

IT-64

UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE MATEMÁTICA E INFORMÁTICA

TRABALHO DE LICENCIATURA

Acesso a Base de Dados via Internet

AUTOR: Flávio Pinto Matsolo

Maputo, Julho de 2002.

IT-64

IT-64

IT-64

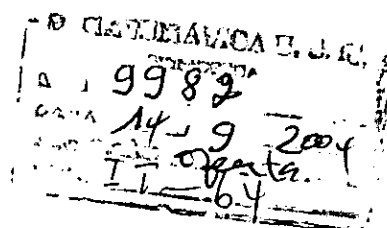
UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE MATEMÁTICA E INFORMÁTICA

TRABALHO DE LICENCIATURA

Acesso a Base de Dados via Internet

AUTOR: Flávio Pinto Matsolo

SUPERVISOR: dr. Carlos Cumbana



Maputo, Julho de 2002.

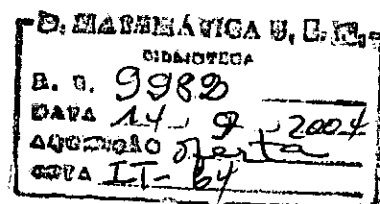
IT-64

DEDICATÓRIA

Aos meus pais.

Qualquer caminho que feche possibilidades futuras pode ser uma armadilha fatal.

Frank Herbert



AGRADECIMENTOS

Quero expressar a minha profunda gratidão em especial ao meu supervisor dr. Carlos Cumbana que arduamente me apoiou na elaboração deste trabalho, a todos os docentes, colegas e trabalhadores do DMI que directa ou indirectamente contribuíram para minha formação académica e para o sucesso deste trabalho, e a minha família e amigos pelo apoio e coragem que me deram ao longo de todos estes anos.

A vós devo muito. Muito obrigado!

Flávio Pinto Matsolo

DECLARAÇÃO DE HONRA

Declaro por minha honra que o presente trabalho constitui resultado das minhas próprias pesquisas e que o mesmo foi realizado para ser submetido como trabalho de licenciatura em Informática na Faculdade de Ciências da Universidade Eduardo Mondlane.

Maputo, Julho de 2002.

O Estudante

Flávio Pinto Matsolo

(Flávio Pinto Matsolo)

RESUMO

O rompimento de barreiras tecnológicas importantes no domínio das tecnologias de informação, experimentado nas últimas décadas, tem viabilizado a implantação de serviços de informação dinâmicos e automatizados que têm contribuído, nos últimos anos, na criação dum novo ambiente de acesso e troca de informações numa escala global, o que leva a uma clara e irreversível tendência para o desenvolvimento de sistemas de informação baseados em tecnologias Web para permitir a interoperabilidade entre sistemas de informação ou a publicação na Internet.

Reconhecendo a importância e o impacto das tecnologias baseadas na Web, o presente trabalho visa avaliar as ferramentas e tecnologias Web, e criar um modelo e protótipo que permite o acesso directo e universal a uma base de dados remota, via Internet.

O trabalho proposto compreende dez capítulos. O primeiro, introduz o tema abordado e apresenta os seus objectivos. O segundo, descreve os materiais e métodos utilizados para alcance dos objectivos preconizados. O terceiro, apresenta conceitos importantes sobre a Internet tais como: definição, filosofia e motivos para o seu uso. O quarto descreve um conjunto de ferramentas necessárias para o desenvolvimento de uma aplicação Web do tipo *Browser (Web)/Server (base de dados)*. O quinto, descreve o ambiente (*site server*) necessário para abrigar a aplicação Web. O sexto, aborda três aspectos importantes sobre gestão do funcionamento da aplicação Web e do *site server*: equilíbrio de cargas e tolerância a avarias, o rastreio do estado do utilizador e a gestão de solicitações. Os constrangimentos causados geralmente pela perda de confidencialidade e ataques diversos numa rede virtualmente livre e de fácil acessibilidade motivaram o sétimo capítulo, que apresenta um conjunto de soluções técnicas para garantir um ambiente mais seguro. O oitavo capítulo apresenta o modelo (*site server*) necessário para suportar a(s) aplicação(ões) Web. O nono, apresenta a implementação de um Web site com o objectivo principal de implementar e simular os pressupostos descritos nos capítulos anteriores. O décimo e último capítulo, apresenta as conclusões e recomendações finais.

ÍNDICE

I. INTRODUÇÃO E OBJECTIVOS	1
1.1. INTRODUÇÃO.....	1
1.2. DEFINIÇÃO DO PROBLEMA	2
1.3. OBJECTIVOS	3
1.3.1. Objectivo Geral.....	3
1.3.2. Objectivos Específicos	3
II. MATERIAL E MÉTODOS.....	4
III. A INTERNET.....	5
3.1. DEFINIÇÃO DA INTERNET	5
3.2. HISTÓRIA DA INTERNET	5
3.3. IMPORTANÇA DA INTERNET	6
3.3.1. Motivos Para a Utilização da Internet.....	7
3.4. FUNCIONAMENTO DA INTERNET	8
3.5. PROTOCOLOS.....	9
3.5.1. Protocolo TCP/IP.....	9
3.5.2. Protocolo HTTP.....	10
3.5.3. Protocolos SLIP E PPP	10
3.5.4. Protocolos ICMP	11
3.6. CONEXAO A INTERNET	11
3.6.1. Conexão Por Linha Telefónica.....	11
3.6.2. Conexão ISDN.....	12
3.7. PROVEDORES DE INTERNET	12
3.8. ENDEREÇOS IP	13
3.9. SUBNET MASK	14
3.10. DOMÍNIOS E URLs NA INTERNET	15
3.10.1. Domínio	15
3.10.2. URL.....	16
IV. ELEMENTOS BÁSICOS NECESSÁRIOS PARA O DESENVOLVIMETO DE UM WEB SITE.....	17
4.1. SISTEMA DE NAVEGAÇÃO	17
4.1.1. Estruturas de Navegação	17

4.2. FERRAMENTAS DE DESENVOLVIMENTO	19
4.2.1. HTML	19
4.2.1.1. Estrutura Básica do HTML	20
4.2.1.2. Alternativas a HTML	21
4.2.2. DHTML	21
4.2.3. XML	23
4.2.3.1. Comparação com a HTML	25
4.2.3.2. Apresentação de Dados XML Através do HTML	26
4.2.4. JavaScript	27
4.2.4.1. Comparação Entre JavaScript, VBScript, Java e ASP	27
4.2.5. Applets	28
4.2.6. Cookies	29
4.2.7. Plug-Ins	30
4.2.8. Sistema de Gestão de Base de Dados	30
4.2.8.1. Segurança e Privilégios no SGBD	32
4.2.9. Páginas Web	32
4.2.10. Gráficos	34
V. ELEMENTOS BÁSICOS NECESSÁRIOS PARA A IMPLEMENTAÇÃO DE UM WEB SITE	38
5.1. SERVIDORES ESPECIALIZADOS	38
5.1.1. Servidor Web	39
5.1.2. Servidor de Aplicações	39
5.1.3. Servidor de Aplicações Web	40
5.2. SISTEMA OPERATIVO PARA O SITE SERVER	42
5.3. HARDWARE PARA O SERVIDOR WEB	43
VI. GESTÃO DO SITE SERVER	44
6.1. RASTREIO DO ESTADO DO UTILIZADOR	44
6.2. EQUILÍBRIO DE CARGAS E TOLERÂNCIA A AVARIAS	44
6.3. GESTÃO DE SOLICITAÇÕES	45
VII. SEGURANÇA	47
7.1. PRINCIPAIS ATAQUES ASSOCIADOS AOS SERVIÇOS FORNECIDOS AOS USUÁRIOS ATRAVÉS DA INTERNET	47
7.2. FIREWALL	48
7.2.1. Funções de um Firewall	49

7.3. IDS.....	49
7.4. PROXY.....	50
7.5. ANTIVÍRUS WEB.....	51
7.6. SSL	51
7.7. OUTRAS FORMAS DE SEGURANÇA.....	52
VIII. MODELO DE SITE SERVER	54
IX. CASO DE ESTUDO.....	54
9.1. INTRODUÇÃO.....	54
9.2. O WEB SITE.....	54
9.3. REGRAS DE ACTIVIDADE	57
9.4. SITE SERVER	58
X. CONCLUSÕES E RECOMENDAÇÕES	59
XI.BIBIOGRAFIA	61
11.1. REFERENCIA BIBLIOGRAFICAS	61
11.2. BIBLIGRAFIA CONSULTADA.....	63
ANEXOS	64
GLOSSÁRIO	72

ÍNDICE DE FIGURAS

Figura 3.1 - Modelo clássico dos meios de comunicação de massas - Lasswell (1948).....	6
Figura 3.2 - Conectividade entre grupos interactuantes via Internet.....	8
Figura 3.3 - Arquitectura Cliente/Servidor.....	8
Figura 3.4 - Conexão <i>Dial-up</i>	11
Figura 3.5 - Classe A de IPs.....	13
Figura 3.6 - Classe B de IPs.....	13
Figura 3.7 - Classe C de IPs.....	14
Figura 3.8 - Classe D de IPs.....	14
Figura 3.9 - Máscara de sub-rede para endereços de classe C.....	15
Figura 3.10 - Domínios de confiança bidireccional.....	15
Figura 4.1 - Web Site de estrutura linear rígida.....	17
Figura 4.2 - Web Site de estrutura linear flexível.....	18
Figura 4.3 - Web Site de estrutura hierárquica de navegação livre.....	18
Figura 4.4 - Transformação de dados XML para uma apresentação HTML ou CSS.....	26
Figura 4.5 - Objectos do MS Access.....	31
Figura 5.1 - Recursos reunidos em um <i>site server</i> para atender uma demanda local e global.....	38
Figura 5.2 - Arquitectura " <i>three-tier</i> ".....	40
Figura 6.1 - Cache em um site server para a Web.....	45
Figura 6.2 - <i>Cache Web</i> do tipo " <i>proxy</i> ".....	46
Figura 7.1 - Configuração do <i>Firewall</i>	48
Figura 7.2 - Servidor de proxy entre um servidor Web e a Internet.....	51
Figura 8.1 - Modelo de <i>site server</i>	53
 Caso de Estudo	
Figura 9.1 - Estrutura de navegação.....	55
Figura 9.2 - Fluxograma da sequência de acções executadas durante a navegação.....	56
Figura 9.3 - DEA.....	57
Figura 9.4 - Modelo de <i>site server</i> do Web site.....	58

I. INTRODUÇÃO E OBJECTIVOS

1.1. INTRODUÇÃO

A convergência e o uso integrado das tecnologias de informação, cujo paradigma é a Internet, a rede mundial de computadores, têm contribuído nos últimos anos na criação de um novo ambiente de acesso e troca de informações numa escala global, transformando a sociedade como um todo, criando bases para novas formas de comunicação e socialização que influenciam profundamente a maneira como as pessoas interagem e geram informações, e na forma em que as colocam a disposição. Estas transformações têm exercido uma profunda influência sobre a concepção e funcionamento dos sistemas de informação automatizados, especialmente aqueles que estão voltados para actividades de pesquisa e negócio (IBICT, 2002).

O rompimento de barreiras tecnológicas importantes, experimentado nas últimas décadas, permite o surgimento de um novo patamar para esses sistemas, antes orientados basicamente em bases de dados isoladas, voltando-os hoje, para interoperabilidade com fontes de informação heterogêneas globalmente distribuídas e na publicação na Internet - que para além de ser um simples espaço público aonde as comunidades vão ocupando e se estabelecendo, é também um meio que pode ajudar pessoas e organizações a simplificar o processo de partilha de recursos tais como informação, *software* e *hardware* (IBICT, 2002).

Porém, o caminho natural para o uso efectivo da Internet compreende a criação de condições favoráveis, como a disponibilidade de ferramentas, de metodologias e de *softwares* próprios e de fácil uso, bem como um volume de recursos que atendam às necessidades de partilha entre os diferentes intervenientes.

É nesta base, que o presente trabalho, visa avaliar as ferramentas e tecnologias Web, criar um modelo e protótipo que permite o acesso directo e universal a uma base de dados remota, via Internet.

1.2. DEFINIÇÃO DO PROBLEMA

As limitações das redes cabeadas e de sistemas *standalone*, em termos de extensão, leva:

- a que muitos usuários finais geograficamente dispersos, não tenham acesso directo e imediato a colecções de informações, sem que seja necessário:
 - que um intermediário interaja com os sistemas de informação para a busca das informações solicitadas;
 - ou que tenham de se deslocar à fonte dos dados.
- a busca de soluções tecnológicas capazes de atender as exigências de uma demanda crescente por serviços, de uma comunidade que busca mais mobilidade e integração.

A falta de unicidade de informação para tomada de decisões certas em tempo útil em diferentes áreas; e por outro lado a necessidade do satisfazer os clientes requer a implantação de um serviço que viabilize de forma simples, dinâmica e automatizada o processo de comunicação intra e extra-organizacional.

O processo de globalização e convergência de mercados aumenta as necessidades de:

- *cooperar para aumentar a competitividade*, o que leva a constante busca de interoperabilidade entre sistemas de informação para reforço do processo de interacção entre os pares;
- *antecipar-se a concorrência*, transformando os sistemas de informação da empresa numa vantagem competitiva.

O processo de distribuição de informação usando os meios de comunicação tradicionais é geralmente caro e muitas vezes não suporta a multiplicidade de formatos requeridos pelos usuários.

Os sistemas tradicionais de distribuição de informação não são capazes de responder as crescentes exigências dos usuários, em termos de volume de conteúdo e de rapidez. Numa altura em que *a informação já não é mais enviada, mas sim é buscada sob demanda*.

À medida que cresce o número de prestadores externos de serviços (por exemplo, lojas reais ou virtuais), parceiros e clientes aumenta na mesma proporção, em qualquer negócio, a necessidade por um lado, de atender globalmente a demanda, e por outro lado de garantir mais confiabilidade e eficiência dos sistemas já instalados.

1.3. OBJECTIVOS

1.3.1. Objectivo Geral

Avaliar as ferramentas e tecnologias Web, e criar um modelo e protótipo que permite o acesso directo e universal a uma base de dados remota, via Internet.

1.3.2. Objectivos Específicos

- Avaliar ferramentas de desenvolvimento de Web sites estáticos e dinâmicos;
- Definir infra-estrutura tecnológica necessária para o suporte de uma aplicação do tipo *Browser (Web)/Server (base de dados)*;
- Criar um protótipo de um Web Site de cadastro, em uma base de dados, de resumos de teses defendidas em Moçambique ou por moçambicanos no estrangeiro.

II. MATERIAL E MÉTODOS

Recolha de Informação

O método de recolha de informação consistiu basicamente na:

- revisão bibliográfica;
- e consultas na Internet.

Método de Trabalho

- Uso do método descritivo e comparativo para avaliação das ferramentas de desenvolvimento e da infra-estrutura tecnológica necessária para implementação de uma aplicação do tipo *Browser (Web)/Server (base de dados)*;
- Uso do método descritivo para a realização da parte teórica do caso de estudo que consistiu na descrição do Web site, do *site server* e das principais regras de actividade (Business Rules);
- Uso das ferramentas e tecnologias avaliadas, para a implementação do protótipo.

A infra-estrutura tecnológica usada assenta fundamentalmente na arquitectura cliente/servidor por esta implementar eficazmente a separação lógica de funções baseada no conceito de *serviço* que estabelece uma relação de dialogo independente entre os processos intervenientes (arquitecturas *two tier* ou *three tier*).

O caso de estudo e protótipo foram implementados a partir de páginas de acesso a dados (PAD) do MS ACCESS versão 2000 que permitem gerar automaticamente páginas HTML padrão (*apresentação*) e possuem comandos XML e controles ActiveX para o "*data biding*" da fonte de dados (*base de dados*).

III. A INTERNET

3.1. DEFINIÇÃO DA INTERNET

A Internet pode ser definida como sendo uma rede gigantesca de computadores de todos tamanhos e tipos que uma vez conectado a ela passa-se a ter acesso a diversos tipos de informação (Sousa, 1997).

Sob o ponto de vista físico, a Internet é uma conexão entre redes, mas sob ponto de vista do usuário, a Internet é um grupo de serviços disponíveis para a troca de informações entre computadores ou indivíduos conectados (Sobreira, 1998).

3.2. HISTÓRIA DA INTERNET

No início dos anos sessenta, o Departamento de Defesa dos Estados Unidos da América, em conjunto com a ARPA (*Advanced Research Projects Agency*), procurava fornecer ao Pentágono um sistema de comunicações que operasse mesmo após um ataque nuclear. A resposta foi dada pela *Rand Corporation*, que propôs uma rede sem uma autoridade central (Sousa, 1997).

Nos fins dos anos sessenta nasceu a ARPANET e seu primeiro nó foi instalado a 1 de Setembro de 1969. E as primeiras ligações internacionais ocorreram em 1973 (Sousa, 1997).

A utilização oficial da ARPANET, nos primeiros anos, foi dedicada para investigação científica, de modo, a tirar-se maior proveito do tempo de utilização, uma vez que os computadores eram muito caros e lentos (Sousa, 1997).

O primeiro protocolo usado pela ARPANET foi o NCP (Network Control Protocol) (Bremmer et al, 1998).

Nos anos setenta, com surgimento de outras redes inicialmente independentes, como a NSFNET (*National Science Foundation Network*), a ARPA foi obrigada a desenvolver o protocolo TCP/IP para permitir a conexão com as diferentes redes (Sousa, 1997).

Com o crescimento, a ARPANET era cada vez mais difícil de controlar, e os militares necessitavam de ter uma rede própria, onde os níveis de segurança fossem maiores. Em 1983, dá-se a separação do segmento militar, com a criação da MILNET (Sousa, 1997).

