

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA  
COMPUTAÇÃO**

**Agnaldo Izidoro de Souza**

**UM AMBIENTE GENÉRICO PARA A  
CONSTRUÇÃO DE BIBLIOTECAS DIGITAIS**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos  
requisitos para obtenção do grau de Mestre em Ciência da Computação

**Murilo S. de Camargo**

Florianópolis, 02 / 2001

# UM AMBIENTE GENÉRICO PARA A CONSTRUÇÃO DE BIBLIOTECAS DIGITAIS

Aluno: Agnaldo Izidoro de Souza

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, Área de Concentração (Sistemas de computação) e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.



---

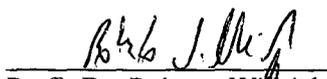
Prof.<sup>o</sup> Dr. Murilo S. de Camargo (Orientador)



---

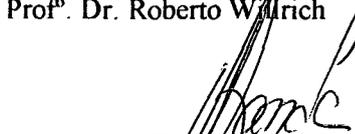
Prof.<sup>o</sup> Dr. Fernando Gauthier

Banca Examinadora



---

Prof.<sup>o</sup> Dr. Roberto Wilrich



---

Prof.<sup>o</sup> Dr. Vitorio B. Mazzola



---

Prof.<sup>o</sup> Dr. Rosvelter Coelho da Costa

Andar e pensar um pouco,  
que só sei pensar  
andando.

Três passos, e minhas  
pernas já estão pensando.  
Aonde vão dar esses  
passos?

Acima, abaixo? Além?  
Ou acaso se desfazem no  
vento sem deixar nenhum  
traço?

Paulo Leminski

Esta dissertação é dedicada àqueles  
a quem sou eternamente grato: Luis  
e Clementina, meus pais, Cristina,  
minha noiva.

## *Agradecimentos*

- Ao Prof<sup>o</sup> Dr. Murilo S. de Camargo, meu orientador, cujos conselhos sempre vieram de forma efetiva e extremamente prática para a concretização de meu objetivo.
- A todos os professores que estiveram ao meu lado durante esta jornada.

## Sumário

<b>LISTA DE FIGURAS.....</b>	<b>x</b>
<b>LISTA DE TABELAS.....</b>	<b>xii</b>
<b>LISTA DE ABREVIATURAS .....</b>	<b>xiii</b>
<b>INTRODUÇÃO.....</b>	<b>17</b>
<b>CAPÍTULO 2 .....</b>	<b>21</b>
<b>BIBLIOTECAS: A SERVIÇO DO SABER .....</b>	<b>21</b>
INTRODUÇÃO .....	22
2.1 BIBLIOTECAS DIGITAIS .....	22
2.2 HISTÓRICO.....	26
2.3 INTEGRAÇÃO: UM DESAFIO .....	29
2.4 VANTAGENS/DESVANTAGENS.....	30
2.5 ARMAZENAMENTO E MULTIMÍDIA.....	31
2.6 DISTRIBUIÇÃO.....	32
CONCLUSÃO .....	33
<b>CAPÍTULO 3 .....</b>	<b>34</b>
<b>MULTIMÍDIA .....</b>	<b>34</b>
INTRODUÇÃO .....	34
3.1 ELEMENTOS MULTIMÍDIAS .....	34
3.2 MAPEAMENTO DE IMAGENS E VÍDEO .....	37
3.3 DOCUMENTOS HIPERTEXTOS .....	39
3.4 DOCUMENTOS MULTIMÍDIA.....	40
3.5 DOCUMENTOS HIPERMÍDIA.....	41
3.6 APLICAÇÕES MULTIMÍDIA INTERATIVAS.....	41
CONCLUSÃO .....	42
<b>CAPÍTULO 4 .....</b>	<b>43</b>
<b>AValiação de sistemas de bancos de dados para multimídia .....</b>	<b>43</b>

INTRODUÇÃO .....	43
4.1 SGBDs OBJETO-RELACIONAIS.....	43
4.2 SGBDs RELACIONAIS.....	45
4.3 SGBDs ORIENTADO A OBJETOS.....	46
4.4 BANCO DE DADOS MULTIMÍDIA.....	48
4.4.1 <i>Objetivos a serem alcançados</i> .....	51
4.4.2 <i>Funções</i> .....	54
4.4.3 <i>Vantagens dos SGBDs Multimídia</i> .....	54
4.4.4 <i>Critérios para avaliação de um Banco de Dados Multimídia</i> ....	55
CONCLUSÃO .....	57
<b>CAPÍTULO 5 .....</b>	<b>58</b>
<b>INTEGRAÇÃO VIA INTERNET.....</b>	<b>58</b>
INTRODUÇÃO.....	58
5.1 <i>WORLD WIDE WEB</i> .....	58
5.2 ORIGEM E EVOLUÇÃO DA <i>INTERNET</i> .....	59
5.3 A <i>INTERNET</i> NO BRASIL.....	61
5.4 ORGANIZAÇÃO DA <i>INTERNET</i> .....	61
5.5 ARQUITETURA <i>WEB</i> .....	62
5.6 INTEGRANDO <i>WEB</i> E BANCO DE DADOS.....	66
5.7 ARQUITETURAS DE INTEGRAÇÃO.....	69
5.7.1 <i>Arquiteturas de Servidores Web</i> .....	71
5.7.1.1 Common Gateway Interface ( CGI).....	72
5.7.1.2 Servidor <i>Web</i> estendido.....	75
5.7.1.3 <i>Application Programming Interface (API)</i> .....	81
5.7.2 <i>Arquiteturas de cliente Web</i> .....	85
5.7.3 <i>Arquiteturas de cliente e servidor Web</i> .....	88
5.7.4 <i>Linguagem de acesso</i> .....	90
5.7.4.1 <i>Personal Home Page (Php)</i> .....	90
5.7.4.2 <i>Active Server Pages (Asp)</i> .....	91
Conclusão.....	93
<b>CAPÍTULO 6 .....</b>	<b>94</b>

<b>PROBLEMAS DE INTEGRAÇÃO.....</b>	<b>94</b>
INTRODUÇÃO .....	94
6.1 TRANSAÇÕES HTTP E <i>BROWSER</i> .....	94
6.2 TRANSAÇÃO <i>WEB</i> E BANCO DE DADOS.....	95
6.3 TRANSAÇÃO <i>WEB</i> - ATRAVÉS DE PÁGINAS .....	97
6.3.1 <i>Várias páginas como uma unidade atômica</i> .....	98
6.3.2 <i>Páginas que não fazem parte de uma transação</i> .....	98
6.3.4 <i>Páginas com dupla funcionalidade</i> .....	99
6.4 FALHAS DE TRANSAÇÕES.....	100
6.5 CONTROLE DE RESTRIÇÕES DE INTEGRIDADE.....	101
6.5.1 <i>Propriedades ACID</i> .....	102
6.6 ACESSO AO BANCO DE DADOS.....	103
6.7 SEGURANÇA DE DADOS.....	104
6.7.1 <i>Autenticação de Usuários</i> .....	106
6.7.2 <i>Criptografia</i> .....	108
CONCLUSÃO .....	109
<b>CAPÍTULO 7.....</b>	<b>111</b>
<b>    AMBIENTE PARA CONSTRUÇÃO DE BIBLIOTECAS DIGITAIS.....</b>	<b>111</b>
INTRODUÇÃO.....	111
7.1 METODOLOGIA APLICADA .....	112
7.2 DESCRIÇÃO DA APLICAÇÃO.....	113
7.3 CONSTRUÇÃO DA APLICAÇÃO .....	115
7.3 DESENVOLVIMENTO .....	127
7.4 SEGURANÇA.....	128
7.5 CONCORRÊNCIA .....	129
7.6 CONSISTÊNCIA .....	129
7.7 RECUPERAÇÃO.....	130
7.8 PORTABILIDADE.....	131
7.9 PROBLEMAS ESTRUTURAIS.....	131
CONCLUSÃO .....	132
<b>CONCLUSÕES FINAIS .....</b>	<b>133</b>

<b>REFERÊNCIAS BIBLIOGRAFICAS.....</b>	<b>137</b>
<b>APÊNDICE A.....</b>	<b>142</b>
<b>APÊNDICE B.....</b>	<b>144</b>
<b>APÊNDICE C.....</b>	<b>144</b>

## Lista de Figuras

### Capítulo 5

Figura 1 - Documento Hipertexto .....	59
Figura 2 - Arquitetura do ambiente Cliente / Servidor .....	63
Figura 3 - Arquitetura do ambiente de integração <i>Web</i> e Banco de Dados. ....	68
Figura 4 - Arquitetura de três camadas.....	70
Figura 5 - Arquitetura de integração <i>Web</i> / Banco de dados .....	71
Figura 6 - Arquitetura de Servidores <i>Web</i> com Interface em Cgi .....	74
Figura 7 - Arquitetura de servidor <i>Web</i> estendido / SSI.....	76
Figura 8 - Arquitetura servidor Web estendido / Acesso Direto ao Banco .....	77
Figura 9 - Arquitetura de servidor Web estendido / via ODBC.....	79
Figura 10 - Componentes de Funcionamento da arquitetura ODBC.....	80
Figura 11 - Funcionamento da arquitetura APIs.....	82
Figura 12 - Funcionamento da arquitetura ISAPI / NSAPI .....	85
Figura 13 - Arquitetura de Cliente Web / Aplicação externa.....	86
Figura 14 - Arquitetura de cliente Web estendido / Modo estendido.....	87
Figura 15 - Arquitetura de Clientes e servidor Web.....	89
Figura 16 - Ambiente de programação ASP/PHP .....	92

### Capítulo 7

Figura 17 - Frameworks: delimitando o problema.....	112
Figura 18 - Tela de entrada do site da biblioteca digital .....	115
Figura 19 - Tela de cadastro de novo usuário .....	117
Figura 20 - Boleto gerado para pagamento.....	117
Figura 21 - Tela de login de usuário .....	118
Figura 22 - Manutenção no cadastro do usuário.....	119
Figura 23 - Opções de acesso no módulo base (Administrador).....	120
Figura 24 - Opções de acesso do módulo base (Funcionário).....	121
Figura 25 - Opções de acesso do módulo usuário.....	122
Figura 26 - Pesquisa geral na base de dados multimídia .....	123
Figura 27 - Pesquisa avançada na base de dados multimídia.....	123

Figura 28 - Resultado da Pesquisa.....	125
Figura 29 - Detalhes do resultado da Pesquisa.....	126
Figura 30 - Verificando o conteúdo da mídia.....	127

## Lista de Tabelas

### Capítulo 4

Tabela 1 - Principais Bancos de Dados .....	48
Tabela 2 - Comparação entre os bancos .....	56

### Capítulo 5

Tabela 3 - Serviços de Segurança.....	105
Tabela 4 - Comparativo entre tipos de criptografia .....	109

## Lista de Abreviaturas

<i>ADT</i>	<i>Tipos Abstratos de Dados</i>
<i>API</i>	<i>Application Programming Interface</i>
<i>ASP</i>	<i>Active Server Pages</i>
<i>BLOBs</i>	<i>Binary Large Objects</i>
<i>CGI</i>	<i>Common Gateway Interface</i>
<i>CPU</i>	<i>Unidade de Processamento Central</i>
<i>DLL</i>	<i>Dynamic Link Librarys</i>
<i>DVD</i>	<i>Digital Versatile Disk</i>
<i>EARN</i>	<i>European Academic and Research Network</i>
<i>FPS</i>	<i>frames per second</i>
<i>FTP</i>	<i>File Transfer Protocol</i>
<i>GIF</i>	<i>graphics interchange format</i>
<i>HTML</i>	<i>Hypertext Markup Language</i>
<i>HTTP</i>	<i>Hypertext Transfer Protocol</i>
<i>IAB</i>	<i>Internet Activities Board</i>
<i>ID</i>	<i>Identificação de Usuários</i>
<i>IDC</i>	<i>Internet Database Connector</i>
<i>IETF</i>	<i>Internet Engineering Task Force</i>
<i>IIS</i>	<i>Internet Information Server</i>
<i>IRTF</i>	<i>Internet Research Task Force</i>
<i>ISAPI</i>	<i>Internet Server Application Programming Interface</i>
<i>JPEG</i>	<i>Joint Photographic Experts Group</i>
<i>MVC</i>	<i>Modelo-Visão-Controlador</i>
<i>MPEG</i>	<i>Motion Pictures Experts Group</i>
<i>MRI</i>	<i>Magnetic Resonance Imaging</i>
<i>NSAPI</i>	<i>Netscape Application Programming Interface</i>
<i>ODBC</i>	<i>Open Database Connectivity</i>
<i>OO</i>	<i>Orientada a Objetos</i>
<i>PHP</i>	<i>Personal Home Page</i>
<i>RNP</i>	<i>Rede Nacional de Pesquisa</i>
<i>SGBDOO</i>	<i>Sistema de Gerenciamento de Banco de Dados Orientado a Objeto</i>
<i>SGBDs</i>	<i>Sistema de Gerenciamento de Banco de Dados Relacional</i>
<i>SIBI</i>	<i>Sistema Integrado de Bibliotecas da Usp</i>

<i>SM</i>	<i>Servidor Multimedia</i>
<i>SQL</i>	<i>Structured Query Language</i>
<i>SSI</i>	<i>Server Side Include</i>
<i>TCP/IP</i>	<i>Transmission Control Protocol / Internet Protocol</i>
<i>URL</i>	<i>Uniform Resource Locator</i>
<i>WWW</i>	<i>World Wide Web</i>

## Resumo

**Palavras-chave:** biblioteca digital, *frameworks*, banco de dados multimídia, integração *Web* banco de dados.

Com a popularização de inúmeros meios digitais, tais como texto, imagem, som, vídeo, etc., e a grande tendência para a digitalização da informação, têm surgido mecanismos que armazenam e permitem acesso a um volume cada vez maior de informação multimídia. Sendo assim, esta dissertação elaborará um ambiente para construção de bibliotecas digitais para multimídia, provendo uma infra-estrutura e interfaces flexíveis, permitindo maior facilidade, integração e manutenção, atendendo assim os requisitos de incorporação e demanda das mídias. Neste contexto, o ambiente irá possibilita, dentre outras coisas, a captura, o armazenamento e a recuperação desses meios digitais. de uma maneira rápida e interativa, possibilitando também a manutenção de vários tipos de mídias, formatos, áreas, bibliotecas e conteúdo, tornando-o totalmente genérico para a construção de bibliotecas digitais, pois sua atualização e feita dinamicamente. Utilizando-se destes conceitos é possível fazer uma analogia com o termo, *framework*: que captura a funcionalidade comum a várias aplicações de um mesmo domínio de problema, que pode ser adaptado para resolver seus problemas específicos. Ao contrário das abordagens tradicionais para a reutilização de *software* que se limitam basicamente na construção de bibliotecas de classes, que permitem reutilizar não apenas componentes isolados mas toda a arquitetura de sistemas computacionais projetadas para um domínio específico. Por isto, toda esta aplicação está moldada visando o seu reuso, flexibilidade e extensibilidade, as quais permitem a construção de qualquer tipo de biblioteca com qualquer tipo de mídia. BUZATO & RUBICA(1998). Para tanto faz-se uso de mecanismos de integração *Web* e banco de dados, fazendo referência à arquitetura que roda no lado do Servidor *Web*. A escolha da aplicação para este estudo deve-se a sua representatividade no contexto atual das arquiteturas disponíveis no mercado e sua relevância no contexto do problema. Optou-se em implementar esta aplicação utilizando tecnologia proprietária da *Microsoft*, por ter uma maior ganho de integridade e desempenho, devido à tecnologia pertencer a uma mesma empresa.

## Abstract

**Key-words:** digital library, *frameworks*, multimedia database, mechanisms of web integration and database.

With the popularization of numberless digital ways, such as text, sound, images, video, etc., and the high tendency for digitalization of information, has arisen mechanisms that store and allow access to a crescent volume of multimedia information. In such case, the present dissertation will elaborate an environment for building of a multimedia digital library, providing infra-structure and flexible interfaces, what allows a larger facility, integration and maintenance, thus answering both requisites of incorporation and request of medias. In this context, the environment will enable among other things, the capture, the storage and recovering of these digital ways, of a rapid and interactive way. It also possibilities the maintenance of several types of media, formats, surfaces, libraries and contents, becoming them totally generic for construction of digital libraries, because its actualization is done dynamically. Through these concepts it's possible to do an analogy with term "framework" which capture the functionality common to several applications to a same dominion of problem. It can be adjusted to resolve their specific problems on the contrary of traditional approaches for reutilization of software that limit themselves basically on the construction of classes library, that permit reutilizate not only isolated components, but also whole architecture of computational systems projected for a specific dominion. That's why all this application is modeled aiming its reapplication, flexibility and extension, which allow construction of any kind of library with any kind of media [BUZATO & RUBICA (1998)]. In this sense we'll make use of mechanisms of *Web* integration and database, alluding to the architecture that roll in the Server *Web side*. The choice of the application for this study it's because of its representativeness in the actual context of architectures availables in the market and its importance in the context of the problem. Technology was opted in implementing this application using proprietor of the Microsoft, for having a bigger profit of integrity and performance, due to technology to belong to one same company.

# INTRODUÇÃO

Segundo DRUCKER(1999), estamos vivendo a era do conhecimento, da informação. Termos como “auto-estradas da informação”, “bibliotecas digitais”, são expressões que entraram em nosso cotidiano. Relatos e previsões da morte do livro impresso e das bibliotecas tradicionais são encontradas com certa facilidade no meio da comunicação, onde é grande a tendência para a digitalização da informação.

Essas novas realidades e perspectivas não deixarão de se refletir nas instituições que têm tido a missão de promover o acesso à informação, através da organização, descrição, preservação e difusão dos documentos: as bibliotecas. Se estamos falando de uma sociedade da informação, a qual traz fantásticas oportunidades que se abrem aos cidadãos, como a educação, a auto-formação, o lazer, a cultura, devemos então também falar de bibliotecas digitais, as quais facilitam o acesso à informação digital e multimídia, através de seus próprios mecanismos. Assim, podemos aceitar uma definição de biblioteca digital tal como proposta por KAHN & WILESKI (1995):

"Bibliotecas digitais são sistemas que armazenam, acessam, disseminam e gerenciam mídias digitais"

A construção de bibliotecas digitais, que armazenam e permitem acesso a um volume cada vez maior de informação multimídia (texto, imagem, som, vídeo, etc.) em diversos formatos, já se iniciou, provendo assim acesso e distribuição eletrônica de materiais de vários tipos e mídias, para pessoas que estejam localmente ou remotamente distantes. É desse modo que é preciso destacar a biblioteca digital como uma solução altamente eficaz, pois com os recursos tecnológicos hoje disponíveis, ela poderá viabilizar o acesso à informação de uma maneira mais rápida, segura e organizada, não tencionando substituir a original, mas sim incorporando e auxiliando as bibliotecas tradicionais para a modernidade.

As Bibliotecas Digitais devem disponibilizar, ainda, recursos, incluindo pessoal especializado para selecionar, construir e oferecer acesso intelectual para

interpretação, distribuição, além de preservar a integridade e assegurar que a persistência do tempo sobre obras no acervo comum estejam garantidas com a digitalização.

## Objetivos

O objetivo geral deste trabalho é, portanto, elaborar um ambiente que consiga construir uma biblioteca digital, as quais provêm uma infra-estrutura e interfaces flexíveis, permitindo maior facilidade, integração, manutenção e confiabilidade no software, atendendo assim os requisitos de incorporação das diversas mídias existentes no mercado de uma forma genérica, utilizando para isso mecanismos de integração *Web* e banco de dados, tendo em vista as atuais demandas.

Esta biblioteca terá como funcionalidade principal, ser um meio de integração de inúmeras áreas de atuação, desde uma pesquisa avançada até uma simples consulta a base de dados, de modo que podemos então definir suas funções básicas:

- Criação e captura;
- Gerenciamento e armazenamento;
- Acesso e busca; e
- Distribuição.

### Objetivos específicos:

- Definir, através de levantamento bibliográfico, os elementos para a implantação de uma biblioteca digital;
- Identificar os conceitos de biblioteca digital, multimídia, banco de dados e sua integração, de acordo com a bibliografia existente;
- Disponibilizar conteúdos ao público em geral, não se restringindo apenas ao meio acadêmico ou um grupo seletivo de pesquisadores;
- Integrar uma base de dados com a possibilidade de consulta simultânea de documentos, imagens, áudios, vídeos e outras;

- Fortalecer os processos de coleta, organização e disponibilização na *Internet* das informações geradas;
- Elaborar uma estrutura genérica à qual se terá acesso via *Internet*, onde será feita toda a distribuição das mídias através da utilização de um banco de dados;
- Manter as bases de dados do sistema;
- Estabelecer critérios de níveis para os usuários; e
- Construir um mecanismo que controle o uso da biblioteca pelo usuário, através de cadastramento.

### Justificativa

Nos nossos dias, existe uma forte tendência de que, cada vez mais, a *Internet* venha a se transformar em um agente de mudança de caráter sócio-cultural. Tal mudança já pode ser observada no mundo acadêmico, e rapidamente vem afetando os outros sub-sistemas. Neste contexto, percebe-se que o volume de informação acessível pela *Internet* cresce exponencialmente, aumentando a necessidade de formas de organização da mesma, pois os mecanismos tradicionais já não suportam mais toda esta informação. Daí a importância de iniciativas que venham a contribuir para melhoria da organização da informação, reunindo questões relativas a um mesmo tema, aprofundando-os e promovendo discussões entre pessoas interessadas. Uma biblioteca digital é uma iniciativa que vem ao encontro dessa necessidade. As bibliotecas tradicionais, que detêm um grande contingente de informação de forma organizada, devem estar preparadas para a utilização das facilidades de acesso promovidas pela *Internet*. E com a globalização, tornar-se-ão as principais fontes de informação mundial. É importante que, desde já, a sociedade se prepare para esse novo contexto.

Em vista disto, a biblioteca digital se faz imprescindível para o processo de aprimoramento de seus serviços, em consonância com a sua missão de apoio ao ensino, pesquisa e extensão universitária, desenvolvidos na Instituição.

## **Estrutura do trabalho**

O trabalho está estruturado de forma que, gradativamente, se obtenha os conceitos necessários à compreensão da amplitude do problema e à verificação de lacunas existentes na bibliografia que contribuem para a existência deste.

Assim sendo, primeiramente será estudado no capítulo 2, através da bibliografia existente, o tema “bibliotecas digitais” em que serão abordadas questões relevantes ao tema proposto, como por exemplo, sua historicidade, possibilidades de integrações, vantagens e desvantagens.

O capítulo 3 irá tratar da multimídia e da forma como seus elementos de estruturação podem dar suporte à bibliotecas digitais. Estamos levando em conta que, hoje, a multimídia é indispensável à construção de um projeto como este.

Em seguida, no capítulo 4, serão tratadas características de banco de dados, tentando estabelecer parâmetros entre bancos de dados multimídia, pois entendemos que esta seria a melhor opção para fazer esta implementação de acordo com a bibliografia disponível. Neste capítulo também serão levantadas algumas questões referentes a alguns produtos de mercado para, se possível, fazer a escolha do melhor banco de dados para esta situação. Assim, estabelece-se a ligação dos capítulos dois e três.

No capítulo 5 serão estudados, através de revisão bibliográfica, os tipos de integração disponíveis atualmente, quais suas vantagens e seu funcionamento, para então escolher uma para a implementação deste trabalho.

O capítulo 6 irá trazer as problemáticas envolvidas neste contexto, restringindo-se os textos nas questões de transações e segurança, pois se trata dos principais problemas para a integração.

Por último, no capítulo 7, será exposta a metodologia aplicada para a construção desta biblioteca e serão apresentadas as soluções encontradas; para finalizar, chegaremos às conclusões obtidas da aplicação da metodologia proposta, bem como recomendações para posteriores trabalhos.

## Capítulo 2

---

### Bibliotecas: a serviço do saber

A mídia impressa sempre foi o principal veículo de registro e disseminação de informações. Nesse meio, as bibliotecas surgiram como instrumento capaz de reunir informações produzidas e oferecê-las ao público em geral. Nela, o deslocamento físico e o local são exigidos para que se possa acessar as informações desejadas.

Com a tecnologia de distribuição de informações e o declínio dos custos dos equipamentos e sistemas informatizados, os quais proporcionam um crescimento na utilização de mídias digitais, a mídia impressa vem sendo substituída em uma grande variedade de aplicações. Ao invés de ir até a estante mais próxima para ler um livro, basta ligar o computador. A época de procurar dados através de enciclopédias, jornais ou microfimes contidos em um acervo já está sendo substituída pela criação das bibliotecas digitais. As bibliotecas surgem em meio a este mundo digital como grandes catálogos e provedores da informação.

As bibliotecas estão geralmente classificadas em quatro categorias:

- Polimídia: utilizado para denotar os diferentes tipos de meios para armazenar informações, o termo polimídia refere-se a bibliotecas que contêm informações de diferentes mídias. Elas são similares às bibliotecas tradicionais contendo livros que convivem com fitas, vídeos, *cd-roms*, microfimes, *Internet*, etc.;
- Eletrônica: é o termo que se refere ao sistema no qual os processos básicos da biblioteca são de natureza eletrônica, o que implica ampla utilização de computadores e de suas facilidades na construção de índices on-line, busca de textos completos e na recuperação e armazenamento de registros;
- Digital: difere-se das demais porque suas informações existem somente no formato digital, não possuindo livros na forma

convencional; dispõem de todos os recursos de uma biblioteca eletrônica, oferecendo pesquisa e visualização dos documentos, tanto local quanto remotamente por meio de computadores; e

- Virtual: está ligada ao conceito de realidade virtual e, portanto, utiliza recursos de *software* que simulam um ambiente de biblioteca na tela do computador, criando imagens em três dimensões que possibilitam entrar e circular pelas prateleiras de uma biblioteca virtual.

Vale dizer que, no entanto, nos ateremos daqui por diante no que se refere às bibliotecas digitais, objeto deste trabalho.

## 2.1 Bibliotecas Digitais

O conceito de Bibliotecas Digitais, não é novo conforme coloca ARMS(2000), pois os avanços nas áreas de telecomunicação fizeram com que muitas universidades criassem informações digitais para serem publicadas neste meio. Segundo GRAHAM(1995), bibliotecas digitais são uma coleção de informações eletrônicas organizadas para serem utilizadas a longo prazo, sendo que as mesmas devem satisfazer a necessidade dos usuários. Coloca-se como prioridade para estas bibliotecas a função de prover informação eletrônica desde a sua organização até o seu armazenamento por períodos que excedam a vida humana.

Uma outra definição bastante interessante foi proposta por KAHN e WILESKI(1995), segundo os quais:

"Bibliotecas digitais são sistemas que armazenam, acessam, disseminam e gerenciam mídias digitais".

Esta definição sintetiza melhor o conceito de Bibliotecas Digitais, pois ela satisfaz todas as necessidades básicas de um usuário, tanto em relação ao armazenamento, quanto ao acesso e a distribuição. Assim, podemos colocar que Bibliotecas Digitais são organizações que disponibilizam recursos, incluindo pessoal especializado para selecionar, construir e oferecer o acesso, interpretação, distribuição,

além de preservar a integridade e assegurar que a persistência do tempo sobre obras no acervo comum esteja garantida com a digitalização.

