oportunidade que me proporcionou em realizar este trabalho, bem como todas as ajudas prestadas ao longo do mesmo.

A drª. Generosa Cossa pela força e apoio prestado na realização deste trabalho.

Ao Ministério de Educação, por ter acreditado neste trabalho, fornecendo seu apoio incondicional, vai o meu especial obrigado.

Ao Engº Jürgen Bund, dr.Carlos Urbano, Engª Rosa Ferreira, Joel Oliveira e o Carlos Tsen, pela paciência e assistência prestada ao longo da realização deste trabalho. O meu agradecimento também a toda a equipe do CCG, especialmente ao seu presidente, Prof. Dr. José Carlos Teixeira, pelo constante apoio e amizade demonstrados.

Finalmente, agradeço em especial aos meus familiares, pois sem eles nada disso seria possível. Para eles o meu grande obrigado.

Índice

RESUMO

Como o título deste trabalho sugere, o processo de integração do comércio electrónico é feito através dum estudo prévio sobre o seu estado. Neste contexto aspectos relacionados com a segurança merecem especial atenção, principalmente no que se refere à segurança dos dados armazenados, mas também das comunicações, visto que para se efectuar compras via Internet é necessário fornecer formas adequadas e seguras de se efectuar os pagamentos. Para esse efeito foram criados vários sistemas monetários seguros.

Este conceito, associado à explosão da World Wide Web, levou à criação dum protótipo de uma Feira virtual de exposição e venda de produtos - "ExventShop", verdadeiramente interactivo e dinâmico. Nele, os visitantes poderão navegar pelas suas lojas, visualizando os produtos diversificados, interagindo com eles, obtendo mais informações sobre cada produto.

O ExventShop fornece aos visitantes a possibilidade de poderem encomendar os seus produtos, sem terem que se deslocarem à Feira, independentemente da sua localização geográfica, funcionando assim num ambiente distribuído.

A existência baseada na Internet reduz assim o tempo e as dificuldades surgidas por distâncias físicas dos vários participantes.

INTRODUÇÃO

"Eis o futuro. As tecnologias de informação aproximam cada vez mais o mundo e podem ser exploradas no conforto do lar apenas com custos de uma chamada local. A explosão das comunicacões começou agora e através dos computadores cada um pode-se tornar dono do mundo."Cyber.Net

Recentemente, o Comércio Electrónico tem vindo a centralizar a atenção da indústria, por um lado, e os grupos de pesquisa e desenvolvimento, por outro lado, no intuito de se implementar soluções cada vez mais eficientes que facilitem as transacções comerciais a partir da Internet ou da Intranet (rede local).

Através do Comércio Electrónico é possível aos estabelecimentos comerciais colocarem toda a sua gama de produtos acessível aos seus clientes a partir duma Loja comercial na Internet (Online Store). Desta forma, os clientes não necessitam de se deslocar ao estabelecimento, podendo realizar as suas compras usando os seus próprios computadores pessoais em casa. Estas compras serão efectuadas por encomenda a partir dum catálogo disponível e poderão ser pagas através de transferência directa usando métodos especializados de pagamento na rede. Este conceito, aliado à explosão da World Wide Web - o serviço mais popular de transporte de informação através da Internet, levou à criação dum protótipo de uma Feira virtual de exposição e venda de produtos - "ExventShop", verdadeiramente interactivo e dinâmico, onde os visitantes poderão navegar pelas lojas, visualizando os produtos disponíveis, interagindo com eles, obtendo mais informações sobre cada produto, podendo ainda encomendá-lo se assim o desejarem. Por outro lado, é facultado ao visitante um mecanismo de pesquisa para que possa rapidamente localizar um determinado produto, bem como algumas informações úteis sobre a navegação e ainda uma visualização gráfica das lojas em 2D para que o visitante tenha uma visão global e possa se integrar facilmente neste ambiente. O ExventShop é parte integrante de um projecto de investigação, estratégica do $CCG/_{ZGDV}$ - Centro de Computação Gráfica denominado Exvent (*Support System For Exposition - like events*), cujas funcionalidades serão descritas mais adiante.

Partindo de soluções existentes no Centro de Computação Gráfica e outras comerciais, tais como os produtos Microsoft, este trabalho tem como objectivo geral desenhar e implementar uma solução estável para uma Loja na Internet, tendo em conta os seguintes objectivos específicos:

- Armazenamento de informação sobre os produtos e sua actualização a partir duma Base de Dados Central dos Produtos;
- Armazenamento de informação sobre os Clientes e sua actualização a partir duma Base de Dados Central dos Clientes;

- Desenvolver formas personalizadas de atendimento electrónico;
- Considerar vários métodos de pagamento;
- Considerar os mecanismos adequados de segurança, suas vantagens e limitações;
- Desenvolver a interface com o utilizador, que deverá ser de manuseamento intuitivo e facilitado;

Este trabalho teve várias fases e para uma melhor compreensão foi dividido em vários capítulos com vista a alcançar os objectivos definidos anteriormente. Passarei a descrever resumidamente:

O Capítulo I dá uma noção geral do estado da arte do Comércio Electrónico, sua evolução, problemas e perspectivas para o futuro, abordando aspectos importantes relacionados com a segurança e formas de pagamento para o mesmo, através duma revisão bibliográfica.

O Capítulo II mostra uma das áreas em que este trabalho se pode enquadrar, através dum exemplo do desenvolvimento duma aplicação prática através do uso de algumas novas tecnologias de informação. Assim faz-se uma breve descrição dum projecto que ainda está em estudo, o Exvent, o qual inclui entre outras, uma solução dum protótipo para o comércio electrónico, o ExventShop, objecto de estudo deste trabalho, cujas funcionalidades serão aqui apresentadas.

No Capítulo III é dado um conceito geral das interfaces e linguagens de programação e serve, também, para descrição e explicação de alguns conceitos importantes definidos em *Active Server Pages* e que interactuam com o servidor e que são usadas neste trabalho.

Capítulo IV apresenta a estrutura funcional e a interface com o utilizador do ExventShop em conjunto com a implementação deste trabalho, onde serão explicadas as técnicas aplicadas, os mecanismos de navegação e interacção tanto da parte do *HiperText Markup Language (HTML) /Active Server Pages (ASP)* bem como dos conceitos organizacionais de toda estrutura interna do ExventShop.

Capítulo V são apresentadas as conclusões e considerações gerais deste trabalho e algumas recomendações para trabalho futuro tendo em vista a evolução que os ambientes virtuais previsivelmente irão ter a nível da Internet.

Como nota os termos em Inglês que aparecem em itálicos, são palavras cujo significado em português não foi possível obter ou são mais frequentemente usados com tal nome, assim como todos os formulários também são apresentados no mesmo idioma, de modo a que possa chegar a um maior número de utilizadores.

Capítulo I
Estudo do estado da arte - Comércio Electrónico

INTRODUÇÃO

À medida que se aproxima do ano 2000 cada vez mais o futuro do comércio não será como no passado. Com este trabalho pretende-se mostrar o porquê, onde e quando o comércio electrónico será importante.

Em particular existirão duas áreas onde ele é fundamental: na ligação cliente - empresa, dentro da empresa, e na ligação empresa - empresa.

A ligação empresa-empresa é cada vez mais fundamental não só para um mais fácil relacionamento em termos de apoio ao cliente (por exemplo através dos "call center"), mas também para que se estabeleça um maior grau de fidelidade nos produtos/serviços da empresa.

O comércio electrónico será também um catalisador nas modificações da organização interna da empresa particularmente através da utilização das tecnologias da Internet e ainda de *Workflow* e *Groupware*, no fundo tudo o que tem a ver com a gestão dos fluxos de informação.

A especial importância terá de ser dada à Internet que fornece uma infra-estrutura tecnológica global fácil de utilizar para o problema da difusão da informação, e como canal de transacções comerciais.

O comércio através da Internet é muito mais barato, permite uma maior amplitude de aplicação e coloca em plano de maior igualdade entre as pequenas e grandes empresas.

Nexto contexto as questões relativas à segurança, merecem especial atenção no que se refere à segurança dos dados armazenados, quer das comunicações, visto que para se efectuar compras via Internet é necessário arranjar formas adequadas e seguras de efectuar os pagamentos. No mundo real existem muitas maneiras de pagar: dinheiro, cartões bancários, cartões de crédito, cheque, senhas, etc... Da mesma forma na Internet foram criados vários sistemas de pagamento.

1.1. DEFINIÇÃO DO COMÉRCIO ELECTRÓNICO

O Comércio Electrónico (CE) poderá ter várias definições dependendo do ponto de vista que se estiver a considerar, isto é, este deverá ser considerado numa perspectiva global, e não apenas como uma nova forma "electrónica" de fazer negócios.
Teremos então uma definição possível:

O CE é o fornecimento relativo a produtos/serviços, ou pagamentos, através de linhas telefónicas, redes de computadores, ou outros meios de comunicação (Ferrão, 1998).

De uma maneira geral o CE tem a ver com a geração e exploração de novas oportunidades de negócio tornadas possíveis através das suas tecnologias associadas. Desta forma é necessário:

- melhorar a organização e comunicação externa (com todos aqueles que têm relações com a empresa);
- utilizar redes informáticas em que as várias formas de comunicação possíveis são cada vez mais centradas na Internet, "a rede das redes" bem como todas as suas tecnologias associadas, como veremos mais à frente.

1.1.1 Motivação

O dia-a-dia actual está impregnado de um número crescente de novas e variadas formas de fontes de informação, seu processamento e distribuição. Como exemplos temos:
Rádio, TV, jornais, CD-R, *Web*, Multimédia, 3D, RV (Realidade Virtual), meios electrónicos de distribuição, que poder-se-ão interligar através do seu processamento adequado e distribuição, de forma a se construir gradualmente a chamada Sociedade de Informação .
Na indústria, os meios electrónicos vieram abrir novas perspectivas e oportunidades:

a) integração de meios electrónicos no planeamento, produção e contacto com o cliente;
b) efectiva distribuição geográfica dos centros de produção e desenvolvimento.

Desta forma, o CE aparece no contexto da construção da Sociedade de Informação com a evolução natural das transacções comerciais.
A tecnologia e os meios electrónicos poder-se-ão transformar em autênticos meios para fazer negócio, com todas as vantagens inerentes.

1.1.2 Evolução histórica

O CE evolui historicamente da Transferência Electrónica de Fundos - TEF, ou seja , a transferência de pagamentos entre entidades que podem ser instituições financeiras, ou mesmo estabelecimentos comerciais que se iniciou na década de 70, e que ainda hoje se utiliza (Marcos, 1998).

No final da década de 70, e início de 80, surgiu o conceito de "mensagem electrónica" que não era apenas uma ordem de pagamento mas assumia a forma de documentos (nota de encomenda, facturas, cheques, documentos de expedição, ...) utilizados nas relações comerciais entre as empresas.

Surgiu assim o *Electronic Data Interchange* - EDI como uma transferência inteligente de dados estruturados, em mensagens pré-estabelecidas e normalizadas. Pressupunha também a normalização dos protocolos de comunicação.

No final da década de 80 e início de 90 apareceu a tecnologia *Groupware*, e da qual o correio electrónico (*E-mail*) era o elemento fundamental.

Esta forma de mensagem electrónica não só permite a troca de informações, mensagens, ou mesmos documentos, mas veio abrir um meio de comunicação informal entre pessoas de várias organizações.

Com o aparecimento da Internet nos anos 90, e em particular da utilização do *World Wide Web* - WWW, o qual de maneira fácil e barata, permite trocar informação à escala mundial, o CE adquiriu uma dimensão e uma expansão no Mercado Global sem limites visíveis. Hoje é possível realizar CE entre os locais mais remotos do planeta - bastando para tal que tenha ligação à Internet.

No entanto, a Internet trouxe também um conjunto de problemas que neste momento se equacionam em todo o mundo, como sejam o pagamento electrónico, a assinatura digital, a segurança da informação quer seja ao nível da empresa, quer quando é transmitida recorrendo a encriptação.

1.1.3 Tecnologias

As tecnologias fundamentais para o suporte do CE dividem-se em dois grandes componentes: a arquitectura e as infra-estruturas.

Arquitectura Cliente/Servidor

Um sistema Cliente/Servidor divide a aplicação que está a processar em três componentes:

- Cliente (lado comprador)

-Interface aos produtos (busca, selecção e encomenda) + tecnologias de pagamento automático;

- Servidor (lado fornecedor)

  - Dados referentes aos produtos + tecnologia de armazenamento + segurança + pagamento automático;

- Rede (canal de transacção)

  - Suporte das (modos de) comunicações entre o Cliente e Servidor;

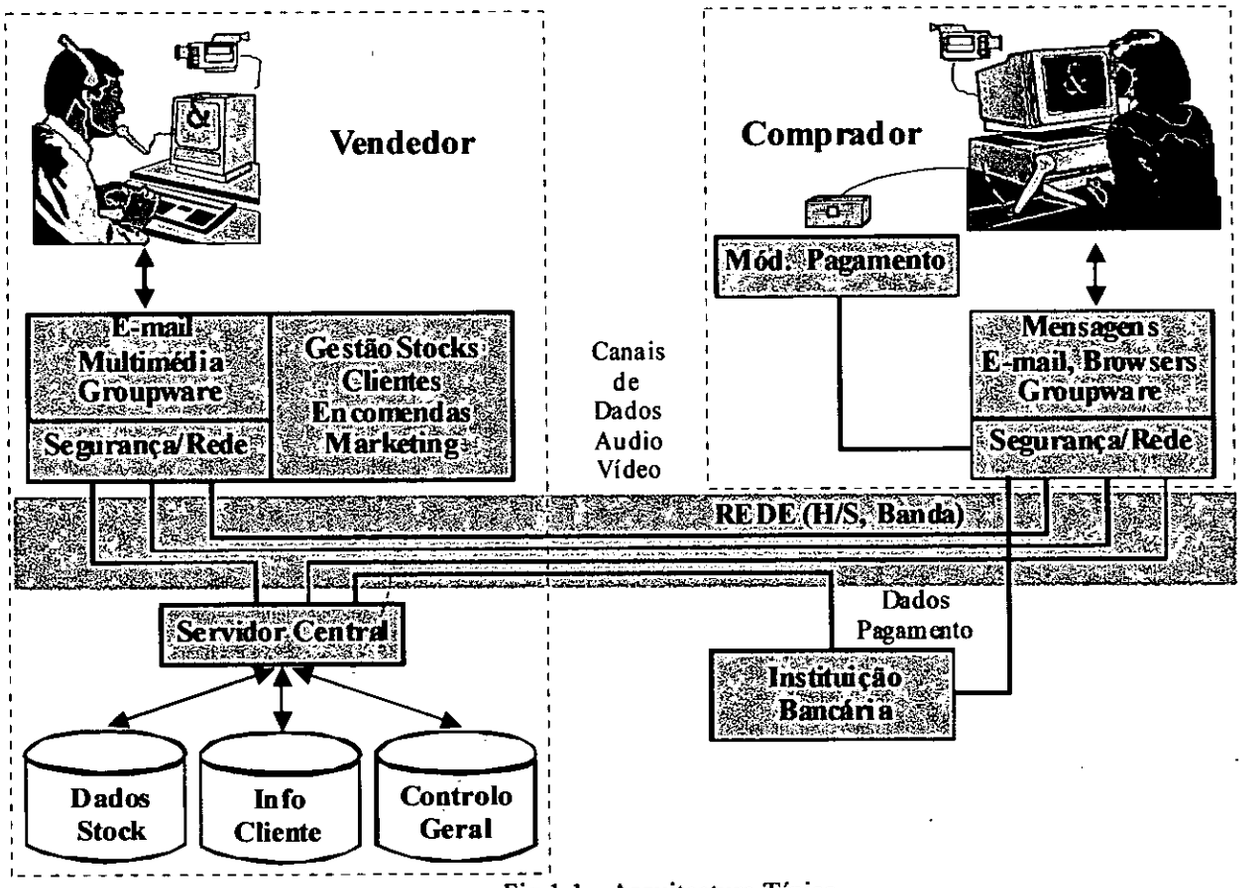


Fig.1.1 - Arquitectura Típica

Infra-estrutura Base

A infra-estrutura base subdivide-se em:

- Serviços comuns

  - Segurança, Autenticação de Documentos, Pagamentos Electrónicos, Gestão de Directorias;

- Mensagem e distribuição de informação

  - EDI, *E-mail*, WWW;

- Conteúdos Multimédia

  - Tecnologia suporte ao desenho das interfaces do CE;

- Software/Hardware de comunicação

  - LAN/WAN, Larguras de Bandas, Redes Multibanco.

1.2 SEGURANÇA

1.2.1 Conceitos básicos sobre segurança

Existe uma série de conceitos, que são essenciais para a segurança na Internet. Esses conceitos, de uma forma geral, constituiram-se numa espécie de requisitos básicos para novos mecanismos de segurança.
Esses conceitos são definidos de seguida:

Integridade dos dados: permite a detecção de modificações não autorizadas nos dados. Vulgarmente, a integridade dos dados permite detectar se os dados foram modificados ou corrompidos durante a transmissão. Pode ser conseguida através da implementação de uma função *one-way hash.*

Confidencialidade: É o processo utilizado para proteger informações secretas de serem reveladas a pessoas não autorizadas. Os dados secretos devem ser protegidos quando são guardados ou transmitidos pela *Net.*
Claramente, essa protecção recorre ao uso da criptografia. A tarefa da implementação da criptografia também requer a distribuição segura das chaves de criptação para o remetente e para o receptor dos dados cifrados.

Identificação: Os utilizadores são identificados perante uma aplicação através de uma identificação do utilizador ou *userid.*

Autenticação: É o processo usado para verificar a identidade reivindicada por um utilizador ou programa. Pode ser feita através do uso de *passwords* por parte do utilizador ou através da troca de chaves e poderá eventualmente envolver uma terceira entidade de confiança.

Controlo de acesso: Concede ou recusa a permissão a um utilizador para aceder um recurso, limitando os acessos para os utilizadores autorizados. O controlo de acessos é frequentemente especificado pelo administrador do sistema ou pelo proprietário do recurso.

Autorização: É o processo de atribuir os acessos permitidos ao utilizador. A permissão de acessos inclui uma especificação, tal como, se o utilizador possui permissão para ler, escrever, ou alterar um dado ficheiro.

Não repudiamento: É a capacidade de provar tecnicamente a origem dos dados e provar a distribuição dos dados, ou seja, demonstra-se que a transmissão ocorreu de facto, entre o remetente e o receptor. Desta forma, impede o remetente negar o envio dos dados, ou o receptor negar a recepção dos dados, e também impede que as entidades envolvidas possam alterar o conteúdo dos dados.

Rejeição de serviço: O ataque de rejeição de serviços, é um ataque do qual o atacante toma posse, ou consome recursos, de forma a que ninguém mais possa usá-lo. Exemplos desses ataques, incluem um vírus, que consome a memória do sistema, ou um ataque na Internet, onde o *host* atacante passa por *host* legítimo.

1.2.2 Criptografia

A Internet pode ser vista como um único super computador, cujos recursos de Hardware e Software estão distribuídos geograficamente à escala mundial. Um componente especialmente importante deste super computador são as redes de comunicação que ligam os computadores. Estas redes são susceptíveis a actividades ilegais por parte de certos utilizadores. As dimensões físicas destas tornam impossível proteger meios de transporte de informação (ex: fios telefónicos) através de medidas de segurança físicas. A classe de métodos mais adequados que podem ser aplicados é a Criptografia.

Criptografia é o estudo da cifragem e decifragem. Vem da palavra grega *kryptos* que significa "escondida", e *graphia* cujo significado é "escrever". Ela consiste na ciência (e arte) da transformação de mensagens numa representação sem significado para qualquer pessoa excepto para quem saiba qual o processo de reverter a transformação (Seberry,1989).

A criptografia já estava presente no sistema de escrita hieroglífica dos egípcios. Desde então vem sendo muito utilizada, principalmente para fins militares e diplomáticos. Sabe-se que os antigos Espartanos cifravam as suas mensagens militares, e uma das cifras mais antigas que se conhece, é a cifra de César, cujo nome advém da sua utilização por Júlio César.

Actualmente, com a utilização generalizada de computadores e redes de comunicação de dados, a necessidade de utilizar criptografia estende-se por diversos domínios, desde a autenticação de utilizadores, à privacidade de comunicações pessoais ou difundidas, em canais de comunicação pouco seguros, ou de acesso livre (por exemplo canais de comunicação via satélite).

Terminologias

Em geral, considera-se necessidade de transmitir uma **mensagem**, entre o **emissor** e um **receptor**. O processo de disfarçar a mensagem chama-se **cifragem** e transforma-a num **criptograma**. O processo de recuperar a mensagem original a partir do criptograma denomina-se **decifragem**.

Por seu turno, a **Criptanálise** é a ciência (e arte) de quebrar criptogramas, ou seja descobrir como fazer a decifragem de um criptograma, sem saber, à partida, como ele foi cifrado (Seberry, 1989).

Os algorítmos de criptografia, também denominados cifras, são as funções matemáticas que fazem a cifragem e a decifragem, tendo, portanto, em geral, dois componentes, respectivamente, o algorítmo de cifragem e o algorítmo de decifragem. Todos os actuais algorítmos seguros são conhecidos e usam, no seu funcionamento, uma **chave**. As chaves devem definir univocamente o criptograma.