Os progressos tecnológicos tanto nos anos oitenta, como nos anos noventa foram uma constante. As ligações aumentaram e tornaram-se mais rápidas para fazer face ao crescimento exponencial de utilizadores (Sousa, 1997).

A ARPANET deixou de existir formalmente em 1989, dando lugar a INTERNET uma rede verdadeiramente global (Sousa, 1997).

3.3. IMPORTANÇIA DA INTERNET

Actualmente, qualquer pessoa ou organizações pode, não só, ter acesso de forma imediata, informação localizada nos mais distantes pontos do globo como também criar, gerir e distribuir informação (texto, imagem e som) em larga escala e no âmbito mundial num suporte essencialmente alternativo ao papel, com claras economias de espaço (de armazenamento), com mínimo de custos (de distribuição) e com melhor acessibilidade. Algo que até então não poderia ser possível usando o modelo clássico dos meios de comunicação de massas, como ilustra a figura 3.1 (Sobreira, 1998).

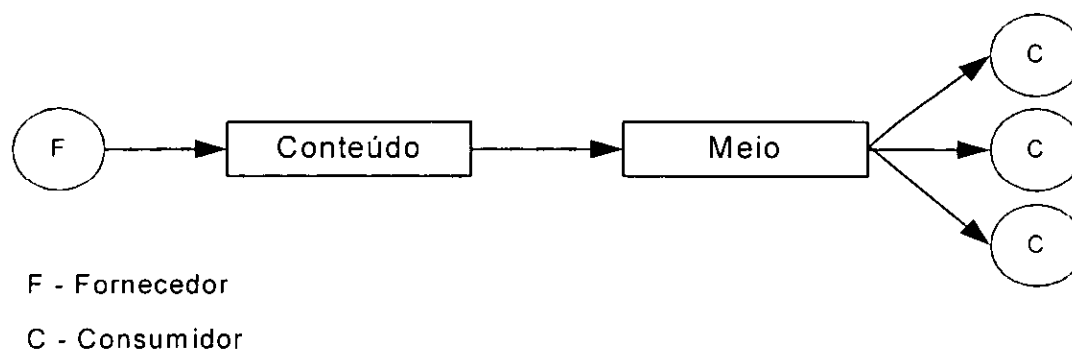


Figura 3.1 - Modelo clássico dos meios de comunicação de massas - Lasswell (1948)

Da figura 3.1, pode-se observar que, a comunicação é feita num só sentido e de “poucos-para-muitos” e não de “todos-para-todos”. A comunicação no sentido inverso, quando existe, só pode ser feita através de outros meios de comunicação.

3.3.1. Motivos para a Utilização da Internet

A Internet é uma estrutura de comunicação “omnipresente”, *onde qualquer um se comunica de qualquer lugar para qualquer lugar*, justamente por (Sobreira, 1998):

- suportar uma diversidade de ambientes computacionais e protocolos de comunicação;
- disponibilidade de browser Web com múltiplos recursos para qualquer sistema operativo cliente;
- os *softwares* existentes para servidores Web apresentam boa interoperabilidade;
- ser um recurso virtualmente livre e de fácil acessibilidade;
- ser importantíssimo pela inovação conceptual: *a informação já não é mais enviada, é buscada sob demanda*;
- a interface com o usuário é agradável e fácil de usar;
- permitir gerir centralmente dados e aplicações em ambiente distribuído;
- permitir integrar e uniformizar dados de fontes diferentes;
- permite integrar ferramentas de análise OLAP e *data mining*;
- implementar uma estrutura de divulgação e pesquisa rápidas (bases de dados abertas) entre usuários geograficamente dispersos;
- existir aplicações que facilitam a criação de páginas Web mesmo por principiantes;
- à medida que cresce a sua popularidade, também cresce a demanda por novas ferramentas e soluções para Web, o que provoca a concorrência entre os fabricantes de *softwares* levando-os a apresentar melhores produtos e mais baratos.

Em resumo, a Internet permite conectividade entre Intranets, ou seja, grupos de uma ou várias organizações (reais ou virtuais) podem trocar informação e interagir operacionalmente via Internet, como ilustra a figura 3.2.

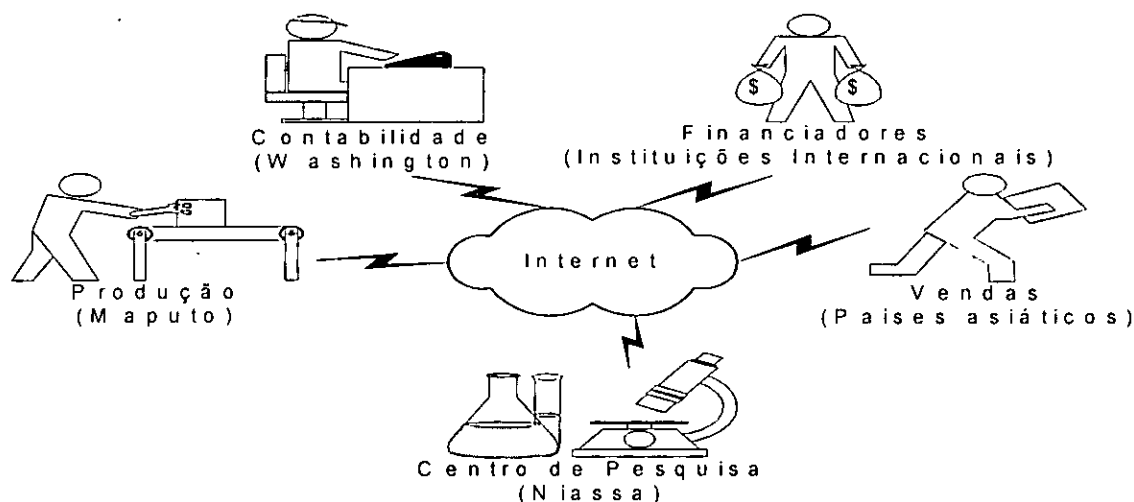


Figura 3.2 - Conectividade entre grupos interactivantes via Internet

3.4. FUNCIONAMENTO DA INTERNET

O funcionamento da Internet baseia-se no conceito Cliente/Servidor. A filosofia Cliente/Servidor consiste basicamente em uma solicitação de informação (pedido) pelo cliente e fornecimento dessa informação (resposta) pelo servidor Web, como ilustra a figura 3.3 (Oliveiro, 2000).

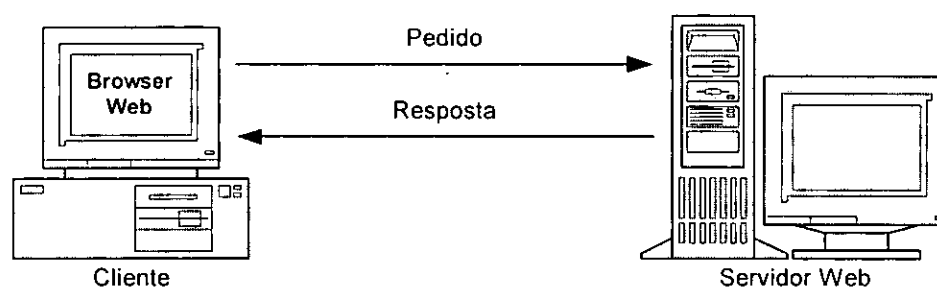


Figura 3.3 - Arquitectura Cliente/Servidor

Da figura 3.3 pode-se observar que, o cliente através do browser Web, envia a solicitação ao servidor Web. O servidor Web recebe a solicitação, verifica inicialmente se o cliente tem acesso aos dados que solicita e, em caso afirmativo, processa a solicitação e depois envia a resposta ao cliente.

3.5. PROTOCOLOS

As transações baseadas na arquitectura cliente/servidor necessitam de protocolos, ou seja, um conjunto de convenções comuns de comunicações entre o cliente e servidor (Oliveiro, 2000).

O controle da Internet em relação a sua operação normal é dividido em diversos órgãos, alguns centrais e outros não (Sobreira, 1998):

- O órgão que faz a gestão de toda política de fornecimento de endereços IP e outros códigos utilizados nos protocolos é o IANA (*Internet Assigned Numbers Authority*);
- A distribuição de endereços IP, e nomes de domínio (DNS), assim como a manutenção da documentação de padronização da Internet é realizada pelo InterNIC (*Internet Network Information Center*) que actualmente é operado por um conjunto de empresas, entre elas AT&T e Network Solutions Inc;
- O GTLD-Mou, um comité criado em 1997 para decidir sobre a padronização de novos nomes básicos da Internet (como *.com*, *.org*, *.gov*, *.arts*, *.web* e outros);
- O IETF (*Internet Engineering Task Force*) é responsável pela definição e padronização de protocolos utilizados na Internet;
- O IRTF (*Internet Research Task Force*) é responsável por criar novas aplicações;
- A definição de padrões de protocolos, criação de novos protocolos, evolução, etc. é controlada pelo IAB (*Internet Architecture Board*) que é um fórum suportado pela ISOC (*Internet Society*).

3.5.1. PROTOCOLO TCP/IP

Geralmente as redes de computadores transmitem e recebem informações por meio de pequenos pacotes de informação (Oliviero, 2000). Por exemplo, quando se envia um E-mail a informação nele contida é dividida em pequenas porções de 200bytes. Essas porções, em primeiro lugar, são

empacotadas para, em seguida, serem enviadas ao destinatário. Cada pacote é enviado separadamente e pode seguir numa rota independente por meio de roteadores. Quando todos pacotes chegam ao destinatário, eles são reagrupados conforme seu formato original.

Assim, a função do TCP é de:

- dividir a informação em pequenos pacotes;
- enviar esses pacotes ao destinatário por meio da rede;
- receber os pacotes enviados;
- verificar se esses pacotes estão correctos;
- e finalmente reagrupá-los no modo original.

O IP, por sua vez, tem a função de cuidar do empacotamento, endereçamento e envio dos dados.

3.5.2. PROTOCOLO HTTP

O processo de comunicação entre o cliente e o servidor é feitos através do protocolo de transferência de Hyper Texto (HTTP), que padroniza o modo de envio e recepção de dados, tendo como característica principal a sua independência total quanto à plataforma, podendo, por exemplo, estabelecer comunicação entre um servidor Linux, e um cliente Macintosh sem qualquer problema.

3.5.3. PROTOCOLOS SLIP E PPP

O SLIP (Serial Line Internet Protocol) foi criado no início dos anos 80 para transmissão pacotes através de uma interface serial (modem) (Zimmerman, 1997).

A falta de verificação de erro e controle de fluxo de dados no SLIP levou ao desenvolvimento do PPP (Point-to-Point Protocol), que inclui também a autentificação e oferece funcionalidades do nível de *link* similares as encontradas em uma Intranet. O SLIP e o PPP são também referenciados como *protocolos de linha* porque foram planeados para garantir a fiabilidade do circuito de rede e operam no nível 2 do modelo OSI¹, enquanto que o TCP e o IP são referenciados como *protocolos*

¹ Modelo criado pela ISO (International Organization for Standardization) com o objectivo de criar padrões de conectividade entre diferentes tipos de computadores (Ver anexos IV. 1).

de dados, porque eles foram planeados para o propósito de transferência de dados e operam nos níveis 3 e 4 do modelo OSI (Zimmerman, 1997).

3.5.4. PROTOCOLOS ICMP

Quando um roteador envia um pacote para outro roteador, ele verifica se este realmente chegou ao destino. Se o roteador destino estiver fora de serviço, o roteador de envio pode receber uma mensagem de redirecionamento para outro roteador. O ICMP (Message Control Protocol) define regras que os roteadores usam para a troca de informações de tráfego.

3.6. CONEXÃO A INTERNET

Actualmente existem basicamente duas formas de conexão física a Internet. A primeira forma é através de uma simples conexão por meio de uma linha telefônica por discagem – *Dial-up* ou dedicada. E segunda forma é a conexão ISDN (Integrated Services Digital Network) (Microsoft, 1996).

3.6.1. Conexão Por Linha Telefónica

Uma conexão *dial-up* ou dedicada exige que esteja disponível uma linha telefónica e um modem conectado ao computador, além de um provedor de acesso à Internet (*IP account*), como ilustra a figura 3.4.

Neste tipo de conexão os *IP accounts* são geralmente do tipo PPP, SLIP ou CSLIP (versão comprimida do SLIP).

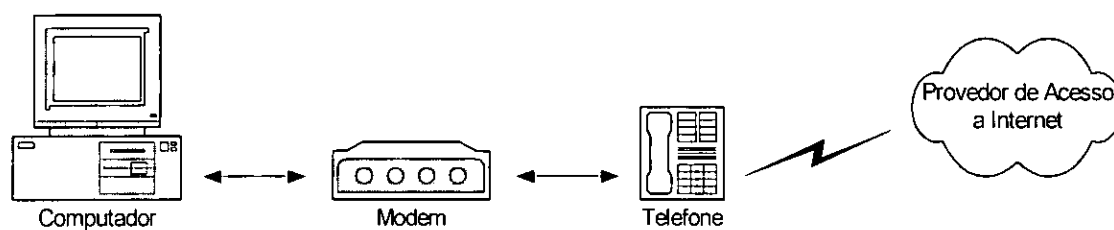


Figura 3.4 – Conexão Por Linha Telefónica

3.6.2. Conexão ISDN

ISDN (Integrated Services Digital Network) é um serviço de telecomunicações que conecta redes através de linhas digitais. Este tipo de serviço permite o tráfego simultâneo de dados e de voz a velocidades até 128 Kbps. O ISDN usa normalmente dois tipos de linhas (Designers, 2002).

- Linhas ISDN BRI;
- Linhas ISDN PRI.

ISDN BRI

A BRI (Basic Rate Interface) é uma linha que chega ao assinante, utilizando o mesmo par de “fios” de telefone. Este tipo de linha multiplexa dois canais de 64 Kbps cada um, um para transportar dados ou voz (canal B) e outro para transportar sinais de controle (canal D). Portanto, permitindo o uso simultâneo de duas linhas em vez de uma - uma para telefone e outra para Internet.

ISDN PRI

As linhas PRI (Primary Rate Interface) são linhas que permitem transmitir, numa única conexão física, 30 canais lógicos de B e 1 de D.

O ISDN proporciona conexões rápidas e estáveis, e apresenta-se como uma solução económica principalmente para LANs (Local Area Network) que conectam múltiplos usuários a uma localização específica na Internet (Microsoft, 1996).

3.7. PROVEDORES DE INTERNET

Para aceder à Internet, é necessário requisitar um canal (*link*) de dados que pode possuir uma velocidade de 256 Kbps, 1 Mbps, 2 Mbps ou 4 Kbps (é óbvio que quanto maior for o *link*, maior será o custo de aluguer) num Provedor de Serviços de Internet – ISP (Internet Server Provider) local, que por sua vez, disponibilizará um endereço único - IP que permitirá o acesso a outros provedores (Oliveiro, 2000).

3.8. ENDEREÇOS IP

Um endereço IP é um número de 32 bits, geralmente representado por um número com 4 partes, cada uma separada por um ponto. Essas 4 partes representam o endereço da rede (Net ID) e o endereço do *host* (Host ID). Cada parte individual pode ter um valor entre 1 e 254.

No endereço IP de 32 bits, o número de bits usados para identificar a rede e o *host* varia de acordo com a classe de rede do endereço.

A classe A, como ilustra a figura 3.5, é usada somente em redes amplas. O bit mais significativo é sempre zero, o que deixa 7 bits disponíveis para definir 127 redes. Os 24 bits restantes do endereço permitem que cada rede Classe A tenha até 16.777.216 *hosts*. A parte de endereço da rede pode assumir valores na faixa de 1 à 126.

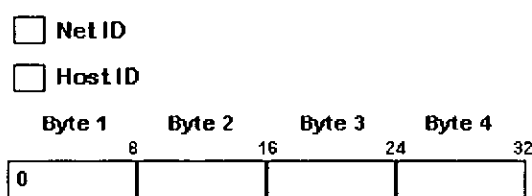


Figura 3.5 – Classe A de IPs

A classe B, como ilustra a figura 3.6, é usada em rede de tamanho médio. Os dois bits mais significativos são sempre 10, e os restantes são utilizados para definir 16.384 redes, cada uma com até 65.535 *hosts* ligados. A parte de endereço da rede assume valores entre 128 à 191.

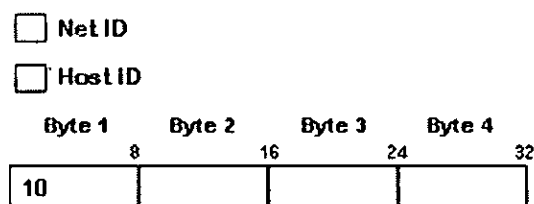


Figura 3.6 – Classe B de IPs

Classe C

A Classe C, como ilustra a figura 3.7, é usada em redes menos amplas. Os 3 bits mais significativos são sempre 110, e os bits restantes são usados para definir 2.097.152 redes, cada uma com no máximo 254 *hosts*. A faixa de endereços de rede está entre 192 e 223.

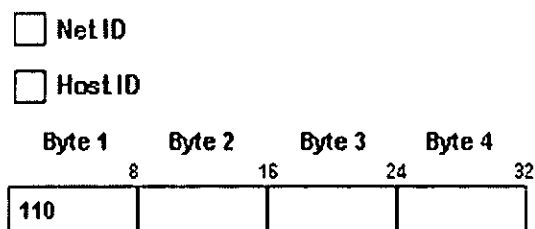


Figura 3.7 – Classe C de IPs

Classe D

E a classe D, como ilustra a figura 3.8, é um endereço especial e não pode ser usado em redes. Os 4 bits mais significativos são sempre 1110, e os 28 bits restantes permitem aceder mais de 28 milhões de endereços.