Por isto, as Bibliotecas Digitais estão sendo criadas, administradas e utilizadas incorporando-lhes políticas inovadoras de desenvolvimento nas suas coleções e novas técnicas de disponibilização e distribuição de seus produtos via *Internet*, com impacto significativo na convergência dos avanços computacionais e nas tecnologias de comunicação. Com sua criação, nada altera as funções das bibliotecas tradicionais, onde identificamos questões fundamentais em seu uso, quais sejam:

- Arquivo de conhecimento;
- Preservação e manutenção da cultura;
- Disseminação e compartilhamento do conhecimento;
- Recuperação da informação; e
- Educação e interação social;

As Bibliotecas Digitais apenas ampliam estes serviços, aproveitando as potencialidades do armazenamento e comunicação digital para desenvolver serviços mais personalizados e amigáveis, para promover o acesso e a utilização de informação multimídia e reduzir as barreiras de distâncias geográficas e organizacionais, também como o tempo de acesso à informação.

Assim podemos afirmar que a passagem das Bibliotecas Tradicionais para Bibliotecas Digitais parece poder proporcionar um apoio real no aumento da qualidade de educação e no desenvolvimento científico e tecnológico de uma sociedade. Com isso, podemos caracterizar algumas tarefas básicas:

- Conectar os usuários as coleções de informação pessoal, coleções encontradas em bibliotecas convencionais e coleções de dados usadas por cientistas;
- Desenvolver interfaces de informação gerais ou especializadas para cada grupo de usuário;

- Prover acesso a um grande número de fontes de informação e coleções de qualidade;
- Promover um ambiente que permita a experimentação e incorporação de novos serviços e produtos;
- Facilitar a provisão, disseminação e uso da informação por instituições, grupos e indivíduos;
- Armazenar e processar informação em múltiplos formatos, incluindo texto, imagem, áudio, vídeo, 3-D; e
- Intensificar a comunicação e colaboração entre os sistemas de informação para benefício da sociedade em geral.

Para BEZY(1997), a biblioteca digital é claramente o paradigma da sociedade de informação, em que as principais funções de desenvolvimento de um projeto de criação de uma biblioteca digital são:

- Criação e captura – A criação envolve a disponibilização de um documento sob forma digital e a captura, a transformação de um documento do formato não-digital para o digital.
- Gerência e armazenamento – o armazenamento no formato digital envolve sempre grandes objetos em quantidade sempre crescente e que devem ser preservados indefinidamente. Esta função envolve a definição de mecanismos de armazenamento que, por razões de performance, devem prever a distribuição dos objetos.
- Busca e acesso - a indexação de objetos no formato digital normalmente é feita utilizando-se bases de dados separadas para os índices e para os objetos físicos. Estes índices, além de permitirem a pesquisa por elementos tradicionais de identificação dos objetos, tais como autores, títulos, assuntos, *abstracts* e palavras-chaves, devem permitir, também, pesquisa no conteúdo dos objetos, como por exemplo, no texto completo.

- Distribuição – esta função trata do planejamento da infra-estrutura física de comunicação necessária para que as bibliotecas digitais possam prover acesso a todos os seus objetos digitalizados, por qualquer pessoa, a qualquer hora e de qualquer lugar.
- Tratamento de direitos autorais - estabelece mecanismos de proteção dos documentos contidos em bibliotecas digitais, prevendo critérios para acesso integral ou parcial a objetos digitalizados, mecanismos para liberação de cópias, remuneração dos autores, etc. Atualmente, este é um dos temas que envolvem grande discussão.

Conforme observado por RODRIGUES(1996), podemos relacionar três pontos essenciais para projetos de desenvolvimento de Biblioteca Digital:

"Em primeiro lugar, as bibliotecas digitais irão armazenar e dar acesso a volumes cada vez maiores de informação multimídia (texto, imagem, som, vídeo, etc.) em suportes digitais e diversos formatos, a par com a existência de documentos noutros suportes (nomeadamente o papel). Em segundo lugar, as bibliotecas digitais estarão acessíveis aos seus usuários a qualquer hora e de qualquer lugar. E, contrariamente ao que até agora era normal, os usuários remotos das bibliotecas poderão obter não apenas informação secundária e de referência, mas também informação primária (o conteúdo integral de documentos textuais, dados e informação factual, documentos multimídia, etc.). Em terceiro lugar, as bibliotecas digitais permitirão, de forma transparente para os seus usuários, a pesquisa e o acesso às suas coleções locais ou a qualquer outra fonte de informação existente nas redes de comunicação onde estejam integradas. A possibilidade de ligação virtual entre todas as bibliotecas transformará cada uma delas num nó de uma biblioteca digital à escala planetária."

Vale salientar que, para que estas bibliotecas alcancem sucesso é fundamental atender às necessidades dos usuários, de modo que sua implantação deva

estabelecer o local de armazenamento de material eletrônico e implementar as ferramentas para sua respectiva utilização.

## 2.2 Histórico

Consoante PEREIRA(1995), a construção das Bibliotecas Digitais foi acontecendo gradativamente, à medida que a tecnologia foi disponibilizando novas ferramentas que podiam ser utilizadas para cada fim específico. Esta construção iniciou-se com o controle do inventário de circulação, depois com a criação de catálogos eletrônicos e a atividade de indexação de alguns produtos. Mais tarde acrescentaram-se versões eletrônicas de obras de referência. E, finalmente, o armazenamento e recuperação de versões eletrônicas da própria informação, como índices e periódicos, sumários e *abstracts*, além de todo o texto e o acesso às bases de dados *on-line*.

Nos meados dos anos noventa, o governo americano reconhecendo a importância crescente dos sistemas de informação e os avanços tecnológicos das bibliotecas digitais, estabeleceu a criação da “*Commission on Preservation and Access*<sup>1</sup>” e a “*National Archives and Records Administration*<sup>2</sup>”, que propunham ser encarregadas pela política, diretrizes técnicas e administrativas, de todas as Bibliotecas Digitais que iriam surgir, bem como a incorporação de todos os projetos já existente neste país.

Na Europa esta iniciativa era repetida, pois as Bibliotecas Digitais são reconhecidas por possuírem a qualidade de reunir um grande número de instituições privadas e públicas na disponibilização de informações.

Já no Brasil, segundo a Rede Nacional de Pesquisa<sup>3</sup>(RNP), as Bibliotecas Digitais têm-se tornado um ponto fundamental para as pesquisas de profissionais de diversas áreas. O comitê gestor *Internet/Brasil* identifica inúmeras bibliotecas digitais construídas e muitas outras em construção, com o objetivo de viabilizar o uso de um maior número dos recursos tecnológicos digitais.

---

<sup>1</sup> <http://www.preservationcommission.org/>

<sup>2</sup> <http://www.nara.gov/>

<sup>3</sup> [www.rnp.gov/](http://www.rnp.gov/)

Dentre os projetos e experimentos que vêm sendo desenvolvidos e que se utilizam do conceito de Biblioteca Digital, destacam-se:

#### **Biblioteca do Vaticano<sup>4</sup>**

A Biblioteca do Vaticano que foi organizada em meados do século XV, tem uma peculiaridade essencial: o acervo é especializado em obras de cunho secular, como os clássicos gregos e latinos, além de possuir obras teológicas e de direito canônico.

#### **Biblioteca Nacional<sup>5</sup>**

Esta Biblioteca desenvolve um projeto de automação que pretende disponibilizar o seu acervo de música, o maior da América Latina.

#### **Biblioteca Nacional da França<sup>6</sup>**

Este projeto pretende disponibilizar um catálogo geral e informatizado de seu acervo com acesso remoto, possibilidades de reserva de lugar e pedido de material à distância e estações de trabalho multimídia.

#### **Biblioteca Digital de Língua e Literatura<sup>7</sup>**

É um banco de textos eletrônicos da RNP. Contém excertos de obras e de estudos literários sobre autores brasileiros, em português e em outros idiomas.

#### **Livros *On-Line*<sup>8</sup>**

Esta página traz mais de 10.000 de obras disponíveis *On-Line*, cobrindo assuntos diversos.

#### **Projeto Gutenberg<sup>9</sup>**

---

<sup>4</sup> [http://sunsite.unc.edu/expo/vatican.exhibit/exhibit/Main\\_hall.html](http://sunsite.unc.edu/expo/vatican.exhibit/exhibit/Main_hall.html)

<sup>5</sup> <http://info.lncc.br/dimas/cgomes.html>

<sup>6</sup> <http://www.bnf.fr/enluminures/accueil.shtml>

<sup>7</sup> <http://www.cr-sp.rnp.br/literatura/>

<sup>8</sup> [www.cs.cmu.edu/books.html](http://www.cs.cmu.edu/books.html)

Criado em 1971, o projeto Gutenberg é uma biblioteca que pretende disponibilizar até o fim de 2001, quase 10.000 obras que já tenha caído em domínio público.

### **Projeto Perseus<sup>10</sup>**

Esta Biblioteca trata de assunto da cultura grega/romana, disponibilizando textos e imagens de vários autores.

### **Projeto Universalis<sup>11</sup>**

Este projeto tende à unificação das bibliotecas dos países do G-7, priorizando documentos antigos com finalidade de divulgação cultural.

### **Sibi – Sistema Integrado de Bibliotecas da Usp<sup>12</sup>**

O SIBI dispõe de um banco de dados bibliográficos da USP o Dedalus, que contém a produção bibliográfica gerada na Universidade e os registros bibliográficos dos acervos das bibliotecas que podem ser acessados por usuários cadastrados via telnet, e permite a localização de obras.

Assim sendo, com esta nova tecnologia, conceitos como os de lugar e tempo ganham uma nova nuance, pois quando se comparam as características de uma biblioteca tradicional com as de uma digital, vemos que o local onde está armazenado o documento não mais importa, pois atualmente a telecomunicação oferece vários recursos para a comunicação, de modo que passa a ser importante o acesso e a confiabilidade das informações. O tempo torna-se ilimitado, e seu conceito local secundário, em vista da perda das orientações espaciais e temporais.

---

<sup>9</sup> <http://promo.net/pg>

<sup>10</sup> [www.perseus.tufts.edu](http://www.perseus.tufts.edu)

<sup>11</sup> [www.culture.fr/culture/bibliuni/](http://www.culture.fr/culture/bibliuni/)

<sup>12</sup> [www.usp.br/sibi/](http://www.usp.br/sibi/)

### 2.3 Integração: um desafio

Um desafio está no fato de que as bibliotecas e centros de informação não poderão contar com os recursos e serviços que o acesso remoto provê se continuarem dentro dos moldes antigos de desenvolvimento de coleções. De acordo com BORGES(1999), para facilitar o uso e a conexão com outras bibliotecas, certas metodologias e normas devem ser seguidas, dentre as quais:

- Integrar as funções básicas da biblioteca (aquisição, organização, recuperação, e manipulação da informação) com as características do acesso em rede, enfatizando a produção e a disseminação da informação.
- Desenvolver cenários que mostrem como as pessoas podem usar a biblioteca digital;
- Disponibilizar ferramentas para uso no *browser*<sup>13</sup> escolhido;
- Criar novos tipos de recursos de informação;
- Desenvolver novas propostas de aquisição;
- Dispor de novos métodos de armazenagem e preservação; e
- Construir novas formas de classificação e catalogação;

No atual modelo de globalização da informação, os países em desenvolvimento precisam disponibilizar seus acervos em grandes quantidades. No Brasil, todas as instituições envolvidas com a educação e a pesquisa no setor privado e público devem se comprometer a prover grandes bases de dados de textos eletrônicos que sejam acessível com maior rapidez através da rede.

Por último, a tendência de dominação dos países desenvolvidos necessita ser contrabalanceada com a participação mais ativa dos países em desenvolvimento, capacidade que, no caso do Brasil, dependerá da maneira como o país irá desenvolver e

---

<sup>13</sup> *Browsers* - são programas especiais para visualizar páginas WWW. Os *Browsers* lêem e interpretam os documentos escritos em HTML, apresentando as páginas formatadas para os usuários.

introduzir novos produtos e serviços de informação baseados na pesquisa e nas suas habilidades tecnológicas.

#### **2.4 Vantagens/Desvantagens**

As bibliotecas digitais são organizações que fornecem os recursos, incluindo o pessoal especializado, para selecionar, oferecer acesso intelectual, traduzir, distribuir, preservar a integridade, e garantir a permanência das coleções digitais, de tal forma que elas estejam disponíveis para a comunidade. LESK(1997), resume uma série de vantagens, dentre as quais:

- Fácil armazenamento e distribuição de dados digitais;
- Acesso ao material de estudo através de uma interface comum – a *Web*;
- O material disponível pode ser visualizado a qualquer hora do dia;
- Economia de espaço físico;
- Redução dos custos de distribuição, pois pela *Internet* não há custos de impressão e transporte;
- São possíveis diversas técnicas de armazenamento e distribuição, tais como texto, imagens, áudio e outros, entre professores, alunos e comunidade;
- Os materiais tornam-se mais acessíveis a um número maior de usuários;
- Maior nível de interatividade do que em uma biblioteca tradicional;
- Grandes quantidades de dados podem ser facilmente pesquisadas e analisadas;
- Maior funcionalidade: disponibilidade, facilidade de uso e rapidez de acesso;
- Maior liberdade e menos burocracia para os usuários; e
- Preservação dos documentos digitais.

Existem algumas desvantagens e desafios em sua utilização, tais como:

- Torna-se mais difícil de validar e autenticar toda informação;
- É muito fácil distrair-se navegando na *Internet*, perdendo assim o verdadeiro objetivo da pesquisa;
- Os métodos de comunicação utilizados na *Internet* podem ser ainda assustadores para muitas pessoas;
- O acesso à *Internet* ainda não é universal;
- Aspectos legais e éticos envolvidos principalmente na distribuição e confecção de documentos; e
- Não existe um formato para digitalização de documentos.

## 2.5 Armazenamento e multimídia

LESK(1997), coloca que o uso da multimídia na Biblioteca Digital é, desde sua criação, foco principal de desenvolvimento. As Bibliotecas Digitais via *Internet* é uma plataforma onde a distribuição de informações multimídia através de redes de computadores é realizada de forma natural. Além do quê, a criação de recursos multimídia tem-se tornado cada vez mais simples, à medida que novas ferramentas de editoração e conversão são desenvolvidas.

Os recursos multimídia referenciados em documentos na *World Wide Web* (WWW), muitas vezes não podem ser executados pelo próprio navegador. Quando o navegador encontra um formato que não consegue processar, ele procura por uma aplicação específica. Os navegadores mais modernos estão sendo desenvolvidos com diferentes aplicações embutidas (e estão se tornando programas cada vez maiores), entretanto, a maioria dos formatos de mídia ainda pode ser processada com as aplicações auxiliaadoras, chamadas *plugins*<sup>14</sup>.

---

<sup>14</sup> *Plugins* – são na verdade um *Software* que é acoplado a um aplicativo para ampliar suas funções. Dessa forma, os programas vão sendo implementados de acordo com as atualizações que os fabricantes fornecem.

Toda ligação feita para algum tipo específico de multimídia ocorre via *Hypertext Markup Language* (HTML)<sup>15</sup> com comandos específicos; isso ajuda o usuário a identificar o recurso como arquivo de áudio, vídeo, etc.. Informações sobre o formato da mídia e o tamanho aproximado do arquivo também são extremamente úteis para que os usuários se informem sobre a aplicação que devem possuir, ou decidam se estão prontos a esperar o tempo necessário para a transferência. Para MAYER e BABER(1999), existem vários tipos de mídias hoje no mercado, dentre as quais podemos citar: textos, gráficos, imagens, animação, vídeo e áudio.

## 2.6 Distribuição

Segundo GONÇALVES(1997), na tentativa de proteger as mídias, tem-se utilizado vários mecanismos de criptografia<sup>16</sup> para proteger a autenticidade das informações, dentre elas:

- Algoritmo de chave pública: onde o remetente e o receptor possuem um par de chaves, uma privada e outra pública. A mensagem que é criptografada utilizando a chave pública (disponível em um servidor de chaves), só poderá ser decodificada com a chave privada correspondente. Assim, uma chave não pode ser dissociada da outra, garantindo maior segurança nas transações via rede.
- Assinatura digital: Serve para atestar a autenticidade de um endereço de *e-mail*, de mensagem, ou mesmo um indivíduo ou empresa. Baseados em algoritmos de chave pública, as assinaturas digitais devem ser reconhecidas por uma autoridade de certificação, que emite um certificado atestando a validade da assinatura e informando a chave pública associada à chave privada da assinatura em questão.

---

15 HTML - é a linguagem utilizada para a criação de páginas para a WWW. Não é, no entanto, uma linguagem de programação.

16 Criptografia - significa convertê-lo um arquivo num código secreto, para que as informações nele contidas não possam ser utilizadas ou lidas até serem decodificadas.

Além destes, existem muitos outros mecanismos para controlar o acesso do usuário às bibliotecas digitais. No capítulo 5, este assunto será tratado com mais detalhes.

## Conclusão

Para a disseminação de bibliotecas digitais, ainda existe um conjunto de problemas técnicos que necessitam ser resolvidos para que o acesso generalizado a documentos multimídia seja feito de maneira rápida como é a perspectiva de seus usuários, problemas esses que podemos destacar: comunicação e tráfego das informações; questões de armazenamento em um banco de dados; a desigualdade no acesso à informação; propriedade intelectual e direitos autorais.

Para tanto, têm-se dirigido esforços para amenizar estas questões, pois o digital proporciona fantásticas oportunidades para o cidadão: educação, auto- formação, lazer e cultura. Sem a menor sombra de dúvida, a *Internet* é quem está propiciando a criação deste novo paradigma da “era da informação”. Neste contexto, as bibliotecas digitais se firmam como uma solução inovadora e interessante, pois sua arquitetura permitirá dar acesso a um volume cada vez maior de informação multimídia de uma maneira mais rápida com diferentes tipos de formatos e em diferentes bancos de dados.

Na parte que se segue, serão introduzidos os conceitos básicos de multimídia. Isto se justifica na medida em que as bibliotecas digitais e multimídia possuem uma forte relação, já que todas as bibliotecas digitais possuem como principal objetivo armazenar vários tipos de mídias, por exemplo, áudio, vídeo etc...

## Capítulo 3

---

### Multimídia

Para MAYER e BABER(1999), multimídia é qualquer apresentação que envolva duas ou mais mídias, como textos, imagens, gráficos ou som. O que torna a multimídia atraente é a interatividade, sendo que o usuário pode escolher seu próprio caminho ao longo da apresentação. A maioria das páginas inclui imagens gráficas junto com o texto, e muitas até oferecem animação, vídeos e sons. O método de navegação hipertexto<sup>17</sup> permite ao usuário navegar por onde ele desejar, com a mesclagem de multimídia e *Web*, tornando possível a hipermídia, pois com o clique em uma parte de uma imagem pode-se acessar diferentes páginas.

Por estes motivos, a multimídia, sem sombra de dúvida, tem uma profunda relação com bibliotecas digitais, pois sua estrutura permite um debate considerável entre os autores de páginas WWW sobre o papel das multimídias em projetos de Bibliotecas Digitais. Inserir um elemento multimídia pode tornar um documento visualmente atrativo para o leitor.

#### 3.1 Elementos Multimídias

Para melhor entendermos a estruturação desses conceitos Multimídia, precisaremos revisar os seus principais tipos de dados. WEGENER(1997) caracteriza assim esses tipos:

- Texto – Representa uma forma de comunicação assíncrona entre humanos, ou seja, textos estruturados em forma de livros, jornais etc...
- Gráficos – São representações que incorporam desenhos e ilustrações, estes tipos de dados podem ser armazenados de forma estruturada dentro de um banco de dados e o conteúdo que existe na forma de

---

<sup>17</sup> Hipertexto - será visto em sessões posteriores.

metadados<sup>18</sup>, tais como linhas, círculos e arcos podem facilmente ser pesquisados.

- **Imagens** - Incluem exemplos tais como figuras e fotografias. A codificação é definida por padrões de formatos tais como o de *Bitmap*<sup>19</sup>, *Joint Photographic Experts Group (Jpeg)*<sup>20</sup> e *Motion Pictures Experts Group (Mpeg)*<sup>21</sup>. A representação do armazenamento de imagens é basicamente uma tradução da imagem, *pixel*<sup>22</sup> a *pixel*, de tal forma que não há o conceito de uma linha, arco ou círculo. Alguns formatos, então, comprimem a representação para reduzir o tamanho do dado resultante.
- **Áudio** – São mecanismos de conversores analógico/digital, que estimulam uma seqüência de números de entradas através de um processamento digital de sinal, e produz outra seqüência de números de saídas. Os dados de áudio têm uma audição temporal restrita ditada pela taxa de amostragem dos dispositivos de gravação para uma gravação fonográfica ideal ao consumo humano.
- **Animação** - Uma animação é uma seqüência temporal de uma dada imagem ou gráficos. Ela especifica uma ordem na qual os conjuntos de gráficos ou imagens serão executados. As imagens ou gráficos são

---

<sup>18</sup> Metadados - são definidos como dados dos dados.

<sup>19</sup> *Bitmap* - tipo de representação de imagem no qual cada ponto da imagem é associado a um valor. Tradicionalmente, esse valor era um *bit* que podia assumir o valor zero ou um, indicando se o ponto correspondente seria representado em preto ou branco. Atualmente, cada ponto da imagem pode ser associado a até 24 bits, permitindo que uma grande quantidade de cores seja associada a cada ponto.

<sup>20</sup> JPEG - são algoritmo para comprimir imagens. Arquivos deste tipo costumam ser menores que os arquivos tipo GIF e possuem alto grau de fidelidade, permitindo armazenar imagens de até 16.7 milhões de cores. Para desenhos de até 256 cores, é recomendada a utilização do padrão GIF que gera arquivos menores neste caso.

<sup>21</sup> MPEG – é uma maneira de comprimir filmes para diminuir o tamanho dos arquivos e facilitar a reprodução. Um MPEG pode reproduzir filmes usando toda a tela.

<sup>22</sup> *Pixel* - é o menor ponto de luz cuja cor e luminosidade podem ser controladas na tela. As imagens são formadas com a combinação de grande número de *pixels*. O termo é usado para se referir a resolução de uma placa de vídeo ou monitor (ex: 800 x 600 *pixels*).

independentemente construídos e então organizados. A despeito de dados simples de imagens, as quais podem ser recuperadas e visualizadas a qualquer tempo, a animação tem uma visão temporal restrita, requerendo que cada imagem ou gráfico seja apresentado na tela e então subseqüentemente substituída pela próxima imagem/gráfico. A restrição pode variar conforme a animação (duas imagens por segundo ou três imagens por segundo).

- **Vídeo** - Vídeo é um conjunto de dados fotográficos seqüenciados temporalmente. O dado utilizado para representar uma gravação de um evento da vida real produzido por um dispositivo tal como um gravador de vídeo digital, é dividido em unidades denominadas *frame*. Cada *frame* contém uma imagem fotográfica simples. Na maioria dos casos, os dados de vídeo são gravados no padrão 24 até 30 *frames per second* (fps). As restrições para visualização temporal são geralmente ditadas pela taxa de gravação no *frame* que possibilita a otimização da visualização que pode ser percebida pelo ser humano.
- **Tipos Combinados** - Dados multimídia combinados são criados pela combinação das formas de tipos de dados multimídia básicos, podendo se combinar com outros dados multimídia compostos. Os tipos podem ser fisicamente reunidos para formar um novo tipo ou podem ser logicamente mesclados, sendo assim a mesclagem física resulta num novo formato de armazenamento, onde os dados tais como áudio ou vídeo é misturado entre si. Uma mesclagem lógica define um novo tipo de dado o qual retém tipos de dados individuais e formatos de armazenamento. Por exemplo, um novo tipo de áudio ou vídeo poderia ser composto de duas partes distintas, entretanto, ao tocar os métodos de execução poderiam ter de deliberar o dado num modo de sincronização fazendo-o parecer como se o dado fosse uma composição.
- **Apresentações** - São complexos objetos compostos que também descrevem as orquestrações de dados multimídia para o propósito de

modificações de dados e apresentação. As orquestrações podem descrever um ordenamento temporal simples, tal como executar vídeo, e em seguida executar um outro.

### 3.2 Mapeamento de Imagens e Vídeo

Para FILHO(1999), uma imagem imóvel é como um quadro estático, onde o armazenamento de informações das cores relativas a uma imagem inclui, na verdade, a combinação de três cores básicas e triplica as exigências de armazenamento.

Os padrões de imagens estática padrão da *Internet* podem assim ser descritos: *graphics interchange format (GIF)*<sup>23</sup>, um formato de arquitetura de 256 cores que utiliza compactação sem perda; *JPEG*, utiliza-se para imagens mais complexas, trabalhando em até 16.7 milhões de cores.

Já para FURTH(1994), o vídeo inclui uma sucessão de imagens chamadas "estruturas" (*frames*) que provêem a ilusão de movimento. O movimento sem falhas requer uma taxa de exibição mínima de 24 fps e a qualidade da imagem aumenta na mesma proporção que a taxa de fps da exibição (*playback*) também é incrementada. O padrão para um mínimo de qualidade, porém, é de 30 fps, conforme ditam MAYER & BABER (1999).

A compressão de vídeo e armazenamentos padronizados, tais como o MPEG-2, MPEG-4 reduzem essa exigência por memória a pelo menos 40:1, utilizando esse padrão, os dispositivos *Digital Versatile Disk (DVD)* serão capazes de armazenar um filme de cinema típico por completo. Além de oferecer taxas de compressão mais altas do que os padrões anteriores, MPEG-1 e MPEG-2, também produzem melhor qualidade de demanda de vídeo e na transmissão por satélite de subscrição, segundo FILHO(1999).

---

<sup>23</sup> GIF – é um tipo de arquivo para armazenamento de imagens, desenvolvido pela CompuServe em 1987 e amplamente difundido na *Internet*. Esses arquivos são identificados pelo sufixo “.GIF”. Em 1989, foi criado o padrão GIF89a, onde foram adicionadas novas propriedades, como a transparência, o entrelaçamento e a animação. O formato de arquivo GIF é limitado a 256 cores.

Diferente das imagens estáticas, os vídeos contêm dados temporais assim como estruturas *frames* com relacionamentos temporais, tais como "antes de", "depois de", e "durante", têm significado além da questão espacial. Por conseguinte, a compressão MPEG se utiliza tanto da compressão espacial (intraframe) como a temporal (interframe) para eliminação de redundância: *frames* são comprimidos espacialmente (baseado em porções de um determinado *frame* que contém dados idênticos) e "temporalmente" (baseado nos *frames* precedentes que contêm dados idênticos). *Interframe* ou temporal, a compressão é possível mesmo quando porções idênticas de *frames* se localizam em diferentes posições nas futuras estruturas (por causa do movimento do objeto ou da câmera, por exemplo). A compressão MPEG é muito poderosa, consome um tempo significativo e é, normalmente, melhor implementada em nível de *hardware*. A descompressão requer menos poder de processamento e pode ser implementada em nível de *software*, mas se efetuada em nível de *hardware* pode oferecer resultados melhores. Em quaisquer dos casos, se ao nível de *software* ou de *hardware*, não puderem manter o ritmo da descompressão durante a exibição (*playback*), as estruturas (*frames*) serão saltadas para não prejudicar a cronometragem precisa, causando uma imagem tremida, conforme exposto em RUSS(1995).

Para FILHO(1999) e RUSS(1995), o áudio, assim como as músicas, a voz humana e o som, incluem frequência e amplitude. Áudio, assim como vídeo, pode ser comprimido e é usualmente intercalado (comprimido e/ou armazenado em conjunto) com o vídeo com o qual se relaciona. MPEG suporta essa junção; armazena múltiplas trilhas de vídeo e áudio, tais como diferentes idiomas ou ângulos de visão da câmera da mesma película, os quais podem ser selecionados durante a exibição (*playback*). Essas intercalações de formatos de arquivos também são críticas com relação à sincronização dos vários fluxos de dados a ela relacionados.