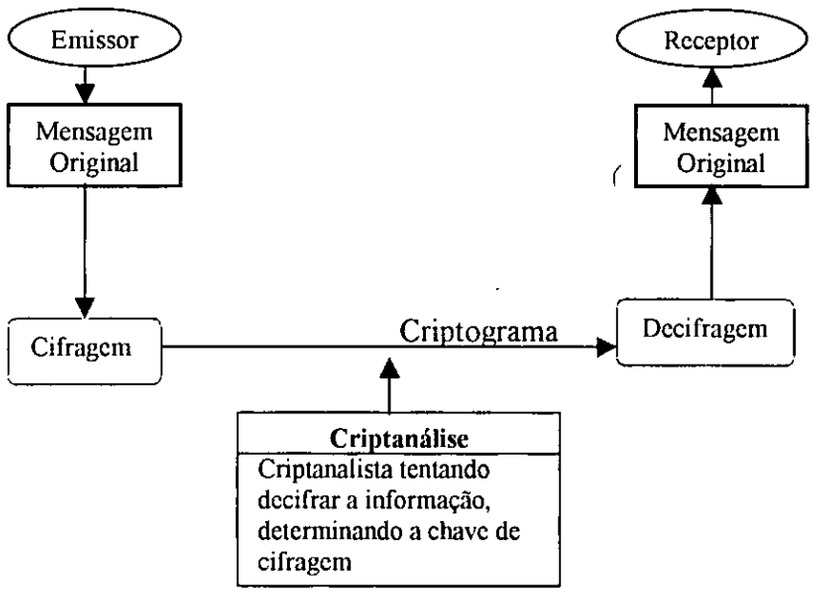


Fig.1.2 - Esquema geral de comunicação segura

1.2.2.1 Algorítmo de chave Secreta

Neste algorítmo, a segurança reside no secretismo da chave. Visto que o funcionamento do algorítmo é conhecido, sabendo a chave, é possível cifrar e decifrar qualquer mensagem. A chave de cifragem pode ser obtida a partir da chave de decifragem, e vice-versa, sendo as duas chaves normalmente idênticas. Quando a chave de cifragem é igual à chave de decifragem o algorítmo diz-se **simétrico.** Em qualquer caso, é necessário que o emissor e o receptor acordem numa chave, antes de poderem usar o sistema. A figura 1.3 abaixo ilustra como tudo se passa:

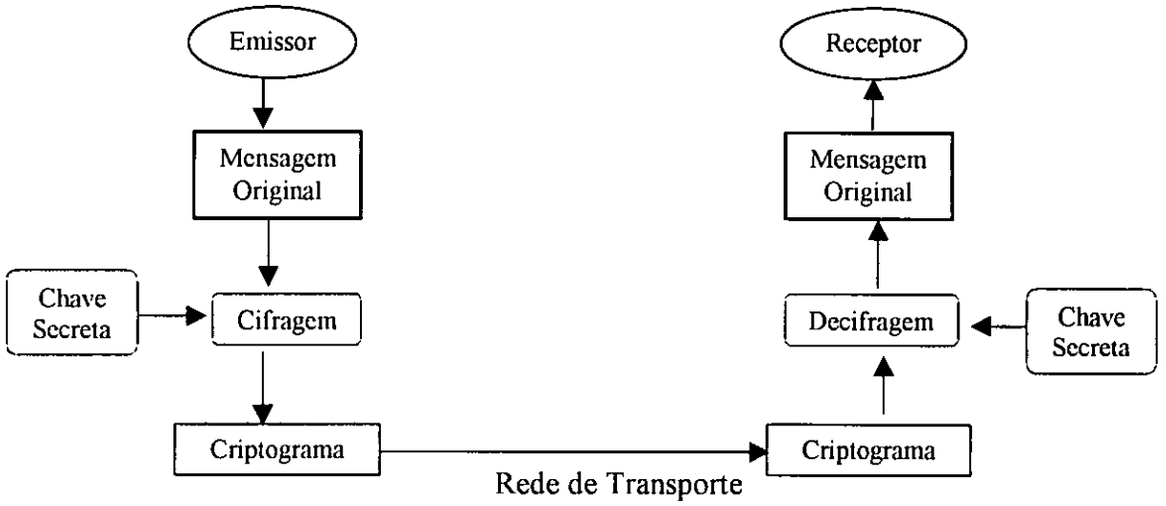


Fig.1.3–Esquema geral da criptografia simétrica

Podem-se citar alguns algorítmos deste tipo: DES, triple-DES, IDEA (*International Data Encryption Algorithm*), RC2, etc. Um dos mais conhecidos é sem dúvida o DES, que por sinal está na base do triple-DES e do IDEA.

O DES *(Data Encyption Standard)* é um algorítmo de blocos que surgiu a partir de um outro desenvolvido pela IBM, que foi posteriormente modificado e adoptado pela *National Bureau of Standards* em 1977.

Este algorítmo diz-se de blocos uma vez que cifra a mensagem dividindo-a em blocos de 64 bits, usando uma chave de 56 bits para cifrar cada um.

Saliente-se o seu uso na cifragem de *passwords* no sistema Unix. É também usado em mecanismos de segurança de *E-mail* como PEM (*Privacy Enhanced Mail*) e o PGP (*Pretty Goog Privacy*) e em vários protocolos de segurança como SSL e no S-HTTP (ver adiante).

Uma aproximação de ataque para este algorítmo seria a força bruta. Assim, todas as chaves possíveis seriam testadas. Deste modo, para chaves de 56 bits, seria necessário considerar 2^{56} chaves diferentes. Tal ataque é sempre possível. As estratégias para resistir a ataques deste tipo é tornar o processo de tal forma moroso que não compense o tempo despendido ou o investimento.

Refira-se como curiosidade que se estima que actualmente demora-se 0.2 segundos para uma chave de 40 bits, 3,6 horas para uma chave de 56 bits e 10^{18} anos para uma chave de 128 bits. Deste modo quanto maior for a chave mais difícil será descobri-la (Rebordão, 1997).

1.2.2.2 Algorítmo Assimétrico ou de Chave Pública

Tradicionalmente, os sistemas de cifragem (algorítmo simétrico) requeriam uma chave secreta comum que era usada tanto para cifragem como para decifragem. De modo a obviar estes

problemas surgiram os **algorítmos assimétricos ou de chave pública**, em que as chaves são obrigatoriamente diferentes.
Os algorítmos baseados em chave pública têm dois importantes atributos:

Primeiro, é computacionalmente inviável derivar a chave de decifragem, sabendo a chave de cifragem e o algorítmo utilizado. Nesta aproximação, cada utilizador tem duas chaves: a chave privada, que só ele conhece, e a chave pública, conhecida por todos.

O segundo atributo de alguns esquemas baseados em chave pública é que qualquer das chaves pode ser usada para cifrar e a outra para decifrar. Um dos algorítmos mais conhecidos que tem esta propriedade é o **RSA** (1978). O nome deriva das iniciais dos seus autores, Rivest, Shamir e Adleman. O RSA utiliza na sua composição dois problemas numéricos complexos, logaritmo discreto e factorização. A segurança deste método baseia-se na dificuldade de factorizar números muito grandes. Só para ter uma ideia, com a tecnologia actual, factorizar um número de 200 dígitos requer aproximadamente 4 biliões de anos de tempo de processamento (Rebordão, 1997).
Trata-se de um dos algorítmos mais importantes no mundo da criptografia e o seu uso é muito variado. Serve de base para esquemas de assinaturas digitais. É usado também em mecanismos de segurança de *E-mail*, como o PEM e o PGP. Este tipo de algorítmos é bastante adequado para autenticações [URL-1.3].

1.2.2.3 Funções de Sentido Único

As funções de sentido único (*one-way hash functions*) são assim denominadas precisamente pelo facto de ser extremamente difícil (ou impossível) obter a descodificação da mensagem.
Este tipo de algorítmo é muito usado para testar a integridade da mensagem recebida, isto é, providenciar uma maneira para o receptor detectar se a mensagem recebida foi alterada por utilizadores não autorizados. O resultado da sua aplicação é normalmente chamado *check sum* ou *message-digest (MD)*, e será um valor de muito menor dimensão do que o valor de entrada, com a propriedade de não ser invertível. Ou seja, conhecendo o valor de saída não é possível tirar qualquer ilação sobre dados de entrada. Tal como no caso dos algorítmos simétricos, requer-se que as entidades comunicantes pré-estabeleçam uma chave secreta (segredo partilhado). Se for aplicada uma função de *hash* segura sobre a mensagem e enviarmos essa *message-digest* juntamente com a mensagem, pode-se verificar com elevado grau de confiança se a mensagem provém da verdadeira entidade e se dados estão íntegros. Basta a entidade receptora recalcular a

message-digest da forma supostamente feita pelo emissor e comparar com o valor recebido. Se coincidirem a mensagem e a identidade emissora são consideras válidas.

Existem vários algorítmos deste tipo dos quais se podem salientar: MD2, MD4, MD5, SHA (*Secure Hash Algorithm*), etc.

Estas funções são normalmente usadas em codificações de *password*. No Unix, por exemplo, são usadas *passwords* de 4 a 8 caracteres. À *password* é ainda concatenado um número aleatório de 12 bits denominado *salt*, com intuito de aumentar o espaço de pesquisa para a Criptanálise. Quando se verificar que uma *password* está correcta, ao contrário do que se poderia pensar, ela será cifrada, com a original, e o resultado será comparado. Isto deve-se ao facto de ser impossível obter *password* original a partir do criptograma guardado no ficheiro de *passwords*.

1.2.2.4 Assinaturas Digitais

São assinaturas electrónicas que não podem ser forjadas. Ela é um *digest* de um texto computado que é encriptado e enviado com a mensagem do texto. O receptor decripta a assinatura e recompõe o *digest* do texto recebido [URL-1.4].

As assinaturas digitais providenciam a prova de autenticidade e origem dos dados.

Actualmente, muitas operações bancárias e transacções tornam-se legalmente válidas depois de serem assinados certos documentos.

Na Internet surge vulgarmente a necessidade de assinar documentos usando computadores locais. Contudo, uma vez que os computadores só aceitam informação na forma digital, qualquer assinatura em questão terá que ser também digital.

No entanto, assinaturas digitais deveriam ter as mesmas propriedades que as que são assinadas a mão. Assim, as assinaturas digitais deveriam ser:

- únicas;
- facilmente autenticáveis;
- não repudiáveis;
- baratas e fáceis de gerar.

Existem três aproximações possíveis para resolver este tipo de problema:

- assinatura digital baseada em chave secreta;
- assinatura digital baseada em chave pública;
- assinatura digital baseada em funções de sentido único.

As duas primeiras, ao contrário da terceira, permitem manter a privacidade da mensagem.

1.2.2.5 Certificado digital

Certificado de Identidade Digital, também conhecido como Certificado Digital, associa a identidade de um titular a um par de chaves electrónicas (uma pública e outra privada) que, usadas em conjunto, fornecem a comprovação da identidade. É uma versão electrónica (digital) de algo parecido a uma Cédula de Identidade - serve como prova de identidade, reconhecida diante de qualquer situação onde seja necessária a comprovação de identidade.

Certificado Digital pode ser usado numa grande variedade de aplicações, como comércio electrónico, *groupware* (Intranet's e Internet) e transferência electrónica de fundos (veja o exemplo recente do Banco Bradesco S.A. na implantação do seu serviço Internet - o BradescoNet) [URL-1.4].

Dessa forma, um cliente que compre numa loja virtual, utilizando um Servidor Seguro, solicitará o Certificado de Identidade Digital deste Servidor para verificar a identidade do vendedor e o conteúdo do Certificado por ele apresentado. De forma inversa, o servidor poderá solicitar ao comprador o seu Certificado de Identidade Digital, para identificá-lo com segurança e precisão.

Caso qualquer um dos dois apresente um Certificado de Identidade Digital adulterado, ele será avisado do facto, e a comunicação com segurança não será estabelecida. O Certificado de Identidade Digital é emitido e assinado (chancelado) por uma autoridade certificadora digital (*Certificate Authority*), como a Thawte (certificadora da ArtNET), que emite o Certificado. Para isso, esta autoridade usa as mais avançadas técnicas de criptografia disponíveis e de padrões internacionais (norma ISO X.509 para Certificados Digitais), para a emissão e chancela digital dos Certificados de Identidade Digital.

Um certificado contem três elementos:

Informação de atributo

Esta é a informação sobre o objecto que é certificado. No caso de uma pessoa, isto pode incluir seu nome, nacionalidade e endereço *E-mail*, sua organização e o departamento desta organização onde trabalha.

Chave de informação pública

Esta é a chave pública da entidade certificada. O certificado actua para associar a chave pública à informação de atributo, descrita acima. A chave pública pode ser qualquer chave assimétrica, mas usualmente é uma chave RSA.

Assinatura da Autoridade em Certificação (CA)

A CA assina os dois primeiros elementos e, então, adiciona credibilidade ao certificado. Quem recebe o certificado verifica a assinatura e acreditará na informação de atributo e chave pública associadas se acreditar na Autoridade em Certificação.

1.3 SEGURANÇA NO CE

Para que uma compra *online* se torne rapidamente uma realidade para a maioria das pessoas e não para alguns, existe um grande obstáculo a ser ultrapassado: o pagamento.

1.3.1 Problemática da Segurança

O dinheiro virtual não vai ser aceite imediatamente. Qualquer sistema necessita de ser seguro e rápido.

A Segurança é uma exigência básica porque a Internet é inerentemente insegura. Milhões de computadores dão forma a uma rede pública onde as comunicações possam ser interceptadas. Enquanto os dados circulam do emissor para o receptor, quase sempre têm de viajar através de diversas outras conexões. Isto é chamado routeamento ou reencaminhamento. Durante o routeamento outros computadores que não sejam o emissor e o receptor, podem aceder aos dados. Mesmo os computadores não envolvidos directamente no routeamento podem alcançar os dados. A segurança é consequentemente um componente crítico de toda a aplicação Internet ou Intranet.

Portanto, emitir dados através de uma rede envolve três riscos básicos da segurança:

- *Eavesdroppings*–intermediários escutando conversas confidenciais (um utilizador falando com outro);
- Manipulação – intermediários alterando informação em uma comunicação confidencial;
- Imitação – um emissor ou receptor comunica-se sob a identificação falsa.

A situação é análoga na compra de bens encomendados por *E-mail* usando o telefone. Os compradores da encomenda-mail querem saber que nenhum terceiro partido pode ouvir seu número de cartão de crédito (*eavesdropping*); que ninguém pode introduzir informação extra da ordem ou mudar o endereço de entrega (manipulação); e se é realmente da companhia da encomenda-mail na outra extremidade da linha e não de um ladrão do cartão de crédito (imitação).

1.3.1.1 A necessidade de segurança

O facto de qualquer que seja a forma de como o número é enviado, - quer seja por telefone, quer pela Internet ou por fax - vai ter de aparecer num computador, cria problemas quanto ao anonimato do comprador.

Um estudo da GTRC de 1995 (Rebordão, 1997), revelava que uma das preocupações dos compradores virtuais era o facto de não ser seguro e até insensato dar o número do cartão de crédito através da Internet. Grande parte dos clientes mencionava a segurança das transacções como a razão principal para não comprarem *online* preferindo linhas telefónicas e de telefaxes gratuitas. Para 82% dos clientes da Internet, a segurança da informação é uma questão vital, sem garantia da qual, será muito difícil implementar verdadeiras soluções para CE.

1.3.1.2 Protocolos de segurança

Um protocolo codifica e descodifica comunicações de mensagens para transmissão *online*. Protocolos de segurança geralmente proporcionam também autenticação. Os protocolos de segurança que têm emergido sobre a *Web* são SSL, NCSA's (*Nacional Center for Supercomputer Applications*) S-HTTP, PCT (*Private Communications Tecnology*) e o IPSec (*IP SECurity*). *Web browsers* e servidores são esperados para suportar todos os protocolos de segurança popular.

1.3.1.2.1 *Secure Socket Layer* (SSL)

SSL é um protocolo líder para garantir a segurança de quaisquer dados que estejam em trânsito na Internet, desde que ambos, o servidor e o cliente, apoiem o protocolo.
A Netscape desenvolveu o SSL para comunicações entre *Web browsers* e servidores, que utiliza criptografia de chave assimétrica, tornando a comunicação entre as partes virtualmente inviolável. Desta forma, se houver intercepção das informações traficadas entre o cliente e o servidor por parte de pessoas não autorizadas, estas informações serão de utilidade zero, já que seria necessário o conhecimento prévio das chaves privadas de criptografia.

SSL é um protocolo de segurança que providencia [URL-1.5]:
autenticação, **confidencialidade** e **integridade dos dados**, sendo planeado para autenticar o servidor e opcionalmente o cliente. O SSL usa como protocolo de transporte o TCP (*Transmission Control Protocol*), que providencia uma transmissão e recepção fiável dos dados. Uma vez que o SSL reside no nível *Socket*, ele é independente das aplicações de mais alto nível. Como tal, o SSL

pode providenciar serviços seguros para protocolos de alto nível, como por exemplo HTTP, Telnet, NNTP, ou FTP e TCP/IP.

O SSL consiste em dois protocolos (Rebordão, 1997):

- *SSL Handshake Protocol* - usado para negociar os parâmetros de segurança na conexão SSL.
- *SSL Record Protocol* - especifica o encapsulamento de todas as transmissões e recepções de dados. Faz parte das negociações entre o cliente e o servidor, o emissor poder identificar qual algorítmo de cifragem suportado.

1.3.1.2.2 Secure Hyper Text Transfer Protocol (S-HTTP)

O S-HTTP fornece a transferência de dados cifrados entre o cliente e o servidor (em ambas as direcções) de uma forma segura e também permite ao próprio servidor autenticar-se perante o cliente.

S-HTTP providencia serviços de segurança para transacções HTTP. Durante as negociações entre o cliente e o servidor uma variedade de algorítmos são providenciados. Por exemplo, o utilizador pode seleccionar se quer a pergunta e a resposta assinadas digitalmente, cifradas ou ambas. Qualquer mensagem pode ser assinada, autenticada, cifrada, ou qualquer combinação destas. Os mecanismos de gestão de chaves incluem também *passwords* e troca da chave pública.

1.3.1.2.3 S-HTTP versus SSL

S-HTTP e SSL usam diferentes aproximações para providenciar serviços seguros para utilizadores *Web*. O SSL executa a negociação do protocolo para estabelecer uma conexão segura ao nível do socket. Os serviços de segurança são transparentes ao utilizador e à aplicação.
Protocolos S-HTTP estão integrados com HTTP e estão ao nível de aplicação. Estes serviços estão disponíveis somente para conexões HTTP, e a aplicação está bem ciente dos serviços S-HTTP.

1.3.1.2.4 Open Trading Protocols (OTP)

A Oracle e quase uma dúzia de outros vendedores estabeleceram recentemente o OTP, um *stand*ard global para todos os formulários do comércio na Internet. Ele permitirá uma estrutura consistente para os formulários múltiplos de comércio na Internet e fáceis de usar.

OTP especifica como as transacções comerciais na Internet podem ocorrer facilmente, com segurança e eficientemente para ambas as partes, independentemente do método de pagamento, e são muito similares ao ambiente comercial do mundo fisico. Ele suporta muitos protocolos existentes, incluindo a Transacção Electrónica Segura (SET), um standard global da indústria para pagamentos seguros sobre a Internet.

OTP complementa estes protocolos fornecendo um conjunto de regras claramente definidas que cobrem (Oracle Magazine, 1998):

- ofertas para a venda;
- acordos para compra;
- pagamento (usando protocolos de pagamento existentes, tais como SET) ;
- transferência de bens e serviços;
- entrega;
- recibos de compras;
- métodos múltiplos de pagamentos;
- suporte para resolução de problema.

1.3.1.3 *Firewall*

Uma barreira de rede (*Firewall*) é um conjunto de sistemas de informação que protege redes internas contra intrusos.

O objectivo de uma *Firewall* é proteger a rede de uma empresa do mundo exterior, ou seja, não dar a todos a possibilidade de aceder aos recursos ou aos serviços explorados em rede TCP/IP ligadas à rede pública. A rede, dita pública, é maioria das vezes a Internet. A vantagem de estar ligado ao resto do mundo é contrabalançada pelo inconveniente de se estar aberto a intrusões criminosas.

1.3.1.4 Garantias de um pagamento seguro

Neste momento, chega-se à conclusão que terá de haver um mecanismo que ofereça segurança na transferência de dados relativos a um pagamento e que suporte diferentes modos de efectuar esse pagamento. A autenticação dos dados, ou seja, provar que o comprador realmente efectuou uma ordem de compra de um determinado bem ou serviço e que o dinheiro virtual tem correspondência com o dinheiro real na posse desse comprador, constitui um aspecto a considerar na criação de sistemas seguros. Estes sistemas deverão assegurar que os consumidores, os vendedores e as

transacções efectuadas permaneçam confidenciais de forma a garantir o anonimato dessas entidades ou eventos.

1.3.1.5 Consequências

Se esta forma de pagamento se tornar segura, o dinheiro electrónico tornar-se-á mais seguro do que o dinheiro de papel. Em caso de roubo do dinheiro electrónico, basta invalidar os números de série que o caracteriza. O utilizador poderá ainda, controlar melhor onde gasta o seu dinheiro em qualquer hora e dia do mês bastando para isso, reconstruir o *log* do *E-mail* para ver para onde enviou o seu dinheiro.

Garantida a segurança na Internet, afluirão muitos clientes e consequentemente muitos mais fornecedores pretenderão também, estar aí presentes.

1.3.1.6 Formas de Facturação

Os primeiros pagamentos que assistimos na Internet efectuavam-se de forma convencional. Os interessados mandavam transferir o montante da sua conta bancária para a conta da entidade que presta o serviço, com o acesso a uma base de dados. Um processo de pagamento moroso, especialmente nas transacções de um País para o outro.

Também se foi vulgarizando na Internet o recurso a cartões de crédito. Ao pretender um determinado bem ou serviço o utilizador tem apenas de enviar dados do seu cartão de crédito para o fornecedor e a organização bancária responsável pelo cartão efectua a transacção. No entanto ao fazer esta operação corre sérios riscos.