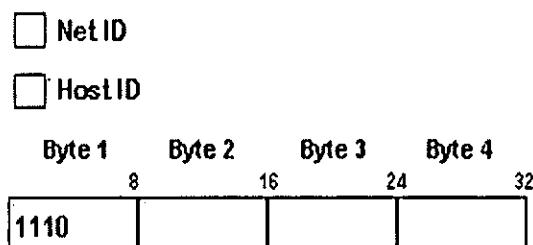


Figura 3.8 – Classe D de IPs

3.9. SUBNET MASK

Subnet Mask é uma máscara que tem a função de simplificar o roteamento de pacotes de informação numa rede.

Uma subnet mask possui 32 bits divididos em quatro bytes, como ilustra a figura 3.9.

Na Subnet Mask utiliza-se o bit 1 para especificar bits de rede e de sub-rede, enquanto o dígito 0 é usado para especificar bits de *hosts*. Por exemplo, o subnet mask 255.255.255.0 especifica que os 3

primeiros bytes identificam a rede e que o último byte identifica o host desta rede, nesse exemplo, nota-se que a *subnet* está a mascarar um endereço IP pertencente a classe C.

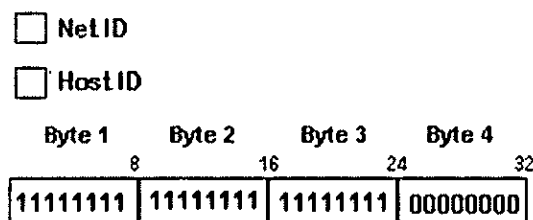


Figura 3.9 – Máscara de sub-rede para endereços de classe C

3.10. DOMÍNIOS E URLs NA INTERNET

3.10.1. Domínio

Domínio é a denominação ou localização que se dá a um site server (conjuntos de servidores na Internet), como ilustra a figura 3.10. O nome de domínio foi concebido com o objectivo de facilitar a memorização dos endereços IP (Microsoft, 1996).

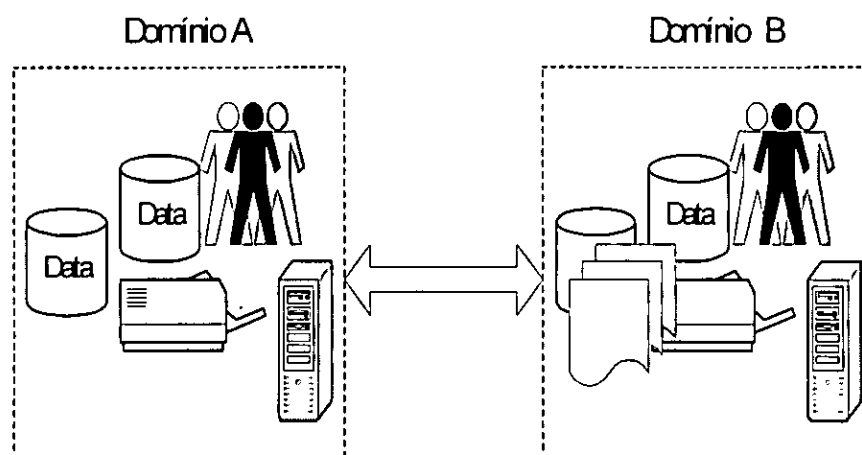


Figura 3.10 - Domínios de confiança bidireccional

3.10.2. URL

URL (Uniform Resource Locator) é um endereço que os browsers Web usam para aceder informações na Internet. Na Internet, existem milhões de documentos – cada um possui o seu próprio endereço (Bremmer et al, 1998).

Um URL é formado por três partes principais:

1. Um identificador de serviços (como *http*);
2. Um domínio (como *www.ups.com*);
3. Um caminho (como *www.ups.com/tracking*).

A primeira parte do URL, o identificador de serviço, diz ao browser Web como aceder o arquivo indicado (qual protocolo deve usar).

O identificador de serviço pode assumir as seguintes formas:

- *http://* - especifica que a conexão usará o HTTP;
- *ftp://* - especifica que a conexão usará FTP (File Transfer Protocol) – normalmente para uma operação *download* (retirada) de um arquivo específico;
- *gopher://* - especifica uma conexão com um servidor Gopher que fornece uma lista gráfica de arquivos acessíveis;
- *telnet://* - especifica uma conexão com uma secção Telnet onde se pode executar programas remotamente.

A segunda parte do URL é o domínio.

E a parte final do URL, o caminho, é uma informação opcional, que indica o *path* para um arquivo específico. Se o caminho não for mencionado, o servidor enviará a página padrão (a homepage).

IV. ELEMENTOS BÁSICOS NECESSÁRIOS PARA O DESENVOLVIMENTO DE UM WEB SITE

4.1. SISTEMA DE NAVEGAÇÃO

Um bom sistema de navegação deve ser claro e eficiente, permitindo ao visitante obter a informação desejada sem que se perca, isto é, deve ser fácil "ir" a uma página, bem como "voltar" ao ponto anterior. De um modo geral, quanto maior for o Web site, maior será o cuidado a ser tomado com o sistema de navegação, porque se ela não for bem estruturada os programas de busca e própria navegação por *hyperlinks* perde a eficácia (Oliveiro, 2000).

4.1.1. ESTRUTURAS DE NAVEGAÇÃO

Existem basicamente três tipos de estruturas de navegação (Oliveiro, 2000):

- Estrutura linear rígida;
- Estrutura linear flexível;
- Estrutura hierárquica de navegação livre.

Estrutura Linear Rígida

Uma estrutura linear rígida é semelhante a um livro em termos de *layout* - a navegação dá-se apenas em dois sentidos, sendo que as páginas estão ligadas de uma forma sequencial, como ilustra a figura 4.1.

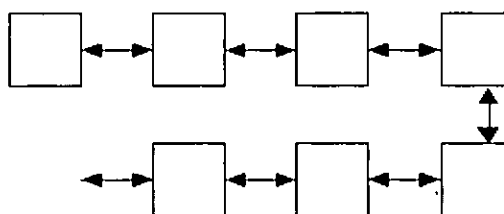


Figura 4.1 - Web Site de estrutura linear rígida

Normalmente este tipo de navegação é usado em questionários e álbuns de fotografias, ou em Web sites com poucas páginas, caso contrário, se um utilizador pretender aceder a apenas última página do Web site terá que percorrer todas páginas, e isso pode provocar que ele desista antes de a visualizar.

Estrutura Linear Flexível

Este tipo de navegação é muito semelhante ao da estrutura linear rígida, porém com a possibilidade de pequenos saltos preestabelecidos para permitir uma navegação mais flexível pelo Web Site, como ilustra a figura 4.2.

Este tipo de estrutura é mais usada na criação de Web sites de tamanho moderado.

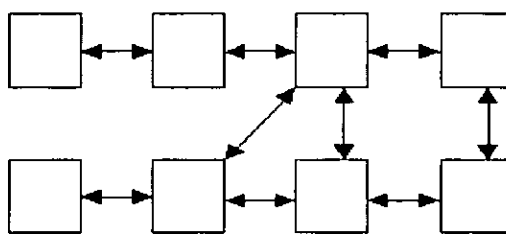


Figura 4.2 - Web Site de estrutura linear flexível

Estrutura Hierárquica de Navegação Livre

Em uma estrutura hierárquica de navegação livre, normalmente, as páginas dispõem de menus de navegação para facilitar, de forma intuitiva, o acesso à informação desejada por parte do usuário, como ilustra a figura 4.3.

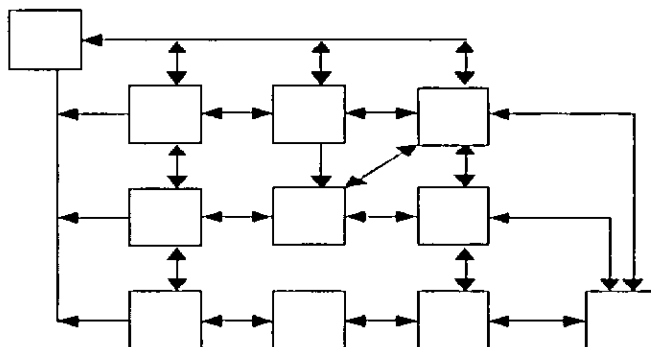


Figura 4.3 - Web Site de estrutura hierárquica de navegação livre

Este tipo de estrutura permite o usuário mover o browser Web, de qualquer página para qualquer página do Web site, de uma forma não sequencial, permitindo que o usuário aceda o que lhe interessa na ordem que deseja.

Este tipo de estrutura é mais usada na criação de Web Sites com muitas páginas e, no entanto, exige um maior cuidado na manutenção, em relação aos outros tipos, devido à complexidade da sua implementação.

4.2. FERRAMENTAS DE DESENVOLVIMENTO

4.2.1. HTML

A HTML (Hiptertext Markup Language) foi concebida originalmente por Tim Berners-Lee em 1989, inicialmente para ser um formato capaz de permitir a cientistas usar computadores diferentes e partilhar informação de forma transparente (Evans, 1998). Para isso, foi necessário considerar as seguintes características:

- A *independência de plataforma*, em que um documento poderia ser exibido de maneira similar em computadores com diferentes capacidades em relação a fontes, gráficos e cores;
- O *hipertexto*, significando que qualquer palavra ou frase em um documento poderia referenciar um outro documento, o que permitiria uma fácil navegação entre documentos muito grandes;
- Os *documentos rigorosamente estruturados* permitiriam a conversão de documentos de um formato para outro, bem como a pesquisa em bases de dados.

A HTML pode ser definida como uma linguagem de elaboração (marcação) de documentos com hipertexto, e não uma linguagem de programação (Evans, 1998).

A função da HTML é de enviar para o browser Web instalado no cliente, comandos de formatação (marcas) que informam ao browser de que maneira os textos e imagens deverão ser exibidos no ecrã.

Por exemplo, ao se pretende transmitir a frase "Seja bem vindo" em negrito (*bold*), teria que se usar um marcador de início do negrito e um outro marcador de fim do negrito. Em HTML temos:

```
<b>Seja bem vindo</b>
```

O resultado exibido seria: **Seja bem vindo**

Para escrever em HTML, não é necessário usar um processador HTML especializado ou uma ferramenta de conversão. Pode-se usar qualquer editor de texto, desde que o texto editado seja salvo com a extensão padrão *htm* ou *html*.

4.2.1.1. Estrutura Básica do HTML

Um documento escrito em HTML possui quatro *tags* (marcas) básicos usadas para descrever a estrutura geral de uma página a ser lida pelo browser Web (Evans, 1998):

```
<Html>
<Head>
<Title>
...Texto do título
</Title>
</Head>
<body>
...Conteúdo da página
</Body>
</Html>
```

- **<html>...</html>**: indicam ao browser Web que o documento a ser lido está escrito na linguagem HTML. Elas devem obrigatoriamente abrir e encerrar uma página;
- **<head>...</head>**: indicam quais os comandos que deverão ser lidos pelo browser Web antes da página ser carregada, ou seja, servem de cabeçalho para a página;

- **<title>...</title>**: indicam o título a ser exibido na barra de título da janela do browser Web. Estas *tags* sempre estão contidas na *tag* **<head>...</head>**;
- **<body>...</body>**: contêm o corpo do documento.

A interpretação de *scripts* e de *tags* HTML é parte do browser Web que analisa gramaticalmente processo. Assim, qualquer *script* ou *tag* dentro das *tags* **<HEAD> ... </HEAD>** é interpretado antes de qualquer *tag* dentro das *tags* **<BODY> ... </BODY>** ser examinado, isto é, objectos criados dentro das *tags* **<BODY> ... </BODY>** não existem na altura em que *scripts* ou *tags* dentro das *tags* **<HEAD> ... </HEAD>** estão sendo analisados.

A manipulação do conteúdo da página deve ser feita por *scripts* ou *tags* criados também dentro do corpo do documento.

4.2.1.2. Alternativas a HTML

A HTML dificilmente fornece todos os recursos de *layout* de páginas que a editoração electrónica demanda actualmente (Zimmernam, 1997).

Novos tipos de formatos surgiram para fazer face a esse facto. E esses formatos são geralmente projectados para aplicações específicas. Por exemplo, o PDF (Formato de Documentos Portáteis) usado no Adobe Acrobat, tem todos os recursos de *layout* de páginas, mas é ainda um formato fechado o que torna difícil a geração automática por *scripts* e outros *softwares*; e os arquivos PDF são muito maiores que os arquivos HTML equivalentes (Zimmernam, 1997).

4.2.2. DHTML

O DHTML (Dynamic HTML) não é uma extensão da linguagem HTML, e nem é uma linguagem de programação, mas sim, é uma linguagem que estabelece um nível superior de interactividade entre *scripts* (VBScript ou JavaScript) e todos os elementos da linguagem HTML incluídos numa página Web (Pires, 2000).

A inclusão, por outro lado, de Java *applets* e controlos ActiveX introduz um nível avançado de interactividade com manipulação directa de dados em base de dados remota a partir do browser Web.

Utilizando, por exemplo, a linguagem VBScript, é possível associar um procedimento a um evento seguindo a convenção: *objecto_Nome do Evento* (*makeRed_onMouseOver*, no exemplo).

```
<SCRIPT LANGUAGE="VBScript">
<--
sub makeRed_onMouseOver
    makeRed.style.color = "red"
end sub

sub makeRed_onMouseOut
    makeRed.style.color = "blue"
end sub
// -->
</SCRIPT>
```

} Muda a cor do texto para vermelho sempre que o mouse estiver sobre o elemento.

} Muda a cor do texto para azul sempre que o mouse não estiver sobre o elemento.

Código VBScript para ilustração de mudanças de eventos em um objecto.

É nesta perspectiva, que qualquer elemento de uma página é modificável em tempo real no browser Web, o que por si só, abre novas possibilidades de interactividade como, por exemplo, a facilidade de mostrar e esconder texto como resposta as acções do utilizador.

Vantagens do DHTML segundo Pires (2000):

- Dado que o DHTML funciona no browser Web - do lado cliente da ligação cliente/servidor - uma vantagem pode ser imediatamente apontada: a redução drástica dos tempos de resposta;
- O DHTML permite a utilização de "pre-caching", que consiste no carregamento em *background* do conteúdo, o que permite uma navegação entre páginas com um tempo de espera mínimo;

- Utilizando DHTML, é perfeitamente possível e simples modificar o conteúdo de uma página sem necessidade de intervenção do servidor;
- O DHTML permite apresentar, manipular e actualizar dados numa página Web, estabelecendo ligações do tipo “Data binding”, entre uma página Web e uma fonte de dados acessível via ODBC ou OLE DB.

4.2.3. XML

A XML (eXtensible Markup Language) é uma linguagem de marcação de dados (meta-markup language) que permite que uma marcação específica seja criada para especificar ideias e compartilhá-las na rede (W3C, 2001).

Características da linguagem XML

- **Representação estruturada dos dados**

O XML provê um padrão que permite codificar conteúdos, semânticas e as esquematizações para uma grande variedade de aplicações tais como:

- documentos do tipo texto;
- objectos Java ou controlos ActiveX;
- registos de dados resultante de uma consulta a uma base de dados;
- interfaces de aplicações de usuário;
- *links* entre documentos.

- **Separação entre dados e apresentação**

A característica mais importante do XML se resume em separar a interface do usuário (apresentação) dos dados estruturados.

Principais vantagens da linguagem XML segundo W3C (2001):

- **Buscas mais eficientes**

Os dados em XML podem ser unicamente "etiquetados", o que permite, por exemplo, que uma busca de livros seja feita em função do nome do autor. Sem o XML, uma busca com o nome do autor poderia levar a qualquer site que referência tal nome.

Sem o XML seria necessário que a aplicação de procura soubesse como foi estruturada cada base de dados que armazena os dados de interesse, o que é impossível.

O XML permitiria definir livros por autor, título ou assunto, por exemplo, o que facilitaria enormemente a qualquer processo de procura.

- **Desenvolvimento de aplicações Web mais flexíveis. Incluindo integração de dados de fontes completamente diferentes; manipulação local dos dados; múltiplas formas de visualização e actualizações granulares de conteúdos.**

- **Integração de dados de fontes diferentes**

O XML permite que dados em múltiplas bases de dados incompatíveis possam ser facilmente combinados via *software*, em um servidor intermediário, estando as bases de dados na extremidade da rede.

- **Manipulação local dos dados**

Os dados XML recebidos por um cliente podem ser visualizados, editados ou mesmo manipulados via *scripts* de acordo com o interesse do usuário.

- **Múltiplas formas de visualizar os dados**

Os dados recebidos por um cliente podem ser visualizados de diferentes formas uma vez que o XML define somente os dados e não a aparência.

➤ **Actualizações granulares de documentos**

Os dados podem ser actualizados de forma granular, evitando que uma pequena modificação no conjunto de dados implique uma nova busca do documento. Dessa forma, somente os elementos modificados são enviados para o cliente.

- **Distribuição dos dados via rede de forma mais comprimida e escalável**

➤ **Escalabilidade**

O facto dos documentos XML separarem os dados da forma com a qual são visualizados, permite ao usuário visualizar os dados da forma que mais lhe agrade. Dessa forma, a interactividade não depende tanto da comunicação cliente/servidor, pois pode ser feita "offline".

➤ **Compressão**

A compressão de documentos XML é fácil devido à natureza repetitiva das *tags* usadas para definir a estrutura dos dados. A necessidade de compressão é dependente da aplicação e da quantidade de dados a serem movidos entre clientes e servidores.