Outros formatos populares de arquivos voltados para vídeo e áudio incluem o *Audio Video Interleaved (AVI)* da Microsoft para Vídeo e o Cinema Instantâneo (*Movies for QuickTime*) da Apple. O padrão JPEG também pode ser aplicado a vídeo na variante JPEG para imagens em movimento, o qual, significativamente, otimiza o desempenho em aplicações onde o acesso aleatório é necessário (JPEG para imagens estáticas e nas aplicações de acesso aleatório, as

imagens ficam inteiramente independentes uma das outras). O padrão JPEG para Imagens em Movimento possibilita a fácil localização dos dados e a correspondente exibição de uma dada estrutura (*frame*) numa determinada localização. Esta tarefa é mais complicada para o padrão MPEG, por causa da questão da compressão *interframes* e suas redundâncias.

### 3.3 Documentos Hipertextos

Segundo DAVID(1997), um hipertexto pode ser definido simplesmente como uma forma de armazenamento das informações, onde estas ficam contidas em núcleos, que se relacionam devido a características comuns e podem ser alcançadas de forma não linear. Há também a possibilidade de percorrer as informações de forma sequencial, como nos sistemas tradicionais de edição, estes dados são armazenados em uma rede de nós conectados por *links*<sup>24</sup>.

Observe-se que o hipertexto é definido em função do armazenamento de informações e não mais apenas em função do texto, podendo conter em seus núcleos não apenas fragmentos de texto, mas também todo o tipo de objeto multimídia, como por exemplo, vídeo, som, gravuras, animações, etc.

Os usuários decidem quanto tempo irão gastar num nodo e quais visitar em seguida. Os Hipertextos são normalmente implementados utilizando-se do “Modelo de Hipermídia de Dexter” que tem três regras: tempo previsto de execução, regra de armazenamento e regra dos componentes internos. A regra do armazenamento consiste em nodos e vínculos; cada nodo inclui ponteiros para as localizações exatas dos vínculos de âncoras conectadas a seu vínculo. Entre as regras de armazenamento e a regra de tempo previsto de execução encontra-se o mecanismo de apresentação e entre a regra de armazenamento e a regra dos componentes internos encontra-se o mecanismo de vínculos que recupera os componentes de mídia onde:

---

<sup>24</sup> *Links* - é a ligação de um item em um hiperdocumento a outros documentos. Este link pode levar a um texto, uma imagem, som, vídeo, outro hiperdocumento ou mesmo outro protocolo, através do seu endereço na Rede.

- Os nós: contém as unidades de informação compostas por texto e outras informações gráficas. Em geral, um nó representa um conceito ou uma idéia expressa de uma maneira textual ou gráfica.
- Os *links* ou elos: definem as relações lógicas entre os nós, isto é, eles definem relações entre conceitos e idéias. A noção de âncora<sup>25</sup> permite a especificação de uma parte da informação que será fonte ou destino de um *link*. Uma âncora hipertexto é descrita por uma seqüência de caracteres ou o nó inteiro, ou seja, são as conexões entre os nós que representam o caminho a ser seguido pelos usuários durante a navegação.

Em geral, quando o usuário de uma aplicação hipertexto clica sobre uma âncora, o *link* é disparado, causando a apresentação da âncora destino. Dessa forma, o usuário pode navegar no documento através das âncoras e dos *links*. Algumas vezes um nó é definido como uma unidade de navegação. Existem outras formas de acesso às informações além da navegação via *links* hipertextos. Como os textos e imagens são informações estáticas (as informações não evoluem no tempo), a noção de tempo não é utilizada quando da especificação de documentos hipertextos clássicos. No hipertexto, o tempo de apresentação é determinado pelo leitor do documento.

### 3.4 Documentos Multimídia

MAYER e BABER(1999) descrevem que um documento multimídia pode ser visto como uma estrutura descrevendo a coordenação e o estilo de apresentação de uma coleção de componentes constituídos de mídias estáticas e dinâmicas. No documento multimídia o autor define os instantes iniciais e finais da apresentação e as relações temporais; esta relação é chamada de cenário multimídia.

Existem dois métodos de interação a partir do qual o leitor pode controlar a apresentação:

---

<sup>25</sup> Âncora - sinônimo de hiperligações (hiperlinks), a âncora refere-se a ligações (links) não lineares entre documentos.

- Um método de controle de apresentação, que permite o ajuste da referência temporal utilizada pela apresentação.
- Um método de interação similar a um *link* hipertexto rudimentar definido por uma âncora e um *link*. Seguindo o *link*, a apresentação salta para uma outra parte do documento.

### **3.5 Documentos Hiperímídia**

BREMER(1997) define documento hiperímídia como uma combinação de documentos hipertexto e multimímídia. Esta classe de aplicação representa uma evolução natural do hipertexto, na qual as noções ou conceitos de nós hipertexto podem agora ser expressos por diferentes tipos de mídia de apresentação. A inclusão de dados multimímídia aumenta o poder de expressão da informação contida em uma aplicação e torna a apresentação mais atrativa e realista.

### **3.6 Aplicações Multimímídia Interativas**

Para MARQUES(1998), aplicações multimímídia são interativas e primordialmente concebidas para essa finalidade; elas não contém meramente materiais multimímídia complementares. As suas principais funções são: organizar, apresentar e sincronizar multimímídia nos ambientes interativos. Em tais ambientes, sincronismo e organização são críticos.

Nas aplicações multimímídia, uma gigantesca quantidade de material de mídia armazenado precisa ser localizado e acessado de forma eficaz. A organização é um fator importante, não somente com vistas às apresentações, mas também para a autoria e produção que incluem pré-produção e pós-produção. Certamente, pelo fato destas áreas contarem com ferramentas de desenho, editoração, direção de arte, além de profissionais gráficos e musicais.

Conseqüentemente, a mídia deve ser organizada corretamente para apoiar atividades paralelas com o mínimo de confusão.

As apresentações multimímídia requerem a exibição das saídas das informações de mídia, onde quer que, quando e como o autor tenha especificado. Essa

capacidade pode ser especificada de forma não procedural por uma Interface Gráfica ao Usuário (GUI) <sup>26</sup>, ou proceduralmente usando um roteiro *script*<sup>27</sup> para melhor controlar a apresentação.

Operações multimídia precisam ser sincronizadas porque elas acontecem simultaneamente devido aos processos concorrentes e aos fluxos de dados.

Além disso, as aplicações multimídia envolvem a interação dos usuários, os quais frequentemente resultam em intervalos na mídia de duração impossível de se prever. Esta possibilidade faz com que a sincronização e os pontos de checagem dos dados multimídia se caracterizem como da maior importância. De fato, sem a interação dos usuários, não haveria necessidade de qualquer aplicação, somente uma sucessão uniforme de mídia.

### **Conclusão**

Todos os elementos multimídia, até agora estudados, necessitam de grande capacidade de armazenamento, pois seu manuseio envolve grande poder de processamento. Por se tratar de um grande volume de dados, temos que atentar também para a questão dos inúmeros tipos de mídias que o mercado hoje possui e com certeza este trabalho não pode englobar todas, apenas parte delas. Assim sendo, para se ter um total controle destas mídias, temos que utilizar mecanismos de banco de dados, pois, segundo SHERRY(1998), eles possuem inúmeros recursos e permitem um melhor acesso, armazenamento e controle de um dado.

A seguir, veremos como são utilizados os tipos de mídia em banco de dados, buscando levantar uma comparação entre os bancos que os suportam, introduzindo o conceito básico de banco de dados multimídia.

---

<sup>26</sup> GUI - interface que une ícones e funções para realizar tarefas e facilitar a vida do usuário.

<sup>27</sup> *Scripts* - a descrição de uma tarefa complexa ou de uma série de tarefas usando uma determinada linguagem, chamada de linguagem de *scripts*. O *script* permite que o procedimento nele descrito seja executado automaticamente.

## Capítulo 4

---

### Avaliação de sistemas de Bancos de Dados para Multimídia

Estamos vivendo um período de profundas mudanças e inovações no que tange à tecnologia de banco de dados. No final desta década, fornecedores de banco de dados anunciavam uma nova tendência para seus produtos: o suporte a tipos de dados adicionais, como:

- Tipos multimídia (texto, imagem, áudio e vídeo);
- Novos tipos de dados : série temporal ou informação geoespacial; e
- Qualquer outro tipo de dado livremente definido pelo usuário.

Alguns fornecedores apontam arquiteturas personalizadas que refletem os rumos futuros de seus produtos, destacando a promoção de uma nova versão da tecnologia de banco de dados relacional denominada Sistema de Gerenciamento de Bancos de Dados (*SGBDs*) Objeto-Relacionais. Começaremos então este trabalho fazendo uma breve análise desta categoria, em relação ao seu antecessor natural, *SGBDs* Relacionais, para então fazer uma análise com o seu rival *SGBDs* Orientados a Objetos.

#### 4.1 *SGBDs* Objeto-Relacionais

KORTH & SILBERCHATZ(1997) afirmam que os sistemas objeto-relacionais chamados *Universal Servers*, agregam uma série de novas facilidades características dos sistemas orientados a objetos, acrescentando uma camada de abstração de dados acima dos métodos relacionais comuns de representação de dados e de programação. Embora esses recursos não proporcionem as verdadeiras capacidades de programação Orientada a Objetos (OO) para banco de dados, mesmo assim eles ampliam os *SGBDs* Relacionais para o mundo dos objetos de uma forma altamente útil.

Eles incorporam alguns dos requisitos exigidos pelas novas aplicações, tais como, paralelismo de *query*, processamento *On-line Analytical Processing (OLAP)*<sup>28</sup>, *Relational On-line Analytical Processing (ROLAP)*<sup>29</sup>, suporte aos padrões de objetos distribuídos *ActiveX*<sup>30</sup>, *Java*<sup>31</sup> e *Corba*<sup>32</sup>, *Internet*, índices *bit-mapped* e suporte ao *Structured Query Language (SQL3)*, onde as novas funções a serem implementadas nos *SGBDs* dependem exclusivamente das novas necessidades de aplicações que terão que ser suportadas pelos mesmos. Funções mais características de OO serão implementadas gradativamente, enquanto os desenvolvedores se habilitam tecnicamente para utilizar todos os recursos da orientação a objeto.

Algumas outras características:

- Os dados são armazenados em tabelas.
- Estrutura de dados complexos podem ser definidos e utilizados através dos Tipos Abstratos de Dados (ADTs). Isto permite que as operações e funções associadas aos novos tipos de dados possam ser utilizadas para indexar, armazenar e recuperar registros baseados no conteúdo do novo tipo de dado, por exemplo: multimídia.
- Não suporta o encapsulamento<sup>33</sup> de operações. Oferece suporte limitado a alguns dos seguintes aspectos: relacionamentos, identidade,

---

<sup>28</sup> Olap - é uma categoria de aplicações e tecnologias usada para agrupar, gerenciar, processar e apresentar dados multidimensionais com o objetivo de análise e gerenciamento.

<sup>29</sup> Rolap - é um produto relacional OLAP que possibilita a análise multidimensional de dados, agrega e armazena dados em um SGBD. O processamento multidimensional pode ser feito dentro do SGBD, ou na camada de servidor ou no cliente.

<sup>30</sup> *ActiveX* - tecnologia da Microsoft, para inclusão de controles interativos e multimídia em páginas *Web*.

<sup>31</sup> *Java* - linguagem de Programação Orientada a Objetos. Criada pela *SUN*, ela foi desenvolvida para ser multi-plataforma. Diferente de outras linguagens de programação, o *Java*, quando interpretados por *browsers*, não tem permissão de gravar dados no disco rígido, sendo, portanto uma linguagem "segura". isto é, os programas escritos em *Java* podem ser executados sem riscos para as informações contidas no computador

<sup>32</sup> *Corba* - Arquitetura de aplicação distribuída.

<sup>33</sup> Encapsulamento - uma determinada entidade esconde as informações que são necessárias apenas à própria entidade. Da mesma forma, quando em programação, utilizarmos um determinado objeto, este deve ser visto como uma entidade completa, com a qual nos comunicamos através de seus métodos, sem a

herança, polimorfismo, composição, extensibilidade (criação de novas classes persistentes) e integração com linguagens OO hospedeiras, tais como Java.

- Suporta uma versão estendida da SQL (às vezes chamada *ObjectSQL*) que permite a definição, manipulação e recuperação de dados. Entre as extensões, estão: tratamento de consultas que envolvem tipos de dados abstratos, atributos multivalorados.

## 4.2 SGBDs Relacionais

DATE(1990), pontua que os primeiros modelos relacionais foram baseados nos modelos hierárquico e de redes. Entre 1969 e 1970 criava-se um conjunto de teorias chamadas de normalização. Estas teorias hoje servem como base para o desenho de bancos de dados baseados no modelo relacional. Em meados da década de 70, a IBM começava a desenvolver um protótipo de *SGBD* Relacional denominado Sistema R. Paralelamente em Berkeley, na Universidade da Califórnia, estava sendo desenvolvido um protótipo chamado INGRES. O sistema R deu origem a dois produtos o IBM SQL/DS e o IBM(DB2). No início dos anos 80, formou-se a empresa *Relational Technology Inc (RTI)*, com o objetivo de desenvolver e comercializar o INGRES. Hoje, a maioria dos *SGBDs* são baseados na abordagem relacional. Muitos fabricantes de *SGBDs* não relacionais trabalham para sobrepor uma camada relacional aos seus bancos de dados.

Um banco de dados relacional consiste em uma coleção de tabelas, cada uma designada por um nome único. Cada uma destas tabelas é composta de linhas e colunas e suas principais características são as seguintes:

- Armazenamento de dados sob a forma de tabelas.
- Os tipos de dados que podem ser armazenados são em número bastante reduzido.

---

necessidade de conhecermos a sua estrutura interna (atributos e implementação de métodos). Na metodologia de orientação a objetos encapsulamos atributos e métodos dentro das classes.

- Os relacionamentos não são explícitos, mas são determinados com base em valores de campos específicos em chaves estrangeiras.

A visão do banco de dados é expressa sob a forma de tabelas. É a linguagem utilizada para definição, gerenciamento, manipulação e recuperação de dados, feita através de SQL. Ao examinar o resultado de uma consulta ou atualizar uma informação, deve-se valer de um cursor para navegar entre os diversos registros retornados.

### 4.3 *SGBDs Orientado a Objetos*

Ele pode ser definido como um banco de dados capaz de armazenar, além de dados, outros tipos diferentes de dados que não podem ser convertidos somente em arquivos lineares ou bidimensionais como tabelas, e sim um tipo especial de objeto. A principal característica de *SGBDs Orientado a Objetos* é modelar estruturas complexas armazenando não somente a estrutura de dados, mas também seu comportamento.

O desenvolvimento dos *SGBDs Orientado a Objetos* teve origem na combinação de idéias dos modelos de dados tradicionais e de linguagens de programação orientada a objetos, onde a noção de objeto é usada no nível lógico e possui características não encontradas nas linguagens de programação tradicionais, como operadores de manipulação de estruturas, gerenciamento de armazenamento, tratamento de integridade e persistência dos dados.

Os modelos de dados orientados a objetos têm um papel importante nos *SGBDs Orientado a Objetos* porque, em primeiro lugar, são mais adequados para o tratamento de Objetos Complexos<sup>34</sup> como textos, gráficos e imagens, e dinâmicos como programas e simulações. Depois, por possuírem maior naturalidade conceitual e, finalmente, por estarem em consonância com fortes tendências em linguagens de programação e engenharia de *software*, em PRESMMAM(1996).

---

<sup>34</sup> Objetos Complexos - são formados por construtores (conjuntos, listas, tuplas, registros, coleções, arrays) aplicados a objetos simples (inteiros, booleanos, strings). Nos modelos orientados a objetos, os construtores são em geral ortogonais, isto é, qualquer construtor pode ser aplicado a qualquer objeto.

KHOSHAFIAN(1994) coloca algumas vantagens em relação ao *SGBDs Orientado a Objetos* :

- As aplicações *Internet* são particularmente adequadas para banco de dados de objetos, já que as maiorias das aplicações são desenvolvidas em Java, que é uma linguagem orientada a objeto;
- São ideais para as aplicações mais populares;
- Crescimento da *Internet*, videogames, aplicações para multimídia e o desenvolvimento de banco de dados distribuídos que não se prestam ao modelo relacional estão trazendo o foco para *SGBDs Orientado a Objetos* menos complexo do ponto de vista do desenvolvimento e utilização;
- Segue os princípios das atuais linguagens de programação;
- Suporta encapsulamento; e
- Utiliza o paradigma da orientação a objeto e permitem a criação de classes, atributos e métodos para os objetos do sistema.

As desvantagens são:

- *SGBDs Orientado a Objetos* não tem embasamento teórico como no caso dos bancos de dados hierárquicos, rede e relacional, os quais empregam a metodologia de árvore, grafos e matemática dos conjuntos, respectivamente. Porém, nem tudo que não tem embasamento teórico é inútil, como o cálculo numérico que é utilizado até hoje;
- Poucos recursos de ferramentas gráficas para desenvolvimento;
- Instável com relação a direcionamento de suas aplicações, já que tudo se resume em objetos;
- Linguagens para consultas de objetos são difíceis e nem um pouco padronizadas; e
- Pouco explorado (tecnologia nova).

As linguagens padrões para seu manipulamento e aplicação são C++, Smalltalk e Java, por garantir um mapeamento direto entre um objeto de aplicação e um objeto armazenado. Opcionalmente, podemos utilizar uma linguagem para consulta de objetos: a *Object Query Language(OQL)*. A maioria dos *SGBDs* Orientado a Objetos suporta SQL via *Open Database Connectivity (ODBC)*.

Para concluir esta comparação podemos destacar os principais bancos de dados de sua categoria. A tabela que se segue relaciona esses principais bancos.

Vendedor	Produtos		
	<i>SGBDs Relacional</i>	<i>SGBDs Objeto-Relacional</i>	<i>SGBDs Orientado a Objetos</i>
Oracle	Oracle 7.x	Oracle 8.x	
Informix	Dynamic Server	Universal Server	
IBM	DB/2	Universal Database (Extensões DB/2)	
Computer Associates	OpenIngres		Jasmine
Gemstone			Gemstone
O2			O2
Object Design			Object Store

Tabela 1 . Principais Bancos de Dados

Neste ponto é importante salientar que os fornecedores de *SGBDs* Objeto-Relacional são muito numerosos e possuem uma grande fatia no mercado com uma infra-estrutura de marketing sólida e forte. Em contrapartida, os fornecedores de *SGBDs* Orientado a Objeto são em número bastante reduzido. Este fato, por si só, faz crer que durante os próximos anos o mercado Objeto-Relacional deverá ser maior que o mercado Orientação a Objetos, o qual está e continuará a crescer num ritmo respeitável. Os usuários deverão escolher dentre os fornecedores de banco de dados baseados em diversos critérios. Está claro que no mundo de hoje, complexo e sujeito a contínuas mudanças, temos que escolher um banco de dados que nos ofereça a opção mais flexível e extensível para companhias que devem responder com agilidade às demandas cada vez mais complexas de novos tipos de dados impostos pelos sistemas de informação.

#### 4.4 Banco de dados multimídia

Para SUB & WEGENER(1998), um Sistema de Banco de Dados é basicamente um sistema de manutenção dos registros, armazenando-os e visando

mantê-los disponíveis sempre que necessário. Um banco de dados multimídia consiste em um conjunto de componentes, o qual pode ser um banco de dados relacional, orientado a objetos ou um armazenamento de dados multimídia (que pode ser como um *SGBDs Multimídia* ou um servidor de mídia (SM)). A diferença entre ambos é que o *SGBDs Multimídia* possui algumas funcionalidades especiais como: persistência e independência de dados, consistência e tolerância à falhas entre outras.

O *SGBDs Multimídia* é um *software* que se coloca entre o banco de dados físico e as aplicações dos usuários. Assim, todas as solicitações desses usuários são manipuladas pelo banco, o qual desobriga as aplicações dessas funções, com simplificação das mesmas e ganho de desempenho. Em outras palavras, ele faz com que os usuários tenham uma visão do Banco de dados acima do nível de hardware.

Atualmente com a *Internet*, a necessidade inicial de se armazenar diferentes formas de mídias, e a possibilidade de acessar grande quantidade de dados multimídia em lugares remotos vêm crescendo assustadoramente, e juntamente a importância deste segmento em outras aplicações, quais sejam:

- Educação;
- Distribuição de publicidade;
- Projetos de engenharia com suporte computacional (*Computer Aided Design*);
- Diagnósticos Médicos;
- Radiografias;
- Eletrocardiogramas;
- Imagens de ressonância magnética ou *Magnetic Resonance Imaging (MRI)*;
- Bibliotecas Digitais;
- Enfim, de uma forma geral, todas aquelas cuja aplicação esteja ligada à multimídia.

Essas aplicações requerem quantidades expressivas de armazenamento de dados: de fato, o armazenamento desse enorme volume de dados e as exigências de processamento dos mesmos são quase idênticos aos problemas dos dados multimídia propriamente ditos. Com isso, efetuar pesquisas no conteúdo dos dados contidos na mídia e fazer o sincronismo dos mesmos, tornou-se excessivamente difícil, pois se tem a necessidade de unir e mesclar dados, imagens e sons de forma transparente para o usuário, que até então as aplicações multimídia e sistemas de autoria incorporavam a sua própria lógica para tratamento dos dados multimídia nos bancos de dados. Mas, recentemente, os novos dispositivos permitem que se utilize diretamente a funcionalidade dos bancos de dados multimídia, desobrigando as aplicações de inúmeras funções que precisavam ser controladas pela mesma; da mesma forma, as primeiras aplicações começaram a se utilizar das facilidades propiciadas pelos bancos de dados convencionais, há aproximadamente três décadas, segundo DAVID(1997).

Todavia, com a importância de se armazenar dados cada vez mais complexos, houve a necessidade de se criar mecanismos que consigam garantir um perfeito armazenamento e uma boa recuperação de dados, pois o futuro nos proporcionará um aprimoramento de textos, vídeos, gráficos, imagens e, fazendo a unificação entre eles, juntamente com esta questão, podemos ponderar mais algumas, quais sejam:

- Tamanho dos dados multimídia – O tamanho dos formatos é bastante significativo, pois podem ter arquivos com inúmeros *gigabytes* de tamanho dificultando o seu tratamento e armazenamento; assim, devido aos requerimentos da taxa de transferência no envio de dados de vídeos, os dispositivos de armazenamento secundário de dados precisarão ser velozes suficientemente para manusear múltiplas solicitações. Portanto, podem ser requeridos sistemas de discos que operem em paralelo. Ao contrário dos dados tipo numérico ou caracter, os dados multimídia requerem significativa capacidade de armazenamento e aumento da velocidade de processamento.
- Escalonamento - Um usuário pode requisitar múltiplos trechos de áudio e vídeo para serem extraídos ao mesmo tempo do disco. Além

disso, inúmeros usuários podem estar requisitando simultaneamente diferentes dados do mesmo disco. Com esta situação, sentimos a necessidade de criar mecanismos que garantam a perfeita recuperação de dados, escalonando-os sem conflitos.

- Sincronismos - Para alcançar os requisitos para executar dados de vídeo e de áudio, o *software* e o *hardware* devem ser otimizados para assegurar que se possa observar as restrições (limites) de temporalidade. A taxa de rendimento do sistema deve ser observada. Em virtude da multimídia ser uma aplicação baseada no tempo, uma grande dosagem de sincronismo é necessária. Se implementado corretamente, os componentes da mídia serão exibidos em perfeito sincronismo. Áudio e vídeo devem estar em sincronismo e, freqüentemente, múltiplos vídeos e trechos silenciosos são exibidos simultaneamente, os quais também requerem sincronização, conforme DAVID(1997).
- Tipos de Dados diferenciados – Há numerosos formatos padronizados disponíveis para imagens, gráficos, áudio, animação e vídeo. Nos *software* de orientação a objeto esses tipos de dados existirão como definindo uma classe de dados – assim, pode acabar existindo um vasto número de definições de classes para suportar todos esse tipos.

#### **4.4.1 Objetivos a serem alcançados**

Para que um *SGBDs* Multimídia atinja os seus objetivos ele deve atender a certos requisitos, com uma arquitetura e interação entre os seus componentes que propiciem os serviços esperados. Podemos enumerar alguns destes serviços:

- Devem dispor, pelo menos, das facilidades dos Bancos de Dados tradicionais;
- Grande capacidade de armazenamento;
- Capacidade de recuperação das informações;

- Integração de mídias, composição e capacidade de apresentação adequada dos dados;
- Suporte a consultas multimídia;
- Interface multimídia e interatividade;
- Suporte ao Diferentes Tipos de Dados Multimídia;
- Manipulação de Significativos Volumes de Dados;
- Largura de Banda suficiente para atender aos Aspectos Temporais;
- Consulta interpretada, com análise por algoritmos das imagens e áudios;
- Conjunto de *Chips* e *Interfaces* Especiais com funções de captura, apresentação, conversão, compressão e descompressão, além de outras operações multimídia;
- Grande capacidade de armazenamento, como *junkbox* e, mais recentemente, os *DVDs* que prometem substituir os *CDs*; e
- Envio dos dados na Rede compatível com o volume de dados transportado.

Contudo, para atingirmos o total aproveitamento dos vários serviços que este *SGBDs* pode nos proporcionar, devemos centrar nossos esforços para as seguintes situações:

- Os modelos de dados estão centrados em sistemas de banco de dados. Um modelo de dados deve isolar os usuários de detalhes de gerenciamento dos dispositivos de armazenamento. Isso requer o desenvolvimento de modelos de dados apropriados para organizar os vários tipos de dados encontrados nos sistemas de bancos de dados multimídia.
- Os modelos de dados multimídia capturam as propriedades estáticas e dinâmicas dos conteúdos dos bancos de dados e assim fornece uma base formal para o desenvolvimento de ferramentas apropriadas necessárias para manipular dados multimídia. As propriedades

estáticas incluem os objetos que dão forma aos dados multimídia, aos relacionamentos entre os objetos, aos atributos dos objetos, etc. Propriedades dinâmicas incluem interação entre objetos, a operação sobre os mesmos, interações do usuário e outras.

Segundo DAVID(1997), a multimídia engloba muitos tipos de mídia, incluindo-se imagens, vídeo, áudio, gráficos, animação, hipertexto e hipermídia. Também pode incluir dados MRI e outros ADTs. Muitos sistemas de banco de dados padrão SQL suportam o armazenamento de ADTs em grandes objetos em formato binário, ou *binary large Objects (BLOBs)*. Assim sendo, baseados em WEGENER(1997), podemos citar alguns exemplos tradicionais de armazenamento, quais sejam:

- Armazenamento fora do banco - Contendo metadados multimídia tais como os nomes, codificação e decodificação utilizada, descrições e palavras chaves de dados vídeo. Os valores substitutivos armazenados dentro do banco de dados descrevem o nome da rota (diretório e subdiretórios) dos objetos multimídia, os quais estão armazenados na forma de um arquivo simples dentro de um arquivo local, ou um arquivo acessado via rede local. Os *SGBDs* Multimídias possuem controle limitado desse tipo de arquivo, uma vez que esses objetos residem fora do banco.
- Armazenamento dentro do banco - Um banco de dados pode prover acesso restrito aos dados e também uma localização centralizada de quais dados podem ser restaurados. Algumas aplicações podem simplesmente desejar que uma central gerencie o armazenamento, ou mesmo a extração de dados multimídia. Os dados são gerenciados por um *SGBDs* Multimídia e podem ser armazenados em discos locais ou em dispositivos de armazenamento terciários, tais como discos óticos. Aplicações extraem os dados multimídia, os utilizam localmente e então efetuam a sua atualização no banco.

#### 4.4.2 Funções

Todo banco de dados multimídia terá que ter também um potencial para entender os dados gerenciados, da mesma forma como os gerenciadores fizeram anos seguidos com tipos de dados simples. No entanto, devemos salientar a inclusão de sua principal habilidade que deverá ser a de pesquisar o conteúdo de dados multimídia.