Como os pormenores dos cartões são enviados pela Internet, o fornecedor não tem a certeza de que o cartão de crédito é daquele cliente. Por outro lado, estes dados ao passarem por uma série de sistemas de informação podem ser interceptados e utilizados de forma criminosa.

Ainda que as empresas de cartões e os comerciantes considerem que a fraude e abuso de cartões não seja em número demasiado assustador, torna-se imperativo criar um sistema capaz de garantir aos utilizadores que a informação de carácter privado não seja interceptada por ninguém.

Numa outra forma de efectuar uma compra pela Internet, o cliente preenche um formulário de encomenda electrónico, mas não inclui o número do cartão de crédito.

O comprador deverá telefonar ao cliente para obter informações sobre o cartão e confirmar a encomenda. Em outros casos, o cliente encomenda produtos *online* de uma empresa onde tem um limite de crédito aprovado previamente. Este tem um número de identificação para que as

encomendas sejam processadas e enviadas. As encomendas serão confirmadas por telefone ao cliente.

1.3.2 Os Sistemas monetários seguros

Para se comprar coisas pela Internet é necessário arranjar uma forma adequada de efectuar o pagamento. No mundo real existem muitas maneiras de pagar: dinheiro, cartões bancários, cartões de crédito, cheque, senhas, etc... Da mesma forma na Internet foram criados vários sistemas de pagamento.

1.3.2.1 Cartão de crédito protegido

Uma das primeiras formas de pagamento na Internet foi o uso de cartões de crédito. Trata-se dum sistema que já existe no mundo real, que é usado por milhões de pessoas e que permite efectuar compras em qualquer parte do mundo, desde que seja aceite pelo comerciante.
A informação do cartão pode ser codificada recorrendo ao uso de criptografia tornando assim possível, a utilização do cartão *online* de forma segura. Contudo, continua a não ser a forma de pagamento ideal para transacções de baixo montante.

1.3.2.2 A terceira entidade

Como pagar para quem não possui ou não gosta de usar cartões de crédito? A resposta reside no dinheiro virtual. O dinheiro foi uma invenção espantosa, pois, antes, todo sistema comercial se baseava em trocas. Para obter um bem era necessário trocá-lo por outro bem, o que tinha inúmeros inconvenientes. No entanto o dinheiro só tem valor porque lhe é reconhecido esse valor (pelo estado, entidades bancárias, ...). No início muitas pessoas preferiam continuar com o sistema de trocas. Levou algum tempo para que os mais precavidos reconhecessem o valor do dinheiro. Foram as vantagens deste (menor peso e volume) assim como uma intervenção por parte da entidade emissora que conduziram ao seu uso generalizado .

A criação de dinheiro virtual torna-se o passo seguinte nesta evolução financeira. O dinheiro virtual tem muitas vantagens: não ocupa espaço, não tem custos de emissão, não se desgasta e não se pode perder. Mas para ser bem sucedido o dinheiro virtual precisa de ser seguro, rápido e simples de usar. Vários sistemas de dinheiro virtual foram criados. Cada um tem as suas vantagens e não é claro qual será aceite de forma generalizada. Como exemplo destes sistemas temos o *NetCash*, o *Netbill*, o *Netchex*, o *Netcheque*, o *Netmarket* e o *Magic Money*, entre outros...

Existem actualmente disponíveis soluções que têm como base a utilização de um sistema de chaves privadas e públicas, associadas a um processo de recolha, certificação e aprovação ou não de um pagamento, rigorosamente supervisionado por uma entidade terceira. Uma delas, o *Ecash*, desenvolvido pela empresa holandesa Digicash (1990), é a mais parecida com dinheiro real (dinheiro físico). O seu funcionamento é o seguinte: bancos operando em *Ecash*, emitem dinheiro com "moedas" digitais, por troca de dinheiro real. O utilizador move as moedas do seu banco para o seu disco rígido e, quando quiser fazer um pagamento de um serviço ou de um produto *online* utiliza-as. O pagamento é rápido e o utilizador pode provar que o efectuou.

1.3.2.2.1 *Ecash*

O *Ecash* engloba os melhores atributos dos vários sistemas e com poucas técnicas de criptografia necessárias para utilizar qualquer um dos outros. O levantamento processa-se como se fosse feito numa caixa multibanco e os pagamentos são como se fossem ao balcão, permanecendo o comprador anónimo. Desenhado para permitir efectuar a partir de um computador pessoal (PC), pagamentos seguros para qualquer estação de trabalho, através da Internet ou *E-mail* oferecendo uma privacidade do dinheiro e uma alta segurança na sua utilização. Quando o utilizador pretende fazer uma compra numa loja presente na Internet que aceite *Ecash*, o PC envia as moedas necessárias para o seu pagamento. Por seu turno, quem as recebe envia-as para o banco e este verifica a sua validade. Se forem válidas, credita-as na conta do receptor do pagamento.

Facilidades oferecidas pelo *Ecash*

Quando um pagamento é efectuado a loja está *online* com o banco e a verificação das moedas é efectuada imediatamente pelo que, os bens e/ou serviços comprados são imediatamente enviados para o cliente.

Com esta forma de pagamento evita-se abrir conta na loja onde se efectua uma compra ou de enviar informação sobre cartão de crédito.

Outro aspecto interessante é o facto de cada moeda só poder ser utilizada apenas uma vez, pelo que não será bem sucedida qualquer tentativa de usar cópias de moedas. Assim, quem receber e pretender gastá-las terá de fazer um novo levantamento.

O dinheiro nunca está perdido: se as moedas forem acidentalmente apagadas do disco rígido, basta enviar os números (únicos) das séries perdidas para que este seja novamente devolvido ao utilizador.

Por último, o facto de o banco poder identificar o receptor permite minimizar as possibilidades de utilização de *Ecash* de forma ilícita. Por ser um sistema simples o *Ecash* tem bastantes probabilidades de ser bem sucedido (Revista Internet da Telepac, 1996).

Exemplos de instituições Bancárias que aderiram ao *Ecash*

O Mark Twain Bank foi dos primeiros bancos na Internet a utilizar o *Ecash*. Trata-se de um banco real a funcionar na Internet e qualquer pessoa pode nele abrir uma conta, enviando dinheiro real e recebendo *Ecash* em troca. O dinheiro fica depositado neste banco como em qualquer outro. Se o cliente em qualquer momento assim o solicitar o banco envia-lhe o seu dinheiro, ou parte deste, por cheque.

O Deutsche Bank aderiu a esta forma de pagamento, ainda que a título experimental.

1.3.2.2.2 *First Virtual*

A *First Virtual* actua, tal como o *Ecash*, como intermédiaria entre clientes e fornecedores.

Para abrir uma conta o futuro cliente preenche um documento por *E-mail* (que contém entre outros dados, o dinheiro que pretende depositar e o endereço do seu correio electrónico) e depois envia o seu número de cartão de crédito por telefone para evitar as questões de segurança na Internet. Assim que receber o número de identificação (ou número da conta), pode adquirir bens físicos ou informação em qualquer servidor *First Virtual* compatível, sendo o pagamento dos mesmos feitos através do cartão de crédito indirectamente por esta terceira entidade não havendo hipótese de ocorrer qualquer tipo de fraude.

Segurança com a *First Virtual*

Ainda que o número da conta *First Virtual* não seja confidencial, dado que todas as transacções feitas com essa conta são confirmadas por *E-mail* pessoal, a segurança reside nessa confirmação: se alguém descobre o número da conta de algum cliente não poderá completar qualquer transacção porque não tem acesso ao *E-mail* desse cliente.

Porém, a possibilidade de fraude ainda existe: é possível começar uma transacção e depois fingir a confirmação da compra esperada por parte do titular da conta. A solução encontrada para impedir estes actos criminosos resultou na atribuição de um número único para cada transacção: o número é enviado para o titular da conta e quando este envia a confirmação da compra autorizando a transacção, envia também esse número único.

Trata-se de um sistema preparado para lidar com fraudes potencialmente grandes sem recorrer ao uso de assinaturas digitais.

É de notar que o serviço de venda de informação que a *First Virtual* oferece tem uma audiência limitada uma vez que parte da atracção da Internet é que a informação sobre quase todos os assuntos é gratuita.

1.3.2.2.3 *NetCheque*

O *NetCheque*, desenvolvido pela Universidade da Carolina do Sul, é outra forma de pagamento electrónico na Internet. Tal como os cheques de papel, os *NetCheque* são simplesmente documentos (electrónicos enviados por *E-mail*) que incluem: o nome do pagador (o titular da conta), a identificação da instituição financeira, o número da conta bancária, o valor do cheque e o nome de quem o vai receber. Para sua autenticação cada *Netcheque* possuí uma assinatura digital, ou seja um código criptografado.

Quando depositamos, estes cheques autorizam a transferência da quantia mencionada, de uma conta para outra.

De notar, que todo o processo está protegido por um sistema de criptografia *Kerberos*.

1.3.2.2.4 *CommerceNet*

A *CommerceNet*, uma organização composta por empresas responsáveis pelo desenvolvimento do comércio na rede, criou um certificado electrónico baseado numa tecnologia de encriptação pública – utilização de assinaturas digitais.

O certificado é emitido pela *CommerceNet* e é utilizado para autenticar o comprador e o vendedor em qualquer transacção. Os computadores dos intervenientes podem processar o certificado e completar ou abortar a transação.

O sistema de segurança que a *CommerceNet* desenvolveu baseia-se no Secure-Hyper Text Transfer Protocol (S-HTTP), acima apresentado.

1.3.2.2.5 *Cybercash*

O *Cybercash*, outra forma de pagamento electrónico, permite o uso de cartões de crédito e dinheiro virtual.

Quando se procede a um pagamento é feita uma ligação ao servidor em que está contida a informação sobre o cartão de crédito. A conta do utilizador é verificada e a transacção aprovada. Os dados do cartão de crédito são enviados para a entidade credora, de forma codificada para que ninguém possa ter acesso à informação.

A lista de serviços que a *Cybercash Inc.* oferece inclui não só transacções seguras com cartões de crédito, mas também com cheques electrónicos e com pequenos movimentos financeiros.

A Internet pode então, suportar grande variedade de mecanismos electrónicos de pagamento tal como uma loja normal onde existem várias formas de pagamento disponíveis. Assim os sistemas do CE deverão ser (Marcos,1998):

- intuitivos e fáceis de usar;
- seguros e garantir a integridade das operações;
- suportar máquinas de navegação e buscas;
- garantir contacto individualizado;
- realizar cálculo exacto e imediato de taxas/impostos;
- realizar tratamento estatístico e de simulação;
- recuperar autonomamente de erros e falhas.

Estas características são essenciais se se pretende alcançar níveis elevados de aceitabilidade da parte dos utilizadores (comprador e vendedor).

1.4. A GLOBALIZAÇÃO DO CE

Para uma melhor expansão do CE são decisivos os seguintes factores:

i) Explosão da Internet com a introdução de um número crescente de computadores com acesso à rede e tecnologia ao alcance do grande público, tornando-a mais barata e fácil de usar.

ii) Evolução das tecnologias de pagamento automático, procurando melhorar os métodos de codificação de dados e de pagamento electrónico (cartões de crédito) via Internet realizável com máxima segurança.

iii) Campanha de Divulgação e Apoio, com a finalidade de procurar um maior envolvimento das Médias, Associações, Universidades e Institutos de I&DT e realizando também programas de apoio nacionais e estrangeiros.

iv) As empresas devem estar consciencializadas para o CE procurando obter maior número presente na *Web*, colocando *online* ofertas de serviços/produtos adquirindo maior confiança nas novas tecnologias e no potencial de negócios em perspectiva.

1.4.1 Evolução de Internet *Hosts*

Uma razão principal da *Web* estar "quente" como um comércio médio é por causa de suas perspectivas actuais de crescimento no tamanho e do futuro desenvolvimento e da demografia extremamente atractivo.

A Figura 1.4 mostra, abaixo, o crescimento em *hosts* Internet de 1989 a 1997. O *host* de nome WWW é o mais predominante na rede implicando que muitos *hosts* são servidores *Web*.

Como de Julho de 1995, havia 6,64 milhões de computadores *hosts* na Internet [URL-1.6]. Este número tem duplicado anualmente desde 1981. A mesma fonte mostra que 2,37 milhões destes são *hosts* internacionais conectados à Internet, representando 150 países, demonstrando que a Internet é verdadeiramente um fenómeno global.

O crescimento em redes Internet-conectadas é também impressionante. Em Janeiro de 1989, havia 213 redes nos Estados Unidos (EUA) e 34 redes conectadas à Internet fora dos EUA. Seis anos mais tarde, em Janeiro de 1995, lá existiam 26.681 redes e 19.637 redes internacionais.

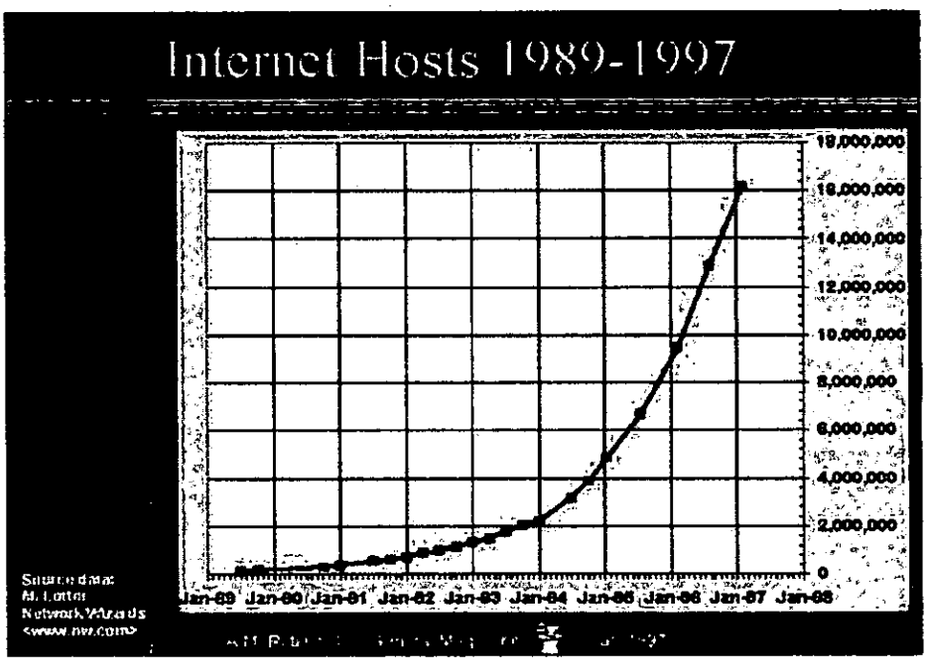


Fig1.4 - Crescimento de Internet Hosts

1.4.2 Evolução da Interacção

Passo 1:

Para toda a empresa, o primeiro e maior passo produtivo para a participação na economia em rede é usar internamente métodos de ligação electrónica para melhorar as comunicações, fluxo de informação, partilha de conhecimento e cooperação. O ambiente interno, que pode ser gerido e controlado com um mínimo de complexidade, representando um lugar previlegiado para o

empregado, gerente e executivo aprender novas técnicas e ferramentas. Intranets fornecem uma base boa para esta aprendizagem.

As ferramentas electrónicas básicas de ligação incluem o correio electrónico (*E-mail*), forums para discussão *online*, planilhas de observação electrónicas e sistemas de partilha de informação (incluindo WWW), ver figura 1.5.

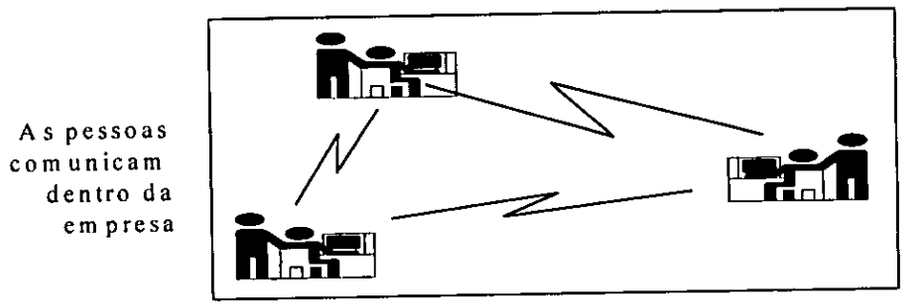


Fig.1.5 - Passo 1

Passo 2:

Uma vez que as pessoas comecem a usar com mais confiança as novas tecnologias, a etapa natural seguinte é começar a estar electronicamente ligado à rede com contactos existentes fora da empresa. Estes contactos podem ser fornecedores, clientes, conselheiros de especialista, oficiais de governo - qualquer um com quem a companhia necessita se comunicar a fim de fazer o negócio e/ou simplesmente negociar serviços.

Certamente, mesmo antes que haja uma decisão para promover internamente a ligação electrónica, cada empresa tem alguns empregados e gerentes que já começaram a ligar-se ao exterior, usando a Internet e outros serviços para fazer contactos e resolver problemas. A estratégia produtiva é estender a experiência destes iniciadores *self-self-starting* de modo que todos no negócio estejam usando estas técnicas (ver figura 1.6).

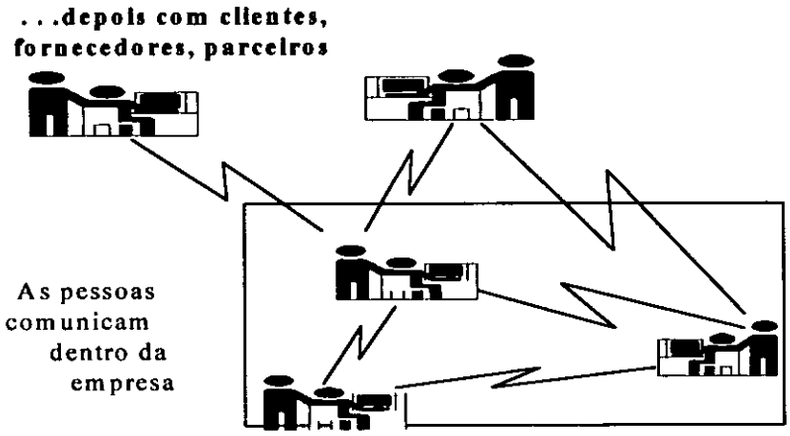


Fig.1.6 - Passo 2

Passo 3:

Após se atingir um estado de familiarização elevada com as novas tecnologias, o passo final é aplicá-las num domínio mais direccionado ao *spectrum* inteiro de comunicações, comércio e marketing. Tipicamente, isto envolve estabelecer um *Web site* interactivo, rico em informação, que actue como a porta "dianteira" da companhia para clientes, fornecedores e sócios de negócio existentes e potenciais clientes, os investidores, etc.

Assegurando-se que todos tenham uma postura tipo "*online* activo", através da ligação electrónica interna e externa, ter-se-á criado uma base de conhecimentos e experiências que permitirá a entrada da companhia na *World Wide Web*, com perspectivas de sucesso (ver figura 1.7).

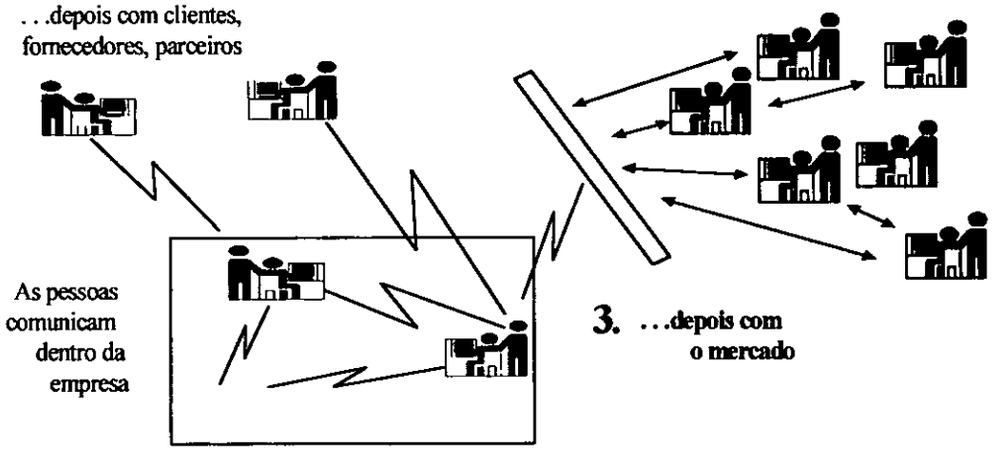


Fig.1.7 – Passo 3

1.4.3 Evolução de Negócios

Em 1997 as pessoas compraram muito pela Internet. Só nos Estados Unidos da América as vendas atingiram mais de 1,4 bilião de dólares. As transacções entre empresas passam de 5 biliões (estudo da DataQuest) [URL-1.2].

Para 1999, prevêem-se receitas na ordem dos 7.7 mil milhões de dólares; 14,8 mil milhões para o ano 2000.

Um relatório elaborado pelo Cebit´98 - um certame tecnológico que se celebra em Hanôver (Alemanha) - estima que no ano 2000 cinco de cada doze escudos do comércio mundial se venderão através da Internet.

A evolução faz-se a grande velocidade. No final de 1997 havia 87 milhões de computadores em todo o mundo, 71% mais que em 1996, e no início da próxima década a cifra mundial de negócios poderá chegar aos 39,6 biliões de escudos, isto é, o comércio na Internet vai atingir mais de US$ 100 biliões até ao final do século e um crescimento contínuo é esperado daí em diante, prevendo-se até 800.000 empresas na Internet.

Em Moçambique o CE está a dar os primeiros passos, estando na fase de simples presença passiva na *Web*.