- **Fácil distribuição na Web**

Assim como a HTML, o XML, por ser um formato baseado em texto aberto, pode ser distribuído via HTTP sem necessidade de modificações nas redes existentes.

4.2.3.1. Comparação com a HTML

HTML e XML derivam do SGML. Ambos identificam elementos em uma página e ambos utilizam sintaxes similares - tags (palavras encapsuladas por sinais '<' e '>') e atributos (definidos com `name="value"`).

A grande diferença entre ambos é que o HTML descreve a aparência e acções em uma página, enquanto o XML descreve o que cada trecho de dados representa. Por exemplo, enquanto em um

documento HTML uma *tag* <p> indica um parágrafo, no XML essa *tag* pode indicar um preço, um parâmetro, uma pessoa, ou qualquer outra coisa que se possa imaginar.

Os arquivos XML como os arquivos HTML podem ser criados a partir de qualquer editor de textos, mas as regras de formatação para documentos XML são muito mais rígidas do que para documentos HTML, isto é, a falta de uma *tag* ou de aspas em um atributo, por exemplo, inutiliza o documento XML, enquanto que no HTML não.

4.2.3.2. Apresentação de Dados XML Através do HTML

Um documento XML não indica por si só como a informação deve ser visualizada, enquanto que o HTML é uma linguagem ótima para a apresentação de dados para o usuário final, isto é, tais dados podem ser representados em XML e visualizados em HTML cabendo o servidor dos dados ou o browser Web converter os registros XML para HTML, como ilustra a figura 4.4.

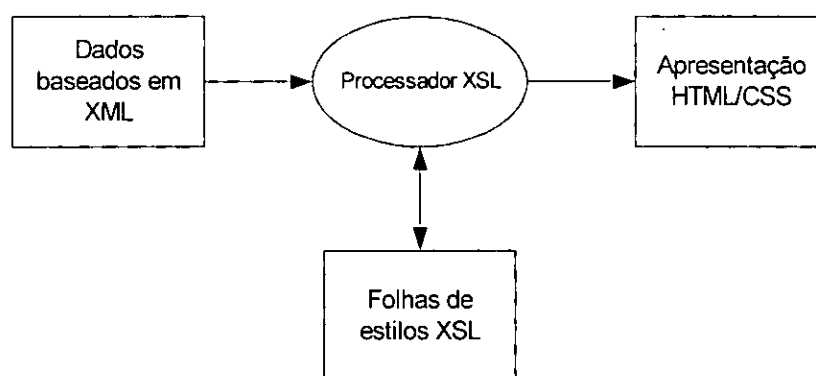


Figura 4.4 - Transformação de dados estruturados em XML para uma apresentação HTML ou CSS

Os mecanismos de ligação de dados e folhas de estilos (*Stylesheet*) podem ser usados para organizar os dados XML em uma apresentação visual e para adicionar interactividade.

A ligação de dados (“data binding”) é um aspecto do DHTML, a qual move itens individuais de dados de uma fonte de dados (por exemplo, XML) para uma apresentação HTML, permitindo que o HTML seja usado como um modelo de exibição de dados XML.

Uma folha de estilos XSL (Extensible Stylesheet Language) contém instruções de como retirar informação de um documento XML e transformá-la para outro formato (como o HTML). Essa transformação é feita de forma declarativa, e não por via de *scripts*.

Os CSSs (Cascade Style Sheets) também podem ser usados para dados XML, só que de estrutura mais simplificada. Entretanto, os CSSs não provêm uma estrutura de apresentação que se difere da estrutura da fonte de dados.

4.2.4. JAVASCRIPT

O JavaScript é uma linguagem interpretada, o que significa que o código JavaScript é executado directamente (sem compilação prévia) quando a página HTML é carregada no browser Web (Coelho, 1997).

As linguagens de programação de tipo interpretado têm, normalmente, a vantagem de ser a mais simples de aprender e de utilizar e a desvantagem de apresentar uma performance mais pobre. No caso do JavaScript, este facto não se faz sentir tanto, porque os *scripts* gerados são compilados directamente na fase de leitura e convertidos em código binário semelhante ao Java (Coelho, 1997).

O JavaScript é uma linguagem baseada em objectos, isto é, não é possível em JavaScript estender o número de classes e assim criar novos modelos de objectos. Não obstante, o facto de o JavaScript derivar, tanto a nível sintáctico como de estrutura do Java que é uma linguagem plenamente orientada à objectos (Coelho, 1997).

O facto de o código JavaScript ser embutido directamente nos ficheiros HTML a sua criação não requer uma aplicação especializada.

4.2.4.1. Comparação entre JavaScript, VBScript, Java e ASP

O JavaScript e VBScript funcionam no lado do cliente e tem a hipótese de funcionarem também ao nível do servidor com *servlets*, enquanto que as ASPs (Active Server Pages) funcionam apenas do lado do servidor. Quanto ao Java, é muito “pesado”, nem todos os browser o suportam.

O VBScript é no final de contas o código base das ASPs, mas que corre do lado do cliente.

O uso de *scripts* JavaScript e VBScript no cliente tem a grande vantagem de algumas operações serem executadas sem a necessidade de novas conexões ao servidor o que diminui o tráfego na rede.

Um das grandes vantagens das ASPs sobre o JavaScript, o VBScript e Java reside no facto do seu código fonte não ser visível a partir do browser Web, isto é, só se vê o HTML puro, daí o código fonte não pode ser usado ilegalmente pelos usuários remotos.

Os códigos JavaScript e VBScript são facilmente identificados no meio do código HTML, sendo de estrutura seguinte:

```
<SCRIPT LANGUAGE="JAVASCRIPT">
{
  history.go(-1)
}
</SCRIPT>
```

O código JavaScript simula o botão de recuo do browser, fazendo com que seja recuada uma página.

```
<SCRIPT LANGUAGE="VBSCRIPT">
{
  format currency("10000")
}
</SCRIPT>
```

Este código VBScript formata o valor 10000 para 10,000\$00.

4.2.5. APPLETS

Applets são mini-aplicações desenvolvidas em Java ou a partir do ActiveX com finalidade de adicionar interactividade em Web sites.

Uma classe Java e um controle ActiveX são módulos de *software* pré-gravados que podem ser modificados para um uso mais personalizado. O ActiveX difere do Java principalmente pela sua forma de execução – enquanto o Java é executado em uma máquina virtual (JVM-Java Virtual Machine), incorporada no browser Web, especialmente desenvolvida para interpretar mini-

aplicações Java. O ActiveX funciona como uma ponte entre programas existentes, ou entre outros programas e o sistema operativo MS Windows permitindo que compartilhem dados sem necessidade de se conhecer o modo de operação das aplicações vinculadas.

4.2.6. COOKIES HTTP

Cookies HTTP são arquivos de texto, muito pequenos, colocados no browser Web por um Web site, como cartão de identificação do cliente. Um *cookie* somente pode ser lido pelo browser Web que o hospedou e servidor Web que o forneceu.

Os *cookies* HTTP podem ser persistentes ou de sessão. Os *cookies* persistentes são aqueles que são gravados no computador cliente e possuem data e hora de expiração. E os de sessão permanecem activos apenas enquanto a sessão não for encerrada.

```
<SCRIPT LANGUAGE="JavaScript">
<!--
// cria uma instância do objecto Date
var now = new Date ();
fixDate (now);
// cookie inspira em um ano (actualmente, 365 dias)
now.setTime (now.getTime () + 365 * 24 * 60 * 60 * 1000);
var visits =getCookie ("counter");
// se o cookie não for encontrado, significa que é a primeira visita
if (!visits) {
    visits =1; // valor para um novo cookie
    document.write ("Esta é a sua primeira visita ao Web site");
} else {
    // incrementa o contador
    visits = parseInt (visits) +1;
    document.write ("Esta é a tua visita " + visits + " Web site");
}
// fixar o novo cookie
setCookie ("counter", visits, now);
// -->
```

</SCRIPT>

Exemplo de um *cookie* HTTP, escrito em JavaScript, a ser incluído entre as *tags* <HEAD> ... </HEAD>, para dizer ao cliente quantas vezes visitou o Web site.

A finalidade principal dos *cookies* HTTP é de informar o servidor Web que determinado cliente retornou a aceder o Web site. Essa informação permite:

Cliente

- exibir e actualizar informações de autenticação (subscrição) já fornecidas;
- fazer ou cancelar subscrições sobre produtos ou serviços;
- localizar facilmente produtos ou serviços novos;
- localizar informações mais relevantes.

Servidor de serviço

- colectar dados para estatísticas de visitas ao Web site;
- informar sobre actualizações e introdução de produtos ou serviços;
- criar e fornecer produtos ou serviços mais personalizados.

4.2.7. PLUG-INS

Os *Plug-ins* são módulos de *software* incluídos em um Web site, normalmente para adicionar suporte a novos formatos de arquivo ou interactividade no browser Web, sem necessidade do cliente adquirir um novo browser Web.

4.2.8. SISTEMA DE GESTÃO DE BASE DE DADOS

O desenvolvimento de uma aplicação de bases de dados utilizando a Web como interface requer a escolha de um SGBD (Sistema de Gestão de Base de Dados) que (Santos, 2000):

- permite explorar as características básicas para a implementação de uma aplicação cliente/servidor;

- é compatível com o padrão SQL (Strutured Query Language);
- fornece acesso eficaz via ODBC ou OLE DB, ou mesmo através da arquitectura de componente MSJDE (Microsoft Jet Database Engine);
- permite aceder qualquer fonte de dados via ODBC ou OLE DB;
- permite uma fácil integração com linguagens de programação de desenvolvimento de Web Sites dinâmicos (XML, JavaScript ou VBScript).

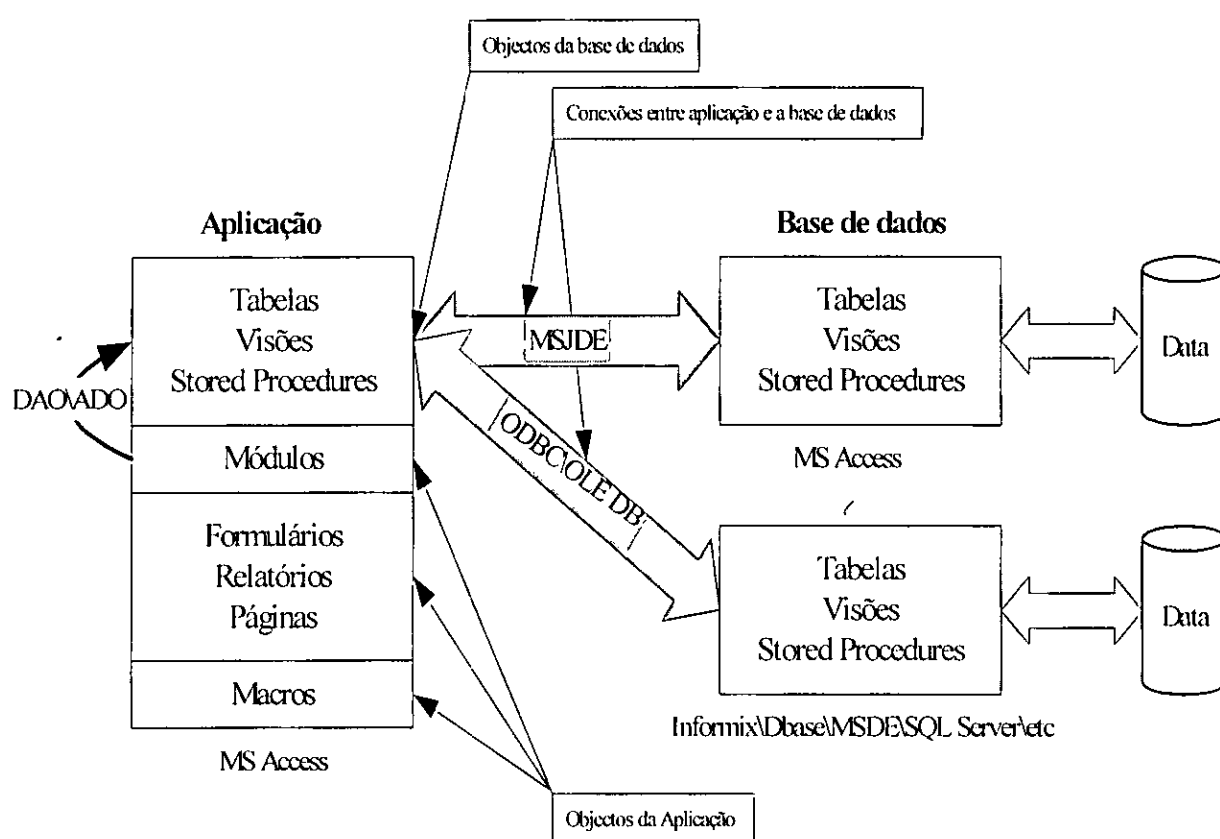


Figura 4.5 – Objectos do MS Access

Da figura 4.5 pode-se observar que utilizando o MS Access é possível criar uma aplicação cliente/servidor a partir duma solução tradicional baseada em formulários, relatórios módulos, macros e de páginas de acesso a dados, que permite manipular uma base de dados MS Access ou qualquer fonte de dados acessível via ODBC ou OLE DB.

4.2.8.1. Segurança e Privilégios no SGBD

Em termos de segurança o SGBD deve permitir basicamente:

- definir quais os privilégios o usuário tem com as instruções *Select*, *Insert*, *Update*, *Delete* em uma base de dados;
- definir quais os privilégios administrativos como a criação de usuários, definição de permissões de acesso, e possibilidades de iniciar e parar o servidor;
- A validação por *host*;
- a criptografia da base de dados.

4.2.9. PÁGINAS WEB

É possível criar três tipos de páginas Web (Mink, 2000):

- Páginas HTML geradas pelo servidor (ASP ou IDC/HTX);
- Páginas HTML estáticas;
- Páginas de acesso a dados (Data Access Pages).

Páginas HTML Geradas Pelo Servidor

Ficheiros HTML gerados pelo servidor, podem ser criados a partir de *tabelas*, *consultas*, e *formulários*. E são exibidos sob forma de tabelas dinâmicas pelo browser Web.

As páginas HTML geradas pelo servidor são usadas quando:

- pretende-se uma maior universalidade de “browse”;
- os dados mudam frequentemente;
- pretende-se visualizar, apenas para leitura, dados nas tabelas conectadas a uma fonte de dados ODBC.

Uma vez gerados os ficheiros ASP ou IDC/HTX, eles devem ser publicados num produto especializado para processamento e a cada instante que o usuário abre ou faz um “refresh” de um ficheiro ASP ou IDC/HTX, o servidor Web cria numa forma dinâmica um ficheiro HTML, e envia-o para o browser.

Páginas HTML Estáticas

Ficheiros HTML estáticos podem ser criados a partir de *tabelas, consultas, formulários e relatórios*. E o resultado é um ficheiro HTML, que é “carregado” apenas uma única vez pelo browser Web.

Em páginas HTML estáticas, para que o usuário visualize novos dados, o ficheiro deve ser exportado novamente, uma vez que ele não permanece conectado a nenhuma fonte de dados ODBC conectada. Não obstante, ser um *Snapshot* dos dados existentes no momento da publicação.

Páginas de Acesso a Dados (PAD)

PAD é um tipo especial de página Web projectado para exibir e manipular dados da Internet/Intranet armazenados em base de dados do MS Access/MS SQL Server. Embora uma PAD seja um objecto do MS Access não é gravada dentro de um arquivo MDB, mas sim em arquivos HTML dinâmicos (DHTML) com uma extensão padrão *htm*.

Uma das vantagens das PADs é o facto de o código HTML da página conter comandos XML para ler o arquivo MDB e um objecto ActiveX responsável pela barra de navegação, o que permite a manipulação da base de dados através do browser Web.

Tipos de PADs:

- **Relatório interactivo:** Este tipo é frequentemente usado para consolidar e agrupar informações armazenadas na base de dados e, em seguida, publica-las sob forma de resumo;
- **Entrada de dados:** Este tipo é usado para exibir, adicionar e editar registros;
- **Análise de dados:** Este tipo permite reorganizar dados para análise.

Manipulação de Páginas de Acesso a Dados

Quando um usuário exibe uma PAD no browser Web, ele visualiza a sua própria cópia da página, o que significa que qualquer filtragem ou classificação é exibida apenas para ele. Entretanto, alteração feita nos próprios dados como, por exemplo, modificar, adicionar ou excluir dados armazenados na base de dados, fica disponível para todos usuários que exibam a PAD.

Segurança no uso das Páginas de Acesso a Dados

Os usuários a partir dum browser Web só podem visualizar, somente o subconjunto dos dados tornados disponíveis como dados de origem na página Web.

Se a base de dados de origem requerer uma senha de acesso, os usuários devem obrigatoriamente introduzi-la para visualizar os dados.

Os usuários não podem a partir dum browser Web alterar a estrutura de dados, objectos, ou *scripts*.

4.2.10. GRÁFICOS

Gráficos são elementos importantes na construção de Web Sites. Além das preocupações comuns relativas à beleza e a criatividade, os gráficos devem ser criados tendo em conta as limitações relativas a cores, tamanho do arquivo gerado, resolução, efeito “aliasing” e forma de apresentação (Barreto, 1998).

Tamanho do Arquivo

Existe uma série de factores que influenciam o tamanho de um arquivo gráfico para a Internet:

- **Número de pontos:** Obviamente, a largura e a altura de um gráfico influenciam directamente o tamanho do arquivo;
- **Número de cores:** Em gráficos que usam cores reais, cada ponto ocupa 3 bytes e é possível representar mais de 16 milhões de cores, mas o mais comum, na Internet, é usar arquivos em que cada cor é guardada em um byte, o que permite representar até 256 cores;

- **Compressão:** A eliminação de repetições dentro do arquivo reduz consideravelmente o tamanho do arquivo. Por exemplo, os arquivos com formato GIF ou JPG são muito menores que os arquivos BMP correspondentes por guardarem bits comprimidos.