SMEULDERS & JAIN(1998), definem as seguintes funções para banco de dados multimídia:

- **Leitura:** recuperação e visualização de dados, apresentações, etc;
- **Atualização:** isso inclui a criação de novos dados multimídia, e a modificação dos dados;
- **Composição:** criação de composições e apresentações utilizando-se de dados multimídia básicos;
- **Pesquisa:** localizando dados em um banco de dados multimídia, quer indiretamente pesquisando o metadado multimídia, ou diretamente buscando o próprio dado;
- **Interação:** isso inclui a interação de um usuário e um *SGBDs* com dados multimídia. Os dados não são mais estáticos, mas dinâmicos pelas ações dos usuários, pela interação de aplicações e sistemas. Além disso, os dados podem interagir e afetar outros dados.

Os Bancos de dados multimídia normalmente incluem linguagem de pesquisa e recuperação dos dados multimídia que foram armazenados e organizados pelo desenvolvedor de aplicação.

#### 4.4.3 Vantagens dos *SGBDs* Multimídia

Segundo GRAY(1998), quando uma aplicação multimídia provoca uma falha no banco de dados, a estrutura de dados é enterrada no *script* onde todo seu valor é perdido. Essa omissão também faz o roteiro mais complicado e menos flexível. Utilizar-se de um banco de dados multimídia faz com que a lógica da estrutura de dados fique disponível a outras aplicações multimídia e simplifica o roteiro de tal forma que muitos deles podem compartilhar o mesmo metadados multimídia. Além disso, quando

um banco de dados multimídia ou de dados abstratos é organizado e declarado para uma aplicação, outras aplicações podem usar essas declarações sem passar pelo mesmo processo degradado de chamada. Essa capacidade acrescenta enorme valor aos dados pela reutilização e controle de redundâncias.

Quando o conteúdo das aplicações multimídia é controlado por um banco de dados multimídia, o conteúdo multimídia pode ser adicionado, apagado ou modificado sem mudar o roteiro da aplicação.

Operações de pesquisa e recuperação são críticas em aplicações multimídia interativas; elas devem ser ao mesmo tempo eficientes e poderosas. Pesquisa e recuperação de dados multimídia e de dados abstratos são temas desafiadores, mas os bancos de dados multimídia tornam isso possível dada a sua flexibilidade de formato de armazenamento e eficiência de operação. Operações de pesquisa e recuperação também dão à aplicação acesso aos componentes mídia de forma que eles possam ser processados dinamicamente e por inteiro, quando necessário.

#### **4.4.4 Critérios para avaliação de um Banco de Dados Multimídia**

Dos vários tipos de critérios encontrados em referências bibliográficas, optamos por trabalhar com o critério quantitativo, onde significa avaliar as características técnicas e funcionais dos produtos. Levantamos um conjunto abrangente de critérios para avaliação dos *SGBDs* que suporte multimídia, visando validar este conjunto. Para tanto, realizamos uma vasta pesquisa bibliográfica que definiu o conjunto de parâmetros em que procuramos levantar os principais aspectos de controle de mídias, como por exemplo: tipo de mídias, suporte, campos aceitos, gerente especializados e outros.

Em função disso, criamos uma sistemática para a avaliação de um banco de dados que suporte as mídias em geral, através de uma seleção criteriosa, em alguns modelos disponíveis no mercado. Vale a pena salientar que toda seleção depende, na grande maioria das vezes, da sua aplicação e de seus requisitos, pois, como já observamos, há inúmeras formas de utilização de um banco de dados: às vezes somente como metadados, metaobjetos ou apenas como repositório de dados, subentende-se que

um banco de dados multimídia tenha que preencher alguns requisitos de funcionalidade, entre elas podemos citar (GRAY[1998]):

- Suporte aos diferentes tipos de dados multimídia;
- Manipulação e armazenamento de grandes quantidades de dados;
- Largura de banda para atender os aspectos temporais;
- Consulta interpretada, com análise de algoritmos para imagens e áudio; e
- Conjunto de funções de captura, apresentação, conversão, compressão e descompressão de mídias.

Esperamos possuir, ao final deste capítulo, critérios objetivos que nos conduzam a uma avaliação de *SGBDs* para o proposto problema, de forma rigorosa, precisa e capaz de adequar a necessidade em se encontrar um banco de dados que garanta as principais funcionalidades acima citadas. Para encontrar a solução desejada, será conduzido um estudo específico, onde serão avaliados alguns bancos de dados objetos relacionais e orientados a objetos, pois os mesmos possuem algumas características necessárias para suportar tipos de dados multimídia, conforme visto anteriormente. Assim serão levantados alguns dos principais bancos de dados do mercado, dentre os quais podemos destacar: Oracle, Db2, Sql-Server, Poet, Object Stores e Jasmine, segue na Tabela 2, o levantamento das principais características desta pesquisa:

	Oracle	Db2	Sql-Server	Jasmine	Object Store	Poet
Suporta BLOBs, som e imagens	Sim	Sim	Sim	Sim	Sim	Sim
Tamanho do arquivo					4Gb	2Gb
Sistema de Arquivo	Sim	Sim	Sim	Sim	Sim	Sim
Possui gerente especializado em multimídia	Não	Não	Não	Não	Não	Não
Modelo temporal	Não	Não	Não	Não	Não	Não
Suporte de linguagem SQL / OO						

	Sim/Sim	Sim/Sim	Sim/Sim	Não/Sim	Não/Sim	Não/Sim
Consulta dados multimídia						
	Sim	Sim	Sim	Sim	Sim	Sim
Acesso direto						
	Sim	Sim	Sim	Sim	Sim	Sim
Tipos de Mídias Suportadas						
	Todas	Todas	Todas	Todas	Todas	Todas

Tabela 2 . Comparação entre os bancos

### Conclusão

As características de um banco podem variar nos atendimentos às funcionalidades mencionadas, mas não fica claro a partir de qual ponto um banco de dados *SGBDs* Objeto-Relacional ou *SGBDs* Orientação a Objeto possa ser considerado como multimídia, pois o mesmo dependerá exclusivamente de sua utilização. A maioria dos bancos pesquisados não estão preparados para suportar grande quantidades de dados de Mídia, pois seria necessário integrar módulos especializados de armazenamento. Esses módulos poderiam proporcionar fluxo necessário para satisfazer os requisitos multimídia.

Por esta razão, nenhum dos bancos pesquisados seriam considerados multimídia por não possuírem um gerente de armazenamento específico para multimídia, pois eles apenas criam alguns mecanismos que na verdade são apenas pseudo-repositórios de dados.

Deve-se escolher o tipo de banco de dados e o produto com base no tipo dos dados e na forma como os usuários finais os acessarão, pois não existe até o presente momento um *SGBDs* compatível para aplicações diversificadas.

No próximo capítulo, veremos como é possível fazer a integração *Web* banco de dados, utilizando as tecnologias existentes no mercado.

## Capítulo 5

---

### Integração via *Internet*

Nos tempos passados, o uso de recursos tecnológicos na sociedade não tinha uma repercussão que apresentasse grande potencial. Com a chamada “Era da informação”, surgiram novos recursos que têm provocado uma verdadeira revolução nas diferentes áreas de nossa sociedade.

CRUMIS (1999), descreve que a comunidade de Banco de Dados demanda novas pesquisas, as quais devem ser direcionadas para o desenvolvimento de tecnologia que ajude a solucionar problemas surgidos nos últimos anos devido à explosão do crescimento da comunicação, incluindo a *Internet* e, em particular, a *Web*. A *Internet* é uma rede global de computadores com cerca de 100 milhões de usuários estimados em 98, e está crescendo assustadoramente, sendo que sua quantidade de informações dobra a cada ano.

#### 5.1 *World Wide Web*

Para MAYER & BABER(1999), a *World Wide Web*, ou simplesmente *Web*, é um recurso de informação globalmente distribuído residindo sob a rede mundial de computadores, a *Internet*<sup>35</sup>, contendo uma grande quantidade de informações relevantes em todas as áreas do conhecimento humano. A *Web*, baseada no conceito de cliente/servidor, com diversos servidores espalhados pelo mundo, possibilita que documentos hipermídia sejam criados e utilizados por um usuário da *Internet*. Além disso, é um sistema dinâmico, independente de uma plataforma específica e inclui recursos multimídia. Para sumarizar o funcionamento deste serviço, pode-se dizer que: um programa "cliente" WWW roda em seu computador; ele apresenta documentos trazidos de outro computador, o servidor; o leitor pode requisitar uma busca, seguir uma

---

<sup>35</sup> [Info.org/guest/zakon/Internet/History/index.html](http://Info.org/guest/zakon/Internet/History/index.html)

ligação por um *link* de hipertexto ou preencher formulários; a resposta do servidor (que com frequência é uma máquina completamente diferente em alguma outra parte do mundo) é um documento que pode ser um texto plano, um hipertexto, ou que incorpore diversas mídias. Veja o exemplo na Figura 1:

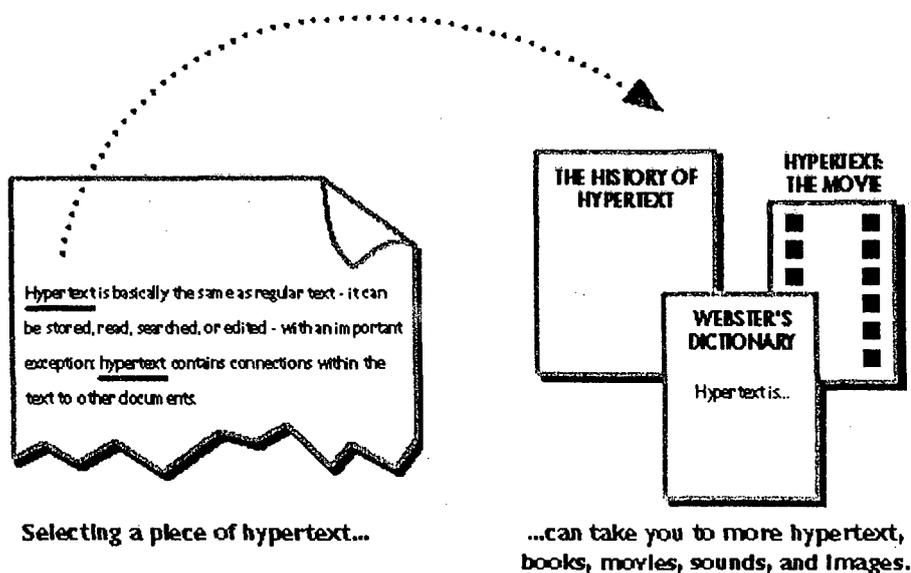


Figura 1. Documento hipertexto

O hipertexto fornece maneiras intuitivas e agradáveis de manejar as informações. Nele existem palavras que são chamadas de *hiperlinks*, onde com um clique do *mouse*, busca a informação associada ao *hiperlink*. A *Web* está atraindo não somente pela facilidade de navegar, mas também pela riqueza gráfica e a integração de texto e imagens.

## 5.2 Origem e Evolução da *Internet*

Segundo MINOLI & SCHIMIDT(1999), a antepassada da *Internet* nasceu em 1969 de um projeto do Ministério da Defesa dos EUA. Chamava-se ARPANET e tinha como objetivo a interligação de computadores utilizados em centros de investigação com fins militares. Após a sua apresentação pública em 1972, e do estabelecimento das primeiras ligações internacionais um ano depois, a ARPANET continuou a crescer lentamente durante os anos 70, mas, por razões de segurança,

continuava a ser uma rede estritamente controlada pelos militares e inacessível a largos setores da comunidade acadêmica internacional e dos Estados Unidos.

Foi no início dos anos 80, mais precisamente em 1983, com a opção dos protocolos *Transmission Control Protocol / Internet Protocol (TCP/IP)*<sup>36</sup> na ARPANET (da qual se separou a componente estritamente militar formando a MILNET), da criação da *Computer Science Network*<sup>37</sup> e da sua ligação à ARPANET, que surgiu a verdadeira *Internet*.

Ao longo dos anos 80, o ritmo de crescimento da *Internet* foi se acelerando, tornando necessária a existência e o funcionamento de estruturas de coordenação e cooperação entre o cada vez maior número de redes e operadores que a integravam. Assim, logo em 1983, foi criado o *Internet Activities Board (IAB)*<sup>38</sup>, dentro do qual se criariam, em 1989, o *Internet Engineering Task Force (IETF)*<sup>39</sup> e o *Internet Research Task Force (IRTF)*. Na década de 80 são ainda de destacar a criação da *European Academic and Research Network (EARN)*<sup>40</sup> em 1983 e da NSFNET (rede acadêmica americana, responsável pela expansão das ligações das universidades à *Internet*) em 1986. Nestes últimos cinco anos merecem destaque a aprovação nos Estados Unidos de medidas tendentes à criação das chamadas auto-estradas da informação em 1991 (*High Performance Computing Act*) e 1993 (*National Information Infrastructure Agenda for Action*), a proliferação de ligações de empresas e organizações governamentais à *Internet*, o desenvolvimento do comércio virtual das emissões de rádio e de diversas outras formas de comunicação interpessoal na *Internet*, e a realização de uma reunião do G7 (Grupo dos sete países mais ricos do mundo), em fevereiro de 1995, sobre o futuro da sociedade da informação.

---

<sup>36</sup> TCP/IP - são protocolos básicos da Internet, usados para viabilizar a transmissão e troca de dados de redes diferentes, permitindo assim que os computadores se comuniquem. Foi criado em 1970 pelo governo americano. Como o TCP/IP foi desenvolvido a partir de fundos públicos, ele não pertence a uma empresa específica e pode ser utilizado por qualquer computador para o compartilhamento de informações com outro computador.

<sup>37</sup> <http://www.netsci.org/>

<sup>38</sup> <http://www.davcsite.com/Webstation/net-history.shtml> ou <http://www.iab.org/>

<sup>39</sup> <http://www.ietf.cnri.reston.va.us/home.html>

<sup>40</sup> <http://www.aarnet.edu.au/>

### 5.3 A *Internet* no Brasil

As primeiras conexões do Brasil com a *Internet* foram feitas em 1988, através da Fundação de Amparo a Pesquisa do Estado de São Paulo e do Laboratório Nacional de Computação Científica (Rio de Janeiro). Em 1989, o Ministério da Ciência e Tecnologia criou a RNP, que construiu o primeiro *backbone*<sup>41</sup> nacional. Em dezembro de 1994, foram iniciados os testes comerciais com linhas discadas. Em agosto de 1995, deu-se o início comercial através de acessos dedicados por provedores de acesso comercial.

### 5.4 Organização da *Internet*

Segundo BREMMER(1997), a *Internet* não é controlada de forma central por nenhuma pessoa ou organização. A organização da *Internet* é feita a partir dos administradores das redes que a compõem e dos próprios usuários. Para isto, existem alguns órgãos internacionais que são responsáveis pela administração mundial da *Internet*.

- *The Internet Society*<sup>42</sup> - é uma organização internacional não-governamental cujo objetivo é a coordenação geral das tecnologias e aplicações da *Internet*. Este órgão é composto por diversos membros (individuais, companhias, agências governamentais e fundações) que participaram desde a criação da *Internet* e suas tecnologias e que tem colaborado no desenvolvimento das novas tecnologias que têm sido desenvolvidas para o uso da *Internet*.
- *IAB* é o órgão que coordena a política da estrutura (arquitetura) do funcionamento da *Internet*, bem como a pesquisa e o desenvolvimento relacionados com o funcionamento da *Internet*. O *IAB* tem como atribuições o seguinte:

---

<sup>41</sup> *Backbone* - Um gerenciador de rede que conecta outras redes à *Internet*. Trata a parte mais intensa do tráfego. É composto por linhas de conexão de alta velocidade, a que se ligam os servidores.

<sup>42</sup> <http://www.isoc.org/>

- A padronização dos protocolos da *Internet*;
- A coordenação das operações da *IETF* e da *IRTF*;
- O desenvolvimento do planejamento estratégico da *Internet*.

No Brasil, foi criado o Comitê Gestor da *Internet* em junho de 1995, composto por membros dos Ministérios da Ciência e Tecnologia, das Comunicações e representantes de Instituições comerciais e acadêmicas.

Com estas noções básicas, é possível compreender um pouco sobre a *Internet*. No entanto, só nos últimos anos é que a *Web* ganhou popularidade, merecendo atenção especial de fabricantes e pesquisadores interessados na expansão de aplicações sob o ambiente distribuído e multi-plataforma proporcionado por esta tecnologia, segundo MINOLI & SCHIMDT(1999).

### 5.5 Arquitetura *Web*

Para BERGHEL(1996), a arquitetura *Web* é composta por duas partes fundamentais:

- De um lado, fica o cliente que é composto por *browsers*, capazes de exibir e solicitar documentos sob a rede, com seus aplicativos externos, para apresentar o documento, caso o *browser* sozinho não seja capaz de interpretar algum tipo de dado (por exemplo, sons ou imagens em movimento).
- Do outro lado, o servidor, composto pelo servidor *Web*, sendo que sua principal função é atender e enviar os pedidos dos clientes *Web*.

Subentende-se então que o servidor *Web* é extremamente útil para os processos, pois ele espera pacificamente os *browsers* solicitarem uma página. Quando o servidor receber a solicitação de informação, esta é enviada para um *browser*. Caso o arquivo não seja localizado, envia-se uma mensagem de erro. Esta arquitetura poderá ser vista na Figura 2.

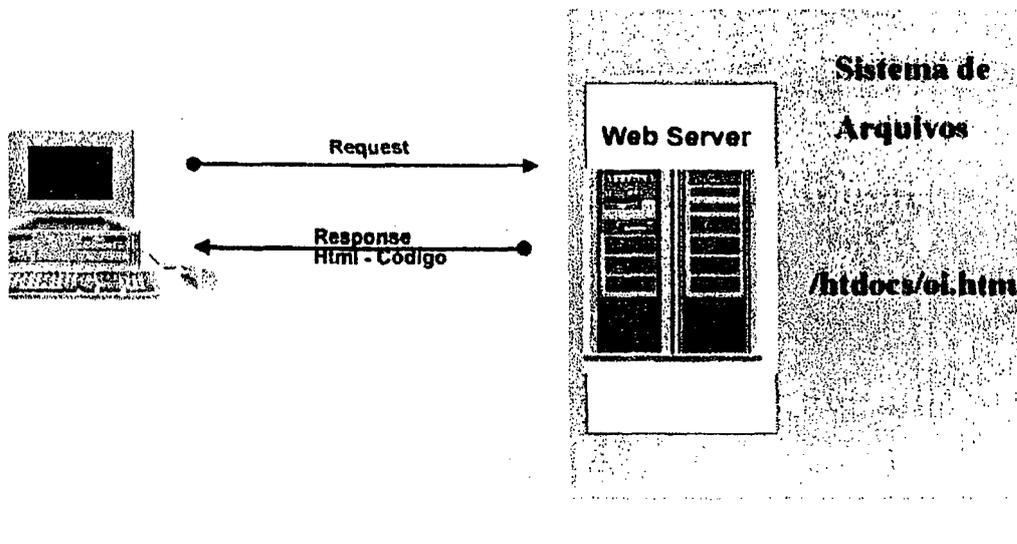


Figura 2. Arquitetura do ambiente Cliente / Servidor.

Esta arquitetura tem como principal característica, o funcionamento em um escopo amplo, que é o caso da *Internet*, composta por uma grande variedade de computadores e redes que interagem entre si. A *Internet* tem um escopo global, e é mantida por canais públicos de comunicação sem qualquer restrição quanto ao conteúdo. Dessa forma, a *Web* incorpora naturalmente a característica de ser um ambiente distribuído e multi-plataforma.

Para MAYER & BABER(1999), com as novas tecnologias atuais que se incorporam na *Internet*, mais especificamente os protocolos, ela se tornou mais amigável e sem dúvida alguma, mais operacional dentre os quais:

- Correio eletrônico: são serviços que permitem a troca de mensagens entre usuários através da *Internet*. São os serviços de maior alcances da *Internet*, pois permitem a troca de mensagens tanto com usuários de outras redes de serviços, como usuários de redes corporativas não totalmente interligados à *Internet*.

- WWW : é um conjunto de documentos espalhados pela *Internet*. Estes documentos têm uma característica em comum: são escritos em hipertexto, utilizando uma linguagem especial, chamada HTML em que os documentos devem ser visualizados através de um tipo de programa chamado *browser*. Através do protocolo *Hypertext Transfer Protocol (HTTP)*<sup>43</sup> fornecem páginas (documentos hipermídia) e, através da *WWW* o usuário tem acesso a uma imensa quantidade de informações, espalhadas por toda a *Internet*, de forma prática e amigável.
- *File Transfer Protocol (FTP)*: é o protocolo<sup>44</sup> utilizado para a transferência de arquivos entre duas máquinas ligadas à *Internet*. É utilizado, entre outras coisas, para atualizar as *Home Pages* hospedadas em máquinas remotas.
- Usenet: é um sistema de discussão via *Internet*.
- Irc: bate papo via *Internet* em tempo real.

Já para YEAGER & MCGRATH(1996), uma outra característica fundamental é que a *Internet* é baseada em três padrões abertos, os quais permitem transferir a informação, descrever sua formatação e localizá-la distribuída pela rede:

- Transferir a informação: toda informação transferida utiliza o protocolo de comunicação HTTP, que é um protocolo aberto e especializado na transferência de documentos.
- Descrever a formatação da informação: a linguagem utilizada para se fazer apresentação de documentos na *Web* é o *HTML*, a qual permite

---

<sup>43</sup> HTTP - é um protocolo padronizado para transferência, através da rede, de arquivos que contém documentos hipermídia. Além de um protocolo para transferência de hipertexto, HTTP é um protocolo para acesso à informação com eficiência necessária para realizar saltos de acordo com a exigência dos hipertextos.

<sup>44</sup> Protocolo – é uma designação formal dos formatos de mensagens e de regras de dois computadores que precisam ser seguidos para que possa haver troca de mensagens, incluindo o controle de fluxo (início-fim), a detecção ou correção de erros e os parâmetros (bits de dados, bits de parada, paridade). O padrão de protocolos que permite computadores de diferentes usuários comunicarem-se, permitindo o intercâmbio de dados entre eles

que os *links* e outros serviços *Internet* sejam incorporados diretamente no formato texto.

- Localizar a informação: para a identificação e localização de documentos *Web* distribuídos pela *Internet* foi proposta a utilização de um formato para endereçamento destes documentos denominado *Uniform Resource Locator (URL)*<sup>45</sup>, onde o protocolo indica o tipo de recurso *Internet* que deve ser usado para a conexão com os servidores.

Toda comunicação entre cliente/servidor é feita através do *request/response*, sempre sendo inicializada pelo cliente. Inicialmente, o cliente envia uma mensagem de sincronização para o servidor pedindo abertura de conexão. O servidor responde aceitando ou não este pedido. Se aceitar, a conexão é estabelecida.

Uma vez estabelecida a conexão, o cliente envia uma mensagem com a requisição feita pelo usuário, representada por uma URL. Esta mensagem é enviada através da rede para o servidor, que busca a informação no disco local ou dispara a execução de uma aplicação externa. Após o processamento do pedido, uma mensagem de resposta é enviada ao cliente que, ao recebê-la, retorna um pedido de fechamento de conexão.

Quando o cliente recebe a informação solicitada, é necessário verificar se os dados são válidos através de informações contidas em campos do cabeçalho da mensagem. Somente após esta verificação, ele pode mostrar o documento para o usuário. Para tanto é preciso saber qual o tipo e o formato do arquivo. Esta informação também é obtida no cabeçalho da mensagem. Todo este processamento impõe um atraso, que é refletido no tempo de resposta observado pelo usuário.

Segundo YEAGER & MCGRATH(1996), podemos generalizar as funções de cada um, ou seja, o cliente interage com outros computadores que atuam como servidores com o propósito de trocar informações. Podemos assim definir as funções básicas de um cliente:

---

<sup>45</sup> URL - é a forma padrão de designação de documentos pela qual é possível acesso a estes através da *Internet*, já que este nome é único.

- É um intermediário;
- Traduz as solicitações;
- Exibe hiperdocumentos;
- Salva, imprime, envia *e-mail* dos hiperdocumentos recuperados;
- Obtém URLs selecionadas;
- Age como cliente de outros protocolos *Gopher*, *FTP*, *News*; e
- Compartilha o processamento.

Já um servidor terá como funções, as seguintes tarefas:

- Continuamente, escuta e atende os pedidos de clientes;
- Fornece hiperdocumentos disponíveis em seu acervo; e
- Executa programas para implementar formulários eletrônicos e imagens clicáveis.

### 5.6 Integrando *Web* e Banco de Dados

MAYER & BABER(1999) descrevem que, atualmente, um dos pontos mais importantes na tecnologia é a integração, isto é, várias aplicações interagem entre si para criar uma estrutura cada vez mais eficiente, como por exemplo, o uso de banco de dados em aplicações *Web*.

Num ambiente *Web* podemos integrar várias aplicações e temos como exemplo, o uso de banco de dados como mecanismos de armazenamento de informações. Neste contexto, a *Web* está passando rapidamente da condição de troca irrestrita de documentos, como projetado inicialmente, para a condição de se tornar uma plataforma de desenvolvimento de inúmeras aplicações baseadas em Banco de Dados. Além disso, vêm crescendo as soluções *Intranet* integradas a Banco de Dados: uma rede corporativa de computadores baseada nos padrões de comunicação da *Internet* pública e da *Web*, cujos dados são armazenados e gerenciados por *SGBDs*. As vantagens desta integração são discutíveis, pois os mecanismos de armazenamento e acesso de dados da *Web* não são muito eficientes, e nem são padronizados. Muitas destas aplicações armazenam dados em arquivos convencionais e, na maioria das vezes, usam linguagens

de propósito gerais como C, C++, Delphi, VisualBasic ou Perl, para armazenamento e acesso aos dados. Quando o volume de informações previsto cresce muito, estas linguagens podem se tornar inadequadas, pois, como se sabe, os *SGBDs* são providos de inúmeros recursos, dentre os quais podemos citar:

- Mecanismos mais eficientes, como estruturas de índices automáticos;
- Execução otimizada de consultas;
- Autorização de acesso;
- Suporte a transações em ambientes multiusuários (como consistência e controle de concorrência);
- Mecanismos de recuperação de dados quando há falhas;
- Facilidade para modelar e organizar tipos de dados complexos e relacionamento entre eles (segundo o modelo relacional, por exemplo); e
- Linguagens específicas de consulta que independem da aplicação (como por exemplo, SQL).

Um outro ponto é que os *SGBDs* possuem uma grande massa de dados sendo gerenciadas pelos seus *softwares* e a *Web* veio abrir a possibilidade de dispor estas informações a um número ilimitado de usuários em redes locais ou remotas. Além disso, ela vem possibilitando o desenvolvimento de aplicações baseadas em *SGBDs* capazes de conectar membros de uma organização sob uma rede heterogênea, local ou remota, além de possibilitar o desenvolvimento e a expansão de novas aplicações baseadas neste sistema. São exemplos dessas aplicações, compras eletrônicas via *Internet* e bibliotecas digitais. Podemos então definir uma arquitetura típica de um ambiente de integração *Web* e Banco de Dados, conforme será mostrado na Figura 3, onde esta possui todos os elementos básicos para esta comunicação.