1.4.4 Vantagens do CE

Para o Cliente:

- Flexibilidade Temporal/Geográfico (casa, serviço, quiosque), proporcionando uma grande redução nas deslocações e no tempo de resposta;
- Leque de escolha ilimitado e permanentemente actualizado;
- Preços mais baixos;
- Relação personalizada com o vendedor fornecendo soluções individuais.

Para a Empresa:

- Acesso à Montra Global (novo canal de distribuição), permitindo a abertura de novos mercados, possibilitando assim o aumento de vendas.
- Melhorar gestão de clientes/produtos implicando um tratamento estatístico mais fácil e com informação bem estruturada;
- Redução de custos fixos (instalação, stocks, armazenamentos);
- Venda self-service;
- Competitividade equilibrada.

1.4.6. Exemplos de Áreas de Sucessos

No Sector livreiro (ex. Amazon.com), a maior livraria do mundo, é um dos melhores exemplos da indústria do comércio electrónico no trabalho. Ela permite que os usuários da Internet pesquisem numa base de dados alargada e usem então um cartão de crédito para requisitar as seleccionadas. A maioria dos 2,5 milhões de títulos da Amazon pode ser entregue ao domicílio dentro de poucos dias.

Através de sua rede de distribuição, pode-se adquirir aproximadamente 400.000 livros dos melhores *best-selling* durante a noite.

A *Web* é um grande equalizador para as novas empresas que enfrentam grandes concorrentes. No caso da Amazon, a imensa população de potenciais clientes sobre a *Web* tem tornado possível para uma loja simples trazer para casa um inventário físico que de outra maneira seria quase impossível iniciar esta operação.

A Amazon mergulhou no mercado de venda livreiro, a níveis só possíveis somente para empresas comerciais muito maiores. Outras áreas de sucesso são [URL-1.2]:

a) Na área de Software/Hardware a Dell, uma empresa Americana através do seu *Web* site vende mais de 3 milhões de dólares em computadores por dia;

b) Nos Serviços Técnicos de Apoio (exemplo - Barnes&Nobles);

c) Para obtenção de Informações úteis acerca de detalhes sobre carros (exemplo - Microsoft - Car Point);

d) Na aquisição de Jogos *Online* (Microsoft – Internet GameZone), que facturou 60 milhões de dólares em 1997 e deve chegar a 670 milhões até ao ano 2001 (pesquisa da Cowles/Simba Information). O "Zone" possui mais de 800.000 usuários registados.

Capítulo II
Enquadramento do Trabalho

O recurso a redes mundiais de computadores para a divulgação de informação é uma realidade que é necessário aderir com rapidez, de modo a acompanhar as evoluções mundiais em todos os campos da investigação. É com base nesta tendência que surge a ideia de implementar uma solução no ciberespaço, para a gestão de feiras e exposições, onde se possa explorar a componente comércio electrónico, surgindo assim o ExventShop, umas das principais componentes integrantes do ambiente Exvents.

AMBIENTE EXVENTS

Nos últimos 3 anos, as tecnologias da Internet têm empreendido um salto quantitativo, devido ao rápido desenvolvimento da mesma, e mais especificamente, a *World Wide Web*. Como consequência, eventos de vários tipos, tais como conferências, exibições, *workshops* e muitos outros - no seguimento designados "**Exvents**" (eventos como exposição) - são baseados em servidores *Web*, especificamente desenvolvidos para tal propósito. Normalmente esses servidores são desenhados por *Webmasters*, cujas tarefas típicas são de construir e manter *Web sites* para empresas, comercial e/ou departamentalmente orientados.

Contudo, a planificação, preparação, organização e execução de um evento exigem normalmente necessidades especiais com complexidades variadas.

Neste contexto tarefas típicas do ponto de vista logístico são:

- A planificação global do espaço para a exposição;
- A alocação de *stands* e sua atribuição aos expositores;
- A preparação e gestão do catálogo electrónico de produtos e serviços oferecidos pelos expositores;
- A planificação do calendário dos eventos, no que diz respeito aos expositores no Exvent bem como o horário de cada visitante individual;
- Um flexível e efectivo sistema de suporte na forma de agentes inteligentes pre*stando* ajuda ao visitante e ao expositor com todas as tarefas acimas relacionadas e assegure a integridade do sistema e de todos os seus dados.

O objectivo principal do projecto Exvent é a análise dos requisitos e necessidades inerentes de um Exvent e desenhar um ambiente distribuído baseado na tecnologia *Web*, proporcionando ferramentas certas para reduzir a complexidade logística inerente a tarefas de gestão global de um Exvent, o que inclui suporte à ligação de associações complexas entre diferentes entidades envolvidas.

Outro requisito importante deste projecto é a capacidade de também suportar eventos que têm somente lugar no "ciberspace" (exposição virtual) e não em qualquer localização física. Além disso, o projecto Exvent deve permitir uma perfeita sincronização entre eventos virtuais e físicos. A existência baseada na Internet reduz também as dificuldades surgidas por distâncias físicas de vários participantes na preparação e gestão de Exvents.

A futura fase é o desenho de um agente inteligente que ajuda com à planificação e preparação da visita, particularmente os horários. Já se começou também a juntar e evoluir ideias acerca de que características avançadas deverão ser incluídas para um sistema Exvent completo.

2.1 ARQUITECTURA DO AMBIENTE EXVENT

A figura 2.1 dá-nos uma visão compacta da arquitectura do sistema, onde o elemento central é o servidor *Web* que proporciona um interface comum entre visitantes, expositores, organizadores e os diferentes serviços fornecidos pelo sistema, a maioria baseada em HTML e Java.

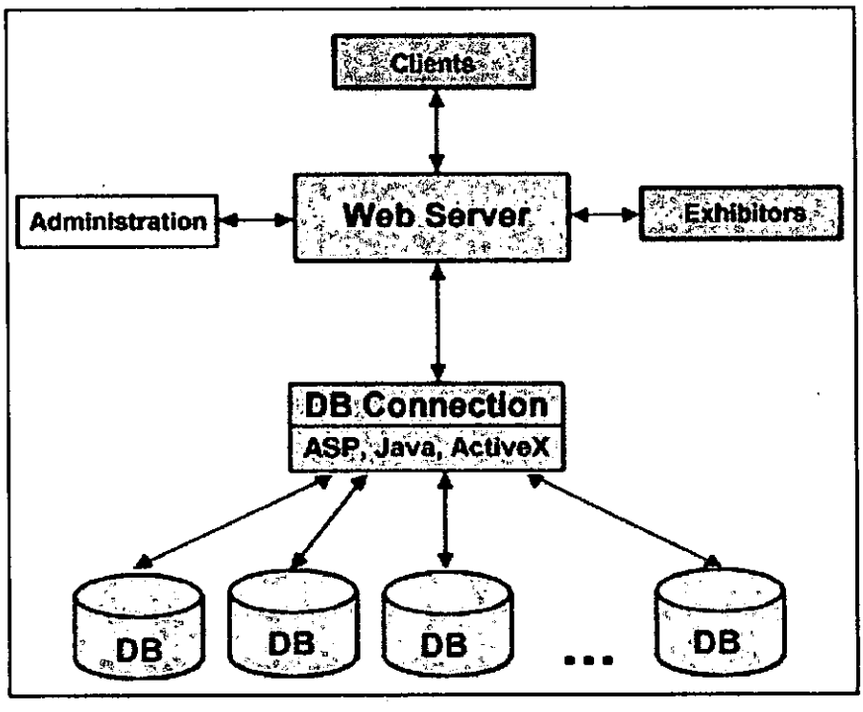


Fig.2.1 - Arquitectura do Exvent

O sistema de base de dados suportado pode ser distribuído sobre uma rede. O servidor *Web* faz a ligação à base de dados usando tecnologias tais como *Java DataBase Connectivity (JDBC)*, *Open DataBase Connectivity (ODBC)*, *Active Server Pages (ASP)*, Java ou *ActiveX*.

De acordo com a acção do utilizador, o sistema proporciona diferentes serviços. Os organizadores do Exvent têm a capacidade de reconstruir o espaço e o horário dum evento através de uma fácil e intuitiva interface que consiste principalmente num *applet* em Java como um *front - end* gráfico (figura 2.2).

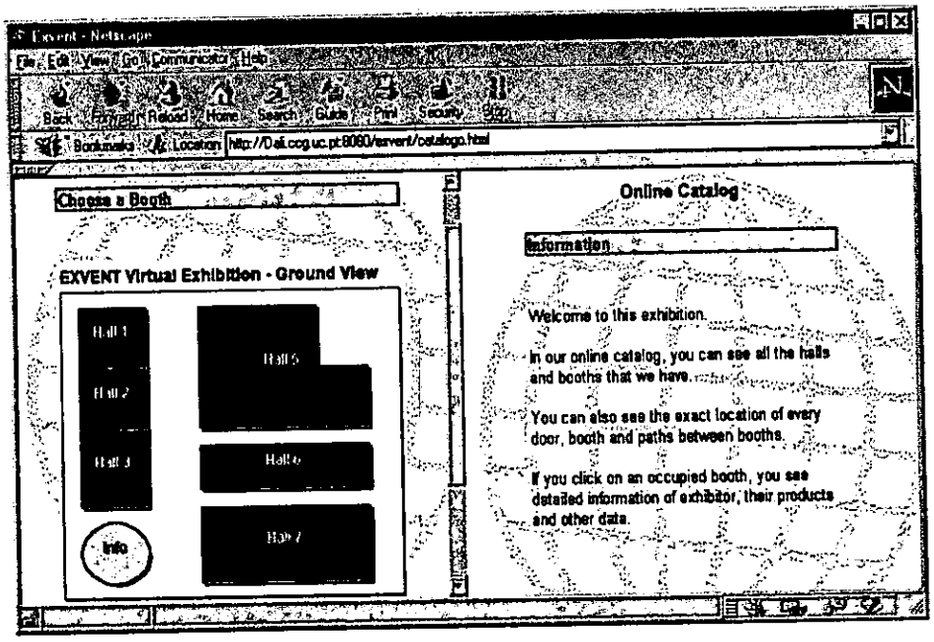


Fig.2.2 - O espaço dos pavilhões

O *applet* recebe os dados relacionados com o pavilhão, *stands* e a exibição, de um servidor de base de dados num formato vectorial. Isto permite uma fácil representação gráfica e escalar dos modelos, presentemente suportado em 2 dimensões. Por outro lado, o sistema fornece ao expositor todos os serviços necessários para realizar muitas das tarefas logísticas relacionadas com a escolha, planificação e gestão de um *stand*. Isto inclui a adição e alteração da informação que diz respeito aos produtos e serviços apresentados e/ou disponível em um *stand*, ou da respectiva empresa em si.

2.2 O DESENVOLVIMENTO DO EXVENTSHOP

O ExventShop é ainda um protótipo de uma loja virtual para exposição e venda de produtos, que se encontra incorporado no Exvent.

Correntemente, um genérico *front-end* visual em Java que permite explorar o espaço do evento para organizadores, expositores e visitantes, está implementado no Exvent e que também é usado

para aceder ao ExventShop a partir do Exvent, uma vez que eles usam uma base de dados comum (veja figura 2.3).

O módulo do visitante permite o planeamento confortável e eficiente de uma possível visita ao evento. O acesso a toda informação relevante ao visitante é feito através dum catálogo *online*, módulo de pesquisa (ver figura 2.4).

Existem também páginas genéricas (e dinâmicas) para pesquisas *online*, bem como para *uploads* da informação e manutenção. Elas são implementados em ASP.

Também está implementada a parte do ExventShop que permite a ligação com a base de dados usando JDBC e ODBC.

No capítulo seguinte serão apresentado em detalhe o desenvolvimento do ExventShop.

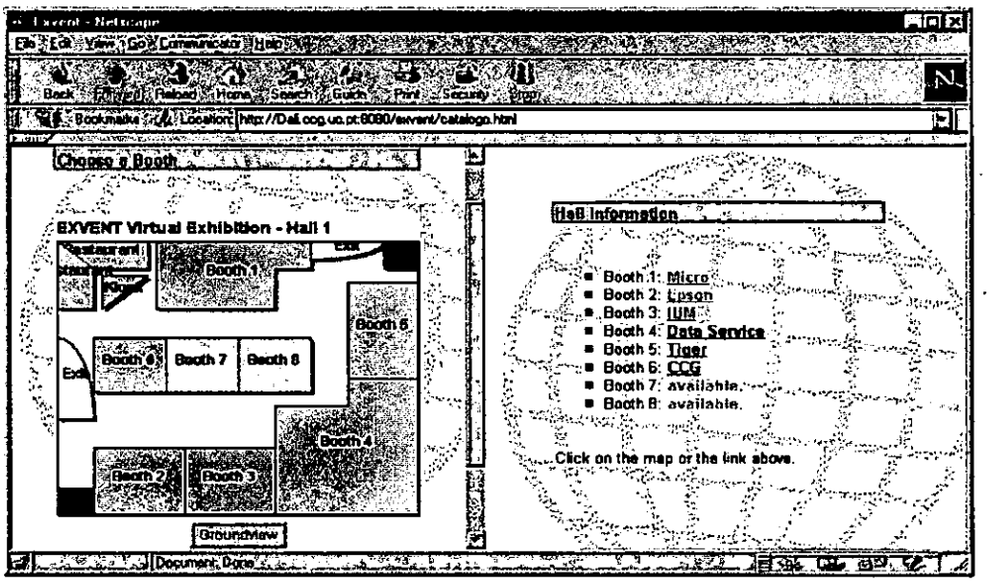


Fig.2.3 - Os *Stands Online*

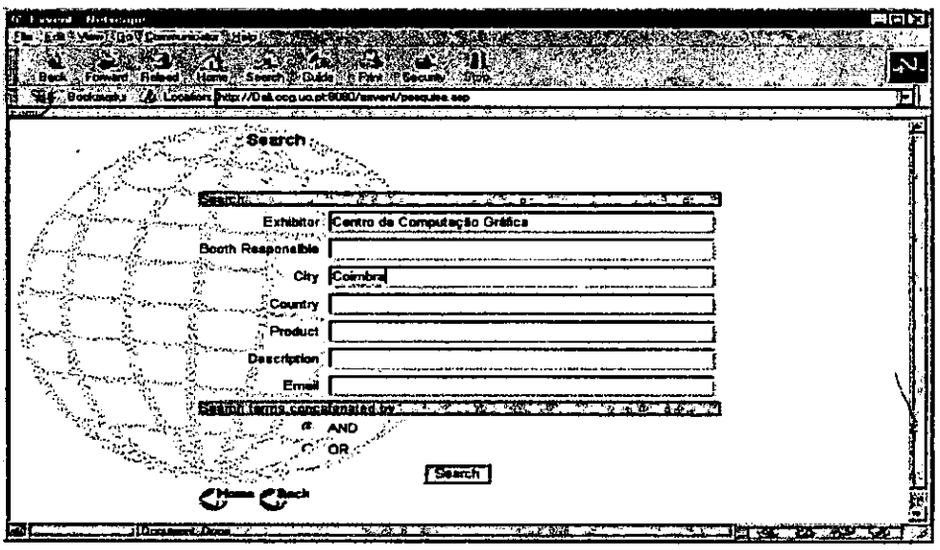


Fig.2.4 - Módulo de pesquisa

Capítulo III
Interface e Linguagens de Programação

3.1. HTML

O HTML (*HyperText Markup Language*) é uma particularização para hipertexto da linguagem genérica SGML (*Standard Generalized Markup Language*).

Desde o início da *World Wide Web*, o HTML foi adoptado como a linguagem *stand*ard para o desenvolvimento de páginas *Web*, devido à sua simplicidade e portabilidade.

A sua simplicidade surge do facto de a programação em HTML ser feita através de *tags* (marcas) que indicam o começo (e o fim) das várias entidades sintácticas (como por exemplo, *links*, tipo de letras, estilo de letras, cores, tabelas, formulários, etc). Sendo assim, não é necessário ter qualquer tipo de conhecimento de linguagens de programação usuais, pelo que o HTML se torna numa linguagem acessível a quase toda gente.

Apesar de os ficheiros HTML serem escritos em modo texto em ficheiros *ASCII* (e por isso bastaria um editor de texto como o *EDIT* do *DOS)*, já existem editores WYSIWYG (*What You See Is What You Get*), que permitem que um utilizador vá visualizando o resultado final à medida que introduz novos elementos na sua página, tornando assim a construção de páginas num método trivial para os utilizadores menos experientes. Claro que, para utilizadores mais experientes e que queiram fazer páginas complexas, o melhor método continua a ser o simples editor de texto.

Por outro lado, o facto de a linguagem HTML ser uma linguagem interpretada, isto é, o código HTML só vai ser interpretado quando o *browser* o recebe, tornando possível que um certo documento disponível num determinado servidor, possa ser interpretado em sistemas tão distintos como DOS/Windows, Macintosh ou Unix.

Com o aparecimento de novas linguagens, interfaces e sistemas multimédia, o HTML teve que evoluir de modo a suportar toda esta variedade de serviços. Neste momento está em fase de desenvolvimento a versão 4.0 desta linguagem, cuja especificação pode ser obtida no documento "*HTML 4.0 Reference Specification*" [URL 3-1].

3.2 JAVA

Após um exame exaustivo dos sistemas de hardware, a nível de vídeo, CD e TV, a equipa Green de 6 programadores de elite da *Sun Microsystems* começou o desenvolvimento de uma nova linguagem de programação orientada a objectos a que James Grosling (o grande mentor da *Sun*) chamou Oak. Tendo como base a linguagem C++, a linguagem Oak foi sendo limitada ao mínimo, de modo a ser compatível com o espaço reduzido oferecido pelos dispositivos de controlo, com o objectivo de permitir aos programadores adicionarem novas funcionalidades a um vídeo ou TV.

O primeiro produto a ser desenvolvido com base nesta linguagem foi um dispositivo de controlo-remoto que continha um pequeno interface visual. Este interface, chamado "*7" apresentava um personagem animado, chamado "Duke", que guiava os utilizadores através de interface gráfico. Este personagem, criado por Joe Parlang, viria a ser a mascote do Java.

Em 1993, quando a NCSA introduz o Mosaic (o primeiro *browser* gráfico), e nascem os gráficos da *World Wide Web*, a sorte da companhia mudou.

Em princípios de 1994, a linguagem Oak passou a ser o produto em vez de parte de um dispositivo e foi adaptada à realidade da Internet. Artur Van Hoff escreveu um compilador de Oak; Naughton e Jonnathan Pryce construíram um *browser* de OAK o chamado "WebRunner". A Sun resolve então apostar forte nesta nova linguagem que passou a ser conhecida como Java. Por motivos legais, o *browser* "WebRunner" teve que mudar de nome, passando a "HotJava"[URL-3.2].

A partir do momento em que o *browser* da Netscape começou a suportar Java, milhões de utilizadores puderam (e podem) usufruir das potencialidades desta linguagem.

Potencialidades da linguagem Java

Os programas Java podem ser embebidos de uma forma natural nas páginas *Web* (conhecidos, neste caso, como *applets*) não necessitando por isso de qualquer tipo de interface entre o cliente e o programa, dando assim a possibilidade de interacções em tempo-real, coisa que não acontecia com CGI (*Common Gateway Interface*).

Uma vez que a linguagem foi adaptada à realidade Internet, existe um conjunto de funções que permitem de modo simples e eficaz efectuar operações que noutro tipo de linguagens (usando CGI) seriam extremamente complexas. Um exemplo disso, são as funções que permitem a comunicação e transmissão de dados através da Internet. Assim torna-se bastante fácil estabelecer um protocolo de comunicação entre o *applet* e um programa Java que actue como servidor.

Por outro lado, o facto de os programas escritos em Java necessitarem de uma compilação, não reduz as potencialidades desta linguagem a nível de portabilidade, pois o código binário que é gerado pelo compilador é um código genérico que irá ser interpretado pelo *browser* (no caso dos *applets*) ou pelo interpretador de Java, esses sim, específicos para uma plataforma em que estão inseridos. Assim, um programa Java compilado numa determinada plataforma pode ser interpretado em qualquer plataforma desde que para essa plataforma haja um *browser* (no caso dos *applets*) ou um interpretador de Java.

Todos esses factores poderão vir a contribuir para que a linguagem Java se torne na plataforma *standard* para o desenvolvimento de aplicações – é esse o objectivo da Sun e da Netscape (que fornece o *browser* para a visualização das aplicações).

Contudo, a concorrência não está a dormir, assim, a Microsoft disponibiliza já um sistema similar (o *ActiveX*) que permite utilizar também algumas funcionalidades do Windows em programas embebidos nas páginas *Web*.

A "guerra" está lançada e só o tempo dirá quem vai vencer.

3.3 JAVASCRIPT

A parceria entre a Sun e a Netscape em torno da linguagem Java, conduziu à criação de uma nova linguagem – JavaScript. Contrariamente a algumas opiniões, a linguagem JavaScript não é uma simplificação da linguagem Java destinada aos utilizadores menos experientes. Esta linguagem fornece uma funcionalidade orientada para uma fatia de mercado diferente da atingida pelos *applets* e aplicações Java. O seu principal objectivo é fornecer uma solução para o desenvolvimento de APIs (*Application Programming Interfaces*) a nível do cliente – especificamente o *browser* da Netscape (*Netscape Navigator*).

O JavaScript elimina a necessidade de se escreverem *applets* Java, que pela sua natureza são bastante pesados, para efectuar cálculos simples ou controlar certas funções do *browser*.

Uma das grandes diferenças entre Java e JavaScript reside no facto de os programas escritos em JavaScript serem imediatamente interpretados pelo *browser* através do código fonte, eliminando a necessidade de uma compilação prévia. Contudo, esta funcionalidade pode levar a que a execução de programas em JavaScript seja mais lenta que a execução de programas em Java, principalmente em máquinas menos potentes.