Cores

O uso de várias cores é fundamental na aparência do Web Site. *Assim o maior desafio na construção de Web Sites consiste em reduzir o tamanho do arquivo sem perder qualidade e variedade de cores* (Barreto, 1998).

Um dos problemas básicos dos gráficos na Web é saber como as cores serão visualizadas pelos usuários. A maioria dos usuários usa placas de vídeo no modo de 256 cores, e algumas dessas cores (40) são reservadas para o uso do sistema operativo e do próprio browser. Assim, se projectista do Web site utilizar gráficos com *palettes* com mais de 216 cores, é provável que os usuários não consigam ver o gráfico como foi projectado (Barreto, 1998).

A solução é utilizar somente cores na *palette* padronizada de 216 cores que inclui todas as cores básicas (verde, vermelho, amarelo, ciano, magenta, preto e branco) e pelo menos quatro graduações entre cada uma dessas cores, de modo que os usuários com placas de vídeo com 256 cores visualizem as imagens exactamente como foram projectadas. E os usuários com placas de vídeo com mais cores ou cores diferentes das indicadas acima visualizem os gráficos de forma diferente, mas com cores adaptadas para a *palette* padronizada.

Outra solução é a utilização de uma *palette* menor, o permite reduzir tanto o cabeçalho do arquivo (a *palette* em si) como o número de bits necessário para armazenar os pontos propriamente ditos.

Resolução

Resolução é o número de pixels que podem ser exibidos numa tela. Assim quanto maior a resolução, mais pixels podem ser mostrados ao mesmo tempo. Assim, como que a maioria dos usuários usa uma resolução de 640x480, é recomendável que os gráficos para a Web não sejam maiores que 600 pixels de largura para permitir que todos usuários incluindo aqueles com telas menores visualizem os gráficos como projectados.

Anti-Aliasing

Um terrível efeito colateral no uso de arquivos *bitmap* é o efeito de “escadas” que surge quando o gráfico inclui linhas inclinadas ou curvas, ou em textos com fontes muito complexas.

Uma forma de evitar este problema é uso apenas de linhas horizontais e verticais. Obviamente, não é nada prático. Outra forma, mais realista, consiste em usar cores intermediárias para suavizar as linhas. E essa solução chama-se “anti-aliasing”.

Forma de Apresentação

- **BMP (Bit Map Format)**

Este formato é necessário quando se utiliza imagens digitalizadas, mas apresenta problemas para sua exibição em diversos tamanhos, pois perde qualidade nos tamanhos muito pequenos ou muito grandes.

- **GIF (Graphic Information Format)**

- Arquivos GIF são melhores para imagens com poucas camadas de cores e são usados em “imagens de apresentação”: gráficos, figuras ou imagens de texto. Quanto menos cores usadas, mais eficiente será o arquivo GIF. Um arquivo do tipo GIF pode conter no máximo 256 cores.
- Arquivos GIF podem ser “entrelaçados”, isto significa que as imagens parecem “*fade in*”, isto é, de uma menor para uma maior qualidade o que dá aos usuários algo para ver enquanto esperam.
- Arquivos GIF podem ser “transparentes”, isto significa que se pode escolher uma ou mais cores para “não serem” exibidas permitindo que as cores do fundo da página apareçam através delas.
- Os arquivos GIF são “*lossless*”, o que significa que a qualidade da imagem não é degradada pelo processo de compressão.

- Os arquivos GIF podem ser animados, o significa que as imagens simulam movimento usando uma série de imagens individuais.
- **JPG (Joint Photographic Experts Group)**
 - Arquivos JPG são melhores para imagens com muitas cores, como fotografias e arte digitalizada. O JPG admite 16 milhões de cores.
 - A compressão do JPG é variável, pode se aplicar maior ou menor compressão a cada imagem individualmente.
 - O sistema JPG é do tipo "lossey", isto é, quanto maior a compressão, menor será a qualidade.
 - As novas versões deste tipo de imagem introduzem a possibilidade de gravar o arquivo JPG como um arquivo entrelaçado, causando a impressão de que o gráfico aparece de uma menor para uma maior qualidade.
- **PNG (Progressive Network Graphics)**

Arquivos PNG são compactos e versáteis e podem combinar as melhores características do GIF e JPG, como a capacidade de ter um fundo transparente ou conter imagens com milhões de cores. Mesmo assim, o formato PNG ainda não é muito usado, principalmente por ser um formato muito recente e não suportado por browsers muito antigos.

V. ELEMENTOS BÁSICOS NECESSÁRIOS PARA A IMPLEMENTAÇÃO DE UM WEB SITE

5.1. SERVIDORES ESPECIALIZADOS

Com um *site server*, qualquer organização pode reunir, de forma eficiente em Web sites, bases de dados, servidores de arquivos, correio electrónico e ferramentas de análise de dados, como ilustra a figura 5.1 (Microsoft, 1996).

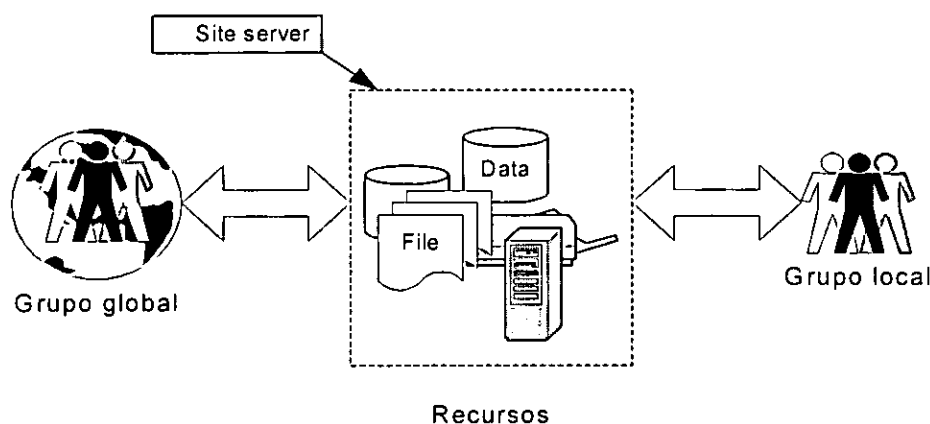


Figura 5.1 – Recursos reunidos em um *site server* para atender uma demanda local e global

Principais vantagens do uso de servidores, segundo Microsoft (1996):

- **Partilha de Recursos**

A partilha de dados baseada em servidor pode ser centralmente administrada e controlada. E os recursos facilmente localizados que recursos aleatoriamente dispersos.

- **Segurança**

A segurança é geralmente a razão primária para a escolha de redes baseadas em servidores. Num ambiente baseado em servidores, a segurança pode ser gerida por um administrador que define políticas para todos os utilizadores da rede.

- **Backup**

Por dados cruciais estarem centralizados em um servidor, é fácil ter a certeza que eles são guardados regularmente para restauro em caso de danos ou percas.

- **Redundância**

Através de sistemas de redundância, dados em qualquer servidor podem ser duplicados e mantidos, para mesmo que aconteça algo aos dados originais, uma cópia pode ser usada para restauro.

- **Número de utilizadores**

Uma rede baseada em servidores pode suportar centenas de utilizadores.

- **Considerações sobre Hardware**

O hardware do computador cliente pode ser limitado para as necessidades do usuário, mas não precisa de memória física e/ou electrónica adicional para suportar os serviços do servidor.

5.1.1. SERVIDOR WEB

O servidor Web é um computador (ou conjunto de computadores) que hospeda um ou mais Web sites e coloca-os acessíveis via browser Web. Por exemplo, quando um usuário solicita uma página Web, o servidor Web actua como um *gateway* para o Web Site, direccionando todas solicitações HTTP e retornando os resultados para a memória electrónica do computador cliente que fez a solicitação (Sun, 2002).

5.1.2. SERVIDOR DE APLICAÇÕES

O servidor de aplicações (servidor da base de dados) faz o lado de servidor nas aplicações cliente/servidor, tornando os dados disponíveis para o lado cliente. Os servidores de aplicações complementam servidores Web que actuam como *gateways* para solicitações HTTP, que requerem dados em fontes legadas (Sun, 2002).

O servidor de aplicações armazena, de forma estruturada, vastas quantidades de dados e quando uma aplicação cliente os solicita, a base de dados permanece no servidor de modo que continue disponível para outros clientes e somente os resultados da consulta são enviados para a memória electrónica do computador cliente que fez a solicitação (Microsoft, 1996).

5.1.3. SERVIDOR DE APLICAÇÕES WEB

Os serviços tradicionais para Web são baseados em uma arquitectura "two-tier", o que significa que dois componentes, tais como um servidor Web e um servidor da base de dados, são usados para criar um serviço.

Para facilitar a separação das regras de actividade são colocados servidores de aplicações Web que representam um claro nível de separação entre o servidor Web e as camadas de acesso aos dados, constituindo eficazmente uma arquitectura "three-tier", como ilustra a figura 5.2, que é neste momento uma arquitectura padrão para aplicações de base de dados baseadas na Web (Amor, 2000).

Os Web sites criados através do servidor de aplicações consistem de pelo menos três camadas de suporte:

- a camada do servidor Web (camada de apresentação - Páginas HTML);
- a camada do servidor de aplicações;
- e a camada de acesso aos dados.

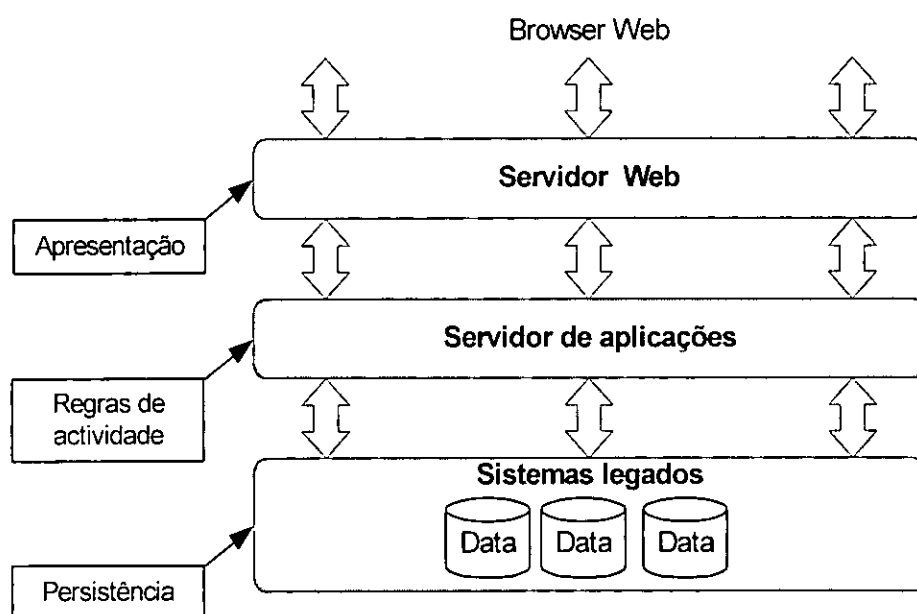


Figura 5.2 - Arquitectura "three-tier"

Neste modelo as regras de actividade estão na camada intermediária, com os servidores de aplicações a gerirem todas as funções de manipulação de dados e funções de criação de páginas HTML. As principais vantagens resultantes desse facto são de facilitar o acesso a dados e serviços que residem em sistemas legados sem conexão directa a Internet evitando que os clientes se conectem directamente a cada subsistema em "*back-end*", e de permitir a criação dum sistema de *messaging* unificado.

Os servidores de aplicações Web não somente separam as regras de actividade, oferecem também características adicionais, tais como gestão de transações, *clustering*, balanceamento de carga e tolerância a avarias (*fault tolerance*).

Os servidores de aplicações, de certo modo, separam o processo de desenvolvimento do processo de distribuição (*Web deployment*), de modo que os administradores de sistemas (e não os programadores) lidem com problemas de gestão.

O código de suporte usado para conectar as aplicações é encapsulado em um componente padrão, que é tipicamente o CORBA ou o COM (implementado apenas em plataforma PC).

Actualmente, a maioria dos servidores de aplicações Web permite que o código de suporte seja escrito em Java ou C++. A linguagem de desenvolvimento é importante porque pode determinar a plataforma de servidor de aplicações e pode afectar na escolha do modelo de computação distribuído apropriado (Amor, 2000).

A maioria dos fabricantes de sistemas de base dados oferece servidores de aplicações Web que integram fortemente aplicações Web em suas bases de dados. Embora isso garanta a velocidade mais alta possível pode forçar um vínculo com um fabricante específico, porque estas soluções usam protocolos proprietários para aceder os dados, e isto, pode tornar-se um problema, se mais tarde pretender-se uma integração de aplicações que usam sistemas de base dados diferentes (Amor, 2000).

5.2. SISTEMA OPERATIVO PARA O SITE SERVER

O Unix é um sistema operativo aprovado e adequado para o modelo de sistema aberto da Internet, mas infelizmente, a maioria dos usuários não acredita que a montagem e a manutenção do Unix sejam fáceis. Além disso, acreditam que usar Unix limita o acesso a várias aplicações para criação de intranets interactivas de preço baixo, tais como Access Basic da Microsoft, Visual Basic da Microsoft ou Delphi da Borland (Bremmer et al, 1998).

Muitas organizações preferem o Windows NT relativamente ao Unix por apresentar uma abordagem passo-à-passo (ambiente assistido através GUI – Graphic User Interface) e por poder funcionar como um sistema operativo de estação de trabalho multitarefa de alto desempenho e também como servidor avançado (Bremmer et al, 1998).

Quanto à funcionalidade, o Windows NT pode comunicar com muitos tipos de computadores. Assim como o Unix. O Windows NT pode dar segurança a dados sensíveis e pode evitar que usuários sem autorização tenham acesso aos recursos do *site server*. Assim como o Unix. Essencialmente ambos satisfazem as exigências mínimas para um sistema operativo que funcionem em ambiente de rede. Mas um servidor Windows NT valida um usuário autorizado apenas para aceder a arquivos e impressoras e sem poder executar nenhuma aplicação no servidor (para tirar o poder superior de processamento que o hardware do servidor), enquanto em Unix pode, se tiver permissão para tal (Kirch, 1998).

Quanto ao desempenho, o poder de processamento é em grande parte em função do hardware em vez do sistema operativo. Não obstante o facto de o UNIX ser historicamente um sistema operativo desenhado para uma arquitectura de hardware de alto desempenho, e o Windows NT exigir muito espaço de memória física e electrónica devido ao seu GUI (Kirch, 1998).

Quanto a custos, neste caso a relação custo - desempenho é meramente trivial. Geralmente, o Unix apresenta um preço mais baixo em relação a outros sistemas por estar associado a hardware mais caro. Mas esse facto não torna o Unix menos poderoso (Bremmer et al, 1998).

Em resumo, a selecção dos sistemas operativos que actuarão nos servidores dependerá fundamentalmente de como o *site server* foi planeado e de vários factores tais como:

- capacidade de responder e gerir solicitações;
- capacidade de interagir ou integrar sistemas “*back-end*”;
- capacidade de integrar ferramentas para criação de aplicações interactivas para Intranet/Internet;mecanismos de segurança;
- compatibilidade com o *hardware*, *softwares* e aplicações já instalados ou por instalar;
- disponibilidade no mercado;
- serviços adicionais incorporados;
- facilidades de “customização”;
- preço global (produto + serviços adicionais) e de pacotes de *Upgrade*;
- conhecimentos técnicos disponíveis;
- garantias de suporte técnico;
- preços de suporte técnico após garantia.

5.3. HARDWARE PARA O SERVIDOR WEB

Praticamente não há limites para a selecção de um sistema computacional no qual abrigar um servidor Web. Quase todos os computadores modernos podem abrigar um servidor Web (Zimmerman, 1998).

A selecção do *hardware* que actuará como servidor Web dependerá de vários factores tais como:

- nível de tráfego previsto;
- compatibilidade com o *hardware*, *softwares* e aplicações já instalados ou por instalar;
- facilidades de “customização”;
- disponibilidade no mercado;
- conhecimentos técnicos disponíveis;
- garantias de suporte técnico;
- preços de suporte técnico após garantia;
- orçamento disponível para compra de equipamentos e de pacotes de *Upgrade*.

VI. GESTÃO DO SITE SERVER

6.1. RASTREIO DO ESTADO DO UTILIZADOR

Um dos problemas mais arduos na construção de aplicações Web consiste na manutenção da informação de sessão de utilizador, devido a ausência do estado do HTTP no acompanhamento de utilizadores individuais durante a execução de uma aplicação Web, factor essencial para a autentificação e manutenção da informação duma sessão (ZDNet, 2002).

Tradicionalmente, o estado da sessão é gerido no nível mais básico através da utilização de um identificador, normalmente um *cookie* HTTP ou um campo HTML, que permite fazer uma correspondência entre uma determinada solicitação HTTP e um conjunto de dados específicos normalmente guardados numa base de dados, facilitando deste modo, a gestão de utilizadores enquanto navegam entre as páginas Web.

O rastreio do estado do utilizador permite a geração de relatórios, que por um lado, reflectem com precisão o uso do Web site, e permitem, por um lado, desenvolver um perfil dos visitantes do Web site e de seus comportamentos.