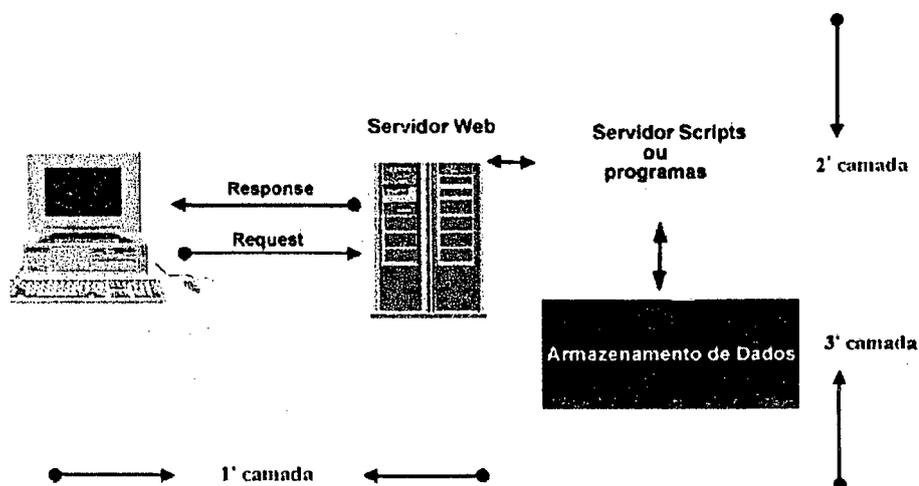


Figura 3. Arquitetura do ambiente de integração Web e Banco de Dados.

O cliente e o servidor *Web* podem estar dentro dos limites de uma *Intranet* sendo somente usados por membros de uma organização ou fazer parte da *Internet* global. Um servidor *script* é o *software* responsável pelo gerenciamento da comunicação e por proporcionar serviços de aplicação entre o servidor *Web* e o servidor de Banco de dados, independentemente da tecnologia aplicada. Ele reside tipicamente ao lado do servidor *Web*, e é composto por um ou mais programas *scripts*. Ao servidor de Banco de Dados cabe a tarefa de gerenciar os dados residindo no Banco de Dados ao passo que àquele cabe a tarefa de receber os pedidos dos clientes *Web* e retornar os dados enviados pelo servidor *script* para serem exibidos pelo cliente *Web* ao usuário final.

Baseando-se na Figura 3 apresentaremos os principais fluxos de dados de uma aplicação no ambiente de integração:

- Inicialmente, o cliente *Web* solicita um pedido ao servidor *Web* via protocolo HTTP, através do *request*;
- O servidor *Web* dispara um processo no servidor *script* enviando os parâmetros recebidos do cliente *Web*;

- O servidor *script* trata os parâmetros recebidos, formula o comando SQL, abre uma conexão com o *SGBDs* e espera pela resposta;
- O *SGBDs* atende a solicitação e retorna os dados ao servidor *script*;
- O servidor *script* trata os dados recebidos e os repassa ao servidor *Web* num formato que o cliente entenda;
- O servidor *Web* retorna os dados enviados pelo servidor *script* ao cliente através do *response*; e
- O cliente identifica o formato dos dados e os exibe ao usuário cliente.

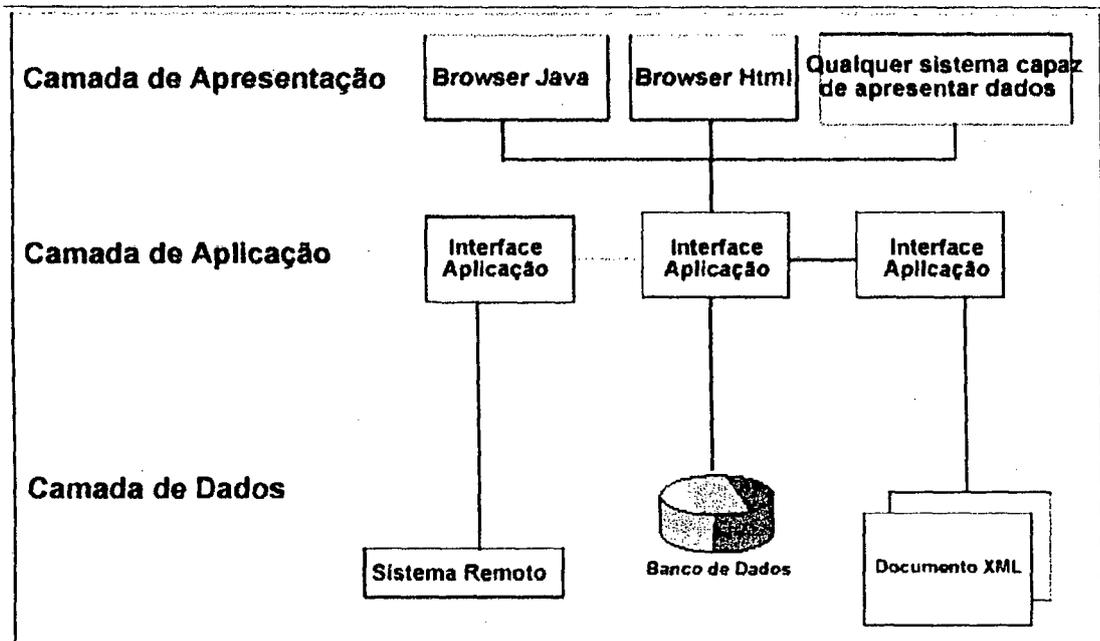
### 5.7 Arquiteturas de Integração

Atualmente, esta tecnologia é uma realidade e encontra-se bastante explorada na *Internet*, através de aplicações em *softwares* educacionais e comerciais, entre outros segmentos. A geração *Web SGBDs* garante várias vantagens por ser um ambiente aberto e multiplataforma, além de facilitar o gerenciamento e a otimização para o acesso ao Banco.

Os servidores de integração *Web* e Banco de Dados se referem à extensa classe de *softwares* que ocupam uma posição intermediária na arquitetura de integração dessas tecnologias, sendo que sua principal função é executar, junto ao *SGBDs*, o pedido que lhe foi solicitado e devolver ao servidor *Web* o resultado do pedido em um formato especificado, como por exemplo, em formato HTML.

Estes *softwares* podem ser classificados em categorias segundo a arquitetura de integração que implementam. Esta classificação pode ser baseada em vários critérios. Em GRAY(1997), por exemplo, o critério utilizado baseia-se na arquitetura cliente/servidor da aplicação que acessa o Banco de Dados: a arquitetura de duas ou de três camadas, conforme Figura 4.

**Estrutura n-camadas**



**Figura 4 .** Arquitetura de três camadas.

Os *softwares* de integração *Web* e Banco de Dados são primeiramente enquadrados em três categorias, conforme a Figura 5.

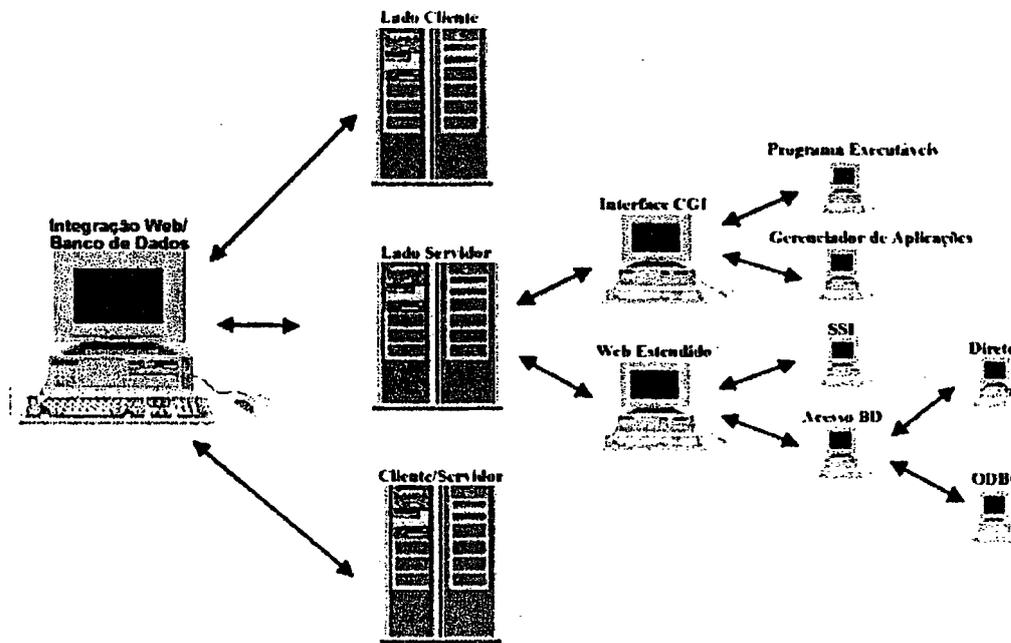


Figura 5. Arquitetura de integração Web / Banco de dados.

### 5.7.1 Arquiteturas de Servidores Web

A arquitetura de integração Web Banco de Dados localizada no lado servidor Web tem como principal característica o fato de ser semelhante à arquitetura Banco de Dados cliente/servidor em três camadas. As três camadas são:

- Cliente Web: onde reside a lógica da apresentação, que controla a interação entre o usuário e o computador. Nesta camada, pode-se ter alguma lógica da aplicação, tal como a validação de dados implementados por ferramentas proprietárias como *Java*, *JavaScript*<sup>46</sup> ou *VBScript*.

<sup>46</sup> *JavaScript* - Linguagem de Script desenvolvida pela Netscape que complementa a linguagem HTML e precisa de um Browser que a suporte (Explorer 3.0 ou Navigator 2.0). É uma linguagem interpretada (o código-fonte sempre é traduzido para uma linguagem de máquina na hora em que for executado) baseada em objetos, ou seja, ela trata todos os elementos de uma página Web como um objeto. No mesmo arquivo

- Servidor *Web*/Aplicação cliente do Banco de Dados: esta camada é composta pelo servidor *Web* e pela aplicação propriamente dita que acessa o Banco de Dados. Ela contém a lógica da aplicação, responsável pelas decisões, cálculos e operações que a aplicação deve realizar, mecanismos de segurança para acesso à aplicação e controle de acesso multiusuário à aplicação. O servidor *Web* pode também agir como um cliente do servidor de Banco de Dados, como será visto em algumas arquiteturas desta categoria.
- Servidor de Banco de Dados: esta camada consiste no servidor de Banco de Dados, tipicamente rodado em máquinas de alto desempenho. Ela contém todos os dados, metadados e regras para integridade referencial dos dados definidos pelo desenvolvedor da aplicação. Esta camada proporciona mecanismos adicionais de segurança para acesso ao Banco de Dados próprios de cada *SGBDs* proprietário.

É importante observar que todo o código da aplicação reside num único local, de forma que toda a manutenção no código da aplicação se reflete imediatamente a todos os clientes *Web*, de modo que podemos enumerar algumas funções básicas desta tecnologia: ela é ágil para ser invocada pelo servidor, tem que receber dados pelo formulário e deve retornar o serviço completado para o servidor *Web*.

#### 5.7.1.1 Common Gateway Interface ( CGI )

CGI é um conjunto de especificações para passagem de informações entre um navegador, um servidor *Web*, e um aplicativo CGI. Um navegador de cliente da *Web* pode iniciar um aplicativo CGI, preenchendo um formulário HTML ou clicando em um vínculo, em uma página HTML em seu servidor *Web*. O aplicativo CGI pode levar as informações que o navegador do cliente *Web* fornece e fazer quase tudo que foi

---

“.HTM” ou “.HTML” em que estão os comandos básicos da linguagem HTML, o código JavaScript é inserido de maneira a ser interpretado quando necessário.

programado; logo após, retorna os resultados do aplicativo em uma página HTML, ou põe as informações em um banco de dados.

O CGI possibilita a execução de aplicações externas ao servidor *Web*. Assim, a interface CGI é a maneira mais natural para o desenvolvimento de aplicações *Web* Banco de Dados, já que são aplicações externas ao ambiente cliente/servidor *Web*. Apesar da implementação da interface CGI diferir em alguns servidores *Web*, estas arquiteturas são consideradas as mais portáteis que existem atualmente.

#### Programas Executáveis

Esta arquitetura é composta por um ou mais programas CGI, geralmente implementados usando-se alguma linguagem de programação hospedeira do *SGBDs*. De fato, para se usar a interface CGI implementada pelos servidores *Web* é necessário uma linguagem que possa ler a entrada padrão, obter as variáveis de ambientes e escrever na saída padrão. Como a grande maioria dos *SGBDs* existentes incorporam alguma linguagem de programação com estas características, praticamente todos eles podem se comunicar com a *Web* por meio de Programas Executáveis CGI sem grandes custos adicionais. A Figura 6 exemplifica esta arquitetura.

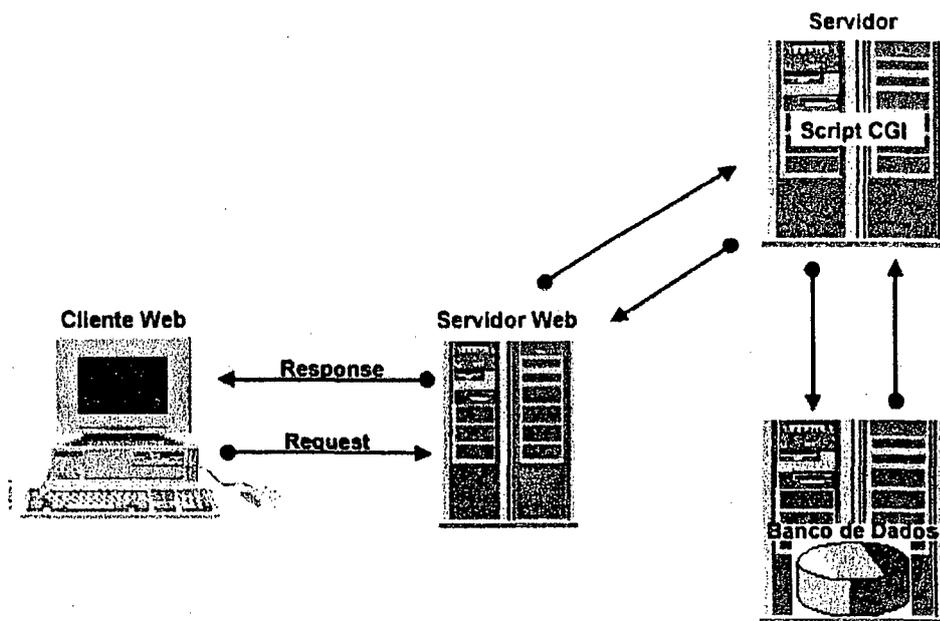


Figura 6 . Arquitetura de Servidores *Web* com Interface em Cgi.

O funcionamento da arquitetura é mostrado a seguir:

- O cliente *Web* solicita ao servidor *Web* a execução de uma aplicação externa, que neste caso é um programa executável na plataforma deste;
- O servidor *Web* dispara um processo para execução do programa através da interface padrão CGI, enviando os dados recebidos do cliente *Web* de acordo com a implementação do servidor *Web* para a interface CGI;
- O programa recebe os dados passados pelo servidor *Web* via interface CGI, decodifica-os, formula o comando SQL e abre uma conexão com o Banco de Dados para sua execução;
- O *SGBDs* retorna os dados e a conexão é finalizada. O programa executável trata os dados recebidos e os repassa ao servidor *Web* via interface CGI, num formato que o cliente *Web* entenda, como por

exemplo, no formato HTML. O processo iniciado pelo servidor *Web* para execução do programa é finalizado neste momento; e

- O servidor *Web* retorna os dados repassados pelo programa ao cliente *Web*.

#### 5.7.1.2 Servidor *Web* estendido

As arquiteturas que se encontram nesta categoria não usam a interface padrão CGI para execução das aplicações *Web* Banco de Dados. Ao contrário, as funcionalidades do servidor *Web* são estendidas de forma a proporcionar acesso mais otimizado e mais rápido a aplicações externas. Ganha-se, portanto, em desempenho quando comparado às arquiteturas que usam a interface CGI, mas perde-se em portabilidade da aplicação, ficando o desenvolvedor limitado às características do servidor *Web* proprietário.

##### *Server Side Include (SSI)*

Segundo BUYENS(2000), SSI é um pedaço de código que está incluído em uma página HTML e é interpretado pelo servidor depois que a página é enviada ao *browser* do cliente. O SSI implementado por alguns servidores *Web* permite a inserção de códigos (ou *tags*) especiais em documentos HTML que são interpretados e executados automaticamente pelo servidor *Web* quando solicitado pelo cliente *Web*. Alguns desses códigos podem incluir funcionalidades para execução de comandos em um *SGBDs*. Entretanto, há um custo de tempo envolvido na performance do servidor. Se todos os arquivos precisarem ser separados e processados, a transação ficará mais lenta. Assim, o SSI deve ser usado com bastante critério.

Qualquer arquivo com extensão ".shtml" será automaticamente separado e processado pelo servidor. Você pode usar um arquivo chamado "index.shtml" ao invés do "index.html" como seu arquivo inicial. A Figura 7 abaixo define esta estrutura.

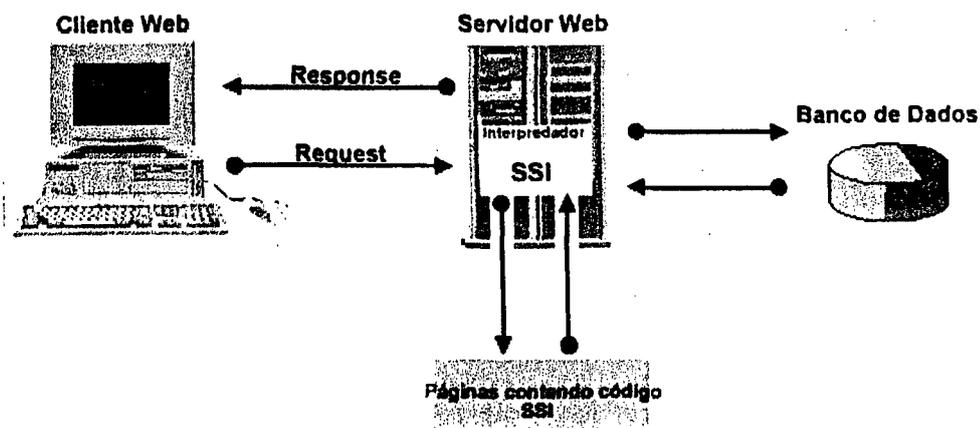


Figura 7. Arquitetura de servidor *Web* estendido / SSI.

O funcionamento da arquitetura é o seguinte:

- O cliente *Web* solicita uma página que possui, além dos elementos da linguagem HTML, códigos SSI para conexão e execução de comandos em um *SGBDs*;
- O servidor *Web* recupera a página e reconhece que a mesma contém códigos SSI para serem executados dinamicamente. Neste momento o servidor *Web* ativa o interpretador SSI, que é um de seus componentes;
- O servidor *Web* se conecta ao *SGBDs* apropriado e solicita a consulta. Os dados são retornados ao sistema e o servidor os formata de acordo com outros códigos SSI de formatação presentes no restante da página HTML; e

- O servidor *Web* encerra a conexão com o Banco de Dados e retorna a página de dados ao cliente *Web*.

#### Acesso Direto ao Banco de Dados

O servidor *Web* pode ser implementado para, além de atender pedidos HTTP, atender pedidos que exijam acesso ao Banco de Dados sem nenhuma interferência da interface CGI ou dos mecanismos API ou SSI. Ele incorpora o protocolo de comunicação do *SGBDs* e adiciona mecanismos próprios para o desenvolvimento de aplicações *Web* Banco de Dados totalmente integrados ao *SGBDs*. Ou seja, o servidor *Web* funciona como um cliente do *SGBDs* que suporta o protocolo HTTP. Esta forma de conexão origina a arquitetura denominada Acesso Direto ao Banco de Dados. Veja a Figura 8.

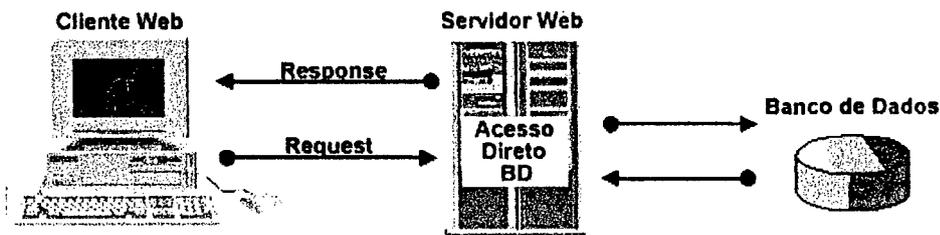


Figura 8. Arquitetura servidor *Web* estendido / Acesso Direto ao Banco.

O funcionamento da arquitetura é mostrado a seguir:

- O cliente *Web* solicita ao servidor *Web* a execução de uma aplicação incorporada ao próprio servidor *Web*;

- Este faz uma série de verificações internas (*login e password*) e inicia uma conexão com o Banco de Dados como se fosse mais um cliente do *SGBDs*;
- Este atende o pedido e retorna os dados ao servidor *Web*; e
- O servidor *Web* formata os dados de acordo com a aplicação desenvolvida no servidor *Web* e retorna-os ao cliente *Web*. Normalmente o servidor *Web* fica esperando por novos pedidos do cliente e não libera os recursos da sessão do usuário no *SGBDs* enquanto o cliente *Web* estiver utilizando a aplicação.

#### *Open Database Connectivity (ODBC)*

Para WOOD(1999), o ODBC oferece um padrão de programação de acesso á dados permitindo aos aplicativos acessarem uma variedade de banco de dados utilizando SQL. Qualquer aplicação que suporte ODBC permite acesso aos dados que estão no banco de dados de um ambiente *Windows*; com esta configuração, podemos, entre outras coisas:

- Criar páginas da *Web* com informações contidas em um banco de dados;
- Inserir, atualizar e excluir informações em um banco de dados baseado na entrada do usuário a partir de uma página da *Web*; e
- Executar outros comandos da linguagem de consulta estruturada (SQL).

De uma maneira mais ampla, podemos entender esta arquitetura conforme a Figura 9.

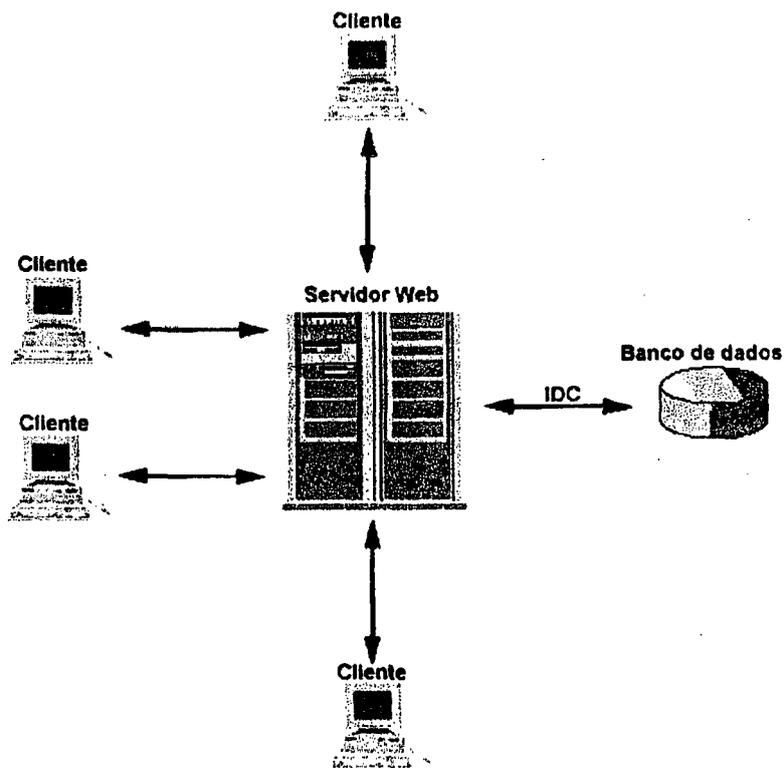
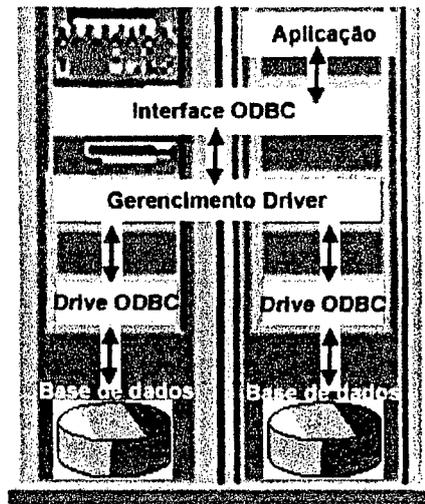


Figura 9 . Arquitetura de servidor *Web* estendido / via ODBC.

Navegadores da *Web* (tais como, *Internet Explorer* ou *Netscape*) submetem os pedidos ao servidor *Internet* utilizando o HTTP. O servidor da *Internet* responde com um documento formatado em HTML. O acesso aos bancos de dados é concluído através de um componente do *Internet Information Server (IIS)* , que e um servidor de páginas *Web*.

Em REED(1996), o ODBC utiliza a arquitetura em camadas para implementar um conjunto padrão de rotinas de acesso a dados que trabalham com uma variedade de banco de dados. A Figura abaixo, mostra as 4 camadas do ODBC: Aplicação, ODBC *driver manager*, *driver* ODBC e *SGBDs* relacional.

A Figura 10 mostra os componentes associados à arquitetura ODBC.



**Figura 10 .** Componentes de Funcionamento da arquitetura ODBC.

O funcionamento desta arquitetura será da seguinte maneira:

- **Aplicação:** executa o processamento e chamadas das funções ODBC para executar comandos SQL;
- **Gerenciamento de *Driver*:** carrega os *drivers* para uma aplicação, e é implementado através da ODBC32.DLL (ODBC.DLL se *Windows*). O *driver manager* permite que múltiplos *drivers* estejam ativos simultaneamente utilizando o nome do *Data Source* fornecido pela aplicação;
- ***Driver* ODBC:** é uma biblioteca de funções (em *Windows* Dynamic Link Library – DLL) fornecida pelo *SGBDs* ou outro fornecedor de *software* para trabalhar com um específico *SGBDs*. Quando uma aplicação faz uma chamada à uma função ODBC para requisitar uma

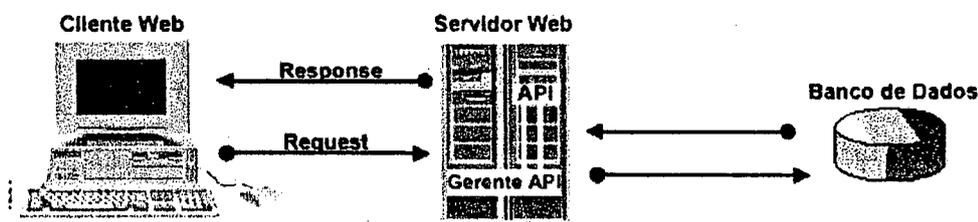
operação ao banco de dados, o *driver* ODBC traduz sua chamada para um formato apropriado *SGBDs* desejado; e

- *Data Source*: contém a definição de um *Driver* ODBC utilizado pela Aplicação para acessar um específico *SGBDs*.

### 5.7.1.3 Application Programming Interface (API)

Trata-se de um conjunto de funções que são incorporadas ao servidor *Web* podendo substituir completamente a interface CGI para a execução de programas *script*, conforme seu funcionamento na Figura 11. As APIs, na realidade, podem realizar muito mais, permitindo, entre outras coisas, melhorar significativamente o desempenho de aplicações externas *Web*. Dentre outras funcionalidades, podemos destacar:

- As API para programação são as mesmas para qualquer *SGBDs*;
- Os programas escritos como APIs de servidores *Web* ficam residentes em memória uma vez iniciados e, portanto, executam muito mais rápido do que quando a interface CGI é usada;
- As APIs permitem o compartilhamento de dados e recursos de comunicação com o servidor *Web*, ao contrário de programas CGI. Aplicações APIs executam no mesmo espaço de memória em que o servidor *Web* está executando e, em algumas implementações, pode-se ter acesso inclusive às estruturas de dados internas do servidor *Web*;
- A aplicação roda em qualquer sistema operacional acessando qualquer Banco de Dados; e
- Permite a construção de páginas WWW que acessa Bancos de Dados, pois dispensa a configuração da máquina cliente.