A escolha de Javascript

Apesar de a linguagem JavaScript ser baseada em objectos, algumas das potencialidades a nível de objectos tiveram que ser eliminadas. Assim, em JavaScript não é possível definir classes, sendo apenas possível definir um objecto, os seus métodos e variáveis. Outro dos aspectos eliminado foi a herança dos objectos que em conjunto com as classes permite o melhoramento e reuso dos objectos de uma maneira simples. Claro que, em pequenos *scripts* é raro o reuso de objectos, pelo que estas funcionalidades, se tornam irrelevantes nestes casos.

Por outro lado, os *applets* Java não interactuam com o código HTML de uma página *Web*. Cada *applet* é limitado a uma sub-área da página. Embora um *applet* possa comunicar com outros *applets* na mesma página, não se pode, contudo, mudar o texto da página HTML onde está inserido. Por sua vez, o JavaScript foi criado de modo a permitir a interacção entre diferentes *tags* e elementos de HTML. Por exemplo, um campo de introdução de dados num formulário HTML pode influenciar o texto dentro da outra página HTML. Outro exemplo, e bastante comum, é o uso de JavaScript para fazer detecção de erros nos campos de entrada de dados de um formulário, não havendo necessidade de ligações a servidores para executar programas (usando CGI) que verifiquem a validade dos dados. O JavaScript possui funções que rapidamente fazem a verificação de erros. Com JavaScript também é possível escrever formulários dinâmicos que mudam o ecrã do utilizador dependendo das opções que o utilizador escolhe.

O JavaScript fornece, assim, «*um meio de executar tarefas simples e úteis de modo a melhorar as funcionalidades de uma página Web*»[URL 3-3].

3.4 ASP

Active Server Page (chamado pela abreviatura, ASP) são páginas *Web* com etiquetas especiais, contendo *scripting* (VBscript, Javascript, Perlscript), integradas em HTML com o qual estabelece um interface de comunicação entre uma aplicação externa e um servidor de informação. Isto é chamado "*server-side scripting*".

Uma vez que os servidores de informação suportados são servidores HTTP, o ASP fornece uma poderosa ferramenta de interacção entre o cliente e uma aplicação num determinado servidor da *World Wide Web*.

O ASP não é uma linguagem de programação, mas sim uma tecnologia desenvolvida pela Microsoft para permitir, que qualquer *scripting* de páginas *Web* possa receber e transmitir dados através da WWW.

Um exemplo bastante comum é a acessibilidade de bases de dados através da WWW. Basicamente existe um *script* que está preparado, de acordo com as especificações do ASP, para receber dados (por exemplo uma chave de pesquisa), processá-los (efectuando uma pesquisa na base de dados) e devolver os resultados em formato HTML. O interface entre o cliente e o *script* é feito através de um formulário onde estão incluídos campos de introdução de dados (que irão ser transmitidos ao programa) e um campo que indica qual o programa a ser executado.

A grande versatilidade do ASP reside nos factos seguintes:

- Sua especificação é independente de linguagens de programação, mas sim de *scripts* (podendo ser VBscript, Javascript e, mais recentemente Perlscript);
- As páginas são compatíveis em qualquer *browser*;
- O código fica escondido e protegido de vistas intrometidas, e para quem o queira roubar deverá despender muito tempo na escrita;
- As páginas não são vistas directamente pelo *browser*. Em vez disso o servidor lê as páginas, executa o *server-side-scripting*, e formata as páginas geradas dinamicamente para o *browser*.

As páginas ASP são identificadas por um ficheiro com extensão ".asp" e os *scripts* pelo respectivo nome (VBscript, Javascript ou Perlscript).

O ASP veio assim dar uma nova dimensão à *World Wide Web* permitindo a interacção entre o utilizador e as páginas *Web*, tornando-se numa ferramenta bastante popular, estando mesmo na base de encomendas de produtos *online*, interface com base de dados, etc.

O ASP possue diversos objectos definidos com o qual executa muitas tarefas. Sem ir para muitos detalhes, os mais importantes são:

Request Object

Este objecto contém informação sobre o pedido realizado ao servidor através de um *hyperlink* ou um formulário. Ele inclui:

- Um conjunto de parâmetros passados com o método POST (ver mais adiante);
- Um conjunto de parâmetros na forma de perguntas anexados, ao método GET (ver mais adiante);
- *Cookies* que são passados para um browser. Ele permite que um conjunto de informações esteja associado com um utilizador;
- Certificados do Cliente.

Response Object

Este é um objecto que controla o resultado da página.

Session Object

O ASP tem um objecto especial que pode armazenar informação através das suas páginas.

O *Session Object* é actualmente um *cookie* especial, permitindo identificar e controlar o ciclo de vida de cada utilizador que está dentro da sessão .

Uma sessão é criada quando o utilizador requer uma página da aplicação, onde ele não havia feito antes uma sessão corrente e geralmente termina quando o tempo expira (ou quando o utilizador não tiver feito pedidos de páginas durante um certo intervalo de tempo).

O ASP usa uma tecnologia chamada ***ActiveX Data Objects*** (ADO) para trabalhar com bases de dados. ADO é tecnologia *ActiveX*, construído dentro do ***Internet Information Server*** (IIS), que é o servidor *Web* usado neste trabalho. Ele suporta características chaves para construção de aplicações Cliente/Servidor e baseadas na *Web* [URL-3.4].

Existem 3 objectos principais em ADO, que servem de interfaces aos dados, do quais deve-se estar informado:

O ***Command Object***, ***Connection Object*** e o ***RecordSet Object***.

Muitas das vezes trabalha-se somente com o *RecordSet Object*. Contudo, para fazer uma operação na base de dados, esses 3 objectos estão presentes, mas ele não é frequentemente necessário criar explicitamente todos os 3, todavia quando um é usado, os outros 2 são implicitamente criados, embora não se deva atribuir uma variável para acedê-las. A sintaxe para criação duma variável de acesso a esses objectos é (em VBscript, por exemplo):

"[variable name]= Server.CreateObject("ADODB.[Object name]")"

RecordSet Object

O *RecordSet Object* é basicamente um cursor (uma tabela temporária que existe na memória) com algumas funções e propriedades definidas para trabalhar com os registos nele contidos. Pode-se criar um *RecordSet Object* explicitamente, ou por execução de um camando através do ***Command Object***.

Command Object

O *Command Object* é o cavalo de batalha do ADO. Ele é o objecto que comanda a base de dados, inicia uma declaração de SQL (*Structure Query Language*) ou executa um procedimento armazenado por si próprio na base de dados. Pode criar *queries* parametrizados, alterá-los bem como alterar as propriedades do *Command Object* por si próprio.

Connection Object

O *Connection Object* é o objecto que actualmente faz a conversação com a base de dados. Ele define o tipo de conexão e suas propriedades são muito numerosas para serem mencionadas. Contém um nome do sistema DSN (*Data Source Name*) que é usado para identificar o *driver* ODBC e o caminho para a base de dados. Ele pode também guardar o nome do utilizador (uid) e à *password* (pwd) que é usado para obter acesso para uma base de dados segura. Em caso de um sistema de base de dados Cliente/Servidor, deve equivaler a uma actual conexão a rede para um servidor.

O *Command Object* e *RecordSet Object* actualmente acedem a base de dados através do *Connection Object*. Contudo, não é frequentemente necessário usar este objecto, uma vez que ele pode ser criado implicitamente pelo *RecordSet Object*.

3.5 INTERFACE COM AS APLICAÇÕES

3.5.1 O servidor e a aplicação

Como existe um interface entre o cliente e uma página ASP - normalmente um formulário ou um *link* directo para a página, obviamente tem que existir troca de informação entre o servidor e a página ASP. Esta informação é transmitida de duas maneiras: através de variáveis de ambiente, que contêm dados sobre o dispositivo de interface, e através do *standard input* usado para transmitir os dados introduzidos pelo utilizador.

3.5.2 Descodificação de dados

Os dados que um utilizador introduz podem ser enviados a uma página ASP de duas formas: usando o método GET, onde os dados se irão encontrar com a variável de ambiente *QueryString* ou usando o método POST, no qual os dados são enviados do *standard input*.

Cada um destes métodos tem vantagens e desvantagem. O método GET permite que uma página ASP seja referenciada directamente, não necessitando por isso de um formulário que sirva de

interface, mas tem o problema de os dados estarem contidos numa variável de ambiente, cujo tamanho é limitado e por isso poderá haver perda de informação. Pelo contrário, o método POST ao possibilitar a leitura de dados através do *standard input* permite que não haja restrições quanto ao volume de dados, mas necessita de um interface – como um formulário – para a transmissão desses dados.

Em qualquer dos métodos, os dados enviados para a página ASP encontram-se na forma de "variável=valor", separados pelo caracter '&'. Contudo, o valor de uma variável encontra-se codificado de forma a resguardar caracteres especiais do formato URL *(Uniform Resource Locator)*. Assim, os espaços em branco são transformados no caracter '+' e todos os outros caracteres especiais são enviados usando notação hexadecimal precedidas do caracter '%' (por exemplo: %22 para indicar o caracter aspas).

No entanto é de notar que apesar de os caracteres especiais estarem protegidos, é preciso ter muito cuidado na elaboração das rotinas de descodificação para que não haja falhas de segurança. Uma atenção especial deve ser dada ao facto de não se executar nenhuma chamada directa ao sistema envolvendo um valor de uma variável sem ter verificado cuidadosamente esse valor, pois se isso não acontecer pode dar origem a que certas sequências *escape* encapsuladas nos valores das variáveis causem o servidor executar comandos arbitrários - e dado que ainda muitos servidores correm com acesso total ao sistema, os efeitos podem ser devastadores.

Um exemplo dum formulário em ASP que usa o método POST é mostrado na figura 3.1. Frequentemente ela inclui um *script*, neste caso, o Javascript, que faz a validação dos dados que serão introduzidos como mostra a figura 3.2.

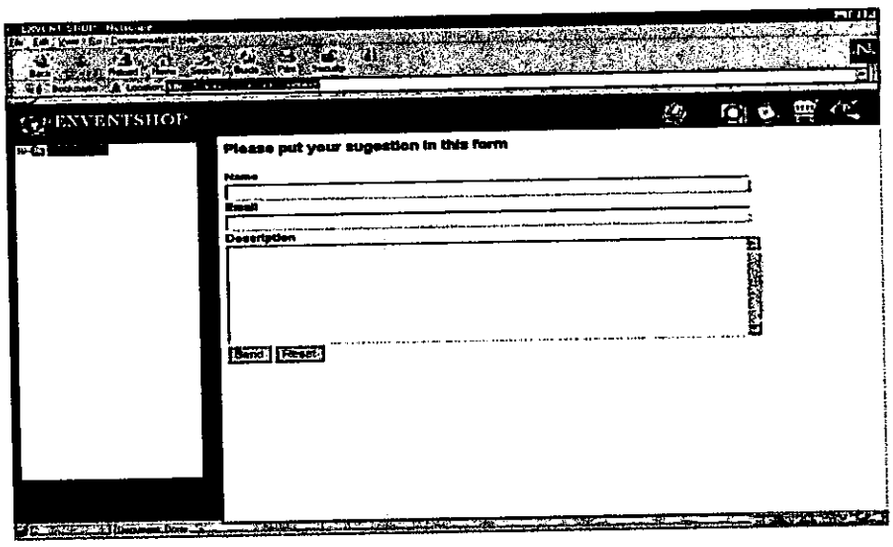


Fig.3.1-Exemplo de formulário

```
<HTML>
<TITLE>SUGESTÕES VIA EMAIL </TITLE>
<HEAD>
<script language="JavaScript">
<!--
function validate_form() {// função usada para validar o formulário
  validity = true; // assume válido
  if (!check_empty(document.form.NAME.value))
      { validity = false; alert('Nome do campo está em branco!'); }
  if (!check_email(document.form.EMAIL.value))
      { validity = false; alert('Endereço de Email é inválido!'); }

  if (!check_empty(document.form.DESCRIPTION.value))
      { validity = false; alert('O campo Description está em branco!'); }
  if (validity)
      alert ("Entrada tem de ser verificado. "
           + "O form agora é passado para seu browser's "
           + "Para entrega do mail ao sub sistema.");
  return validity;
                        }

function check_empty(text) {
  return (text.length > 0); // Se está em branco devolve falso
                              }

function check_email(address) {
  if ((address == "")
    || (address.indexOf ('@') == -1)
    || (address.indexOf ('.') == -1))
      return false;
  return true;
                                  }
// -->
</script>

</HEAD>
<BODY>

<Font face="Arial"size="4"><Strong> Please put your sugestion in this form </Fonte></Strong>
<form name="form" method="POST" action="mailshop.asp" onSubmit="return validate_form()">
<Font face="Arial"size="3"><Strong>Name</Fonte><br><input type="text" size=50
name="NAME"></Strong><br>
<Font face="Arial"size="3"><Strong>Email</Fonte><br><input type="text" size=50
name="EMAIL"></Strong><br>
<Font face="Arial"size="3"><Strong>description</Fonte><br><Textarea Name="DESCRIPTION" Rows=8
Cols=50 wrap=virtual></Textarea></Strong><br>
<input type="submit" name="submit" value="Enviar">
<input type="reset" value="Limpar">
</form>
</BODY>
</HTML>
```

Fig.3.2-Código que gera o formulário acima

3.5.4 Devolução dos dados ao cliente

Geralmente, após a execução de uma página ASP, há a necessidade de se enviarem dados para o utilizador. Existem três modos de enviar dados para o utilizador: ou é enviado um documento, ou uma referência para outro documento, ou uma mensagem de erro. Estes três modos são descritos na página ASP que devolve para o *browser*.

Toda a informação a enviar para o servidor deve ser feita através do *standard input*, sendo o cabeçalho a primeira a ser enviada ao servidor, seguindo-se, obrigatoriamente os dois caracteres de mudança de linha, para distinguir do corpo (caso exista) da informação a ser enviada. Os três tipos de cabeçalho que existem são *Content-type, Location e Status.*

Content-type

Quando se quer enviar um documento - que pode não ser só texto, mas sim uma imagem, som ou qualquer outro tipo de informação - o cabeçalho a indicar ao servidor deve ser do tipo *Content-type*.

Como já foi referido, o parâmetro *Content-type* é usado pelos *browsers* para saberem interpretar correctamente o tipo de informação recebida.

Geralmente, os dados que uma página ASP devolve são documentos HTML, pelo que o *MIME-Type* a indicar será: text/html. Um caso bastante popular onde isto não acontece é no caso dos contadores de acesso gráficos, onde a informação enviada pela página ASP é uma imagem e não um documento HTML.

Location

O *Location* serve para indicar um URL que irá ser automaticamente lido pelo cliente.

Status

O *Status* é usado quando se quer enviar mensagens de erro pré-definidas. Geralmente existem páginas que são carregadas quando existe algum problema na aplicação, como por exemplo não encontrar nenhum dado de uma pesquisa, dando a possibilidade de o utilizador tentar de novo. Contudo, se não se optar por esta via, podem-se enviar códigos de erro ao servidor, que enviará ao cliente uma página definida por defeito para erro correspondente.

Uma descrição completa dos códigos de erro aceites pelos servidores HTTP, deve ser consultado o documento "*Status Codes in HTTP*" [URL- 3.5].

Capítulo IV
Estrutura Funcional e Interface do ExventShop

Neste capítulo serão explicados os mecanismos de navegação e interacção com a parte HTML/ASP, as interfaces usadas para interagir com o utilizador, bem como a estrutura por trás da organização funcional do Exvent.

4.1 MODELO CONCEPTUAL

O esquema da figura 4.1 dá uma ideia da estrutura do interface global escolhido para visualizar as várias páginas construídas em HTML/ASP, a partir de um *browser*, podendo ser da Netscape ou Internet Explorer.

Este modelo representa o estado actual em que se encontra o ambiente Exvent, sendo de destacar a parte que estabelece a ligação com o ExventShop, apartir do módulo de serviços para visitantes. Este interface dá também, ao visitante a opção de ir directamente às compras, caso não queira visitar os pavilhões disponíveis, bastando accionar o *link* para o ExventShop.

Estando no ExventShop, o visitante, ao visualizar um determinado produto, ser-lhe-á mostrado uma página com apenas uma breve descrição com um *link* para ver mais detalhes sobre o produto e também para encomendá-lo, se assim o desejar.

O visitante tem também a possibilidade de escolher novos produtos, ou mesmo entrar para o espaço das exposições, visitando assim os pavilhões.

Apesar do ExventShop estar incorporado no Exvent, também foi concebido para poder funcionar como um sistema *stand alone*.

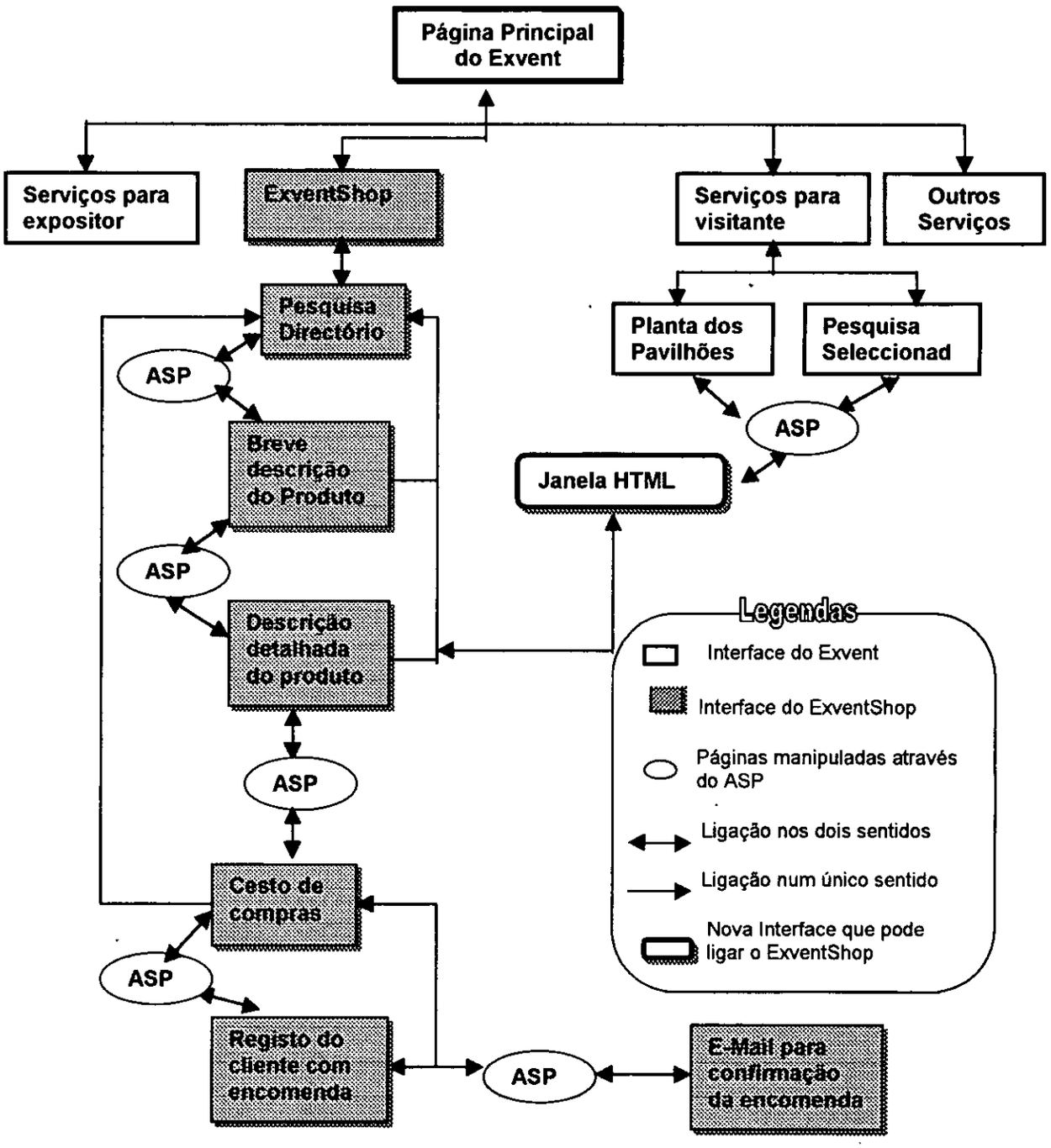


Fig.4.1-Modelo conceptual da interface

4.2 MODELO DA BASE DE DADOS

O desenvolvimento do ExventShop foi baseado num modelo relacional de bases de dados dado pelo diagrama da figura 4.2, onde mostra o relacionamento entre as entidades envolvidas.