6.2. EQUILÍBRIO DE CARGAS E TOLERÂNCIA A AVARIAS

A premissa básica por detrás do equilíbrio de cargas consiste em distribuir cópias de todo o código da aplicação por diferentes servidores de aplicações e ter o servidor Web a efectuar a distribuição de solicitações equitativamente pelos servidores de aplicações envolvidos (ZDNet, 2002).

O equilíbrio de cargas pode melhorar consideravelmente o desempenho ao proporcionar escalabilidade através da incorporação de *hardware* adicional ou por permitir que aplicações críticas usem mais memória operativa.

Em ambientes de servidor único, pode-se configurar vários processos de servidor para servir os utilizadores em caso de um processo de servidor falhar, fazendo com que o servidor recomece automaticamente a execução dos processos falhados.

Nos ambientes de servidores múltiplos, pode-se fazer com que outros servidores do grupo assumam as responsabilidades quando um dos servidores pára de funcionar. O problema aqui consiste em determinar os pontos onde possam ocorrer falhas e em reforçar a tolerância a avarias nesses pontos. Pode-se, no entanto, utilizar um algoritmo para determinar a melhor forma de distribuição de cargas tendo em conta que cada servidor pode gerar um valor de desempenho global e transmiti-lo essa informação a todos os servidores disponíveis. Esse valor tem por base muitos parâmetros que incluem a carga do sistema (CPU, Input/Output e memórias), solicitações por segundo, tempo médio de execução, e cache resultante, o que permite determinar qual o servidor está mais apto a gerir o próxima solicitação.

6.3. GESTÃO DE SOLICITAÇÕES

Conforme as solicitações HTTP aumentam, o congestionamento se torna inevitável. Para usuários eventuais isso pode significar frustração, mas para os serviços, os problemas causados podem converter-se em aumento de custos e queda na produtividade (3Com, 2000).

Uma solução é o aumento da largura de banda, mas isso tem custos extremamente altos da própria aquisição da consequente aquisição de pacotes *upgrades* de *hardware* para servidores devido a sobrecarga.

Outra solução é armazenar o conteúdo frequentemente acedido, onde ele pode ser fácil e rapidamente encontrado pelos usuários, como ilustra a figura 6.1. Esse processo chama-se *cache Web*.

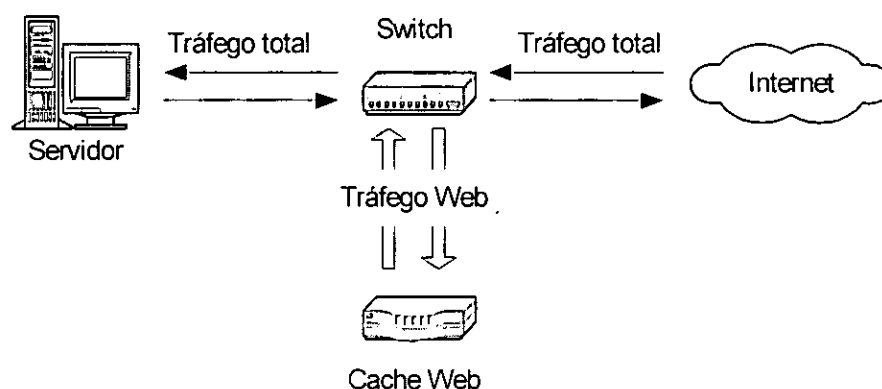


Figura 6.1 - Cache em um site server para a Web

Os ISPs utilizam o cache de rede a muitos anos para aumentar a sua capacidade de responder os “picos” de demanda por conteúdo específico e para converter conteúdos para outros formatos como, por exemplo, páginas Web repletas de gráficos em dados texto puro que pode ser transmitido para *paggers*, telefones celulares e dispositivos *handheld*.

Da figura 6.1, pode-se observar que a própria natureza do *cache Web* oferece um certo nível de tolerância a avarias. Se o cache falhar, o conteúdo da Internet continua disponível a partir do servidor de origem, e se os dados em cache forem perdidos ou corrompidos eles podem ser reconstruídos a partir dos dados no servidor de origem.

O *cache Web* pode ser usado como um ponto de verificação de todo tráfego de entrada e de saída da rede. Isto significa que todos os browsers Web irão direccionar as solicitações ao *cache Web* (Figura 6.2). Se o cache Web contiver o conteúdo solicitado, o conteúdo será imediatamente enviado para o usuário. Caso contrário, o *cache Web* funcionará como um *proxy* e buscará o conteúdo no(s) servidor (es) de origem em nome do usuário, mantendo uma cópia do conteúdo no cache Web para responder a futuras solicitações.

O *cache Web* oferece normalmente *logs* detalhados rastreando as solicitações de modo que os gestores visualizem claramente o estado e o modo do uso do Web site para um melhor planeamento da capacidade e gestão de recursos.

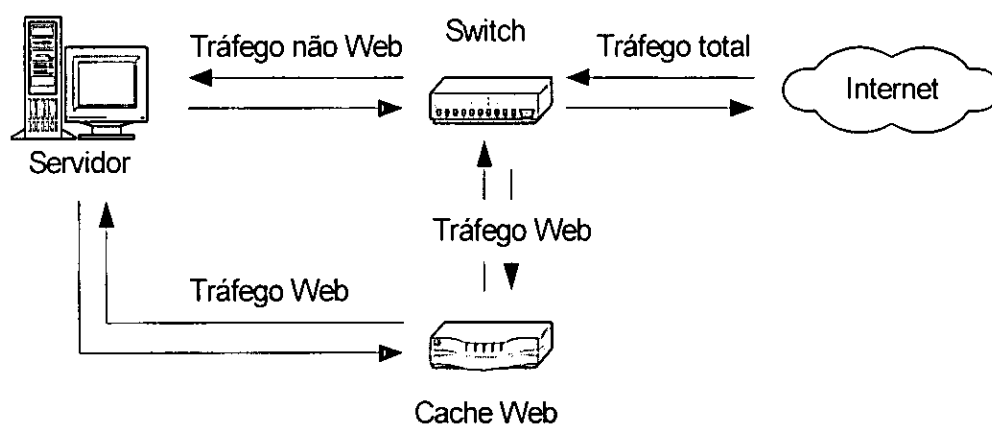


Figura 6.2 - Cache Web do tipo "proxy"

VII. SEGURANÇA

A preocupação com o controle, presente em quase todas organizações não se deve apenas as necessidades de determinar quais informações, exactamente, serão úteis na prática para melhorar o desempenho ou produzir mais eficiência operacional. Há também preocupação de preservar sigilo ou de evitar a utilização caótica de dados e sistemas. Pois a Internet apesar de abrir um mundo de possibilidades as pessoas e empresas levanta, essas preocupações que podem limitar ou mesmo pôr em causa a sua utilização.

Todavia, instalada apropriadamente a segurança, a Internet pode enriquecer uma Intranet ou servidor Web com serviços e recursos que de outro modo seria impossível fornecer.

7.1. Principais ataques associados aos serviços fornecidos aos usuários através da Internet

- *Hijacking*: situação em que alguém rouba uma conexão depois da autentificação do usuário legítimo;
- *Packet Sniffing*: situação em que alguém lê dados confidenciais enquanto eles transitam pela rede, sem interferir com a conexão;
- *False authentication*: situação em que alguém que não é um usuário válido convence o sistema de que ele é um usuário válido;
- *Trojans rootkits*: são ferramentas que substituem partes fundamentais do sistema com aplicações que se comportam como as originais, mas que: a) facilitam o acesso futuro do intruso; b) escondem a sua presença;
- *Nukes*: são programas que finalizam prematuramente uma conexão TCP/IP através do envio de pacotes ICMP com mensagens de erro. Tais programas enganam o cliente ou servidor com mensagens de detecção de problemas de conexão;
- *Virus*: programas (geralmente, *applets* - mini-aplicações Java ou controles ActiveX) desenvolvidos para alterar clandestinamente ficheiros ou *softwares* instalados num

computador. Eles têm um comportamento semelhante ao do vírus biológico: multiplicam-se, precisam de um hospedeiro, esperam o momento certo para o ataque e tentam se esconder para não ser eliminados.

Os ataques mencionados podem:

- destruir ou modificar informações ou aplicações;
- quebrar a privacidade;
- parar fornecimento de serviços.

7.2. FIREWALL

Antigamente, paredes de tijolos eram construídas entre construções em complexos de apartamentos de forma que se ocorresse um incêndio ele não poderia se espalhar de uma construção para a outra. De uma forma completamente natural, as paredes foram chamadas de *firewall* (Zimmerman, 1998).

É nessa perspectiva, que em redes de computadores, *firewalls*, como ilustra a figura 7.1, são barreiras interpostas entre a rede privada e a rede externa com a finalidade de evitar intrusos ou ataques.

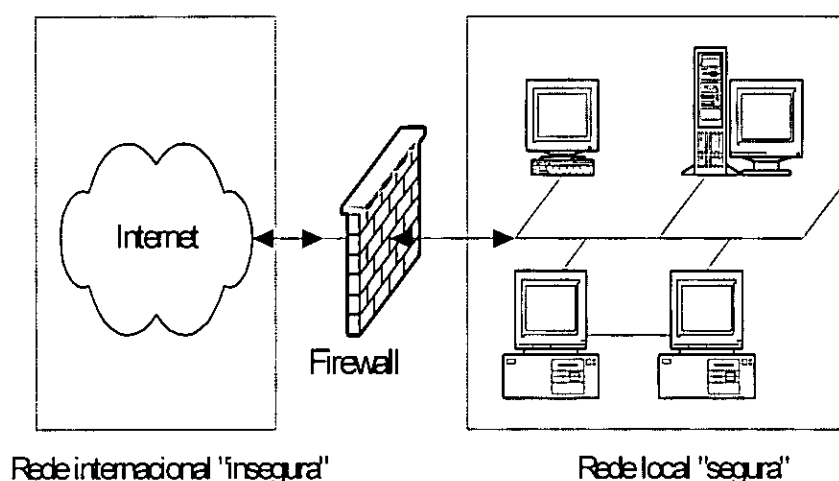


Figura 7.1 - Configuração do Firewall

A figura 7.1 mostra uma configuração simples no qual o *firewall* fica entre a rede interna e a Internet. Um *firewall* pode ser um *software*, um hardware ou uma combinação de ambos.

Numa conexão de rede local típica para a Internet, o servidor Web, o servidor de rede local, o servidor Web e o servidor firewall podem ser todos reunidos em uma única máquina ou em máquinas separadas.

7.2.1. Funções de um firewall

Ao se pretender manter uma conexão Internet segura, deve-se considerar a obtenção de protecção por meio de um *firewall* para (Zimmerman, 1998):

- controlar os pacotes TCP/IP que circulam entre o servidor Web e o resto da Internet;
- regular o tráfego da rede e descartar os pacotes provenientes de localizações indesejáveis na Internet (baseado no tipo de pacote TCP/IP e/ou na análise do arquivo de relatório de acessos);
- realizar a *filtragem de IP* dos pacotes de entrada;
- filtrar certos pacotes ICMP de saída, para que o servidor não possa divulgar informações sobre a estrutura do servidor Web.

7.3. IDS

O principal defeito da grande maioria dos *firewalls* é de não procederem a verificação do conteúdo dos pacotes em tráfego na rede interna. Um IDS (Intrusion Detection System) é uma solução complementar à instalação de *firewalls*, que permite identificar os ataques mais cedo e permitir uma tomada de decisão em tempo útil (ZDNet-Redes, 2002).

O IDS tem a função de analisar permanentemente o tráfego da rede (interno e externo) e compara-lo com padrões conhecidos comportamentos de intrusos.

Um IDS pode analisar o tráfego na rede de diferentes perspectivas, cada uma com objectivos e resultados diferentes (ZDNet-Redes, 2002):

- Análise de assinaturas;
- Análise de comportamentos;
- Análise de anomalias nos protocolos.

Análise de Assinaturas

Análise de assinaturas (*Signature detection*) consiste na procura de padrões específicos de tráfego, correspondentes a determinado ataque. A desvantagem é que o padrão de ataque tem de ser conhecido e programado antecipadamente no IDS.

Análise de Comportamentos

Análise de comportamentos (*Behaviour detection*) consiste na análise e procura de padrões de comportamento, através da identificação de anomalias estatísticas. A ideia é que uma rede segue determinados padrões de comportamento que resultam em determinadas estatísticas. Alterações dessas estatísticas devido, por exemplo, ao registo de maior tráfego a horas poucos usuais ou ao aumento do número de pacotes de terminado tipo de protocolo, resultam na identificação de um possível ataque.

Análise de Anomalias nos Protocolos

Análise de anomalias nos protocolos (*Protocol anomaly detection*) consiste na análise de conformidade dos padrões de pacotes de determinado protocolo. Por exemplo, o uso de caracteres inválidos para a subversão do funcionamento normal de um servidor Web.

7.4. PROXY

O *proxy* é usado para mascarar todos endereços IP nos pacotes de saída como se fossem originários do próprio servidor *proxy* para evitar a detecção externa da estrutura da rede. Caso contrário, os *hackers* que estejam a controlar o tráfego que sai da rede eventualmente poderão determinar os endereços IP individuais e em seguida falsificar os IPs (*IP spoofing*) e devolve-los ao servidor.

Normalmente o servidor de *proxy* suporta os protocolos HTTP, HTTPS (HTTP Secure) e FTP.

7.5. ANTIVÍRUS WEB

Pode ser qualquer aplicação usada para proteger o *site server* contra ataques de vírus. O antivírus pode actuar normalmente, também, como um *gateway* entre uma rede (Intranet) ou um servidor *standalone* e a Internet. Toda a informação é inspeccionada sempre que transaccionada entre o *site server* e a Internet, de acordo com uma configuração específica baseada nas políticas de segurança estabelecidas.

Os programas antivírus contemplam bases de dados com objectos hostis conhecidos, que uma vez detectados durante a navegação são mantidos isolados enquanto os outros continuam activos.

7.6. SSL

Usar um servidor com suporte SSL (Secure Socket Layer) é uma boa maneira de providenciar segurança para um Web site, pois permite criptografar os dados que fluem entre o cliente e o servidor garantindo a autenticidade tanto do cliente como do servidor, ou seja, um usuário de um *site server* com suporte ao SSL tem a certeza da origem dos dados (Santos, 2000).

O SSL usa uma técnica de criptografia de chave-pública, onde o servidor envia uma chave pública para o cliente critografando informações na qual somente o servidor poderá descriptografar com uma chave privada.

7.7. OUTRAS FORMAS DE SEGURANÇA

Para além das formas referidas existem muitas outras formas de segurança complementares para garantir a segurança do Web Site que passam por:

- desenvolver política de segurança para usuários. Quais os programas permitidos para os usuários executarem? Com que frequência eles devem alterar suas senhas? Os usuários na rede local terão a permissão de discar para a Internet a partir de um modem privado?
- rotular todos os recursos (informações, aplicações ou equipamentos) críticos do sistema de acordo com a sua sensibilidade e estabelecer um nível de segurança específico sobre cada um deles;

- controlar por meio de senhas o acesso a índices de base de dados pesquisáveis por meio de interfaces Web;
- monitorar cuidadosa e frequentemente os arquivos de relatório do sistema;
- se permitir acesso via FTP, deve-se definir permissões restritivas de arquivos e directório FTP;
- implementação de controles de segurança física que permitem restringir o acesso a áreas críticas (*Site server room*) para prevenir acessos não autorizados que podem causar danos ou roubos dos equipamentos;
- implementação de mecanismos de rastreio de utilizadores;
- implementação de mecanismos de autentificação mais rigorosos para o acesso a sites especiais com acesso limitado ou na aquisição e registro de produtos;
- implementação de mecanismos de controle ambiental que permitem evitar danos nos equipamentos derivados de factores tais como: calor, humidade, poeira, fogo ou falhas eléctricas;
- uso de técnicas de criptografia para dados críticos transaccionados na rede;
- implementação de sistemas de *backup* para dados críticos e aplicações;
- nunca partilhar informações de autentificação de utilizadores com terceiro sem o seu consentimento;
- uso de programas antivírus actualizados e de versões mais recentes de browsers Web, para permitir a detenção mais eficaz de *applets* hostis ou de origem não conhecida.

VIII. MODELO SITE SERVER

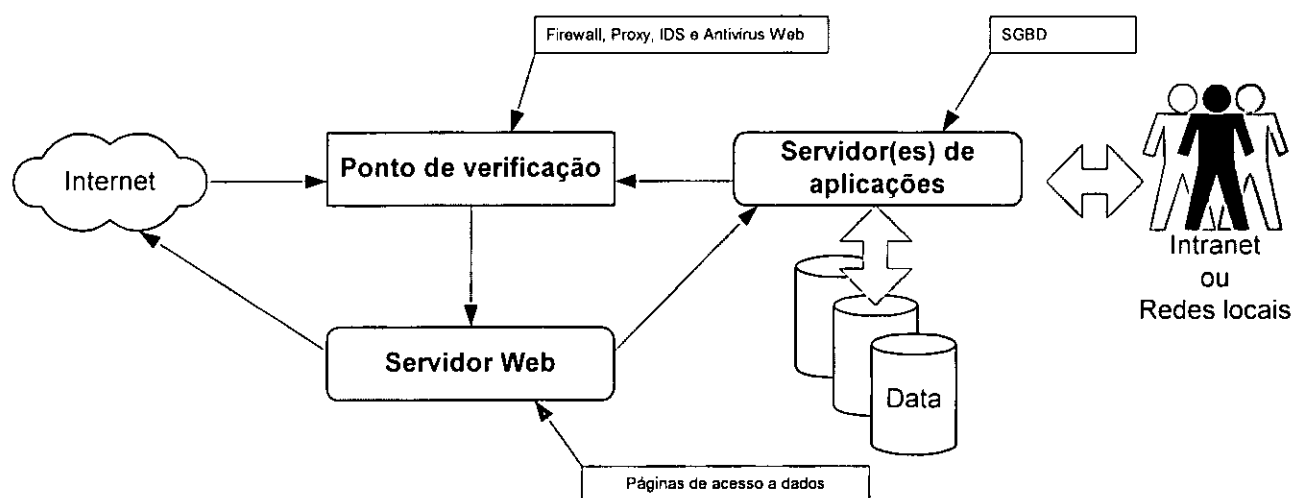


Figura 8.1 – Modelo de *site server*

Descrição

Da figura 8.1, pode observar-se que:

- todo tráfego de entrada e de saída para a Internet passa pelo ponto de verificação o que garante maior segurança;
- o servidor Web actua como um *gateway* entre Internet e as bases de dados ou servidores de aplicações legados, significando que este modelo pode constitui-se eficazmente numa arquitectura "*two-tier*" ou "*three-tier*".