**Figura 11 .** Funcionamento da arquitetura APIs.

O funcionamento da arquitetura é mostrado a seguir.

- O cliente *Web* solicita ao servidor *Web* a execução de uma aplicação API, que pode ou não já estar executando no mesmo espaço de memória em que o servidor *Web* está executando;
- Este inicia a execução da aplicação API se ela ainda não estiver executando e repassa os dados enviados pelo cliente *Web* de acordo com a implementação API do servidor *Web*, ou seja, não se usa a interface CGI;
- A aplicação API inicia a conexão com o Banco de Dados ou, dependendo da implementação da aplicação, pode-se substituir um CGI (como na arquitetura anterior) por um API que envia um pedido para um Gerenciador de Aplicação;

- A aplicação API realiza a conexão com o Banco de Dados e, em seguida, repassa os dados ao servidor *Web* por canais próprios da implementação API do servidor *Web*; e
- O servidor *Web* retorna os dados ao cliente *Web*. A aplicação API normalmente fica residente em memória esperando por novo pedidos até que o servidor *Web* decida quando ela pode ser finalizada para liberar recursos do sistema.

Geralmente, nesta arquitetura, o servidor *Web* oferece bibliotecas *Dynamic Link Librarys (DLLs)* específicas para conexão com o Banco de Dados, sendo que a conexão com o Banco de Dados pode ou não ser finalizada ao final de um pedido do cliente *Web*. Isto vai depender das funcionalidades presentes na API proporcionada pelo servidor *Web*.

Muitos produtos comerciais que usam uma das duas arquiteturas descritas anteriormente possuem uma versão de forma a suportar uma ou mais APIs de servidores *Web* proprietários, como por exemplo, *ColdFusion*, que já suportam as APIs NSAPI da *Netscape Corporation* e ISAPI da *Microsoft Corporation*.

#### ISAPI e NSAPI

Estes são padrões de APIs, a da Microsoft foi denominado como *Internet Server Application Programming Interface (ISAPI)*, ao passo que o padrão da *Netscape* foi chamado de *Netscape Application Programming Interface (NSAPI)*, veja o exemplo de seu funcionamento conforme a Figura 12.

Estes padrões nada mais são que DLLs. Uma das principais diferenças entre uma DLL e um EXE é que se uma DLL for usada por cinco aplicações diferentes, ela será carregada na memória apenas uma vez, e ficará servindo simultaneamente às cinco aplicações. Isto representa uma grande economia de recursos do computador. E, com base nisto, os padrões ISAPI e NSAPI são, na verdade, a inclusão da capacidade de executar DLLs, respectivamente nos servidores *Web* da *Microsoft* e da *Netscape*.

Utilizando-se anteriormente os *scripts CGI*, percebe-se que os aplicativos ISAPI têm um desempenho muito melhor porque são carregados na memória em tempo

de execução do servidor. Eles pedem menos *overhead* porque cada pedido não inicia um processo separado, ou seja, a lógica de funcionamento é idêntica ao do padrão CGI, sendo a única diferença o fato de que a DLL é carregada uma única vez no servidor, não importando quantos clientes *Web* a estejam usando em um dado momento.

Estas DLLs precisam receber parâmetros, processar a informação (incluindo acesso a bancos de dados no servidor) e retornar uma página HTML para o cliente *Web*. O acesso a estas duas APIs é providenciado via linguagem de programação.

Mas trabalhar com ISAPI ou NSAPI em servidores *Web* ainda não é satisfatório. O problema é que uma DLL quando carregada no servidor *Web* não pode ser descarregada da memória enquanto estiver rodando. Desta forma, não é possível eliminar a DLL do servidor ou substituí-la por uma nova versão (o que acontece muito durante a fase de desenvolvimento) sem antes derrubar o servidor *Web*.

Um outro problema é que se houver um erro na DLL rodando no servidor *Web*, isto pode até causar a paralisação deste servidor. Um erro muito comum que pode ocasionar isto é fazer *loop* em uma tabela.

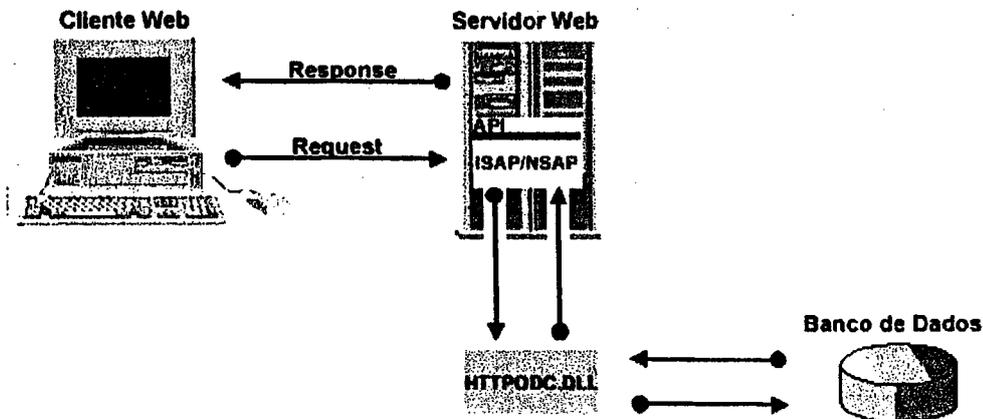
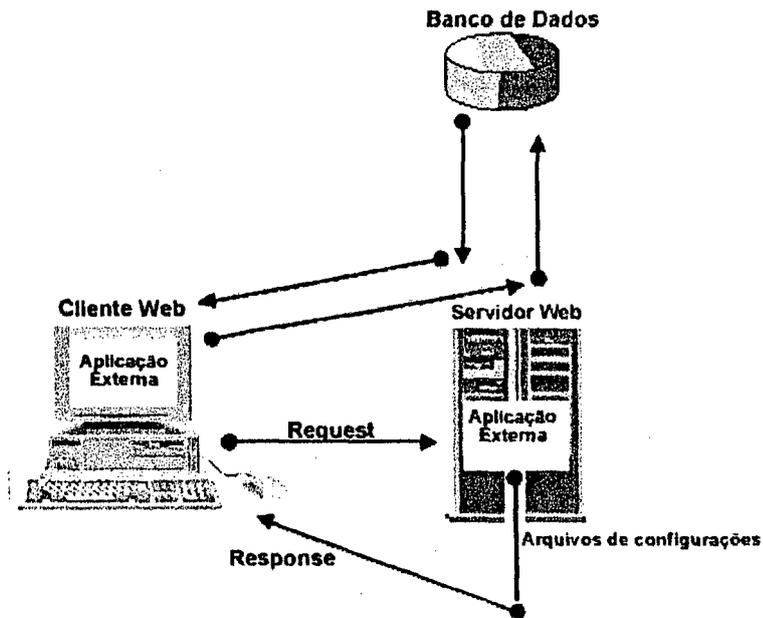


Figura 12 . Funcionamento da arquitetura ISAPI / NSAPI.

### 5.7.2 Arquiteturas de cliente *Web*

Esta arquitetura tem como característica o fato do aplicativo acessar o Banco de Dados usando recursos do lado cliente *Web*. Embora o código resida no servidor *Web*, todo o processamento da lógica da aplicação é realizado na máquina do cliente, deixando assim de sobrecarregar o servidor com o processamento. A Figura 13 exemplifica esta arquitetura.



**Figura 13 .** Arquitetura de Cliente *Web* / Aplicação externa.

O funcionamento da arquitetura de uma Aplicação Externa é mostrado a seguir:

- O cliente solicita um pedido ao servidor *Web*;
- Este atende ao pedido enviando instruções para aquele executar uma Aplicação Externa. Geralmente são enviados também arquivos de configuração da aplicação com uma extensão que a Aplicação Externa consiga interpretar;
- O cliente *Web* recebe o pedido e imediatamente inicia a execução da Aplicação Externa para acesso ao Banco de Dados;
- A Aplicação Externa assume o controle da aplicação, se conecta ao Banco de Dados, solicita um pedido e recebe os dados enviados pelo *SGBDs*; e

- Os dados são exibidos no cliente *Web* pela Aplicação Externa. Esta é finalizada quando o usuário seleciona uma opção apropriada no cliente *Web*.

Esta arquitetura de integração pode ser implementada configurando-se programas de Banco de Dados já existentes para serem executados no cliente *Web*. Cabe a este solicitar a execução dos programas de aplicação, conforme Figura 14.

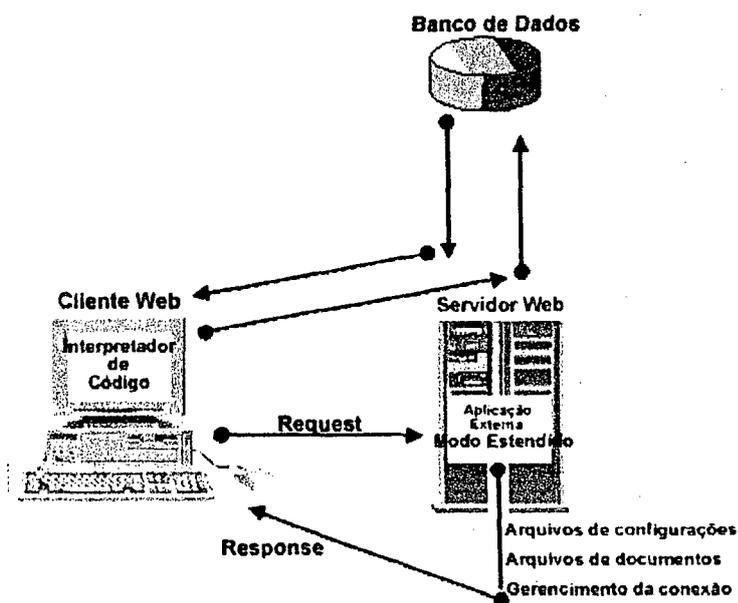


Figura 14 . Arquitetura de cliente *Web* estendido / Modo estendido.

Já o funcionamento da arquitetura de modo estendido é mostrado a seguir:

- O cliente *Web* solicita um pedido de um documento que contém ou faz referência ao código da aplicação para acesso ao Banco de Dados;
- O servidor *Web* envia o documento e o código da aplicação. Juntamente com o código da aplicação pode ser enviado o código para gerenciamento da conexão com o Banco de Dados, como por exemplo, um *driver* gerente de conexão semelhante a um *driver* ODBC;

- O cliente *Web* recebe o documento e o código da aplicação e imediatamente reconhece que deve iniciar a execução do código da aplicação com o auxílio de um interpretador que é um componente do *browser*;
- O cliente *Web* se conecta ao Banco de Dados usando as funcionalidades de acesso ao Banco de Dados enviada junto com o código da aplicação ou incorporadas no cliente, executando o pedido do usuário; e
- O *SGBDs* envia o resultado ao cliente *Web* e este, de acordo com o código da aplicação, exibe o resultado da operação ao usuário. A conexão com o Banco de Dados é, geralmente, finalizada após cada pedido ter sido atendido.

### 5.7.3 Arquiteturas de cliente e servidor *Web*

As arquiteturas de integração *Web* e Banco de Dados que se enquadram nesta categoria têm como principal característica o fato de que parte da aplicação que acessa o Banco de Dados executa no lado cliente *Web* e parte no lado servidor *Web*, conforme a Figura 15, pois o código da aplicação reside neste juntamente com a maior parte da lógica da aplicação. No entanto, pode-se ter uma parte da lógica da aplicação executando no lado cliente como complemento da aplicação que roda no servidor e, diferentemente das arquiteturas anteriores, os dois lados podem trocar mensagens entre si, sem a intervenção direta do servidor *Web*.

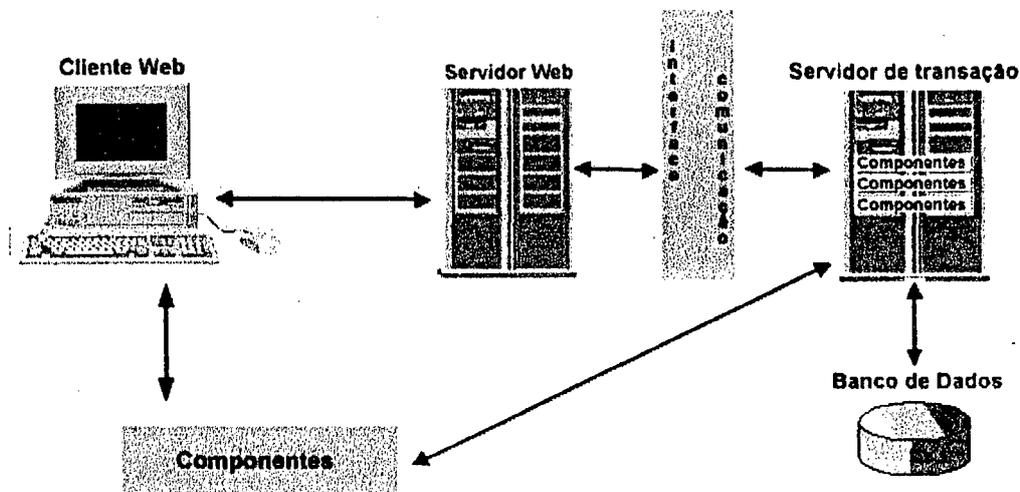


Figura 15 . Arquitetura de Clientes e servidor *Web*.

O funcionamento desta arquitetura é mostrado a seguir:

- O cliente *Web* solicita ao servidor *Web* a execução de um componente da aplicação no Servidor de Transação. Geralmente, alguns componentes são descarregados no cliente e tornam-se ativos neste momento, podendo ser acionados para realizarem ações em algum ponto da aplicação.
- O servidor *Web* atende ao pedido disparando o componente via interface de comunicação. Esta interface pode ser algum mecanismo desenvolvido sob uma API do servidor *Web* ou algum mecanismo próprio suportado pelo Servidor de Transação para comunicação com um determinado servidor *Web* proprietário;
- Em algumas implementações, os componentes são autenticados para se verificar se podem ser disparados pelo usuário da aplicação. Eles são tratados como unidades atômicas de processamento e coordenados pelo Servidor de Transação. Os dados do usuário e da transação que

estão sendo submetidos são mantidos persistentes até que uma ação *commit* ou *abort* seja realizada;

- O componente em execução pode ou não realizar uma ação sobre um ou mais Bancos de Dados. Caso seja solicitada uma ação sobre um Banco de Dados, o Servidor de Transação trabalha em cooperação com o *SGBDs* para que o pedido seja atendido e que sejam garantidas as propriedades ACID, estas propriedades serão vistas no capítulo 5; e
- Os componentes podem executar concorrentemente se usuários diferentes estiverem submetendo as mesmas ações ao mesmo tempo. O Servidor de Transação garante o isolamento e a consistência das ações realizadas. Ao final de uma solicitação do usuário, os dados são retornados via servidor *Web* ao cliente. É importante ressaltar, mais uma vez, que os componentes no lado deste podem trocar mensagens diretamente com componentes no Servidor de Transação.

Como exemplos desta tecnologia, podemos destacar:

- *Microsoft Transaction Server*<sup>47</sup>.
- *Oracle WebServer 3.0* em BUYENS(2000).

#### 5.7.4 Linguagem de acesso

A função básica desta tecnologia é permitir que desenvolvedores da *Web* escrevam rapidamente páginas geradas dinamicamente, dentre as quais podemos destacar o PHP e o ASP.

##### 5.7.4.1 *Personal Home Page (Php)*

O PHP é uma linguagem de *script* colocada dentro de páginas HTML, sua sintaxe é semelhante ao C, JAVA e Perl com alguns comandos específicos do PHP. A principal vantagem desta linguagem é proporcionar a desenvolvedores da *Web*, escrever rapidamente páginas geradas dinamicamente. Os princípios desta tecnologia

---

<sup>47</sup> [www.microsoft.com](http://www.microsoft.com)

são semelhantes ao *ColdFusion* da *Allaire*, *Mod\_Perl*, *JSP* da *Sun*, ou *ASP* da *Microsoft*. Esta linguagem está em expansão e vem para competir com o *ASP* da *Microsoft*.

Vale lembrar que o *script* do PHP é diferente de CGI escritos em Perl ou C, por exemplo. Ao invés de escrever um programa com vários comandos para gerar um HTML, você escreve um *script* com algum código dentro da própria página. O código PHP é encapsulado em uma *tag* o que o habilita a entrar e sair do modo PHP. O que distingue o PHP de *scripts* cliente como o *JavaScript* ou *VBScript*, é que o código é executado no servidor. Isso possibilita, entre outras coisas que o usuário não tenha acesso ao código escrito no servidor (conteúdo da *tag*).

Trabalhar com PHP conduz a uma arquitetura em três camadas de uma forma tão coesa que vários desenvolvedores nem notam a sua presença. Na primeira camada, existe um cliente que pode ser o *browser*. A camada intermediária é obviamente PHP e o servidor *Web* como aplicação hospedeira, enquanto que a terceira camada é um gerenciador de bancos de dados.

O funcionamento deste ambiente é semelhante á arquitetura ASP, ver Figura 16.

#### 5.7.4.2 Active Server Pages (Asp)

ASP é um ambiente de programação onde podemos combinar HTML, *scripts* e outros componentes para criar aplicações dinâmicas. Este ambiente foi criado pela Microsoft para o seu servidor *Web* IIS, e tem suporte nativo para *VBScript* e *Jscript*. Mas a linguagem mais usada é *VBScript*.

Os arquivos criados com ASP têm a extensão “.ASP”, ao invés do tradicional “.HTML”, e devem ser executados em um servidor *Web* compatível com ASP, observe seu funcionamento na Figura 16.

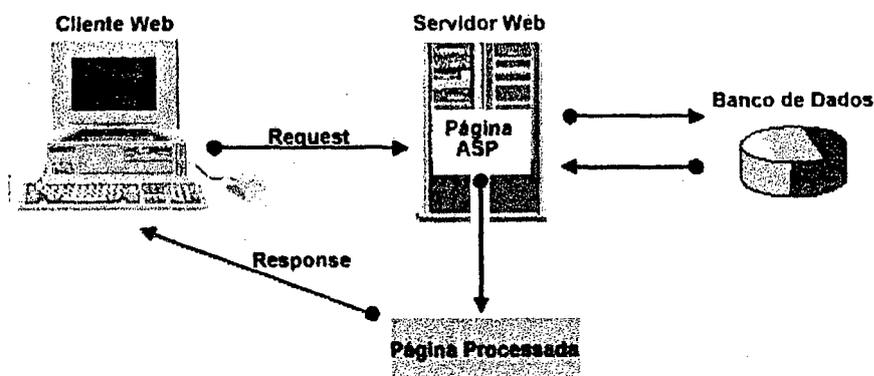


Figura 16 . Ambiente de programação ASP/PHP

O funcionamento desta arquitetura é mostrado a seguir:

- O *browser* do usuário faz a chamada a uma página “.ASP” localizada no servidor *Web*;
- O servidor lê a página ASP e inicia o processamento;
- A página é processada, executando os *scripts* localizados na página; e
- Como resultado do processamento, sai uma página HTML, que é enviada para o *browser* do usuário.

Como os *scripts* rodam no servidor, este faz todo o processamento e, na verdade, o que chega ao usuário são páginas HTML padrão. A partir disso, podemos chegar a duas importantes conclusões:

- Qualquer *browser* que suporte HTML pode ver páginas “.ASP”;
- O código-fonte dos *scripts* em *VBScript* ou *Jscript* não é visto pelo usuário. O que ele vê é o HTML gerado pelo servidor *Web*;

Com ASP, é bem mais fácil criar conexões entre o *browser* e os dados em formatos normalmente incompatíveis com HTML, como bancos de dados. ASP não cria um processo no servidor para cada pedido do usuário (um processo ISAPI), como acontece com o CGI. Usando ASP ao invés de CGI, um servidor pode atender a um grande número de pedidos de usuários de forma mais rápida e usando menos memória. Além disso, criar páginas ASP é em geral muito mais fácil do que criar aplicações CGI.

### **Conclusão**

Apesar das inúmeras vantagens hoje encontradas na integração dos atuais *SGBDs*, os desenvolvimentos de aplicações dirigidas são limitadas e muito dependentes das ferramentas e plataformas proprietárias distribuídas pelos fabricantes.

Além do quê, os *SGBDs* comerciais foram projetados e implementados levando apenas em consideração a questão do armazenamento e recuperação da informação, ou seja, atendem a pedidos de usuários que se conectam ao banco de dados em sessões contínuas, ficando o usuário conectado enquanto durar a sessão, o que não ocorre com a *Web*.

Portanto, verifica-se que os modelos atuais *SGBDs* não são adequados para um ambiente *Web* Banco de Dados.

No próximo capítulo, trataremos das questões referentes aos problemas encontrados durante a integração, isto se justifica pelo fato de termos chegado a conclusão que estes modelos estão muito longe de serem considerados perfeitos para a integração.

## Capítulo 6

---

### Problemas de integração

Proteger e integrar o conteúdo das páginas *Web*, são sem dúvida os problemas mais difíceis de resolver, pois a *Internet* não oferece nenhuma segurança de comunicação entre os usuários e o servidor. Devido a esta situação, a segurança de um *site Web* envolverá duas áreas distintas, segundo GONÇALVES(1997):

- A segurança da rede: refere-se aos computadores, discos rígidos e outros, ligados diretamente ao servidor *Web*; e
- A segurança das transações: refere-se à habilidade que o seu *site Web* tem em conduzir uma transação privada, dentro da *Internet*, sem nenhuma interferência indesejada.

Estas duas áreas podem ser divididas em basicamente quatro áreas de riscos:

- Risco de segurança: seu *site* pode ser acessado por pessoal não autorizado;
- Risco nas transações: informações que são interceptadas por pessoal não autorizado;
- Risco de informações: informações de senha e *login* são adquiridas por pessoas que invadem o *site*; e
- Risco de *bug*: são falhas dentro dos sistemas que abrem alguma porta para alguém entrar em sua máquina.

Por serem questões amplas, foge ao escopo deste trabalho referenciar todas estas questões, pois entende que as questões mais relevantes para a continuidade deste trabalho são com relação à parte transacional e a parte de autenticação.

#### 6.1 Transações HTTP e *BROWSER*

Os aspectos de segurança envolvendo os servidores e os clientes na *Web* são muitos. Eles variam em formato e forma, dependendo da política de segurança que rege

e o tipo de *browser* utilizado. Contudo, essas exigências de segurança estarão sempre interdependentes da relação cliente/servidor.

## 6.2 Transação<sup>48</sup> *Web* e Banco de Dados

Sem dúvida alguma, uma questão fundamental que deve ser levantada em um ambiente *Web Banco* é a definição de transação para este contexto. Assim, podemos exemplificar da seguinte forma: iniciam-se no cliente *Web* e pelo servidor de Banco de Dados.

Na *Web* um usuário pode, por exemplo, abandonar a navegação sem que isso seja notificado ao servidor *Web* ou ao *SGBDs*. Além do mais, há possibilidade do usuário navegar por outras páginas que em princípio não fazem parte do contexto transacional, sendo submetido pelo usuário, podendo retornar mais tarde à transação que estava em andamento, ou seja, uma transação é baseada em páginas *Web*, de modo que mesmo quando uma página de origem contenha vários comandos para serem realizados no Banco de Dados, seria desejável que todos eles fossem considerados como uma operação atômica de processamento. Isto indica que uma transação na *Web* é formada por um grafo de navegação por páginas *Web*, geradas dinamicamente ou não. Assim, elas são baseadas no modelo navegacional da *Web*, sendo que não existe como impor um rígido controle na ordem das transações. O usuário pode, por exemplo, submeter uma transação e logo em seguida retornar à mesma transação por meio da opção *back* do *browser* e resubmetê-la.

Sendo assim, a transação é baseada no modelo navegacional hipertexto, composto por uma ou mais páginas *Web*, de modo que o conceito de uma transação *Web Banco de Dados* se diferencia do conceito tradicional de transação de Banco de Dados como uma unidade básica de trabalho no *SGBDs*.

---

<sup>48</sup> Transações - são as unidades básicas de trabalho dos *SGBDs*. Uma transação é vista como uma unidade atômica de processamento consistente e confiável, composta por uma seqüência de operações elementares sobre os dados. Como produto da sua execução, uma transação pode mudar o estado do Banco de Dados ou não provocar qualquer alteração, dependendo dos efeitos de sua execução serem aceitos (*commit*) ou cancelados (*abort*).

Portanto, as transações devem ser executadas concorrentemente, de forma transparente e confiável para a aplicação, deixando o banco de dados de forma consistente, mesmo na presença de várias transações executadas simultaneamente. A confiabilidade refere-se à capacidade do sistema de Banco de Dados em resistir aos diversos tipos de falhas que podem ocorrer no ambiente durante a execução de uma transação.

Segundo ELMASRI & NAVATHE(2000), a avaliação da correção da execução concorrente de transações é feita com um critério que indica se o resultado final é ou não aceitável. O critério tradicionalmente empregado é o de seriabilidade do escalonamento das ações das transações, isto é, verifica-se se o resultado final é equivalente ao de uma execução serial dessas ações. Para isso podem ser usados diferentes mecanismos, sendo que o método mais difundido é o baseado em bloqueios.

Uma avaliação cuidadosa de transações *Web* Banco de Dados indica que os mecanismos tradicionalmente empregados para verificação da execução serial das transações em *SGBDs* tradicionais podem não ser apropriados ao ambiente *Web* Banco de Dados. De fato, os critérios de correção e seriabilidade de uma transação *Web* Banco de Dados devem levar em consideração os seguintes aspectos:

- Ser orientado a páginas *Web*, onde possivelmente uma página pode conter várias operações no Banco de Dados ou ser parte de uma única operação no Banco de Dados;
- Ser baseado em um controle de concorrência para interação com transações de longa duração. Isto é decorrente do fato de que os mecanismos atuais, como bloqueios, estão implementados neste modelo;
- Ser flexível o suficiente de forma a englobar combinação de várias páginas *Web* para execução de uma única transação; e
- Ter mecanismos para recuperação por falhas na comunicação entre cliente, servidor e servidor de Banco de Dados *Web* para garantir a integridade das transações.

Para SILBERCHATZ & SUDARSHAM(1999), todos os servidores *Web* implementam políticas idênticas aos *SGBDs*, tornando-se insuficientes para garantir, em algumas arquiteturas de integração, a consistência e o isolamento de transações *Web* Banco de Dados.

### 6.3 Transação *Web* - Através de Páginas

Concordando com GRAY & REUTER(1993), um ambiente de integração *Web* banco de dados obrigatoriamente deverá seguir o modelo de transação através de páginas, pois elas são a interface do usuário; assim, vale salientar o seguinte:

- Modelos de transação têm por objetivo fornecer um contexto definido para padrões de processamento em uma aplicação. Estes padrões variam de acordo com os requisitos que motivaram a proposição do modelo; e
- As transações gerenciadas pela maioria dos *SGBDs* existentes seguem o modelo de transações planas<sup>49</sup>. O modelo é assim chamado porque existe uma única camada de controle pela aplicação;

A forma mais simples de uma transação *Web* é aquela em que uma página é composta por uma ou mais operações sob o *SGBDs*. Uma vez que uma página *Web* deve ser considerada como uma unidade básica de operação, se a página contiver um conjunto de operações sob o *SGBDs*, é desejável que estas operações sejam consideradas como uma unidade atômica<sup>50</sup>.