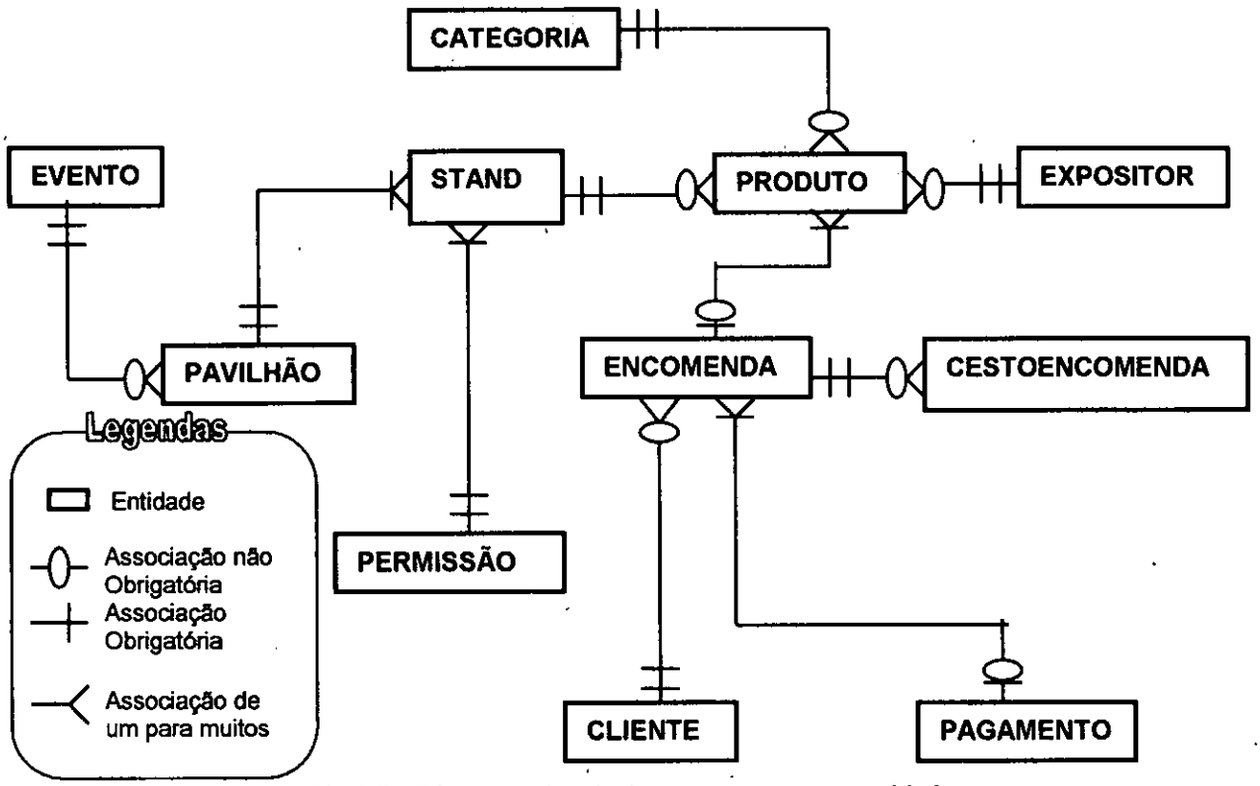


Fig.4.2 - Diagrama de relacionamento entre as entidades

A descrição de cada entidade e o papel que desempenham é a seguinte:

Cliente (ou Comprador)

O Cliente representa a entidade fundamental para existência do ExventShop. Para tal, ele deverá fornecer todas as informações úteis para que não sinta dificuldades na sua navegação. O cliente tem a possibilidade de inscrever os seus dados pessoais, caso esteja interessado em fazer parte dos clientes do ExventShop. Assim esta entidade guarda todos os registos dos clientes.

Encomenda

Caso o cliente se decida pela compra, o (s) produto (s) escolhido (s) serão colocados num ficheiro de encomendas, com a possibilidade do cliente remover o produto consultando outros ou sair da loja se assim o decidir.

Produto

Os produtos fazem parte da loja e estes poderão estar disponíveis, em stocks ou por encomendar. Este pode se localizar num determinado *stand*, com uma determinada Categoria e pertecendo a um Expositor, podendo ainda ser encomendado. Assim, a sua respectiva entrega poderá ser feita mediante a sua disponibilidade e localização do Cliente.

Expositor (ou Vendedor)

Os produtos disponíveis no ExventShop têm proveniência por parte duma entidade fornecedora. Tem importância para o Comprador saber informações referente às características do produto que, por sua vez, deverão ser fornecidas pelo Expositor. Para o ExventShop é também importante guardar os dados do Expositor bem como do seu representante oficial para posteriores esclarecimentos acerca de um determinado produto ou para casos de reordenamentos dos stocks da loja.

CestoEncomenda

Representa o cesto e os seus conteúdos. Cada registo desta tabela referencia um cesto individual contendo um produto e a sua respectiva quantidade feita pelo cliente e que fica correntemente guardado sobre a encomenda. A ligação com a tabela Encomenda permite identificar o cesto apropriado para um determinado cliente.

Pagamento

Entidade relacionada com as encomendas confirmadas. Através dela pode-se saber quais as encomendas que poderão estar na eminência de serem processadas. Ela estabelece uma ligação com a tabela Encomenda, com a finalidade de poder identificar essas encomendas.

Evento-Pavilhão-Stand-Permissão

Um Evento identifica o tipo de exposição. Ela pode acorrer em um ou vários Pavilhões, contendo um ou mais *stands*, que por sua vez, precisa duma permissão para poder ser acedida, para efeitos de manutenção dos produtos.

Para o armazenamento dos dados, primeiro foi usado o sistema da base de dados SQL Server. Devido à dificuldade em executar os *applets* feitos em Java, teve que se recorrer a um outro

sistema de bases de dados que estava disponível e que pudesse executar os *applets* para a visualização gráfica. Deste modo passou-se a usar a tecnologia da IBM, o DB2, cuja estrutura de dados para cada entidade envolvida é apresentada no anexo 1.

4.3 INTERFACE COM O UTILIZADOR

4.3.1 O menu principal

A figura 4.3 mostra o menu principal, que representa a página de entrada para o ExventShop. Ele funciona segundo a acção do utilizador. É a partir dele que o visitante e os expositores têm os *links* para as restantes páginas bem como um *link* directo ao ExventShop.

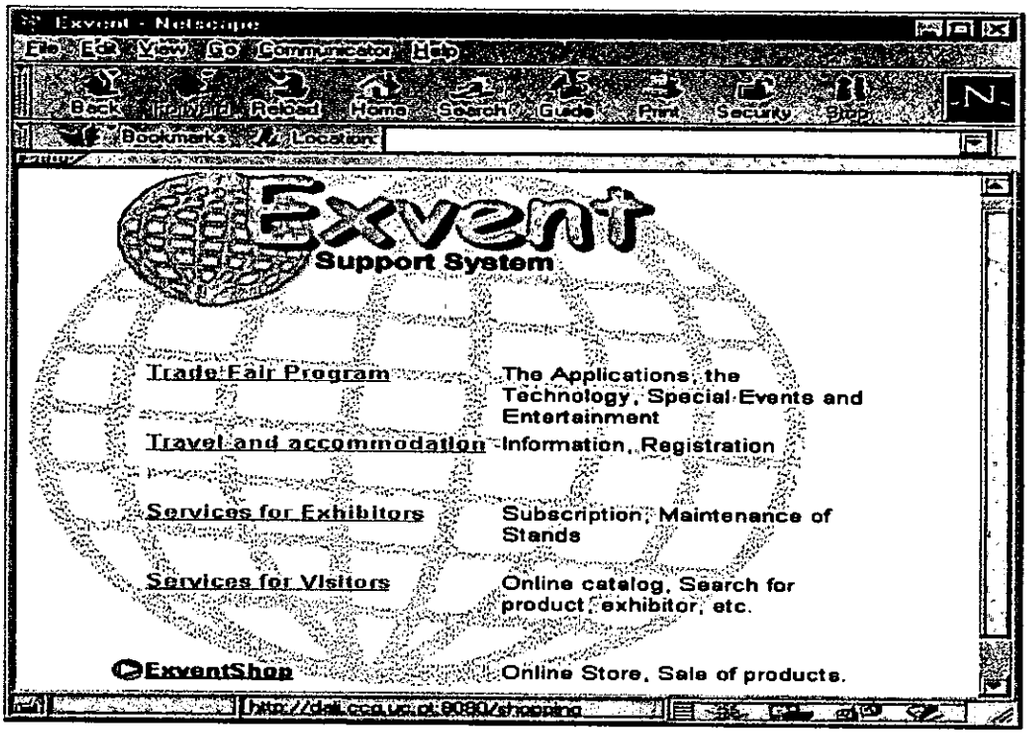


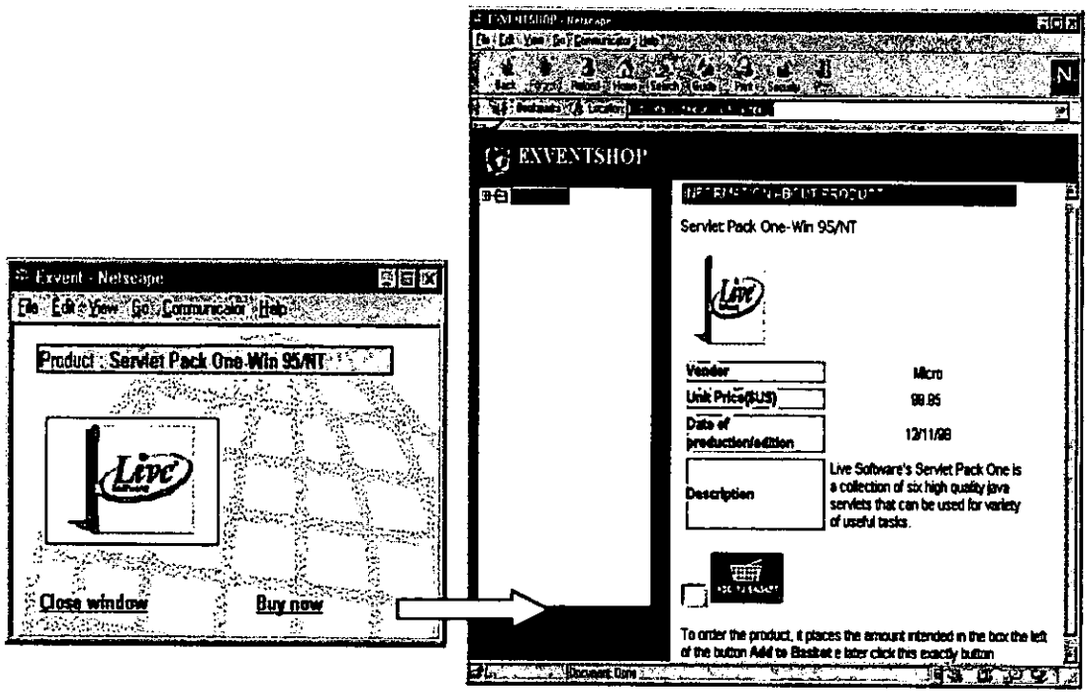
Fig.4.3 - Menu de entrada

4.3.2 Serviços para Expositores

Este é um *link* que dá possibilidade ao expositor para a escolha dos Pavilhões e os respectivos *stands* a fim de colocar os seus produtos para a exposição. Para o efeito, ele deverá indicar o seu *login* e *password*, e se estiver a aceder a página pela primeira vez, deverá preencher um formulário com campos tais como: nome da empresa, responsável pela exposição, morada, caixa postal, cidade, país, telefone, *E-mail*, etc.

4.3.3 Serviços para Visitantes

Este *link* possibilita aos visitantes percorrer os Pavilhões em exposição, havendo também um *link* para pesquisar os produtos, as empresas ou outros campos do seu interesse. O mecanismo de pesquisa é bastante simples, existe uma base de dados que contém informações sobre os produtos, os expositores, *stands* e outros cujos campos aparecem no formulário de pesquisa. Um visitante ao aceder a página de pesquisa e ao introduzir uma chave de pesquisa irá fazer correr a aplicação ASP que pesquisará à base de dados as chaves que o visitante introduziu. Caso encontre, serão mostrados numa forma de *link* o produto correspondente à chave. Se o visitante escolher um deles, será apresentada numa janela, uma breve descrição do produto (nome e a imagem) e um *link* para o ExventShop onde poderá ser apresentada uma descrição detalhada do produto (veja figura 4.4). Caso o visitante esteja interessado no referido produto, pode assim, fazer a sua respectiva encomenda. Este processo será explicado posteriormente.



Janela com breve descrição do produto seleccionado e uma nova janela com mais informações para compra.

Fig.4.4 - Interligação entre o Exvent e o ExventShop

4.2.4 Funcionalidades do ExventShop

ExventShop é uma abordagem duma aplicação desenvolvida em ASP para funcionar sobre a Internet. Ela está constituída de várias partes sendo de destacar as seguintes:

- Um catálogo *online* de produtos;

- O Cesto de Compras (cujo termo mais usado é o *Shopping Basket*);
- A parte da verificação das encomendas para pagamentos; e
- A manutenção do catálogo.

Quando o visitante entra para a página principal poderá visitar o ExventShop de duas formas. A primeira já foi referenciada atrás, enquanto que a segunda é ainda mais simples, uma vez que o visitante não precisa passar pelos pavilhões podendo ir directamente às compras. Esta ideia surgiu para facilitar aos visitantes que, não tendo muito tempo a perder, estejam só interessados em fazer compras.

4.2.4.1 Catálogo *Online*

Existem várias estratégias para desenhar um catálogo *online* funcional. No caso do ExventShop, a ideia foi construir um serviço que fosse bastante simples, eficiente, dinâmico e familiar para o visitante, onde pudesse localizar rapidamente qualquer produto. Para o efeito foi criado um serviço de directório.

O Directório

O Directório é uma estrutura que tem a forma de uma árvore constituído por um conjunto de informações guardadas numa base de dados, cuja organização dos dados e sua manipulação origina o exemplo da figura 4.5 composta pelos elementos abaixos:

Fig.4.5 - Estrutura do Directório

Como já havia referenciado acima, esta é uma construção dinâmica, uma vez que as páginas *Web* são geradas, consoante os pedidos dos utilizadores. Esta estrutura pode crescer/decrescer à medida que as categorias dos produtos também vai crescendo/decrescendo.

O princípio para construção dessa árvore é baseado na relação de que um produto pertence a uma determinada categoria e num algorítmo dado pela função *tree*, criado para este propósito, manipula um objecto (tabela Categoria) duma base de dados, cuja definição obedece uma relação de dependência entre os campos da própria tabela e, através da tecnologia *ActiveX*, desenha a estrutura da árvore.

Porquê o Directório?

Portanto, a escolha desta estrutura dependeu principalmente de como os dados estão estruturados e organizados na base de dados e o tipo de informação que se pretendia visualizar para o utilizador.

O objectivo foi de procurar incluir no directório produtos diversificados, partindo do princípio que a maioria dos produtos têm características comuns, diferenciando apenas a sua descrição e a facilidade de pesquisa que esta estrutura oferece. Assim, a definição da estrutura de dados em que se baseou esta construção foi a partir da relação entre as entidades directamente envolvidas, os Produtos e as Categorias.

A tabela Produto guarda todos os produtos disponíveis na base de dados. Ela contém um campo, CodCateg que faz referência à tabela Categorias, que permite guardar os produtos segundo uma dada categoria. O campo NomeProd estabelece um *link* para a página de informação detalhada de cada produto onde podem ser visualizados todos os campos desta tabela.Os anexo 1 e 2 fornecem a definição das tabelas para a base de dados e o código para a construção dessa estrutura respectivamente.

4.2.4.2 Criação do *Shopping Basket* (Cesto de Compras)

Shopping é uma das actividades que nenhuma das outras nações faz melhor do que os Estados Unidos da América. De facto, pequenos *shopping* estão "virtualmente" em todo o sítio.

Uma das muitas metáforas vindas de experiências do mundo físico de *shopping* é o *shopping basket*. O *shopping basket online*, é tal como numa loja, serve como um lugar para guardar os items dos compradores antes da verificação ou pagamento. É como no mundo real, pode-se dar uma olhada nos conteúdos do cesto antes de ir para a caixa de pagamentos.

Existem diferentes aproximações para criar um *Shopping Basket* usando o ASP. O método aplicado no ExventShop foi usando um *Session Object* para identificar o comprador e o armazenamento de todos os items que o mesmo adiciona para o cesto numa base de dados. A desvantagem deste método é que o carregamento adicionado guarda-se sobre o servidor *Web* (por causa dos frequentes acessos à base de dados). As vantagens dessa aproximação incluem:

- *Shopping basket* pode conter um rico conjunto de dados;
- A sessão do *shopping* pode ser acessível de qualquer computador.

  (Isto requereu o uso de um esquema de *E-mail/password*)

Algumas características deste *shopping basket* incluem a abilidade para: adicionar novo item, alterar a quantidade de um item e recalcular o valor do produto, remover um ou mais items, cancelar ou confirmar encomenda. A figura 4.6 mostra o *shopping basket* usado no ExventShop.

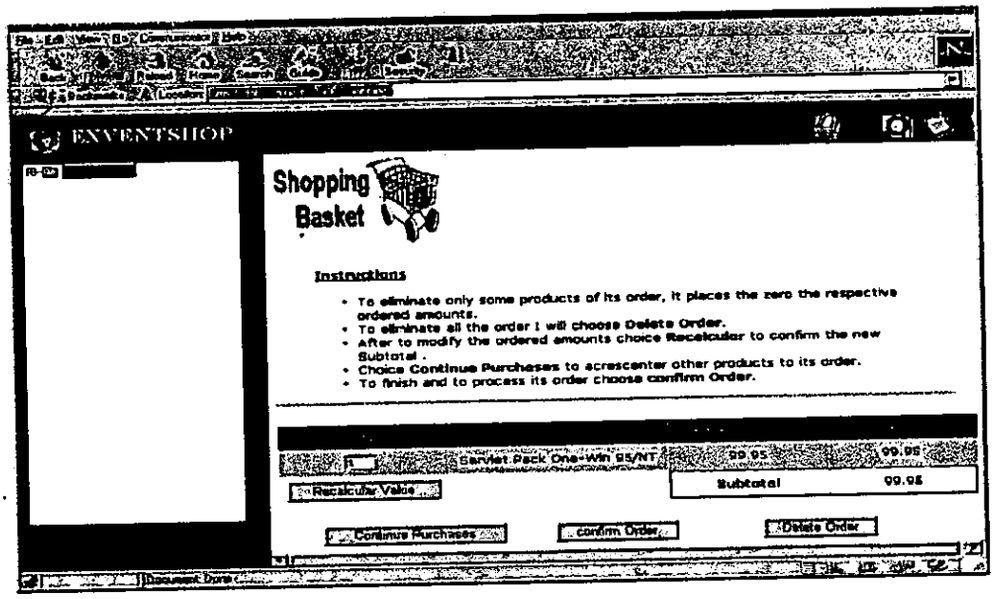


Fig.4.6- Estrutura do *Shopping Basket*

O diagrama de transição de estados do ExventShop, da figura 4.7, mostra o ciclo de vida do *Shopping Basket* desde o seu início até o seu término, incluindo as modificações que ele possa sofrer, bem como a aceitação ou rejeição de produtos.

A partir do momento que o comprador coloca o produto no *Shopping Basket*, este pode fazer várias operações já referenciadas acima.

De referir que após a confirmação da encomenda alguns dos dados do *Shopping Basket* são transferidos para o Cesto de encomendas, onde fica registado o número da encomenda, do cliente, as quantidades e a data, para uma posterior identificação caso o pagamento seja concretizado

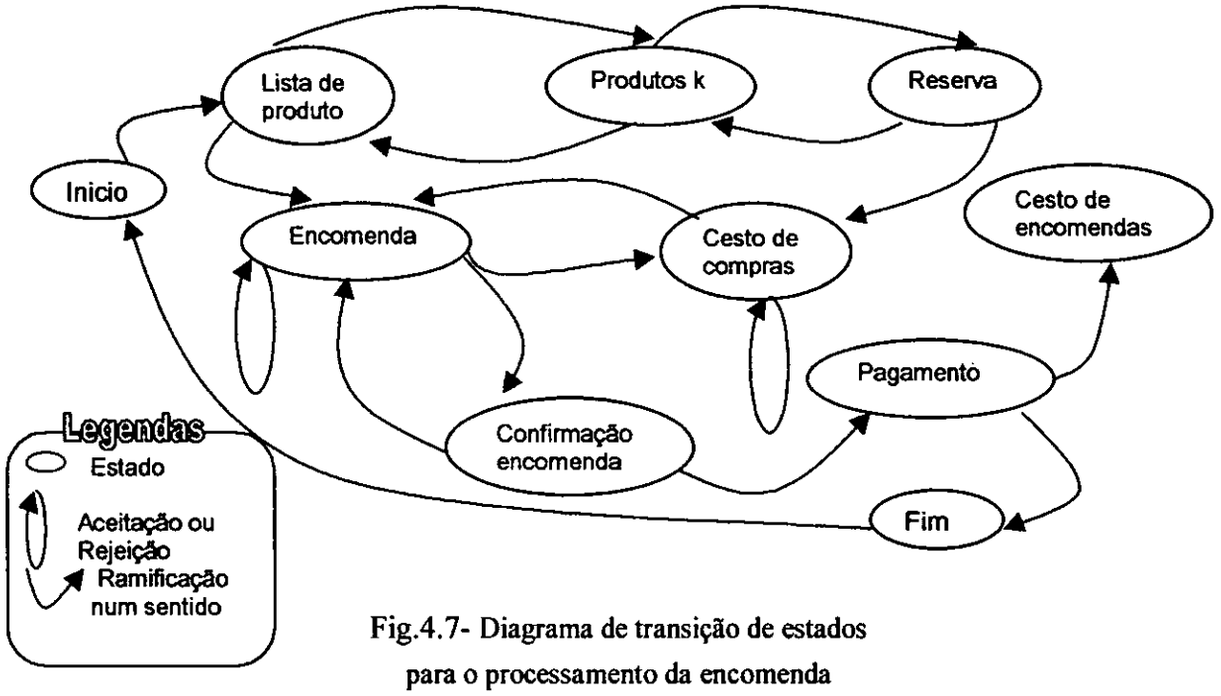


Fig.4.7- Diagrama de transição de estados para o processamento da encomenda

O anexo 3 apresenta o código do *Shopping Basket* desenvolvido em ASP. De salientar que o Cesto de compras é uma das partes mais complexas de um catálogo *online*.

4.2.4.3 Verificação e confirmação da encomenda

Numa loja *online*, o "*check out*" é onde a informação do cliente é pedida. Isto é, onde o endereço, número do cartão de crédito, e outras informações são todas recolhidas do cliente.

Uma das primeiras coisas a considerar quando desenhamos a funcionalidade da verificação geral é se vai ou não fornecer um ambiente seguro (via SSL).