Os SGBDs nos servidor de aplicações podem ser diferentes e a unificação das fontes de dados pode ser feita a partir de páginas de acesso a dados dum Web site no servidor Web ou a partir uma aplicação que faz *links* ODBC ou OLE DB as fontes de dados.

IX. CASO DE ESTUDO

9.1. INTRODUÇÃO

Os trabalhos académicos, especificamente trabalhos científicos, teses e dissertações, cumprem um papel de grande importância no desenvolvimento científico, tecnológico e cultural de um país. Constituem uma das expressões mais verdadeiras da actividade de pesquisa face a complexidade e diversidade dos problemas que a sociedade impõe (IBCIT, 2002).

Actualmente, quando se fala em sistemas de informação para pesquisa, verifica-se uma clara e irreversível tendência para o desenvolvimento de sistemas baseados em tecnologias Web, padrões e protocolos abertos, que incluam a disponibilidade de texto completo em formato digital e que garantam a preservação do conteúdo, publicação na rede pelo próprio autor, integração a redes internacionais, além da busca unificada que supere a heterogeneidade e especificidade de sistemas (IBCIT, 2002).

É neste contexto, que a incorporação desse conceito tecnológico aos sistemas de teses abre novas perspectivas no que se refere não só a disposição de dados para pesquisa como também para a consolidação do espaço académico moçambicano na Internet.

9.2. O WEB SITE

Objectivos do Web Site

O Web Site tem como objectivo principal o cadastro virtual, em uma base de dados, que consiste de resumo de teses produzidas em Moçambique ou por moçambicanos no estrangeiro.

Público Alvo

Estudantes e professores.

Ferramentas

- Uso do MS Access para a implementação da base de dados;
- Uso de PADs para geração de páginas HTML e consultas à base de dados;
- Uso de funções JavaScript para a validação dos dados.

Estrutura de navegação

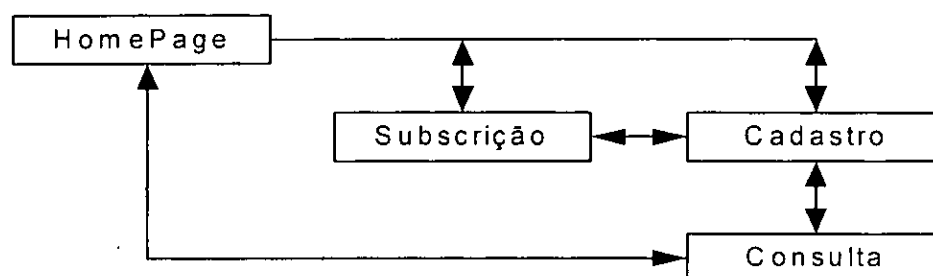


Figura 9.1 - Estrutura de navegação

Descrição:

Home page

A partir da Home Page (ver anexo II.1) os usuários podem:

1. Subscrever-se no Web site;
2. Realizar o cadastro de suas obras (resumo de teses);
3. Realizar consultas à base de dados contendo os cadastros efectuados.

Subscrição

A partir desta página os usuários têm a possibilidade de fazer o seu registo na base de dados através do preenchimento dum formulário colocado a sua disposição (ver anexo II.2). Só depois do usuário estar inscrito é que pode efectuar o cadastro das suas obras.

Cadastro

A partir desta página os usuários subscritos podem realizar o registo das suas obras através do título, área, data, instituição, país, supervisor e resumo (ver anexo II.3).

Consulta

A partir desta página os usuários podem realizar consultas à base de dados, através da digitação duma palavra chave e/ou através de consultas mais criteriosas (ver anexo II.4).

Na opção consultar por palavra chave, deve-se simplesmente digitar no campo respectivo a palavra que se deseja encontrar na base de dados. Com isso accionar a busca na base de dados de todos os registros que contem a palavra especificada. Optando-se por uma consulta mais criteriosa, o usuário deve realizar a busca por meio de outros campos além da palavra chave.

Fluxograma

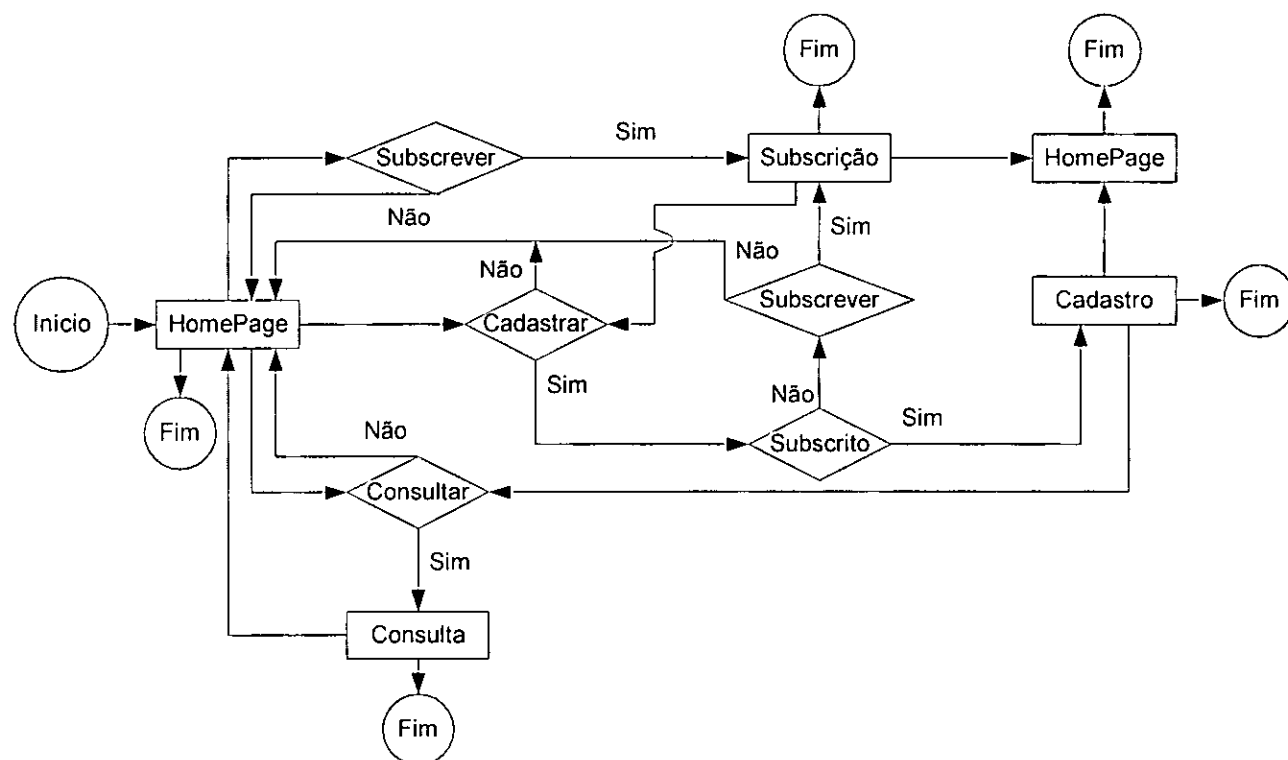


Figura 9.2 - Fluxograma da sequência de acções executadas durante a navegação

9.3. REGRAS DE ACTIVIDADE (BUSINESS RULES)

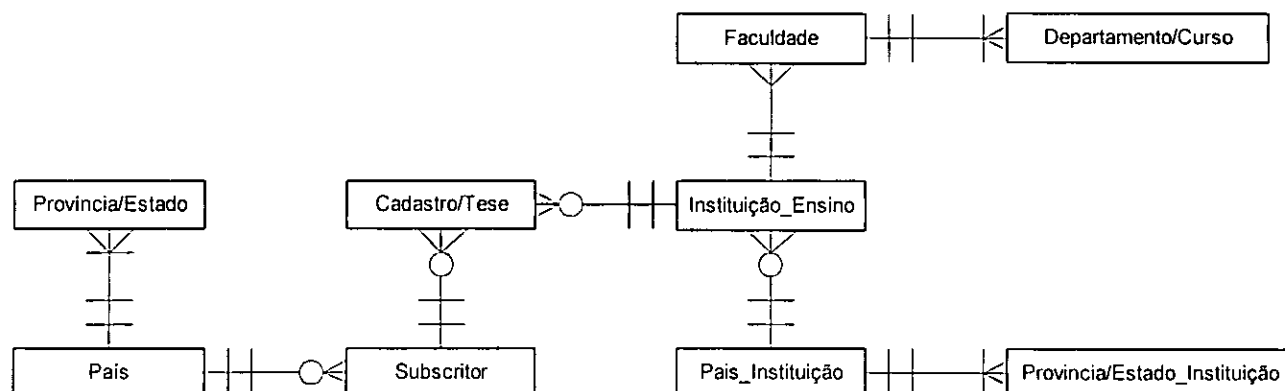


Figura 9.3 - DEA (Diagrama de Entidade e Associação)

Descrição:

Um subscritor pode efectuar vários cadastros. Caso contrário, sujeita-se a inutilização da sua subscrição no prazo de 5 dias (ver anexo III. 1).

Uma obra cadastrada pertence apenas a um único subscritor. E não deve existir um cadastro sem subscritor.

No acto da subscrição o subscritor deve introduzir uma senha a sua escolha, com mínimo de 5 caracteres alfanuméricos, que a partir dela efectuará o cadastramento das suas obras (ver anexo III. 2).

Para a confirmação da senha introduzida, o usuário deve reintroduzi-la a partir dum campo da página (não residente na base de dados) (ver anexos III. 4 e III.5).

O subscritor apenas pode cadastrar a tese se ele for de nacionalidade moçambicana ou se tese foi defendida em Moçambique (ver anexo III. 3).

9.4. SITE SERVER

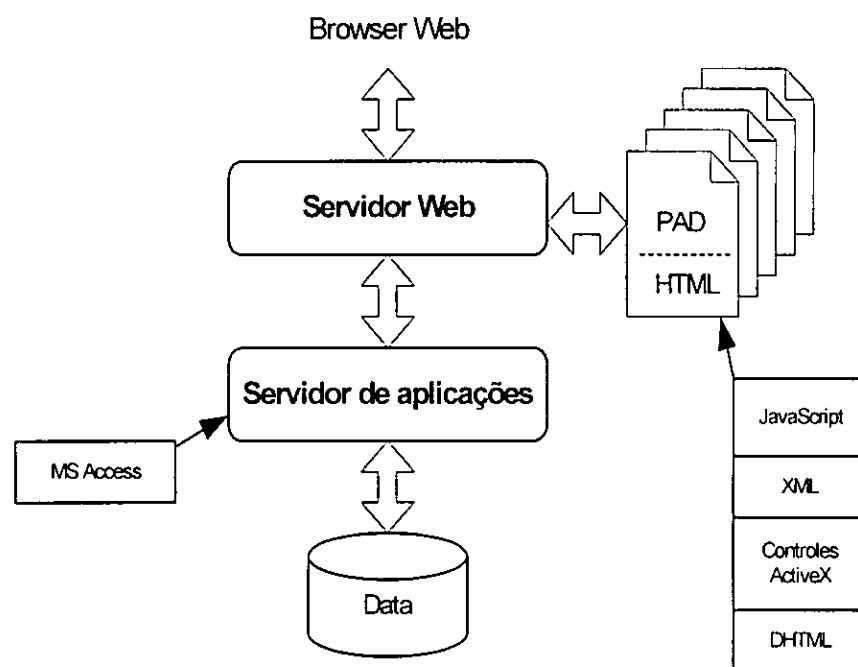


Figura 9.4 - Modelo de *site server* para Web site descrito acima

Descrição:

O Web Site, a interface entre o cliente browser e a base de dados, é implementado através de PADs do MS Access, que acedem através do XML e do Controle ActiveX DAO a base de dados implementada em MS Access.

O servidor Web armazena as PADs sobre forma de ficheiros HTML para permitir o acesso multi-usuário via HTTP.

O servidor de aplicações provê o acesso a base de dados respeitando as regras de actividade (Business Rules) da base de dados.

Os comandos JavaScript são incorporados dentro do código HTML das PADs para permitir a validação, no cliente browser, dos dados (preenchimento) antes de serem enviados, o que permite minimizar o número de solicitações HTTP e consequentemente o tráfego na rede.

X. CONCLUSÕES E RECOMENDAÇÕES

Na do estudo teórico e prático realizado, pode-se concluir que:

- a Internet para além de ser um sistema de integração pública constitui também uma estratégia de comunicação organizacional. Pois é possível que membros de uma organização interajam operacionalmente entre si ou com o exterior via Internet;
- *a grande utilidade duma base de dados em um Web Site*. Pois permite a implementação de várias soluções que dinamizam o conteúdo de um Web Site, facilita sua administração, possibilita a criação de aplicações cliente/servidor completas, comércio electrónico e todas as facilidades que uma base de dados pode oferecer;
- *a grande utilidade de interfaces Web em uma base dados*. Pois permite o acesso e a análise *on-line* de conjuntos de dados complexos – *Data werehouse*, facilita a captura e disseminação *on-line* de dados, facilita a unificação de fontes de dados homogéneas ou heterogéneas, e propicia a existência de relações horizontais entre pessoas dentro das organizações, e entre clientes e fornecedores, sobretudo através da eliminação de gestores intermediários;
- o facto de o MS Access na versão 2000 apresentar o objecto PAD permitiu a criação de páginas Web que interagem de forma dinâmicas com uma base dado MS Access, sem se recorrer à codificação HTML. O que reduziu de maneira drástica o tempo de desenvolvimento;
- a integração das ferramentas de desenvolvimento descritas ao longo do trabalho, com o sistema operativo Windows NT possibilitou uma implementação optimal do Web site descrito no capítulo IX;
- a incorporação de diversas ferramentas está a tornar progressivamente a Internet mais interactiva e mais fácil de usar. O grau de conhecimentos informáticos necessários para o seu uso tende a diminuir.

E recomenda-se que:

- para garantir uma maior segurança e um maior suporte de registos da base de dados deve-se criar as PAD em MS Access conectado a uma base de dados Microsoft SQL Server ou Informix protegida pelo sistema integrado de segurança do Microsoft Windows NT/2000 e por um *firewall*;
- a base de dados MS Access mostrou-se uma ferramenta de muito fácil acesso, rica em documentação e fornece grandes facilidades no que se refere a criação e manipulação de base de dados, mas porém com grandes limitações no número de registos a suportar, sendo que o seu uso como servidor avançado de base de dados para Web não é recomendável;
- se não houver previsão de tráfego muito intenso na rede pode-se dedicar até um PC para abrigar o servidor Web. Caso contrário, deve dedicar uma máquina de alto desempenho projectada para tal fim e o uso de *cache Web* para o alívio do congestionamento;
- o servidor Web e os restantes servidores no *site server* (caso existam) podem ser todos reunidos em uma única máquina se o orçamento disponível for escasso, caso contrário é recomendável separa-los para garantir um ambiente mais seguro e escalonado;
- as organizações ou instituições produtoras e/ou disseminadoras de informação façam um esforço em termos de investimento, na aquisição ou desenvolvimento de sistemas de base de dados virtuais com vista à implantação futura de um sistema integrado de bibliotecas ou livrarias virtuais no país.

XI. BIBLIOGRAFIA

11.1. REFERÊNCIAS BIBLIOGRÁFICAS

- (3Com, 2000) 3Com Corporation (2000). Cache da Web: Uma solução Para Aliviar o Congestionamento na Internet. <http://www.3com.com> 07/06/02.
- (Amor, 2000) Amor, D. (2000). A(R)evolução do E-business. São Paulo, Makron Books.
- (Barreto, 1998) Barreto, A. (1998). Desenvolvendo Web Sites – Scripts, Ferramentas, Conceitos e Dicas. <http://www.tutorial.zzn.com>, 05/04/2001.
- (Bremmer et al, 1998) Bremmer, L. M.; Lasi, A. E.; Servati, A. (1998), A Bíblia da Intranet. São Paulo, Makron Books.
- (Coelho, 1997) Coelho, P. (1997). Animação de Páginas na World Wide Web com JavaScript. Lisboa, FCA.
- (Designer, 2002) Network Designers (2002). Integração ISDN X Brooktrout NetAccess. <http://www.networkdesigners.com.br/Aritgos/isdn/isdn/isdn.html>. 06/06/2002.
- (Evans, 1998) Evans, T. (1998). Construindo uma Intranet. São Paulo, Makron Books.
- (IBICT, 2002) IBICT (2002). Biblioteca Digital Brasileira. <http://www.ibict.br> 09/04/2002.
- (Kirch, 1998) Kirch, J. (1998). Servidor Microsoft Windows NT Versus UNIX. <http://www.microlink.com.br/~buik/> 04/04/2001.
- (Microsoft, 1996) Microsoft Corporation (1996). Networking Essentials. Washington, Microsoft Press.