Tendo em vista esta consideração, para KORTH & SILBERHATZ(1997), é importante restringir alguns mecanismos dos atuais *SGBDs* que possibilitam que cada operação no *SGBDs* seja considerada uma unidade atômica. Por exemplo, em diversos

---

<sup>49</sup> Transações Planas - O modelo é assim chamado porque existe uma única camada de controle pela aplicação: todas as operações entre os delimitadores de início (*begin transaction*) e término (*end transaction*) possuem o mesmo nível, e são aceitas ou rejeitadas em bloco (atomicidade). A expansão deste modelo para o ambiente de integração deve levar em consideração o escopo ou abrangência de uma transação *Web* Banco de Dados.

<sup>50</sup> Operação Atômica – É a garantia que todos os efeitos de uma transação se refletirão no banco de dados, ou nenhum refletirá, uma falha pode deixar o banco em um estado no qual uma transação tenha sido parcialmente executada.

*SGBDs* relacionais é possível configurar o ambiente de forma a permitir que a cada comando SQL submetido, o próprio *SGBDs* emita automaticamente um *commit*, indicando que a operação não será desfeita (*rollback*) caso alguma falha ocorra no próximo comando SQL. A desativação deste mecanismo no ambiente de integração possibilita dar uma semântica mais adequada ao escopo de transações no ambiente de integração. Dessa forma, se uma página *Web* contiver várias operações garante-se que todas serão efetivadas pelo *SGBDs* ou nenhuma delas será.

### 6.3.1 Várias páginas como uma unidade atômica

Conforme RAHMEL(1997), a limitação do escopo de uma transação *Web* Banco de Dados a uma única página *Web* é desejável para efeitos de desempenho, já que assim é minimizado, por exemplo, o tempo de duração de bloqueio aos dados no *SGBDs*, favorecendo a concorrência. No entanto, desenvolvedores de aplicações freqüentemente precisam que várias interações com o usuário sejam realizadas como se fosse uma única transação atômica. Cada interação origina no ambiente *Web* uma página e é desejável que cada página faça parte de uma transação global atômica. Para que isso seja possível deve-se expandir o escopo de uma transação *Web* Banco de Dados de forma a se aceitar várias páginas *Web* como uma unidade atômica no *SGBDs*.

Para implementar esta característica na *Web* pode-se, por exemplo, indicar na primeira página da operação o início da transação (*begin*) e na última página da operação o fim da transação (*commit*). Todas as operações que ocorrerem entre estes demarcadores serão consideradas como parte de uma única transação atômica global. Caso alguma delas falhe, a transação inteira é cancelada (*abort*) e um *rollback* é realizado pelos *SGBDs*; assim é necessário implementar uma sessão contínua de usuário sob o *SGBDs* para que se possa demarcar o início e o fim da transação.

### 6.3.2 Páginas que não fazem parte de uma transação

SHERRY(1998), coloca a possibilidade de inclusão de várias páginas formando uma única transação atômica. Isto é útil para resolver o problema de transações que exijam várias interações com o usuário. No entanto, a solução de delimitar um conjunto de páginas para compor uma transação atômica pode ser muito restritiva no ambiente *Web*. De fato, podem existir situações em que a navegação

hipertexto possibilite, durante a execução de uma transação composta por várias páginas *Web*, que o usuário saia do escopo transacional que deveria ser seguido, para realizar, por exemplo, consultas contidas em páginas que em princípio não fazem parte do contexto da transação que está sendo submetida.

Para implementar esta opção na *Web*, pode-se, além de demarcar a página que dá início e a página que origina o fim de uma transação, identificar quais páginas utilizadas durante a ação do usuário devem ser consideradas no contexto transacional. Isto pode ser feito, por exemplo, por meio da definição de um indicador transacional para a página. Quando uma página sendo submetida pelo usuário contiver um indicador de que ela não pertence à transação em andamento, o *SGBDs* pode, por exemplo, desconsiderar as ações desta página no contexto da transação global. Observe que a implementação desta característica no ambiente *Web Banco de Dados* é importante, pois permite que a característica hipertexto da *Web* não fique limitada.

#### 6.3.4 Páginas com dupla funcionalidade

A solução de contextualizar uma página *Web* como parte de uma transação ou não, ainda pode ser restritiva no contexto navegacional da *Web*. De fato, aí existem situações que nos permitem identificar momentos em que uma mesma página pode ora fazer parte de uma transação *Web Banco de Dados*, ora ser considerada como qualquer outra página do ambiente *Web*.

Para implementar esta característica no ambiente de integração pode-se enumerar, durante a fase de desenvolvimento da aplicação, quais são as páginas participantes da transação. Isto pode ser feito, por exemplo, por meio da URL que gera a página. Um conjunto de URLs definindo a transação atômica é delimitado por indicadores de início e fim da transação. Quando uma página sendo solicitada pelo usuário não estiver delimitada pelos indicadores de transação, o *SGBDs* pode, por exemplo, desconsiderar as ações desta página no *SGBDs*. Esta solução permite que páginas tenham dupla função: ora são consideradas na transação, ora não. Vai depender do desenvolvedor tê-las ou não incluído no contexto transacional para uma determinada aplicação.

## 6.4 Falhas de transações

Uma transação entra num estado de falha depois de ser determinado que ela não pode mais prosseguir com sua execução normal. Para KORTH & SILBERCHATZ(1997), uma transação pode falhar devido a erros de *hardware* em um dos componentes do sistema, erros na comunicação dos componentes do sistema (*links* perdidos), erros lógicos (dado não encontrado) e erros do sistema, como na detecção de *deadlocks*<sup>51</sup>. Nesse caso, devem ser ativados mecanismos de recuperação do sistema evitando estados inconsistentes no *SGBDs*. Nos atuais *SGBDs*, este é um problema já bem estudado e resolvido.

Já para PAZANDAK & SRISVASTAVA(1997), em uma arquitetura típica *Web* podem ocorrer inúmeras falhas; assim comentaremos algumas delas:

- Falha na máquina do cliente ou no *browser*, onde podem ocorrer falhas de *hardware* ou de *software* tornando a aplicação cliente inoperante por um determinado tempo. Se a conexão com o cliente não se reiniciar (ou seja, ocorrer *timeout* definido, por exemplo, por um administrador da aplicação *Web* Banco de Dados) deve-se garantir que todas as ações da transação submetidas pelo usuário sejam desfeitas (*rollback*).
- Falha na rede entre o cliente *Web* e o servidor *Web*, onde *links* de comunicação podem ser perdidos ou ocorrer falhas na comunicação entre o cliente e o servidor.
- Falha no servidor *Web*. Um *rollback* será realizado se o servidor não conseguir se restabelecer após um determinado tempo.
- Falha no *SGBDs*. Ele garante mecanismos de consistência e recuperação de transações submetidas quando for restabelecido.

---

<sup>51</sup> *Deadlocks* - Surge quando os objetos do banco de dados são bloqueados em tal seqüência que um grupo de transações possua as duas seguintes propriedades: Cada uma das transações é bloqueada, esperando por um objeto; A execução de cada uma das transações que não pertencem ao grupo, não permite o desbloqueio de qualquer transação no grupo.

ADJEROH & NWOSU(1997) questionam que, desde o momento em que se inicia uma transação no cliente até seu fim, uma transação *Web* Banco de Dados pode estar em três possíveis estados. São eles :

- Transação iniciada mas não confirmada: o usuário solicitou o início de uma transação e nenhuma resposta de confirmação foi recebida;
- Transação em processo: o usuário recebeu a confirmação de que uma transação se iniciou e está em execução e, portanto, uma sessão para ele será mantida pelo servidor junto ao *SGBDs*. Nenhuma confirmação de fim de transação (*commit*) foi solicitada; e
- Transação finalizada, mas não confirmada: o usuário decidiu confirmar o fim da transação (enviou um *commit*), mas nenhuma resposta de confirmação foi recebida. Neste caso o usuário fica, em princípio, sem saber se sua transação foi ou não efetivada no Banco de Dados.

De acordo com estes três estados, podemos implementar uma semântica para que seja ativado o processo de recuperação da transação. Não é intenção deste trabalho salientar estes pontos, mas em KORTH & SILBERHATZ(1997) e SILBERCHATZ & SUDARSHAM(1999), serão encontrados detalhes sobre este assunto.

### 6.5 Controle de restrições de integridade

As restrições de integridade presentes na grande maioria dos *SGBDs* têm por objetivo assegurar que as eventuais atualizações respeitem as regras estruturais e semânticas definidas para o Banco de Dados.

KORTH & SILBERHATZ(1997), definem basicamente dois tipos de restrições de integridade, restrições de modelo e restrições de domínio.

- restrições de modelo são geralmente fáceis de serem testadas e dizem respeito à organização interna do Banco de Dados, não sendo afetadas pelo ambiente *Web*.

- restrições de domínio são restrições sobre os valores que podem ser assumidos, por exemplo, por atributos de tabelas de *SGBDs* relacionais. Embora sejam fáceis de serem testadas pelo *SGBDs* isoladamente, podem gerar problemas quando se usa o ambiente *Web*.

### 6.5.1 Propriedades ACID

O suporte à integridade transacional e a sessão contínua, são requisitos essenciais para o desenvolvimento de qualquer aplicação *Web* Banco de Dados que requeira propriedades similares às aplicações desenvolvidas sob sistemas de Banco de Dados cliente/servidor.

SILBERCHATZ & SUDARSHAM(1999) e ELMASRI & NAVATHE(2000) salientam que a integridade transacional diz respeito às transações que satisfazem certas características tradicionalmente agrupadas em quatro propriedades:

- Atomicidade - se refere aos efeitos das transações sobre os estados do Banco de Dados como unidades de operação únicas e indivisíveis. Isto significa que todos ou nenhum dos efeitos são considerados;
- Consistência - requerimento que estipula o estado consistente do sistema ao final da transação. É expressa de duas maneiras: a consistência da transação em si mesma e a consistência do Banco de Dados em face da execução concorrente de transações;
- Isolamento - implica que cada transação deve ver um estado consistente do Banco de Dados sempre que for executada. Isto requer que os resultados intermediários de uma transação não sejam observáveis por outras transações e/ou usuários; e
- Durabilidade - significa que os efeitos de uma transação, uma vez completada, são tornados permanentes no Banco de Dados.

Uma transação pode ser finalizada de duas maneiras: com sucesso (*committed*) ou cancelada (*aborted*). Quando uma transação é finalizada com sucesso,

todas as mudanças feitas por ela são tornadas duráveis no Banco de Dados. Quando uma transação é cancelada, todas as mudanças feitas por ela são desfeitas (*rollback*).

Assim, o problema específico que é o de garantir a integridade transacional no ambiente de integração ainda foi pouco estudado. Entre os fatores que contribuem para isso está a complexidade de uma solução que combine a limitação natural do ambiente *Web* com a série de interações entre o cliente *Web* e o *SGBDs* de forma a torná-las atômicas, consistentes e não afetadas por estados intermediários de outras interações concorrentes. Além disso, é desejável que qualquer proposta considere os mecanismos atuais dos *SGBDs*, que já são providos de técnicas para garantir as propriedades ACID para transações sob seu controle.

## 6.6 Acesso ao banco de dados

Para ADJEROH & NWOSU(1997), outro problema importante quando se discute a integração *Web* e Banco de Dados é relativo à segurança nas transações entre o cliente *Web*, o servidor *Web* e o *SGBDs*. Dois pontos se destacam: a segurança na comunicação entre cliente e servidor e a segurança neste.

A segurança e o acesso ao Banco de Dados dizem respeito a uma das importantes funcionalidades dos atuais *SGBDs*. Pode-se ver a *Web* como um usuário do *SGBDs* como qualquer outro. Nestes termos, os mecanismos de segurança e acesso ao Banco de Dados poderiam funcionar como os já existentes. Conforme ELMASRI & NAVATHE(2000), o uso de *visões* proporcionado pelos *SGBDs* no ambiente de integração pode ser bastante útil. Além disso, alguns fatores devem ser levados em consideração com relação à segurança e acesso ao Banco de Dados, como os destacados a seguir:

- Gerenciador de acesso: a maioria dos *SGBDs* mantém permissões para acesso ao Banco de Dados (senhas de *login*) e aos objetos do Banco de Dados (*grants*).
- Acessos simultâneos: o número de acessos simultâneos ao Banco de Dados é geralmente limitado. Dependendo dos requisitos da aplicação,

um usuário pode não conseguir completar sua transação devido a sobrecarga no gerenciador de acessos do *SGBDs*.

- **Autorização:** em algumas integrações *Web* e Banco de Dados, a autorização para execução de uma aplicação *Web* que consulta ou atualiza o Banco de Dados é dada ao *software* usuário que ativou o *gateway* (geralmente o próprio servidor *Web*). Assim, qualquer cliente *Web* poderia, por exemplo, disparar a aplicação via servidor *Web*. Mecanismos de configuração deste servidor e do *SGBDs* devem estar presentes de forma a contornar este problema.
- **Recuperação:** um esquema de recuperação em Banco de Dados tradicionais é ativado como consequência de diversos tipos de falhas, como por exemplo, erros lógicos, paradas de sistema e falhas de disco. A extensão destes mecanismos para o ambiente *Web* não é imediata e os *gateways* devem implementar mecanismos adicionais para tratar de falhas dos componentes da aplicação para que possa, por exemplo, ser comunicado ao usuário o resultado de uma transação solicitada. Hoje não existe nenhum mecanismo de recuperação de paradas e falhas de clientes e servidores *Web*. Só estão presentes os mecanismos tradicionais dos *SGBDs*, que podem ser insuficientes no contexto transacional *Web*.

## 6.7 Segurança de Dados

A garantia dos direitos autorais de objetos digitais passa pelo uso da própria tecnologia digital. Dessa forma, faz-se necessária uma análise dos métodos atualmente empregados para proteção de dados a fim de propor formas de aplicação ou até mesmo adaptações para o uso em bibliotecas digitais.

BERNSTEIN(1997), identifica diferentes serviços de segurança, relacionados a partir da Tabela 3.

Serviços de Segurança	Exemplo
<b>Identificação e Autenticação</b> Determinar a identidade de uma entidade (usuário, aplicação ou sistema) e confirmar se a entidade é quem ou o que afirma ser.	Exigir que um usuário final informe uma ID de Usuário e uma senha ao estabelecer <i>logon</i> com a rede e a máquina de <i>firewall</i> .
<b>Controle de Acesso</b> Uma vez que uma entidade tenha sido identificada e autenticada, o ato de decidir quais serão as concessões, atribuições, autorizações, direitos ou permissões associadas à execução da tarefa solicitada.	Atribuir direitos e privilégios a um usuário de uma conta corporativa.
<b>Integridade dos Dados</b> Garantir que dados não tenham sido modificados, acrescentados ou eliminados ao serem transportados ou armazenados e que estão completos.	Executar uma operação de soma de verificação em um arquivo que foi transferido para garantir que ele não foi alterado durante o processo de transmissão.
<b>Confidencialidade dos Dados</b> Garantir a proteção e a não-revelação de dados devido à natureza (legal, reguladora, patenteada ou confidencial) do recurso de informação. Criptografar dados confidenciais que serão enviados através da <i>Internet</i> .	Utilizar assinatura digital para confirmar quem é o proprietário dos dados; Incluir aviso de <i>copyright</i> em dados patenteados.
<b>Não-Repudição</b> Fornecer a integridade e a origem dos dados em uma relação que não deve ser falsificada e que pode ser confirmada por terceiros a qualquer momento.	Complementar o processo de identificação e de autenticação de transações baseadas na <i>Internet</i> com tecnologias como a criptografia com chave pública, os serviços de tabelião digital e a assinatura digital.
<b>Disponibilidade dos Dados</b> Garantir que os dados estão presentes, acessíveis e podem ser obtidos rapidamente de acordo com pré-requisitos reguladores. Isso também implica a utilidade dos dados, pois eles devem estar prontos para serem usados.	Garantir a implementação de procedimentos de <i>backup</i> adequados para aplicações de servidor da <i>Web</i> .

Tabela 3 . Serviços de Segurança.

Para um sistema de bibliotecas digitais identificamos a necessidade de quatro dos sete serviços acima indicados, a saber:

- Identificação e autenticação: a fim de evitar que usuários não autorizados tenham acesso à biblioteca ou mesmo como forma de limitar níveis de acesso;

- Controle de acesso: a fim de que se possa ter um controle de quem está fazendo uso dos serviços da biblioteca, a que categoria de usuário pertence, quais os seus direitos ou restrições de acesso, etc;
- Integridade dos dados: a fim de garantir que o objeto digital não seja alterado durante o transporte via rede;
- Confidencialidade dos dados: a fim de garantir que os dados, mesmo se interceptados na comunicação entre dois sistemas, não possam ser revelados, além de garantir a origem dos mesmos.

Para que tais serviços sejam implantados de forma segura e confiável, pode-se fazer uso de diferentes mecanismos de segurança como, por exemplo, identificadores, senhas, assinaturas digitais, entre outros.

#### 6.7.1 Autenticação de Usuários

Segundo BERNSTEIN(1997), os serviços de autenticação são um elemento importante em qualquer sistema de segurança na *Internet*. Diversos métodos e aplicações oferecem serviços de autenticação com diferentes graus de certeza. Em geral, quanto maior for a certeza necessária para identificar um usuário, maior será o custo e mais difícil será a utilização do método.

No lugar de senhas reutilizáveis, quatro técnicas principais são utilizadas para autenticação. Essas técnicas se baseiam na localização de uma pessoa ou computador, ou no que essa pessoa ou computador conhece, tem ou é. Ao lidar com pessoas é necessário utilizar mais de uma dessas técnicas para autenticar a identidade delas.

- Autenticação baseada na localização: muitos sistemas de rede se baseiam em um identificador de usuário (ID) e no endereço de rede do sistema original para fazer a autenticação. Ou seja, o autenticador assume que a identidade da origem pode ser inferida com base no endereço (de rede) de onde os pacotes foram enviados. A autenticação se baseia unicamente no fato de que a comunicação com um ID de

usuário tem origem em um determinado *host*<sup>52</sup> com um endereço IP específico.

- Autenticação baseada naquilo que alguém conhece: em geral, esse método é implementado através da utilização de uma combinação de identificador e senha do usuário. O problema é que não há uma forma infalível de garantir que apenas pessoas confiáveis conheçam a senha adequada. As senhas ocasionais representam uma nova variante dos tradicionais esquemas de senhas. Para acessar um *host*, o usuário tem de informar um ID e uma senha, exatamente como acontece em um sistema de senha comum; no entanto, a senha é válida apenas uma vez. Nesses sistemas, uma lista de senhas é computada antecipadamente, e os resultados são impressos para facilitar a entrada do usuário.
- Autenticação baseada naquilo que uma entidade é ou representa: essa categoria abrange os atributos físicos (como as impressões digitais) de pessoas ou computadores. Infelizmente os sistemas biométricos são muito caros e ainda estão em fase de experiência. Além disso, poucos são os sistemas existentes no mercado que fazem autenticação verificando os atributos do computador. Por exemplo, o sistema de autenticação pode registrar o tipo da CPU, o tamanho do disco rígido, as aplicações instaladas e outros itens para criar uma “impressão digital” do computador. Quando um usuário estabelece *login*, o sistema verifica sua “impressão digital” antes de permitir o acesso.
- Autenticação baseada naquilo que uma pessoa ou entidade possui: o fato de alguém ter uma placa especial de computador pode ser usado

---

<sup>52</sup> *Host* - Um computador da Internet ou de uma rede local no qual você pode se conectar usando Telnet para obter arquivos usando FTP ou de qualquer outra forma fazer uso dele. O termo *Host Address* é o endereço propriamente dito que identifica um *Host* e que possibilita ao seu computador solicitá-lo quando você estiver na Rede.

para verificar sua identidade. Em geral, esses dispositivos são protegidos por senhas que, após sucessivas adivinhações erradas, deixam de funcionar.

### 6.7.2 Criptografia

A criptografia é tão antiga quanto a própria escrita. Os romanos utilizavam códigos secretos para comunicar planos de batalha e, atualmente, a criptografia é reconhecida como ferramenta indispensável para a proteção da propriedade intelectual, da integridade e confiabilidade de informações financeiras, da integridade da segurança individual e pública, além da segurança nas operações do comércio eletrônico.

Basicamente, a criptografia consiste na transformação de uma mensagem em uma forma ininteligível. Assim, a mensagem transmitida via rede somente será lida pelo destinatário que possuir a “chave” capaz de recompor a estrutura inicial da mensagem. Existem basicamente dois tipos de criptografia, cada tipo podendo ser implementado através da utilização de variados métodos e algoritmos. Resumidamente temos o seguinte:

- Criptografia Simétrica: conhecida também como criptografia de chave simples, funciona bem em aplicações limitadas, como as militares, onde o emissor e o receptor podem se preparar antecipadamente para trocar chaves. As mensagens são codificadas com uma chave secreta ou compartilhadas e decodificadas utilizando a mesma chave.
- Criptografia Assimétrica: conhecida também como criptografia de chave dupla, onde uma chave é utilizada para a transmissão de mensagens, de conhecimento exclusivo do remetente, e outra para a decodificação desta mensagem, de conhecimento exclusivo do destinatário. Uma chave não pode ser deduzida de sua correspondente. Implementa também uma variante, conhecida como criptografia de chave pública. Neste tipo de sistema cada pessoa tem duas chaves: uma chave pública e uma chave privada. As mensagens criptografadas com uma das chaves do par só podem ser decriptografadas com a outra chave correspondente; portanto, qualquer mensagem

criptografada com a chave privada só pode ser descriptografada pela chave pública e vice-versa. Como o nome sugere, normalmente a chave pública é mantida universalmente disponível em um “servidor de chaves”, e a chave privada é mantida em segredo.

Um comparativo entre estes dois tipos de criptografia pode ser acompanhado a partir da Tabela 4:

Vantagem	Desvantagem
<b>Chave Simétrica</b> <ul style="list-style-type: none"> <li>• Rápida</li> <li>• Pode ser facilmente implementada em <i>hardware</i></li> </ul>	<ul style="list-style-type: none"> <li>• As duas chaves são iguais</li> <li>• Distribuição das chaves é difícil</li> <li>• Não aceita assinatura digital</li> </ul>
<b>Chave Pública</b> <ul style="list-style-type: none"> <li>• Utiliza duas chaves específicas</li> <li>• Distribuição das chaves é relativamente fácil</li> <li>• Oferece integridade e não-repudição através da assinatura digital</li> </ul>	<ul style="list-style-type: none"> <li>• Relativamente lenta</li> <li>• Computacionalmente intensiva</li> </ul>

**Tabela 4 .** Comparativo entre tipos de criptografia

Através da combinação destes dois tipos básicos de criptografia, obtém-se diferentes mecanismos de segurança.

## Conclusão

A tecnologia *Web* e, em particular, a tecnologia de integração *Web* e Bancos de Dados ainda estão em pleno desenvolvimento. Como ressaltado ao longo deste capítulo, ainda existem vários problemas técnicos para que a integração seja satisfatória.

Alguns destes problemas são inerentes ao ambiente *Web* ou ao *SGBDs* isoladamente e, portanto, sua solução depende de alterações funcionais das tecnologias envolvidas ou a busca por ferramentas proprietárias. Por exemplo, o suporte a um modelo de transações adequado ao ambiente *Web* implica em mudanças significativas no modelo de transações atualmente suportados pelos *SGBDs*. Outro exemplo é a solução para a limitação da interface *Web* (HTML) que não proporciona suporte a interfaces orientadas a dados como atualmente suportado pelas aplicações de Banco de

Dados cliente/servidor. Neste caso, deve-se estender a linguagem HTML ou usar linguagens proprietárias gráficas suportadas por *browsers*, como Java.

A alteração das tecnologias envolvidas não é a solução mais adequada para os problemas abordados, a não ser se for patrocinada por organizações interessadas na padronização do ambiente de integração ou para efeitos de pesquisa de ponta. Diante deste fato, cabe ao desenvolvedor da aplicação identificar as limitações que a integração das tecnologias podem gerar e solucioná-las da melhor forma possível de acordo com as ferramentas disponíveis. Contudo, o problema decorrente da integração de sistemas *Web* com sistemas de Banco de Dados é o problema da manutenção das propriedades ACID de Banco de Dados no ambiente de integração.

No capítulo posterior, faremos uma analogia traçando um paralelo entre *frameworks*, e a construção de um modelo para biblioteca digital multimídia.

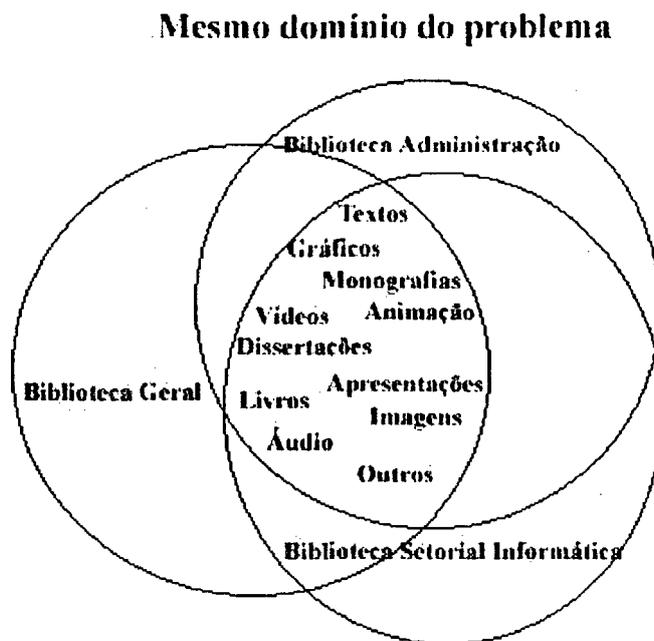
## Capítulo 7

---

### Ambiente para construção de Bibliotecas Digitais

Neste capítulo, analisaremos toda uma aplicação, onde todo o projeto será feito de forma genérica delimitando um domínio específico do problema, para então tornar se molde para futuras atualizações. Por este motivo neste trabalho segue a definição proposta por BUZATO & RUBICA(1998): o *framework* é uma meta-estrutura vista como uma arquitetura de sistemas computacionais semi-acabada para um domínio que pode ser adaptado para resolver seus problemas específicos. Assim, segundo os mesmos autores, ao contrário das abordagens tradicionais para a reutilização de *software* que se limitam basicamente na construção de bibliotecas de classes, meta-estruturas permitem reutilizar não apenas componentes isolados mas toda a arquitetura de sistemas computacionais projetadas para um domínio específico.

Utilizando-se destes conceitos é possível fazer a seguinte afirmação: um *framework* captura a funcionalidade comum a várias aplicações, sendo que as mesmas devem ter algo razoavelmente grande em comum e pertencem a um mesmo domínio de problema, conforme exemplificada na Figura 17, os tipos de mídia encontrados nesta modelagem, tais como, vídeos, áudio, textos, etc., possuem características similares. Por isto, toda esta aplicação está moldada visando o seu reuso, flexibilidade e extensibilidade, as quais permitem a construção de qualquer tipo de biblioteca com qualquer tipo de mídia.



**Figura 17.** *Frameworks*: delimitando o problema

Com isso, conclui-se que os problemas citados acima estão relacionados, por isso é possível fazer a analogia que após sua implementação o sistema vai fornecer formas de reuso de análise, designe e código, assim como as aplicações de *frameworks*.

## 7.1 Metodologia aplicada

O planejamento da interface para a biblioteca digital é de suma importância, pois é através dela que é possível detectar e prever, em tempo de desenvolvimento, os possíveis problemas ocorridos e a distribuição das informações através da referida biblioteca. Este planejamento requer habilidades em diversas áreas, incluindo computação, designe gráfico, manipulação de dados de mídia contínua e multimídia.