Naturalmente, *sites* de grandes dimensões irão enviar o cliente para um servidor *Web* seguro quando se iniciar com o processo de verificação geral. Os passos para tornar um SSL é explicado no capítulo I.

A segunda coisa a considerar, embora deva parecer óbvio, é o tipo de dados que deverá ser recolhida do comprador quando ele faz o verificação geral. Questões como:

comprador é passado para uma página, que de seguida envia-se ao mesmo para alertar e confirmar a respectiva encomenda. Uma página final de agradecimento e com um alerta do envio desse *E-mail* é mostrado ao comprador.

To: joel@ccg.uc.pt
From: ExventShop@ccg.uc.pt
Subject : Order
Reply-To: ExventShop@ccg.uc.pt
Date: Mon, 22 February 1999 12:00 am
X-Mailer: ABMailer , ActiveX Mailman ver 1.0, by Andy Blanchard

Content:

You have received this email as a result from its register with the ExventShop personal information.
We accuse to the reception with your order. Please send us an email as reply of confirmation of order!

With our best regards

ExventShop

Fig.4.9 - Recepção do *E-mail* do lado do comprador

4.2.4.4 Manutenção do catálogo

Começar um catálogo *online* ascendente e sua execução pode ser uma tarefa de um trabalho intensivo. Mesmo depois do trabalho parecer completo, as mudanças continuarão a ser feitas dentro. Um elemento do catálogo é certo que mudará, mais frequentemente é os dados do catálogo. Os preços dos produtos mudarão, os produtos eles mesmos mudarão, novos produtos serão adicionados, etc.

A manutenção dos dados do ExventShop focará principalmente em adicionar, modificar e em apagar produtos. Esta operação estará provavelmente ao cargo de pessoas que lidam com o inventário dos produtos, por motivos de segurança, evitando assim que qualquer utilizador tenha acesso `a base de dados. Para esse efeito o acesso `a manutenção só poderá ser feito através do uso dos campos *userid* e à *password.*

Os dados relacionados com as compras do Cliente, permanecerão na base de dados durante um certo período (podendo ser 3 ou 5 dias) `a espera da confirmação da parte deste para posterior elaboração dum relatório de compras, sendo depois removidas a fim de libertar espaço no disco, resolvendo assim a desvantagem que havia citado atrás em termos de espaço de armazenamento.

O acesso ao relatório é feito através de um formulário que pesquisa à base de dados os *E-mail* dos compradores. Este permite fazer a listagem dos dados do comprador e a sua respectiva

Irá o comprador ser permitido fornecer uma conta e endereço de compra diferente? Irá o comprador ser permitido múltiplas formas de pagamento? Irá ser permitido um comprador fora do país de origem?

Outro importante passo a considerar é como a loja *online* irá proporcionar o processamento do pagamento. Assumindo que ela irá permitir pagamentos com cartões de créditos (*Visa, Mastercard*, etc), como irá um encargo ser dado sobre o cartão do comprador? Diversas empresas referenciadas no capítulo I, fornecem opções para o processamento de pagamentos. Uma associação entre a loja *online* e um banco é também necessária na encomenda para o processo de pagamentos.

As funcionalidades da verificação geral para o ExventShop irá ser bastante simplificado em que o *site* irá permitir um endereço a ser introduzido para ambos os endereços de compra e encargos. Após a confirmação da encomenda ao comprador é fornecido um formulário para introdução do seu endereço de *E-mail* e a *password*, mostrado pela figura 4.8. Este formulário tem duas opções:

- Se o cliente é novo, deverá deixar os campos do formulário em branco, clicando o botão **Register**, então aparecerá um formulário com os campos em brancos para serem preenchidos os dados pedidos;
- Se o cliente não é novo, deverá preencher os campos em brancos e clicar o botão **Register**, então aparecerá um formulário com os campos preenchidos com os seus dados. Esta informação vai-se buscar à base de dados e o cliente deverá confirmá-la.

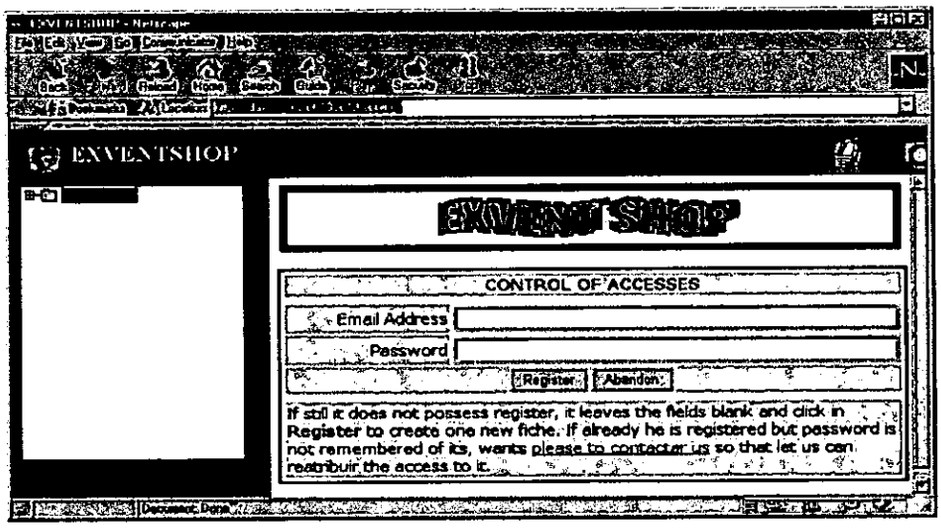


Fig.4.8-Formulário para garantir a segurança

O preço total com os encargo da sua encomenda é visualizada e o cliente deverá estar pronto para introduzir informação de pagamento. Após a confirmação da encomenda, o campo *E-mail* do

encomenda. Duas opções são fornecidas neste formulário, uma para visualizar o relatório e a outra para apagar todos os dados relacionados com as encomendas feitas pelo comprador seleccionado O relatório de compras mostrado na figura 4.10, contém os dados de cada Cliente e a sua respectiva encomenda, para efeitos de controle pela loja, caso o mesmo confirme a encomenda.

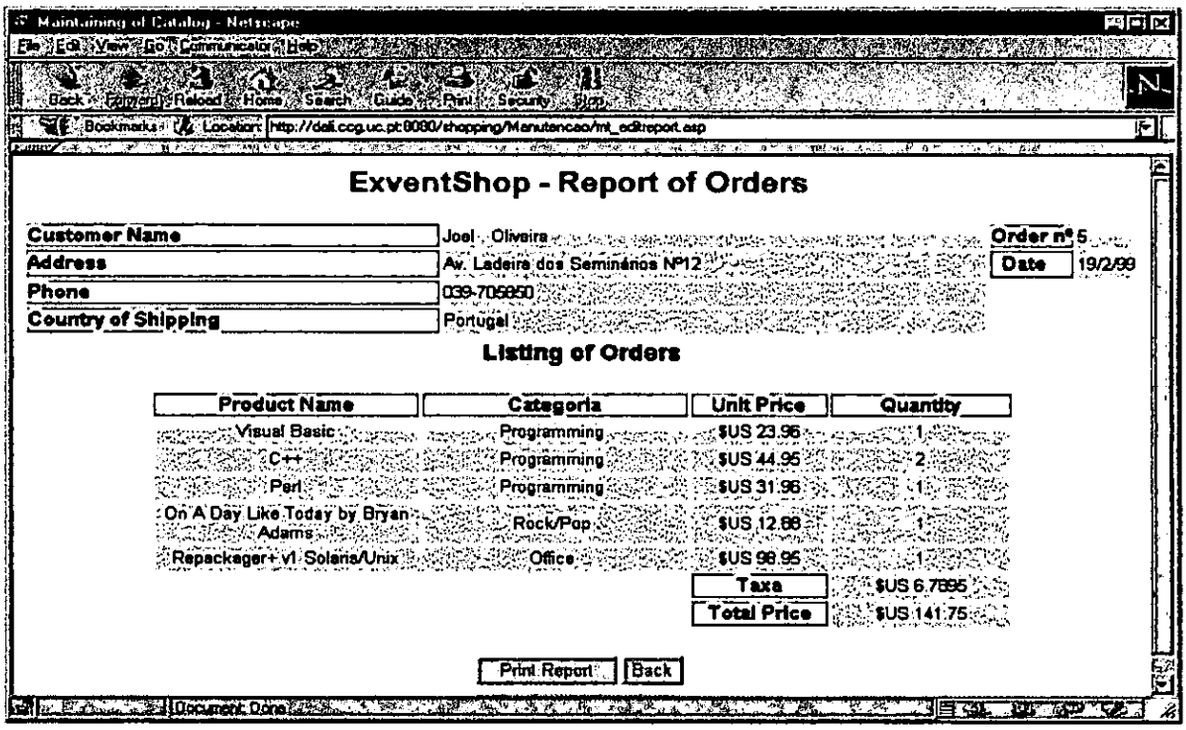


Fig.10 – Relatório de compras do cliente

Desenvolvimento do código de manutenção

A ferramenta de manutenção actualizará principalmente as tabelas dos produtos e das categorias. A organização do código em ASP consiste em três principais páginas dadas pelos ficheiros: mt_list.asp, mt_edit.asp e mt_commit.asp. Estas páginas permitirão listar, editar um produto e fazer as actualizações respectivamente.

Um ficheiro common.js foi incluído em todas as páginas. Ele permite o uso de uma função comum para todas as páginas. Esta é uma função simples chamada *trim* (ver figura 4.11 em anexo), que limpa com os espaços em branco nas *strings* passados por ele.

Para a visulização do relatório de encomendas é feita uma pequisa `a base de dados através dos ficheiros mt_listclient.asp, que vai buscar o *E-mail* do cliente e o mt_editreport que vai buscar à base de dados os dados relacionados com o *E-mail* escolhido, permitindo assim saber quem é o cliente e o que é que encomendou ou eliminá-lo da base dados.

Software de Implementação

Para a programação, propriamente dita, no Exvent foram usadas as seguintes linguagens:

- HTML/ASP - para a construção de páginas *Web* de todo o Exvent;
- JAVA - para a construção dos *applets* gráficos dos pavilhões (incluídos os *stands*);
- JAVASCRIPT/VBSCRIPT - para construção e formulários dinâmicos.

Além das linguagens de programação, foram usadas aplicações para desenvolvimento de várias componentes deste projecto:

- Netscape Navigator 4.05 e Internet Explorer 4.0 - para a navegação na *Web;*
- Adobe Photoshop 5.0 - para criação do desenho gráfico e concepção das imagens visualizadas nas páginas HTML/ASP;
- Homesite 3.0 e Microsoft Visual Studio - InterDev 6.0 - para desenvolvimento das páginas ASP;
- IBM DB2 5.0 - sistema de gestão de base de dados;
- Microsoft Internet Information Server 3.0 - Servidor *Web* usado para executar a aplicação na Internet;
- Windows NT 4.0 – usado como ambiente de trabalho;
- Microsoft Office 97 - ferramenta usada para desenvolvimento do relatório.

5.CONCLUSÕES E RECOMENDAÇÕES

São várias soluções já disponíveis para tornar os serviços da Internet seguros. Pensa-se que todo o trabalho desenvolvido ultimamente esteja a captar o interesse da vasta comunidade interessada na utilização da Internet para fins comerciais.
Todas as medidas de segurança sobre diversos modos de pagamento via Internet são um contributo essencial para o desenvolvimento de uma economia digital, onde empresas e particulares podem trocar informações, bens e serviços usufruindo de todas vantagens oferecidas pelas novas tecnologias de informação.

Contudo apesar do real desenvolvimento na economia digital, as compras feitas pela *Web* são apenas um pequeno indício nos indicadores da economia digital.
Ainda falta muito até que este comércio contribua substancialmente para o produto nacional bruto (PNB) de um País. Mas, há factores que indicam que os vendedores mais convencionais estão a pensar seriamente na *Web*.

Nesta economia digital, o dinheiro deixará de ser controlado exclusivamente por autoridades centrais condicionadas por operações políticas. Em substituição terão de haver companhias funcionando na Internet responsáveis por manter o dinheiro (electrónico) seguro e autenticado.
Como nem todos os utilizadores abandonaram os cheques e o dinheiro quando os cartões de crédito se tornaram populares, nem todos irão optar pelo dinheiro virtual - mesmo que haja um sistema que predomine. Existirão sempre aqueles que simplesmente não confiarão na segurança da Internet qualquer que seja o tipo de métodos de criptografia usados e aqueles que por princípio, nunca utilizarão novos sistemas. A aceitação de um conceito de dinheiro intangível por parte dos utilizadores (consumidores) poderá constituir uma questão séria a contornar. No entanto para a grande maioria, o dinheiro virtual tornar-se-á a forma mais segura, fácil e rápida de efectuar uma transacção.

O comércio através da Internet é muito mais barato, permite uma maior amplitude de aplicação e coloca em plano de maior igualdade as pequenas e grandes empresas.

Os utilizadores, por seu lado, virão a ser os grandes beneficiados, passando a ter ao seu dispôr um crescente mercado *online* de bens e serviços. A tecnologia de segurança para a WWW está a catapultar a realidade anunciada do comércio electrónico generalizado.

O Exventshop é assim uma das muitas soluções para o comércio electrónico existentes na *Web* e que pode fazer uso dessas ferramentas de segurança.

Já se começou também a juntar e evoluir ideias acerca de que características avançadas deverão ser incluídas para um sistema Exvent completo. A seguinte lista dá-nos alguma visão de algumas dessas futuras características:

- Um agente inteligente que notifica imediatamente o visitante quando uma informação relevante duma visita planeada muda;
- Suporte *online* de patrocinadores;
- Características de comunicação estendida que suporta a notificação de visitantes via telemóvel;
- Um sistema de guião electrónico inteligente;
- Integração de Quiosque Multimédia;
- Visualização em 3D das lojas.

Para o futuro, a loja desenvolvida (ExventShop), no acto do pagamento, estabelecerá uma ligação *online* com o Banco a fim de confirmar os dados referentes ao cartão de crédito fornecidos pelo comprador, processo não verificado até então, por ainda se tratar de um protótipo.

Pensa-se também alterar o sistema de bases de dados usado até então, para o sistema Oracle visto que fornece melhores desempenho e possibilita a execução dos applets em java com maior rapidez.

Elaboração de páginas com estatísticas de compras diárias, mensais, trimestrais ou anuais relacionados com:

- Os produtos mais vendidos;
- Os clientes que fazem mais compras;
- As páginas com mais acessos.

6.REFERÊNCIAS BIBLIOGRÁFICA

[Marcos, 1998] Marcos, A.F.(Junho1998). Comércio Electrónico: Perspectivas no Mercado Global.

[Ferrão, 1998] Ferrão, F.(Julho 1998). Sojornal: Comércio do Futuro.

[Oracle Magazine, 1998] Oracle Magazine, (January/February 1998). Electronic Commerce. Volume XII/Number 1.

[URL-1.1] Booz – Allen & Hamilton, (Abril 1997). G7 Global Market Place For SMEs. http:// www.ispo.cec.be/ecommerce/doc2.htm

[URL-1.2] Balanço da Internet em 1997 http:// www.Barreto.com.br/*Web*/*Web*97.htm

[Rebordão, 1997] Rebordão, H. (1997). Tópicos sobre Segurança.

[Revista Cyber.net, 1996] Revista Cyber.net, edição de Novembro de 1996.

[Revista Internet da Telepac, 1996] "Ecash", Net, Revista Internet da Telepac, nº 1, edição de Setembro de 1996.

[URL-1.3] http://www.techweb.com/Encyclopedia

[URL-1.4] Pistelli, D. Criptografia. http://www.nucc.pucsp.br/novo/cripto/cripto.html

[URL-1.5] Secure Socket Layer http://home.it.netscape.com/products/security/ssl/index.htm

[URL-1.6] Network Wizard.
http://www.nw.com

[Seberry, 1989] Seberry, J. & others (1989). Cryptography – An Introduction to Computer Security, 375 pp, Prentice Hall.

[URL-3.1] "HTML 4.0 Reference Specification"
http://www.w3.org/TR/REC-html40/

[URL-3.2] "What is Java?"
http://www.javasoft.com/nav/whatis/index.html

[URL-3.3] "Beginner's Javascript"
http://www.javaworld.com/javaworld/jw-03-1996/jw-03-javascript.intro.html

[URL-3.4] http://Activeserverpages.com

[URL-3.5] "Status Code in HTTP"
http://www.w3.org/pub/www/Protocols/HTTP/HTRESP.html

ANEXO 1

Estrutura da base de dados

EVENTO

Nome do campo	Tipo de dados	Tamanho	Descrição
CodEv	Integer Not Null	-	Primary key
NomeEv	Varchar Not Nul	60	Indica o nome do evento
Local	Varchar	200	Indica o local da sua realização
Notas	Varchar	200	Indica a descrição de um evento

PAVILHÃO

Nome do campo	Tipo de dados	Tamanho	Descrição
CodPav	Integer Not Null	-	Primary key
CodEv	Integer Not Null	-	*Foreing key* que faz referência a tabela Evento
X	Integer	-	Indica o valor a tomar para compriment
Y	Integer	-	Indica o valor a tomar para a largura

STAND

Nome do campo	Tipo de dados	Tamanho	Descrição
CodStan	Integer Not Null	-	Primary key
CodPav	Integer Not Null	-	*Foreing key* que faz referência a tabela Pavilhão
Coordenadas	Integer	-	Recebe os valores de um ponto
Area	Integer	-	Indica o zona limitada por um *stand*

PERMISSÃO

Nome do campo	Tipo de dados	Tamanho	Descrição
CodPerm	Integer Not Null	-	Primary key
CodStan	Integer Not Null	-	*Foreing key* que faz referência a tabela *Stand*
Login	VarChar	15	Nome a ser usado para identificação
Password	VarChar	8	Chave usada para obter acesso ao *Stand*

EXPOSITOR

Nome do campo	Tipo de dados	Tamanho	Descrição
CodExpo	Integer Not Null	-	Primary key
Responsavel	VarChar	30	Indica o nome do responsável do *Stand*
Nome	VarChar	25	Indica o nome da empresa/expositor
Descricao	VarChar	100	Fornece a descrição da empresa
Morada	VarChar	30	Indica a localização da empresa
CodPostal	VarChar	12	Indica a caixa postal do expositor
Localidade	VarChar	30	Localidade do expositor
Telefone	VarChar	12	O número de telefone do empresa
Telemovel	VarChar	12	O número do telemóvel
Fax	VarChar	15	O número do fax
Cidade	VarChar	20	O nome da cidade onde se situa a empr.
Pais	VarChar	25	O País proveniente
Email	VarChar	15	O email da empresa/expositor
URL	VarChar	30	Seu endereço se tiver alguma página/ficheiro

CLIENTE

Nome do campo	Tipo de dados	Tamanho	Descrição
CodClient	Integer Not Null	-	Primary key
Apelido	VarChar	30	Indica o seu último nome
Pnome	VarChar	60	Indica os primeiros nomes do Cliente
Morada	VarChar	60	Indica o localização física do Cliente
CodPostal	VarChar	10	Indica a sua caixa Postal
Telefone	VarChar	12	Indica o seu número de telefone
Cidade	VarChar	20	Indica a cidade de entrega do produto
Pais	VarChar	25	Indica o País de entrega do produto
Email	VarChar	30	Indica o seu correio electrónico
Password	VarChar	8	Indica a sua password para futuro acess
ConfirmePassw	VarChar	8	Campo para confirmar a password

PAGAMENTO

Nome do campo	Tipo de dados	Tamanho	Descrição
CodCartao	Integer Not Null	-	Primary key
CodEnc	Integer Not Null	-	*Foreing Key* que faz referência a tabela Encomenda
TipoCartao	VarChar	30	Indica o tipo de cartão de crédito usado
Validade	Date	-	Indica a data de validade desse cartão

CATEGORIA

Nome do campo	Tipo de dados	Tamanho	Descrição
CodCateg	Integer Not Null	-	Primary key
NomeCateg	Varchar Not Nul	60	Identificação do nome da categoria
Descrição	Varchar	200	Fornece a descrição da categoria
CodSup	Integer Not Null	-	Identifica a classe superior dessa categ.