- (Mink, 2000) Mink, C.; Anibal Junior (2000). Microsoft Access 2000. São Paulo, Mackron Books.
- (Oliveiro, 2000) Oliveiro, A. J. C. (2000). HTML - Orientado Por Objecto 4.0. São Paulo, Editora Érica.
- (Pires, 2000) Pires, H. (2000). Programação - Dynamic HTML.
<http://www.tutorial.zzn.com> 04/04/2001.
- (Santos, 2000) Santos, L. L. (2000). Manipulação de Banco de Dados via Internet Usando Ferramentas Gratuitas. Erechim/RS.
- (Sobreira, 1998) Sobreira, F. (1998). Apostila de Internet.
<http://www.sidenet.com.br/user/flavio.sobreira/> 09/03/98.
- (Sousa, 1997) Sousa, I. D. (1997). Negócios & Internet. Lisboa, FCA.
- (Sun, 2002) Sun Microsystem (2002). Sun Deployment of E-Commerce Solutions.
<http://www.sun.com/solutions/> 31/05/02.
- (W3C, 2000) W3C (2000). XML – Extensible Markup Language.
<http://www.w3.org/xml/> 22/05/2002.
- (ZDNet, 2002) ZDNet (2002). Servidores de Aplicação Para Web.
www.zdnet.pt/pcmagazine/tecnologias/9901/servidores.shtml
24/05/2002.
- (ZDNet-Redes, 2002). ZDNet-Redes (2002). Redes: Trancas e Alarmes na Internet.
<http://www.zdnet.pt/redes/0202/tecnologias2.shtml> 31/05/2002.
- (Zimmerman, 1997) Zimmerman, S.; Brown, C.L.T. (1997). Kit de Construção de Sites Web. São Paulo, Makron Books.

11.2. BIBLIOGRAFIA CONSULTADA

Carriço, J. A. (1998). HTML e Sheets – Técnicas de Criação de Páginas e Estilos Para a Web. Lisboa, Cti Edições.

INT Media Group (2002). JavaScript Cookies. <http://www.webreference.com/js> 07/06/02.

Microsoft Corporation (2000). Microsoft Access 2000. <http://www.msdn.microsoft.com/library>. 15/9/2001.

Torres, A.B. (1997). JavaScript – Guia de Referências. www.geocities.com/siliconvalley/way/3105 13/05/2002.

ANEXOS

LISTA DE ANEXOS

Anexos I

Anexo II. 1- DEA

Anexos I

Anexo II.1 - Home Page

Anexo II.2 – Subscrição

Anexo II.3 – Cadastro

Anexo II.4 - Consulta

Anexo III

Anexo III. 1 - Código SQL para eliminar subscrições fora do prazo.

Anexo III. 2 - Código JavaScript para a verificação do comprimento da senha.

Anexo III. 3 - Código Access Basic para a certificação da tese.

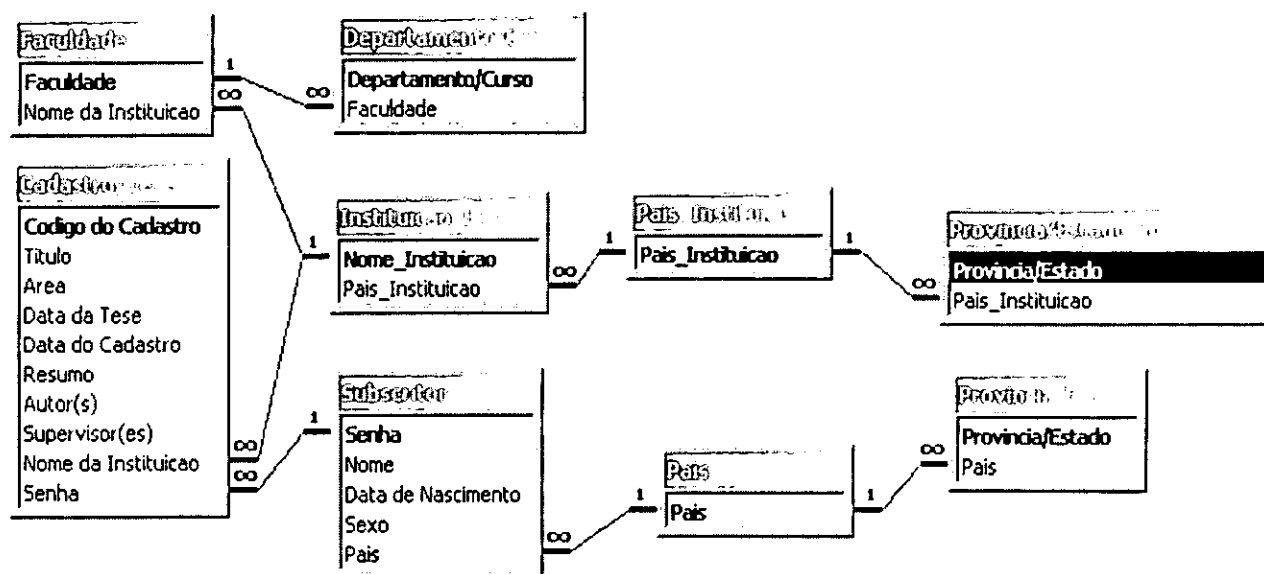
Anexo III. 4 - Código HTML para a criação de um campo na página para a confirmação da senha.

Anexo IV

Anexo IV. 1 – Modelo OSI

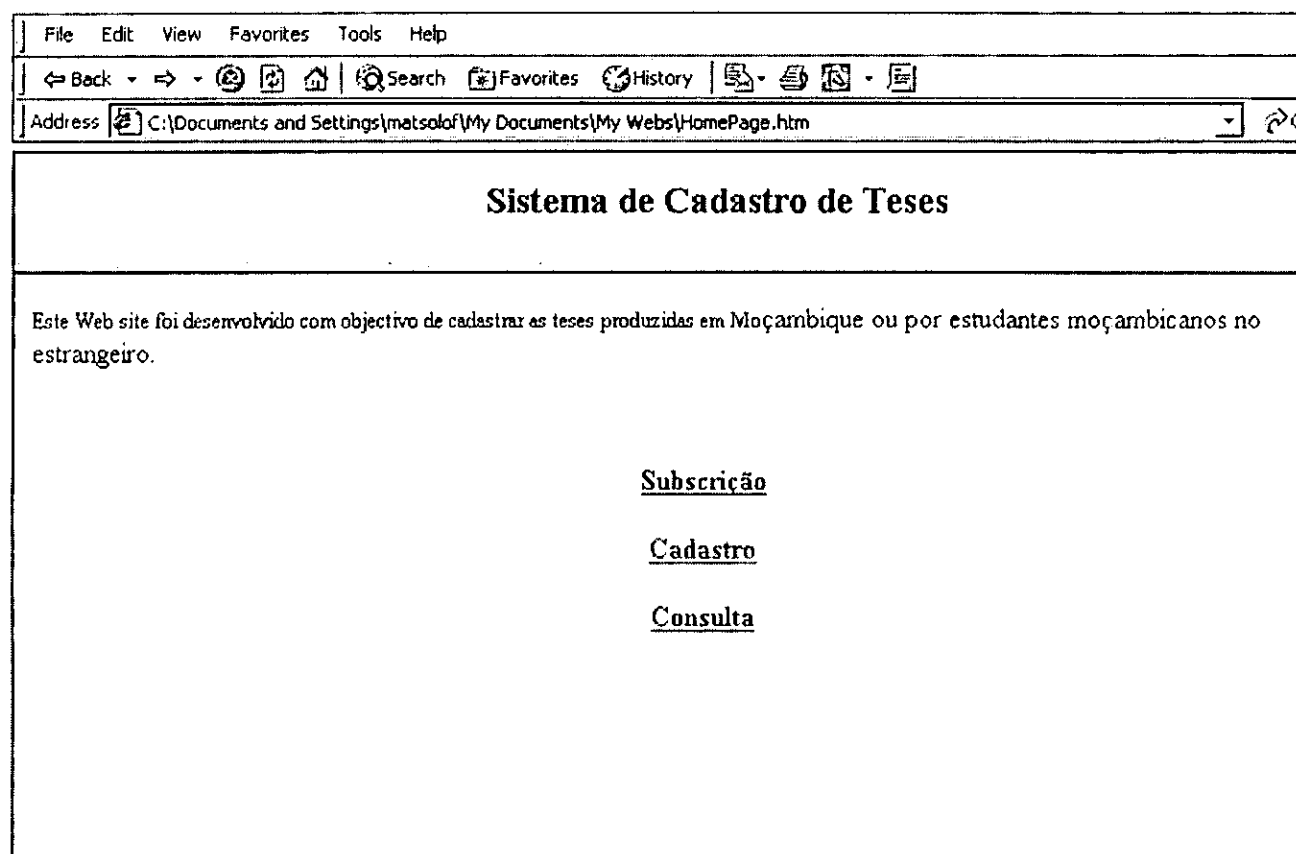
ANEXOS I

Anexos I.1 - DEA (Diagrama de Entidade e Associação)



ANEXOS II

Anexos I 1.1 - Home Page



Anexos II.2 – Subscrição

File Edit View Insert Format Tools Window Help

Subscricao

Senha	xxx (A deve ter entre 5 e 10 caracteres)	Reintroduza a senha
Nome Completo	Ana Carlos da Costa	
Nacionalidade	Mocambicana	
Data de Nascimento	3/3/1975	
Sexo	Femenino	
Pais	Mocambique	
Provincia\Estado	Maputo	
Data da Subscricao	17-05-2002	
Endereco Electronico	Ana@hotmail.com	
Licenca	O Web site ora desenvolvido permite o cadastro virtual de resumo de teses defendidas em Mocambique ou por mocambicanos no estrangeiro. Qualquer subscritor tem no maximo 5 dias para fazer a seu primeiro cadastro, caso contrario sera eliminado da base de dados. Os direitos.	

Subscricao 1 of 2

≤≤Voltar
Cadastrar

Anexos II.3 - Cadastro

File Edit View Insert Format Tools Window Help

Cadastro

Senha: xxx

Codigo do Cadastro: 1 Data de Cadastro: 02-06-2002

Instituicao: Univesidade Eduardo Mondlane

Faculdade: Faculdade de Ciencias

Area: Informatica

Titulo: Acesso a Base de Dados Via Internet

Pais: Mocambique

Data da Tese: 02-06-2002

Autor(es):

Supervisor(es):

Resumo: A convergência e o uso integrado das tecnologias de informação, cujo paradigma é a Internet, a rede mundial de computadores, têm contribuído nos últimos anos na criação de um novo ambiente de acesso e troca de informações numa escala global transformando a sociedade como um todo, criando bases para novas formas de comunicação e socialização que influenciam profundamente a maneira como as pessoas e/ou empresas interagem e geram informações, ao mesmo tempo que as colocam a disposição. Estas transformações têm exercido uma

<< Voltar

Consultar >>

Cadastro 1 of 1

Anexos II.4 - Consultas

A consulta será feita através do botão  da barra de navegação da página.

ANEXO III

Anexo III. 1 - Código SQL executado no servidor de aplicações para eliminar subscrições fora do prazo:

```
DELETE *  
FROM Subscricao  
WHERE (((Date()-[data da subscricao])>5) AND ((Exists (select senha from cadastro))=False));
```

Anexo III. 2 - Código JavaScript para a verificação da senha (mínimo 5, maximo 10 caracteres alfanuméricos).

A idéia consite na criação de um conjunto de “descritores de validação” associados a cada elemento da página (ou formulário). Cada “descriptor de validação” será apenas uma cadeia de caracteres alfanuméricos que especificam o tipo de acção a ser executada em caso de violação de determinada cláusula.

```
...  
ArrValidationDes[id++]=  
    [  
        ["required", "Introduza a senha"] // não permite senha vazia  
        ["minlen=5", "A senha não deve ter menos que 5 caracteres"] // não permite senha com menos de 5  
caracteres alfanuméricos]  
        ["maxlen=10", "A senha não deve ter mais que 10 caracteres"] // não permite senha com mais de  
10 caracteres alfanuméricos  
    ];  
...
```

Anexo III. 3 - Código Access Basic a ser executado no servidor para verificar se o resumo de tese a cadastrar elaborada por moçambicano ou foi defendida em Moçambique:

...

VarX as Variant

If DataAccessPages!Cadastro.Pais_Instituição <> "Moçambique" then

VarX = Dlookup ("[Pais]", "Subscritor", "[Senha] = DataAccessPages!Cadastro.Senha")

If VarX <> "Moçambique" then

' Cadastro não autorizado!

...

Else

'Cadastro bem sucedido!

...

End If

Else

'Cadastro bem sucedido!

...

End If

...

Anexo III. 4 - Para a confirmação da senha introduzida, o usuário deve reintroduzi-la apartir dum campo da página (não residente na base de dados). Eis o código HTML para a criação do campo na página:

<Form method=Post>

Confirmar Senha: <Input type="Text" name="Confirmar Senha" size=20 maxlen=20>

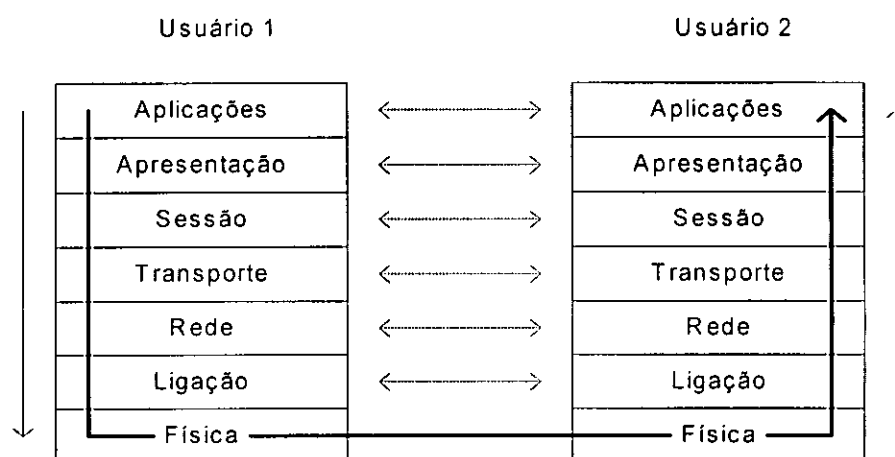
</Form>

Anexo III. 5 - Código JavaScript para a comparação das senhas introduzidas:

```
<SCRIPT LANGUAGE="JAVASCRIPT">  
  
if (Senha != Confirmar Senha)  
    {  
        alert.window("Reintroduza a senha");  
        ...  
    }  
  
</SCRIPT>
```

ANEXO IV

Anexo IV. 1 – Modelo OSI



GLOSSÁRIO

Browser Web: *Software* que interpreta as *tags* em um ficheiro HTML, formata-as para o formato Web, e exibe-as aos utilizadores. Alguns browsers Web podem ainda permitir que os utilizadores enviem e recebam e-mail, leiam newsgroups, e executem ficheiros de som e vídeo embebidos num Web site.

Data Warehouse: A idéia básica é extrair dados de diversas fontes, transformando, filtrando e consolidando-os de acordo com a necessidade e armazenando-os em uma base de dados (*Data Warehouse*) modelada adequadamente para atender à análise multi-dimensional.

Encriptação: Codificação de dados que para sua leitura é necessária a respectiva chave decodificada.

Gateway: É um computador de rede que corre um *software* para mais de uma interface de rede. O gateway gere o fluxo de dados entre duas redes de acordo com tabelas de roteamento.

Hacker: O termo hacker significar mestre, perito, e é empregado para referenciar pessoas que invadem sistemas de informação usando ferramentas modernas.

Hiperlink: Imagens ou textos que dão acesso a outros documentos hipertexto.

Hipertexto: Texto constituído por *hiperlinks*.

Home Page: É o termo normalmente de usado para nomear a primeira página de um Web Site.

Host: É um componente da arquitectura TCP/IP (que não realiza a função de roteamento) que é apenas a origem ou destino de um datagrama IP.

Internet: É um conjunto de vários computadores conectados através do protocolo TCP/IP e conexões discadas, ou dedicadas.

Link: Ligação a um objecto por *hiperlink*.

Modelo OSI: O modelo OSI (*Open System Interconnect*) foi criado em 1977 pela ISO (*International Organization for Standardization*) com o objectivo de criar padrões de conectividade para interligar sistemas diferentes de computadores locais e remotos.

ODBC: É um padrão para acesso a bases de dados relacionais.

OLAP: É o conjunto de ferramentas que possibilita a exploração *on-line* dos dados contidos em um *Data Warehouse*.

OLE DB: É um padrão de acesso a dados projectado para aceder dados que estejam armazenados em diversos formatos, incluindo arquivos sequenciais-indexados e bases de dados hierárquicas, e-mails, sistemas de armazenamento remoto, texto e gráficos.

Script: É um programa intermediário que permite executar a partir do browser Web muitas operações de rede como: operações de base de dados, pedidos *on-line* e outras operações que requerem uma interacção extensiva entre o usuário e o servidor Web.

VBScript: É uma versão "light" do Visual Basic destinada à Internet. O código VBScript é inserido directamente entre as HTML para dar maior efeito às páginas.

Web: Área da Internet que contem documentos em formato multimédia (texto, som, imagem e/ou vídeo) interligados entre si.

Web Site: É um conjunto de páginas que pertencem a um mesmo grupo de informações, ou seja, a um mesmo endereço de Internet.

Workstation: É um computador a partir do qual o usuário acede informações no servidor (base de dados etc.) e rodam aplicações locais (Word, Excel, etc.).