ENCOMENDA

Nome do campo	Tipo de dados	Tamanho	Descrição
CodEnc	Integer Not Null	-	Primary key
CodClient	Integer Not Null	-	*Foreing key* que faz referência a tabela Cliente
CodProd	Integer Not Null	-	*Foreing Key* que faz referência a tabela Produtos
EstadoEnc	Byte	-	Indica o estado da encomenda, se foi comfirmada ou não.
Taxa	Double	-	Valor acrescentado ao preço unitário devido as deslocações para entrega
EncTotal	Double	-	Preço total da encomenda com a taxa incluída

CESTOENCOMENDA

Nome do campo	Tipo de dados	Tamanho	Descrição
CodCesto	Integer Not Null	-	Primary key que indica um único cesto
CodEnc	Integer Not Null	-	*Foreing Key* que faz referência a tabela Encomenda
CodProd	Integer Not Null	-	*Foreing Key* que faz referência a tabela Produtos
Qtidades	Integer	-	Indica a quantidade encomendada
Data	Date	-	Indica a data em que a encomenda foi feita

PRODUTOS

Nome do campo	Tipo de dados	Tamanho	Descrição
CodProd	Integer Not Null	-	Primary key
NomeProd	Varchar Not Null	60	Identificação do nome do produto
PrecoUnitario	Double	-	Preço por unidade de cada produto
Stocks	Integer	60	Guarda as quantidades existentes
Data	Date	-	Fornece a data de produção/publicação
Image	Varchar	100	Fornece a imagem para cada produto
Notas	Varchar	200	Fornece a descrição do produto
CodCateg	Integer Not Null	-	*Foreing key* referente a tabela Categorias
CodExpos	Integer Not Null	-	*Foreing key* referente a tabela Expositor
Cod*Stand*	Integer Not Null	-	*Foreing key* referente a tabela *Stand*

ANEXO 2

O códico que gera o Directório do ExventShop

----------Ficheiro tree.asp----------

```
<%@ LANGUAGE="JSCRIPT" %>
<HTML>
<HEAD>
<%
function tree (Root,RSet)
{
  RSet.MoveFirst();
  RSet.Filter="CodSup="+Root;
  if(!RSet.EOF)
   {
     Response.Write('<UL>\n');
      while(!RSet.EOF)
      {
            Response.Write   ('<LI><OBJECT   type="text/sitemap"><param   name=
            "Name" value = "'+Rset("NomeCateg")+'"><param name ="Local" value =
            "asp/category.asp?CodCateg ='+Rset("CodCateg")+'"></OBJECT>\n);
            var Branch = RSet.Clone ();
            tree(RSet("CodCateg"),Branch);
            RSet.MoveNext();
      }
      Response.Write ('</UL>\n');
}
   RSet.Close();
}
%>
<!-- Sitemap 1.0 -->
</HEAD>
<BODY>
<OBJECT type="text/site properties">
<param name="FrameName" value="main">
```

```
<param name="Window Styles" value="0x800025">
<param name="ImageType" value="Folder">
<param name="Font" value="Arial,8,0">
<param name="Background" value="0xA45200">
<param name="Foreground" value="0xFFFFFF">
</OBJECT>
<UL>
<LI><OBJECT type = "text/sitemap"><param name= "Name" value = "PRODUCTS"><param
name = "Local" value = "principal.htm"></OBJECT>
<%
rs = Server.CreateObject ("ADODB.RecordSet");
q = "select * from Categoria order by NomeCateg";
rs.Open (q,"DSN=SQLShop;UID=kismael;PWD=shop",3,3);
tree (0,rs);
%>
</BODY>
</HTML>
```

----------Ficheiro tree.htm usado para chamar o tree.asp--------

```
<HTML>
<head>
<title>Directory</title>
<base target="main">
</head>
<body bgcolor="#0052A4" text="#FFFFFF">
<p>
<applet width="100%" height="90%" code="HHCtrl" codebase="class/">
<param name="Command" value="Contents">
<param name="Item1" value="asp/tree.asp">
<param name="Flags" value="0x0,0x17,0xA45200">
</applet>
</p>
</body></HTML>
```

ANEXO 3

O código que gera o Shopping Basket

----------Ficheiro Shop.asp----------

```
<%@ LANGUAGE="JSCRIPT" %>
<HTML>
<head>
<title>Shopping Cart</title>
</head>
<body>
<script language="javascript">
function valida()
{
      i=1;
      while(i <= document.encomenda.NProd.value)
      {
            Qtd=eval( "window.document.encomenda.Qtidades"+i+".value;");
            Stock=eval( "document.encomenda.Stocks"+i+".value;");
            if(Qtd+i<0){alert("Please introduces valid quantity!");
            window.history.back();}
            else{
            if(Qtd+i>Stock)
            { alert("Quantity unavailable!");window.history.back();}}
      i++;
      }
}
</script>
<h2 align="left">Shopping Basket</h2>
<p><img src="../images/shopbask.gif" width="100" height="94">
<%
//FAZER A CONEXÃO COM A BASE DE DADOS
conn = Server.CreateObject("ADODB.Connection");
conn.Open("dsn=sqlshop;uid=kismael;pwd=shop");
```

```
var CodEnc;
var CodCesto;
var CodClient;
var anonymus;
CodEnc = Session('CodEnc');
CodClient= Session('CodClient');

// RECALCULAR O VALOR DA ENCOMENDA
if (Request.Form('Action') =='Recalcular Value')
{
  i=1;
  while(i <= Request.Form('NProd'))
  {
      if (Request.Form('Qtidades'+i)==0)
            {cestosql = "delete CestoEncomenda Where CodCesto = "+ Request.Form
            ('Cesto'+i) +" and CodEnc = "+ Request.Form('Enc')+" and CodProd =
            "+ Request.Form ('Prcd'+i);}
      else
      {cestosql = "Update CestoEncomenda set Qtidades = "+Request.Form
      ('Qtidades'+i) + " Where CodCesto = "+Request.Form ('Cesto'+i) +" and
      CodEnc="+Request.Form ('Enc')+" and CodProd = "+ Request.Form
      ('Prod'+i);}
      cur_qtd = conn.Execute(cestosql);
      i++;
  }
}

if (!Session('anonymus')) Session('anonymus')=0;
if (!CodEnc) CodEnc=0;
if (!CodClient) CodClient=0;
// ADICIONAR UM NOVO ITEM NO SHOPPING BASKET
if (Request.QueryString('CodProd') != 0 && Request.QueryString('Encomenda') ==
'Adicionar')
{
```

```
// CRIAR UMA ENCOMENDA E UM CLIENTE CASO ELAS NÃO EXISTAM
if (Session('flag')==1){
Session('flag')=0;
      if (CodEnc == 0)
      {
            Encomendasql = "select max(CodEnc) as mCodEnc from Encomenda";
            cur_CodEnc = conn.Execute(Encomendasql);
            CodEnc= cur_CodEnc('mCodEnc');
            CodEnc=parseInt(CodEnc);
            if (isNaN(CodEnc)) CodEnc=0;
            cur_CodEnc.Close();
            CodEnc=CodEnc+1;
            Session('CodEnc') = CodEnc;
            if (CodClient==0){
                  Clientesql = "select max(CodClient) as mCodClient from
                  Cliente";
                  cur_CodCliente = conn.Execute(Clientesql);
                  CodClient= cur_CodCliente('mCodClient');
                  CodClient = CodClient + 1;
                  Session('CodClient') = CodClient;
                  addsql="insert into Cliente(CodClient) values
                  ("+CodClient+")";
                  cur_CodClient= conn.Execute(addsql);
                              }
            addsql="insert into Encomenda(CodEnc, Estado_Enc, CodClient) values
            ("+CodEnc+",0,"+CodClient+")";
            cur_CodEnc = conn.Execute(addsql);
      }
//INSERIR ITEM NO CESTO DE ENCOMENDAS
if (!CodCesto) CodCesto=1;
Cestosql = "select max(CodCesto) as mCodCesto from CestoEncomenda where CodEnc
= " + CodEnc;
cur_CodCesto = conn.Execute(Cestosql);
CodCesto = cur_CodCesto('mCodCesto');
```

```
CodCesto = CodCesto + 1;
sql="Select * from CestoEncomenda Where CodProd = "+Request.QueryString
('CodProd') +" and CodEnc = "+CodEnc;
res = conn.Execute(sql);
// SE A ENCOMENDA JÁ EXISTE, ACTUALIZAR A QUANTIDADE
if (res.BOF) {
      sql = "insert into CestoEncomenda (CodEnc,CodCesto,CodProd,Qtidades)
      values("+CodEnc+","+CodCesto+","+Request.QueryString('CodProd')+","+Reque
      st.QueryString('Qtidades') + ")";
            }
else {
      CodCesto--;
      sql="update CestoEncomenda set Qtidades =" +Request.QueryString
      ('Qtidades') +" + Qtidades Where CodCesto = "+res ('CodCesto')+" and
      CodEnc ="+res('CodEnc')+" and CodProd = "+res ('CodProd');
      }
cur_Cesto = conn.Execute(sql);
}

// APAGAR TODAS AS ENCOMENDAS
}
if (Request.QueryString('Encomenda') == "cancel")
{
      deletesql = "delete from CestoEncomenda where CodEnc= " + CodEnc;
      cur_Cesto = conn.Execute(deletesql);
      deletesql = "delete from Encomenda where CodEnc = " + CodEnc;
      cur_Cesto = conn.Execute(deletesql);
      Session('CodEnc') = 0;
      CodEnc = 0;
      deletesql = "delete from Cliente where CodClient = " + CodClient;
      cur_Cesto = conn.Execute(deletesql);
      Session('CodClient') = 0;
      CodClient = 0;
   }
```

```
if (CodEnc != 0 ) {
      if( CodClient!= 0){
      cont=1

// APRESENTAR TODOS OS ITEMS NO SHOPPING BASKET

 shopsql = "select Produtos.CodProd,Stocks, NomeProd, (PrecoUnitario *
Qtidades) as Qtidades_PU, Qtidades, PrecoUnitario, CodEnc, CodCesto from
Produtos,CestoEncomenda where CestoEncomenda.CodProd = Produtos.CodProd and
CestoEncomenda.CodEnc = " + CodEnc;
cur_shop = conn.Execute(shopsql);
 %>
</p>
<div align="center"><div align="center"><center>
<table border="0" width="90%" cellspacing="1" cellpadding="5">
<tr>
  <td width ="100%"><font face = "Verdana, Arial, Helvetica" color = "#000000"
  size= "2"><b><u>Instructions</u></b><br>
   <ul>
      <li>To eliminate only some products of its order, it places the zero the
      respective ordered amounts. </li>
      <li>To eliminate all the order I will choose <strong> Delete Order
      </strong> </li>
      <li>After to modify the ordered amounts choice<strong>Recalcular</strong>
      to confirm the new Subtotal .</li>
      <li>Choice<strong>Continue Purchases </strong> to acrescenter other
      products to its order.</li>
      <li>To finish and to process its order choose <strong> confirm Order
      </strong>.</font></li>
   </ul>
  </td>
</tr>
</table>
</center></div>
<hr align="center">
```

```
</div>
<form name="encomenda" Action="Shop.asp" Method="POST">
<%
if(!cur_shop.EOF){
%>
input type = "hidden" name ="Enc" value = "<%=cur_shop('CodEnc')%>"><div align
= "center"><center><table cellpadding = "5" border ="0" bordercolo r= "#ffffff"
cellspacing ="1" width ="673">
<tr>
      <td bgcolor="#000000" align="center" width="153"><font face="Verdana,
      Arial, Helvetica" color="#ffffff" SIZE="2">Qty</font></td>
      <td bgcolor ="#000000" align="center" width="199"><font face ="Verdana,
      Arial, Helvetica" color ="#ffffff" SIZE="2">Items</font></td>
      <td bgcolor ="#000000" align="center" width="144"><font face ="Verdana,
      Arial,Helvetica" color ="#ffffff" SIZE="2">Unitary Price($US)</font></td>
      <td bgcolor ="#000000" align="center" width="129"><font face ="Verdana,
      Arial,Helvetica" color ="#ffffff" SIZE="2">Partial Price($US)</font></td>
</tr>
<%
cont=1;
while(!cur_shop.EOF){
%>
<tr>
  <td bgcolor ="#e3e3e3" align="center" width="153"><font face ="Verdana,
  Arial,Helvetica" size="2"><input Name= "Qtidades <%=cont%>" Value = " <% =
  cur_shop('Qtidades')%>" size ="2" maxlength="3"></font>
  </td>
  <td bgcolor ="#e3e3e3" align="center" width="199"><font face ="Verdana,
  Arial, Helvetica" size ="2"><%=cur_shop('NomeProd')%></font>
  </td>
  <td bgcolor ="#e3e3e3" align="center" width="144"><font face ="Verdana,
  Arial, Helvetica" size ="2"><%=cur_shop('PrecoUnitario')%></font>
  </td>
  <td bgcolor ="#e3e3e3" align="center" width="129"><font face ="Verdana,
  Arial, Helvetica" size ="2"><%=cur_shop('Qtidades_PU')%></font>
```

```
 <input type="hidden" name="Cesto <%=cont%>" value="<%=cur_shop('CodCesto')%>
"><input type = "hidden" name="Prod<%=cont%>" value="<%=cur_shop('CodProd')%>">
 <input type="hidden" name="Stocks<%=cont%>" value="<%=cur_shop('Stocks')%>">
 </td>
</tr>
<%
      cur_shop.MoveNext();
      cont++;
      }
            }
 else  {
      Response.Write("<center><h3>Does not have items in Shopping Basket </h3>
      </center> \ n");
      }

//MOSTRAR O SUBTOTAL
shopsql = "select sum((PrecoUnitario * Qtidades)) as Preco_Total from
CestoEncomenda, Produtos where CestoEncomenda.CodProd = Produtos.CodProd " +
"and CestoEncomenda.CodEnc = " + CodEnc;
cur_shop = conn.Execute(shopsql);
%>
<tr>
      <td align="center" width = "153" ><input type = "submit" name = "Action"
      value = "Recalcular Value" onClick = "valida()"><input type = "hidden"
      name = "NProd" value = "<%=cont-1%>">
      </td>
      <td align="center" width="199"></td>
      <td bgcolor="#ffff00" align="center" width="144"><font face="Verdana,
      Arial, Helvetica" color="#000000" size="2"><b>Subtotal</b></font></td>
      <td bgcolor ="# ffff00" align="center" width="129"><font face ="Verdana,
      Arial, Helvetica" size="2"><%=cur_shop('Preco_Total')%></font></td>
</tr>
</table>
</center></div>
</form>
```

```
;
<div align="center"><center>
<table border="0" width="90%" cellspacing="1" cellpadding="5">
<tr>
      <td width="33%"><form method="POST" action="Consulta.asp">
      <div align = "center"><center><p>
      <input type="submit" name="Action" value="Continue Purchases"></p>
</center></div>
</form>
</td>
<td width="33%"><form action="acesso.asp" method="POST">
<input type="hidden" name="enc" value="<%=CodEnc%>">
<input type="hidden" name="clien" value="<%=CodClient%>"><div align="center">
<div align="center"><center><p>
<input name="Encomenda" value="confirm Order" type="submit"></p>
</center></div>
</div>
</form>
</td>
<td width="34%"><form Action="shop.asp?Encomenda=cancel" method="POST">
<div align="center"><center><p><input value="Delete Order" type="submit"></p>
</center></div>
</form>
</td>
</tr>
</table>
</center></div>
<%
conn.Close();
            }
                                          }
      else{
%>
<center><h3>Does not have items in Shopping Basket</h3></center>
```

```
<Fonte face="Verdana,Arial" size="4">We inform that its order was annulled.<A
href="http://Dali.ccg.uc.pt:8080/Shopping" target="_top">It follows this Link
to come back the Store.</Fonte>
<%
      }
%>
</body>
</html>
```

```
function trim(instr)      {
                    str = new String(instr);
                                        retstr = "";
                    for (k = 0; k < str.length; k++)
                          if (str.charAt(k) != " ")
                                retstr = retstr + str.charAt(k);
                    return retstr;}
```

Fig.4.14- Função *trim*

Glossário

ActiveX: Uma marca da Microsoft de várias tecnologias baseadas na componente modelo de objectos, muitos dos quais destinados para Internet.

ADO: Active Data Object – um interface de programação, que é designado como o padrão de Microsoft para aceder dados.

Applet: Pequenas aplicações de software escritas em java, com alta capacidade de interacção.

API's: Aplication Program Interfaces – um formato de linguagem e mensagem usado por um programa de aplicação para comunicar-se com um sistema operativo ou outro programa de sistema tal como um sistema de gestão de bases de dados.

ASCII: American *Stand*ard Code for Information Interchange – o código básico que é usado pelos computadores para produzir texto.

ASP: Active Server Pages – páginas *Web* contendo código de programação escritos em Vbscripts, Javascripts ou Perlscripts.

CGI: Common Gateway Interface – é um interface para programadores que constroem scripts que correm como aplicações num servidor *Web* que comuniquem com clientes.

Cookie: Dados criados por um *Web* server e que são armazenados no computador de um usuário. Fornece uma maneira de estar a par de tudo o que acontece com o usuário, com a cooperação do *Web browser*.

DSN: Data Source Name – usado para configuração sobre o servidor *Web* antes de usar o ADO para poder conectar a uma base de dados em ASP.

***E-mail*:** Electronic mail ou correio electrónico, um processo de enviar mensagens e ficheiros anexos através da Internet para quem tem uma conta *E-mail*.

FTP: File Transfer Protocol – processo de enviar ficheiros através da rede.

Host computer: O computador (anfitrião) ao qual nos ligamos quando entramos num sistema *online*.

HTML: Hyper Text Markup Language – a linguagem baseada em texto usada para criar páginas *Web*.

HTTP: Hiper Text Transfer Protocol – o protocolo de comunicação usado pela *Web*.

IIS: Internet Informtion Server – software servidor *Web* da Microsoft que funciona sob Windows NT. Suporta o protocolo de segurança SSL em torno de um PC baseado em rede dentro de um *Web* site.

Internet: Milhares de redes de computador espalhados por todo o mundo e interligados, que se servem de protocolos comuns para comunicar entre si.

Internet Explorer: O *Web browser* da Microsoft, compatível com o Windows e Macintosh.

IP ou Internet Protocol: Um dos protocolos de comunicação que constitui o alicerce fundamental da Internet.

IPSec: IPSECurity – um protocolo de segurança que fornece autenticação e encriptação sobre a Internet.

Java: Nova linguagem de programação, que possibilita o download de aplicações chamadas applets numa página *Web*. Foi criada para estender as capacidades da *Web*.

JDBC: Java DataBase Connectivity – um interface de programação que deixa aplicações java aceder a base de dados através da linguagem SQL.

Kerberos: Um sistema de segurança que autentica usuários. Ele não fornece autorização para base de dados ou serviços.

MIME: Multipurpose Internet Mail Extensions – um sistema que converte ficheiros binários para formato ASCII, usado para os anexos das mensagens *E-mail*.

Mosaic: Um *Web browser* bastante conhecido, desenvolvido pelo NCSA. Talvez o primeiro grande browser.

NCSA: National Center for Supercomputing Aplications – Entidade Norte-Americana que tem por missão desenvolver alta tecnologia para investigação. Está muito ligada a *Web*.

Net: Abreviatura da Internet.

Netscape Navigator: O *Web* browser mais famoso para PC, Macintosh e Unix. Desenvolvido pela Netscape Communications Corp.

NNTP: Network News Transfer Protocol – um protocolo usado para conectar aos grupos da Usenet.

ODBC: Open DataBase Connectivity – um interface de programação da Microsoft que fornece uma linguagem comum para

aplicações Windows para aceder as bases de dados sob uma rede.

Página: Um simples documento HTML na *Web*.

Password: Um código usado para entrar no sistema. As passwords podem (e devem) conter letras e números.

PCT: **P**rivate **C**ommunications **T**echnology – um protocolo da Microsoft que fornece transacções seguras sob a WWW.

PEM: **P**rivacy **E**nhanced **M**ail – um *stand*ard para segurança de *E-mail* sobre a Internet.

PGP: **P**retty **G**ood **P**rivacy – software de criptografia de chave pública baseada no método RSA.

Protocolo: Os protocolos são um conjunto de standard de comunicação para sistemas informáticos, que permitem a transferência de dados entre eles.

Script: um pequeno programa escrito numa linguagem de programação para um propósito especial.

Servidor: Um computador servidor que partilha informação com outros computadores ligados em rede. Os *Web* servers, guardam páginas, Os FTP servers guardam ficheiros.

SET: **S**ecure **E**lectronic **T**ransaction – um protocolo padrão para mastercard e visa para segurança *online* de pagamentos de cartões de crédito via Internet.

Server-side script: Um pequeno programa que funciona sobre o servidor que controla e automatiza certas funções ou referências de um programa para outro.

Session: Em comunicação, uma conexão activa entre o usuário e o computador ou entre dois computadores. Usando um programa de aplicação (período entre o início e fim)

SGML: **S**tandard **G**eneralized **M**arkup **L**anguage – o percursor e super conjunto do HTML.

Site: Um computador na Internet. Se estivermos a navegar na *Web*, vemos as páginas de um *Web* site.

Socket: Um método de direccionar dados para uma aplicação apropriada em rede TCP/IP.

SQL: **S**tructure **Q**uery **L**anguage – linguagem usada para interrogar e processar dados numa base de dados relacional.

SSL: **S**ecure **S**ocket **L**ayer – um protocolo líder para segurança sobre a Internet. Quando uma sessão SSL é iniciada o browser e o servidor troca dados através da encriptação de chave secreta durante tal sessão.

TCP: Transmission Control Protocol – um dos principais protocolos de transmissão de dados usados na Internet. Normalmente referido em conjunto com o seu par IP, isto é o TCP/IP.

Telnet: Um processo para nos ligarmos a sistemas remotos e acedermos aos ficheiros e aplicações que ali estão guardados.

Unix: Sistema operativo no qual foi desenvolvido a Internet. Muitos servidores correm em Unix.

Upload: transferir um ficheiro do nosso computador para um FTP ou para um gropher site.

URL: Uniform Resource Locator - o endereço que define a rota de um ficheiro sobre a *Web* ou qualquer outra facilidade da Internet.

***Web browser*:** Aplicação que nos permite ver páginas *Web*.

***Webmaster*:** Responsável pelo servidor de um *Web* site.

WYSIWYG: What You See What You Get – refere-se a visualização de textos e gráficos sobre o écran na mesma forma como eles irão ser imprimidos.

7.BIBLIOGRÁFIA NÃO REFERENCIADA

Danesh A. & Tatters W. (1996). JavaSript 1.1 Developer's Guide, 1ª Edição, 591 pp, by Sams.net publishing.

Gunnit S. Khurana & Balbir S. Khurana (1996). *Web* Database Construction Kit, 662 pp, Published by White Group Press™.

Coleman D.(1997), "Groupware - Collaborative Strategies for Corporate LANs and Intranets", Prentice Hall PTR, London, Sydney, Toronto, Rio de Janeiro, ISBN 0-13-727728-8.

Daconta M.(1996), "JAVA for C/C++ Programmers", Wiley Computer Publishing, ISBN 0 471-15324-9.

Vogel A., Duddy K.(1997), "Java Programming with CORBA", Wiley Computer Publishing, ISBN 0 471-17986-8.