Para a construção desta biblioteca foram seguidas algumas etapas, quais sejam:

- **ANÁLISE** - A análise funciona melhor quando a metodologia adotada ajusta-se bem às características de cada usuário e aos recursos disponíveis; é importante que se realize com cuidado a fase de análise, que consiste em identificar todos os fatores que provavelmente influenciarão a construção da biblioteca digital.
- **PLANEJAMENTO** - Aqui foram seguidos alguns passos, com o objetivo de disciplinar o planejamento da biblioteca digital e garantir sua qualidade. Nele deve ser claramente identificado o que se pretende realizar e como fazê-lo, sendo que o planejamento sugerido consiste de dois passos: a identificação das formas a serem alcançadas e o estabelecimento de prioridades nessa lista de soluções.
- **PROJETO** - A fase de projeto é intermediária entre o planejamento e a implementação. Um bom projeto resulta em material que atrai visitantes, mantém o interesse, evita o problema de desorientação no hiperespaço, maximiza a audiência, facilita expansões, etc.
- **IMPLEMENTAÇÃO** - A fase de implementação consiste em dar subsídio para uma rápida e satisfatória implantação e implementação do *software*.

Neste escopo é necessário, primeiramente, cumprir uma etapa para então começar outra.

## 7.2 Descrição da Aplicação

Conforme exemplificado no apêndice A, a aplicação da biblioteca digital se desenvolveu num ambiente *Web* Banco de dados. Para tanto, faremos referência à arquitetura que roda no lado do Servidor *Web*, onde este trabalho está todo enquadrado. A escolha da aplicação para este estudo deve-se a sua representatividade no contexto atual das arquiteturas disponíveis no mercado e sua relevância no contexto do problema. Apesar das inúmeras opções existentes no mercado, optamos em implementar esta aplicação utilizando tecnologia proprietária da *Microsoft*, por ter uma maior ganho de

integridade e desempenho, devido à tecnologia pertencer a uma mesma empresa. Vale salientar, que esta aplicação não representa todas as atuais implementações neste estilo; uma outra solução também bastante aceita são as aplicações CGI, pois possuem a vantagem de rodar em qualquer plataforma, conforme capítulo 4.

Com as novas tecnologias, qualquer computador rodando um sistema operacional é capaz de atuar como servidor de páginas. Porém, em grandes aplicações, devido à velocidade necessária para a carga das páginas e dos tipos de páginas desenvolvidas, o melhor é optar por um sistema operacional próprio para estes serviços. Aí então, o *Windows NT Server*, em relação aos seus concorrentes, tem se mostrado possuidor de muitas qualidades, tais como segurança, performance e uma série de ferramentas para o desenvolvimento incluso no sistema operacional. Por este motivo, o servidor de página de *Internet* escolhido para a implantação desta biblioteca digital foi o *Internet Information Server* (IIS) por ser um dos mais utilizados na atualidade e por não ter custo na sua instalação, além de ir de encontro à idéia básica da integridade, pois o mesmo vem com o *Windows NT Server*; além disso, é muito simples de ser instalado, configurado e mantido. O IIS permite ainda a criação de páginas dinâmicas, ou seja, aquelas que mudam o seu conteúdo conforme parâmetros recebidos. Este projeto, por ser possuidor de conceitos de *frameworks* e estar utilizando tecnologia que roda no servidor, foi necessário adotar um gerador de páginas dinâmicas: assim utilizaremos o ASP (*Active Server Pages*), que se utiliza da tecnologia *ActiveX*. Nestas mesmas páginas, pode-se incluir código de programação em *VBScript* ou *Jscript* para alteração do conteúdo das páginas no momento de sua carga.

No tocante a servidores de banco de dados, optou-se pelo *SQL Server 7.0*, por ser um produto da mesma linha da Microsoft, onde o mesmo está integrado com o sistema operacional, de modo que terá acesso mais rápido aos dados, além de ferramentas de administração e desenvolvimento mais amigável.

Esse estudo de caso pretende discutir detalhes de características já abordadas e outras ainda não citadas de forma mais objetiva e direcionada à arquitetura escolhida para o desenvolvimento da aplicação em questão.

### 7.3 Construção da aplicação

Através desta aplicação foi possível fazer todo o cadastramento para o controle e manutenção da biblioteca digital, além de diversas consultas parametrizadas sobre toda a base de dados disponível.

A aplicação se inicia quando o usuário selecionar a URL que contém a página principal para então escolher o serviço desejado, conforme é possível observar na Figura 18 da tela de abertura da biblioteca digital.

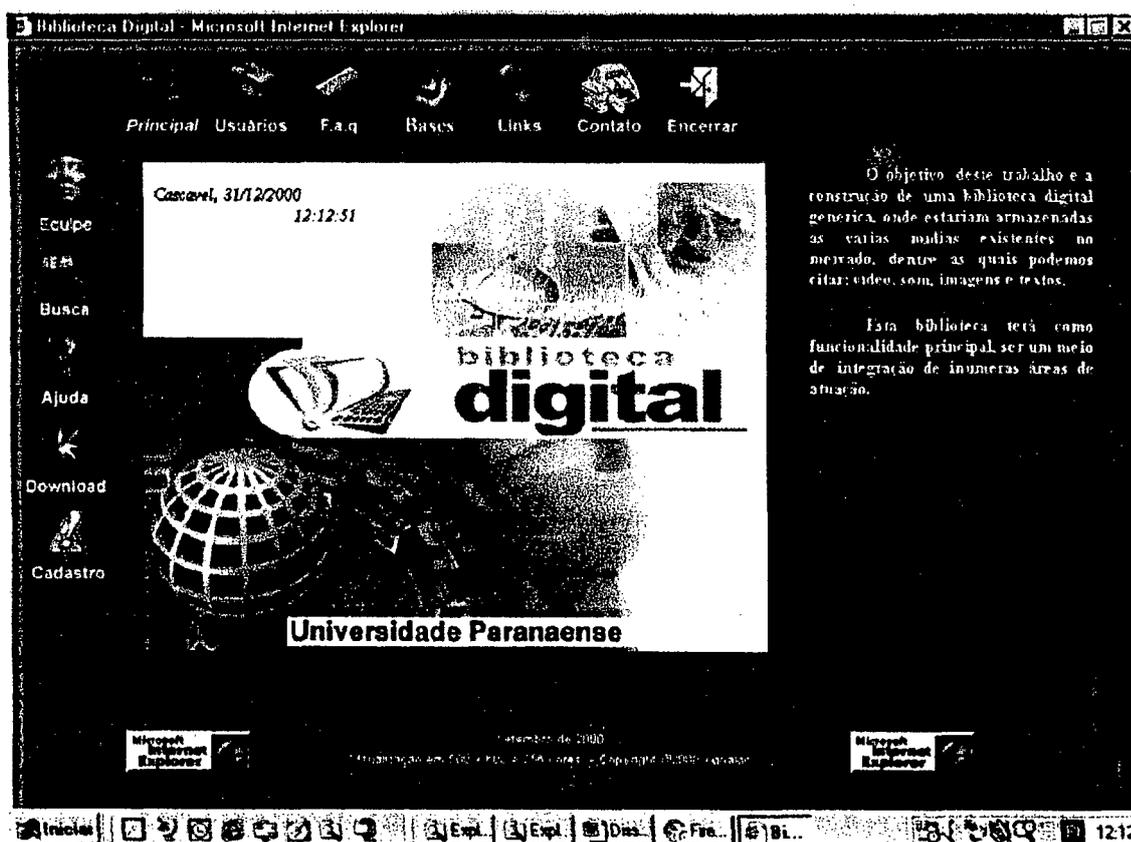


Figura 18 . Tela de entrada do site da biblioteca digital.

Esta biblioteca digital teve como preocupação ser uma base de dados multimídia genérica. Segue abaixo um resumo dos itens da tela principal:

- *Links*: apresenta um catálogo de URLs de projetos que estão em desenvolvimento.
- *FAQ*: um dos melhores meios de tirar dúvidas referentes à utilização da página.

- *Download*: este tem por objetivo apresentar uma lista de *softwares* que são usados para visualização das mídias encontradas na página.
- *Busca*: aqui serão encontrados os principais robôs de buscas internacionais e nacionais.
- *Ajuda*: através desta opção é possível ter algumas definições básicas para que seja possível navegar na biblioteca digital.
- *Equipe*: neste módulo é possível entrar em contato com os desenvolvedores deste *software*.
- *Principal*: ao clicar nesta opção o usuário retornará à tela principal, com o fechamento da sessão.
- *Encerrar*: nesta opção será efetuado o fechamento da sessão.
- *Bases*: no caso de haver mais alguma base de dados a ser acessada pelo usuário.
- *Contato*: aqui serão definidos os possíveis contatos do usuário com os respectivos responsáveis pelas bibliotecas digitais.
- *Cadastro*: neste módulo, caso o usuário que não possua cadastro, o sistema fornecerá meios para que este seja feito, conforme a Figura 19. Após feito todo o preenchimento será emitido então um boleto bancário para o usuário efetuar o pagamento (conforme a Figura 20) o qual, uma vez efetuado, será liberado o acesso ao usuário.

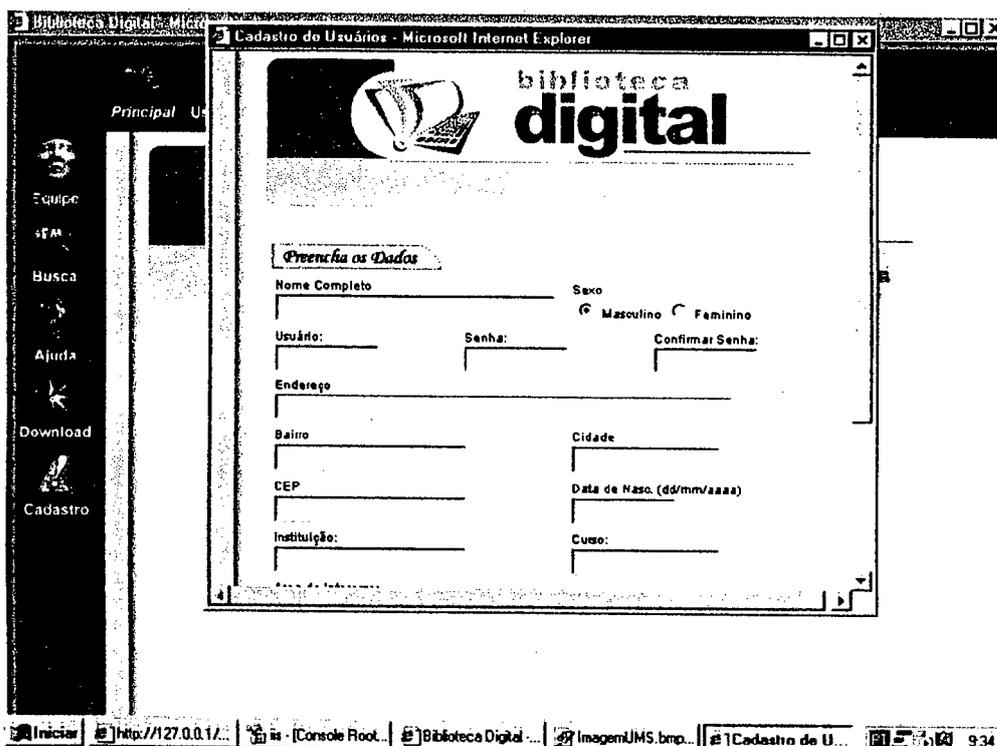


Figura 19 . Tela de cadastro de novo usuário.

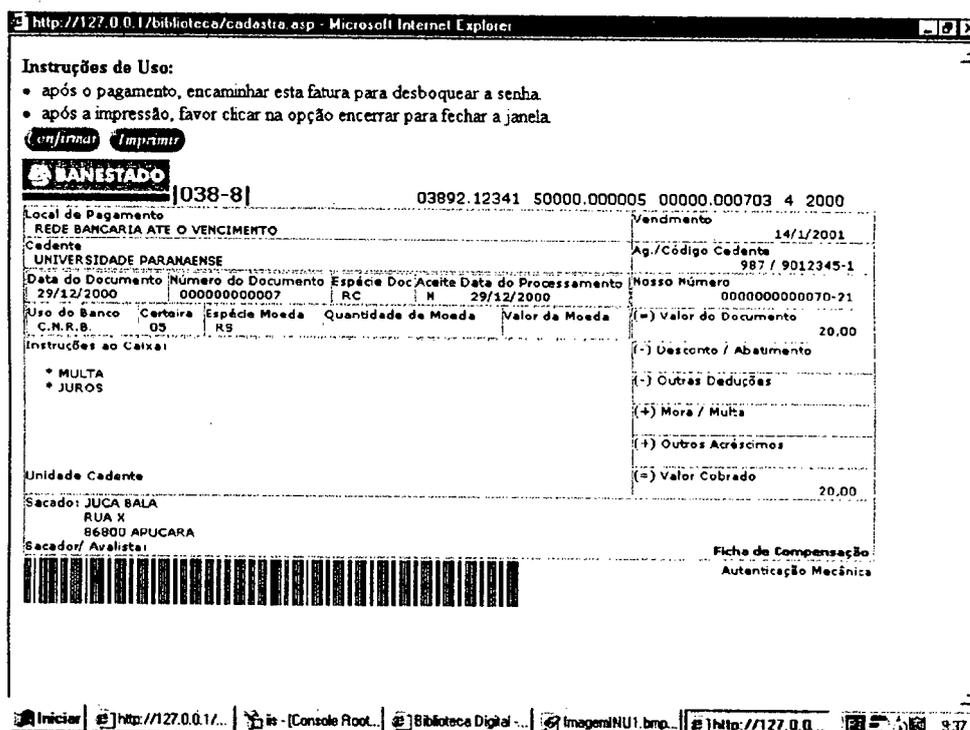


Figura 20 . Boleto gerado para pagamento.

- **Usuário:** nesta opção o usuário terá acesso direto ao sistema se já for cadastrado através da verificação do *login* e do *password*, conforme a Figura 21. Com esta opção é possível ao sistema controlar o nível do usuário, utilizando o conceito de permissões. Vale salientar que são feitos todos os controles de exceções.

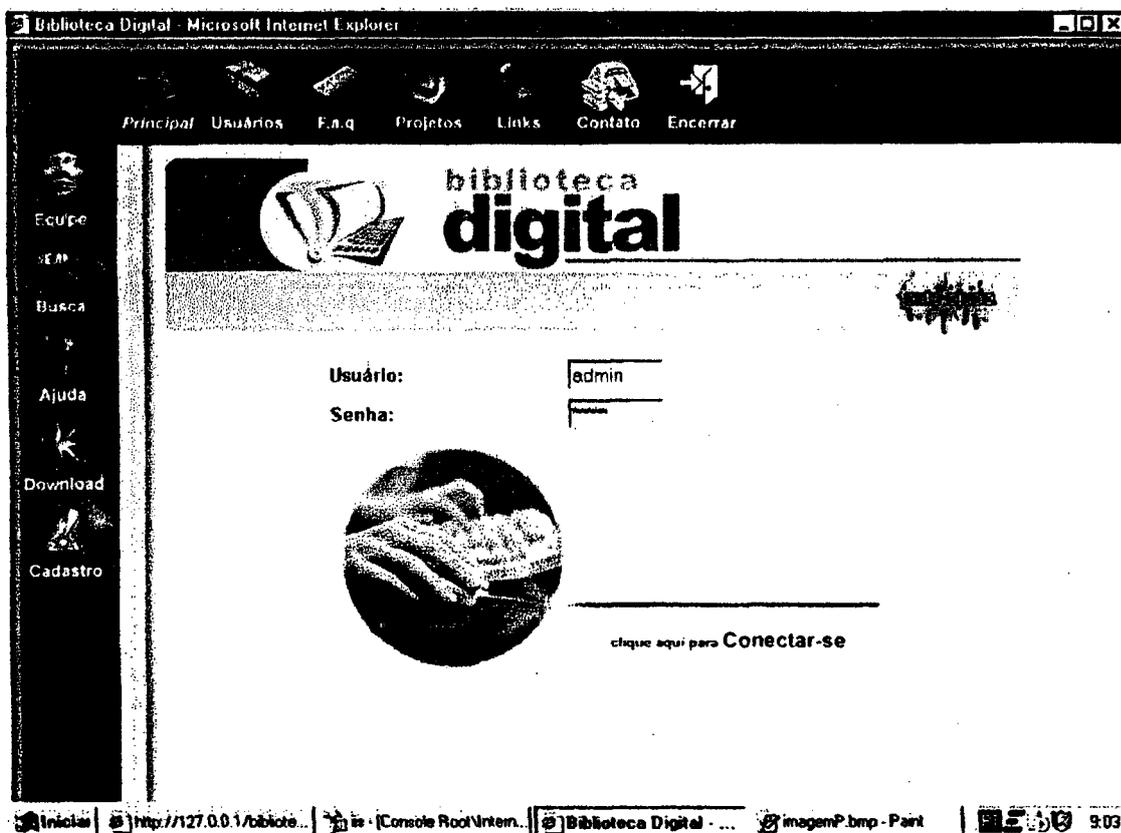


Figura 21 . Tela de *login* de usuário

Após a verificação do *login* e *password* do usuário, o sistema controlará três níveis, quais sejam:

- **Nível 1 – Administrador.** O administrador terá acesso a dois módulos: módulo de usuário, conforme a Figura 22 e módulo da base de dados da biblioteca, conforme exposto na Figura 23.
- **Nível 2 – Funcionário.** As permissões do funcionário seguem o mesmo padrão do administrador só que com restrições. As Figuras 22 e 24 detalham este procedimento.

- Nível 3 – Usuário. O usuário terá direito às seguintes permissões, conforme a Figura 25.

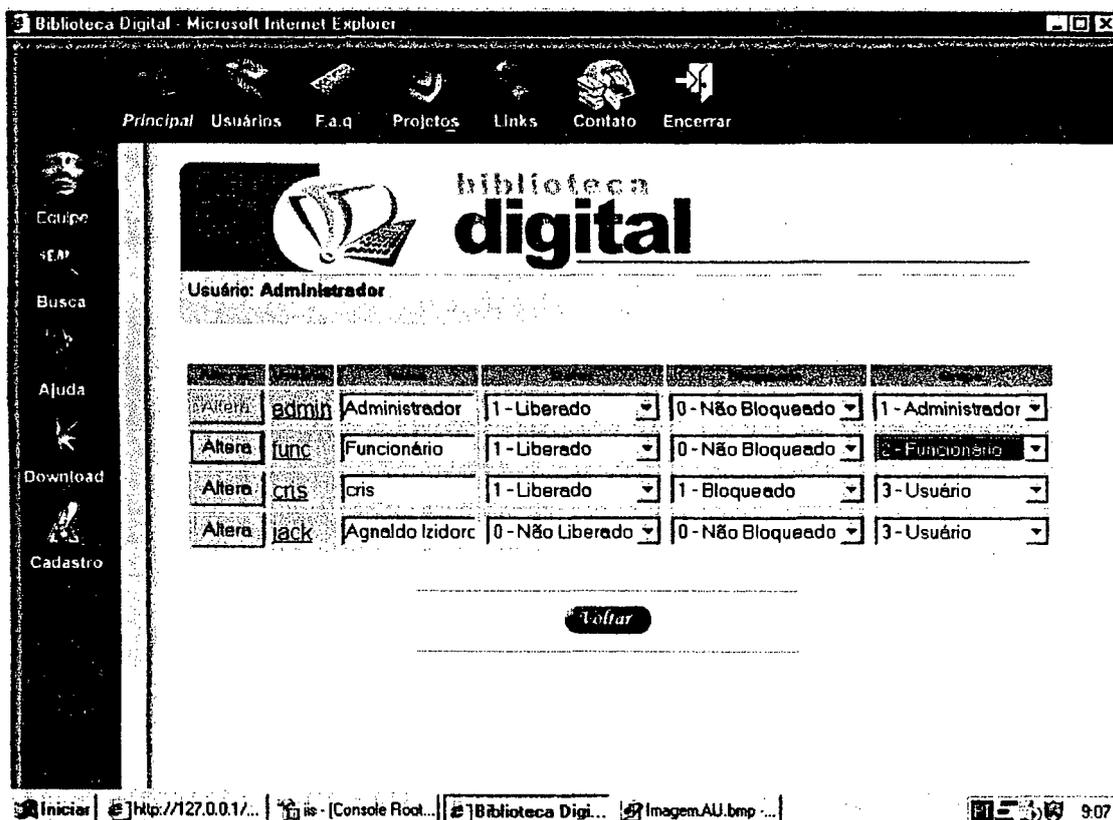


Figura 22 . Manutenção no cadastro do usuário.

No módulo usuário será permitido, no primeiro instante, que sejam alterados os principais itens da base. Não satisfeita esta condição, poderemos alterar o resto dos campos clicando no campo “usuário”, de modo que assim será possível manter o resto da base. Além disso, o módulo “usuário” permite que se faça uma busca rápida na base de dados. Esta opção aparecerá logo após entrar neste módulo.

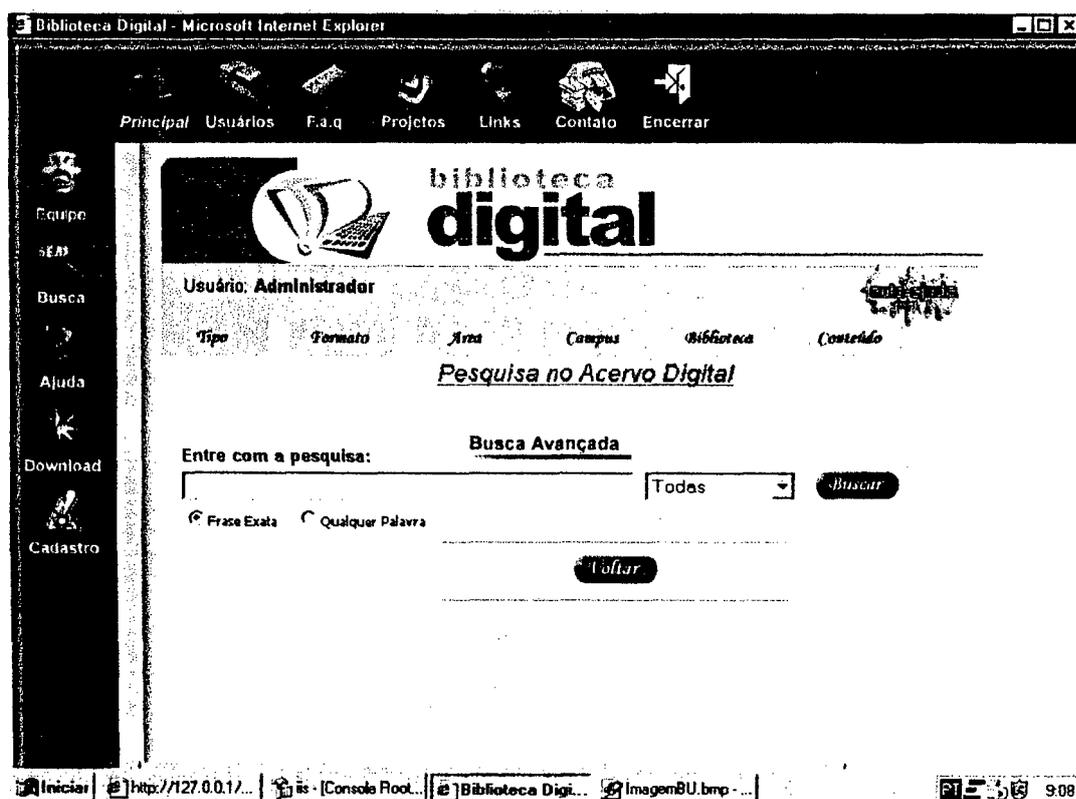


Figura 23 . Opções de acesso no módulo base (Administrador).

Neste módulo o administrador poderá manter os seguintes itens:

- Tipo - aqui será cadastrado o tipo do documento ou mídia, por exemplo, Imagens, Sons, Apresentações, Artigos e outros. (de acordo com o que se deseja armazenar).
- Formato - local onde é incluído o formato das mídias, por exemplo, PPT, PS, DOC e outros.
- Área - onde serão cadastradas as várias áreas de cada documento ou mídia.
- Campus - sessão destinada à manutenção dos vários campos que poderão estar sujeitos à pesquisa.
- Biblioteca - destina-se ao cadastramento das bibliotecas onde estarão guardadas as respectivas mídias.

- Conteúdo - local onde será feita a manutenção de todo o conteúdo das bibliotecas.

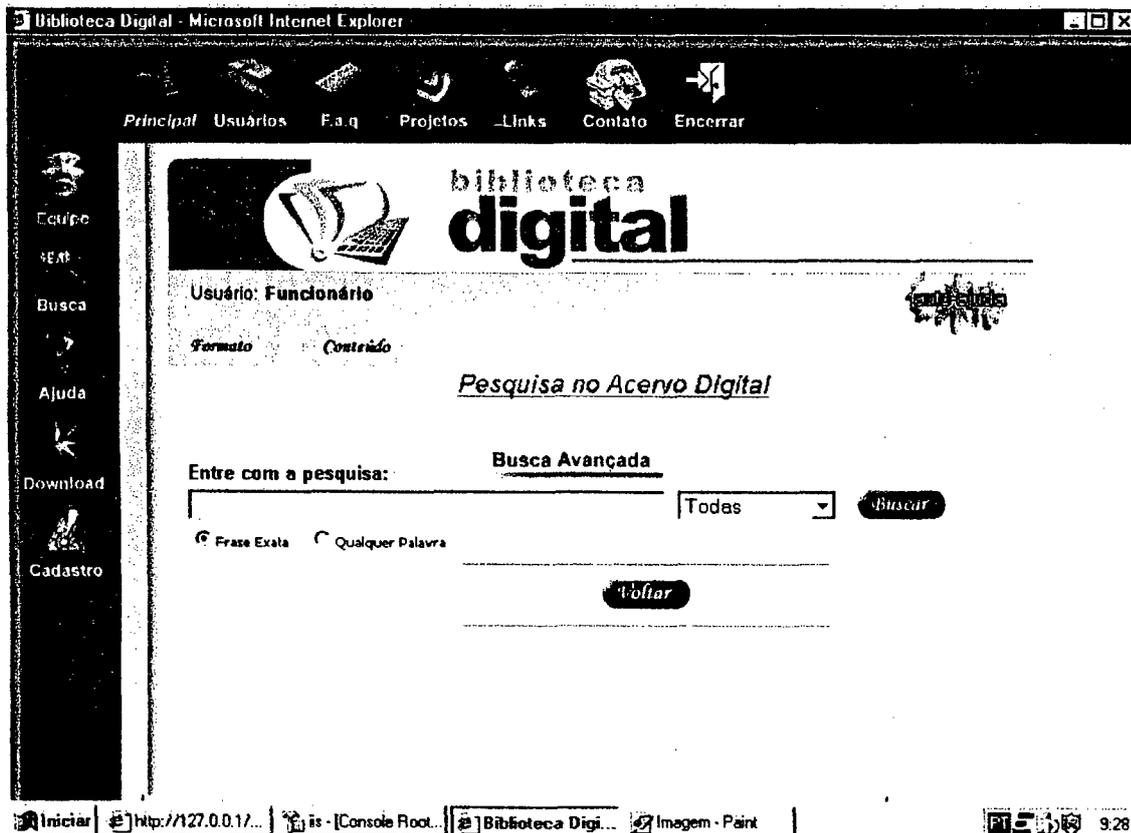


Figura 24 . Opções de acesso do módulo base (Funcionário)

Neste módulo, o funcionário poderá manter os seguintes itens:

- Formato - local onde incluiremos o formato das mídias, por exemplo: PPT, PS, DOC e outros.
- Conteúdo - local onde será feita a manutenção de todo o conteúdo das bibliotecas.

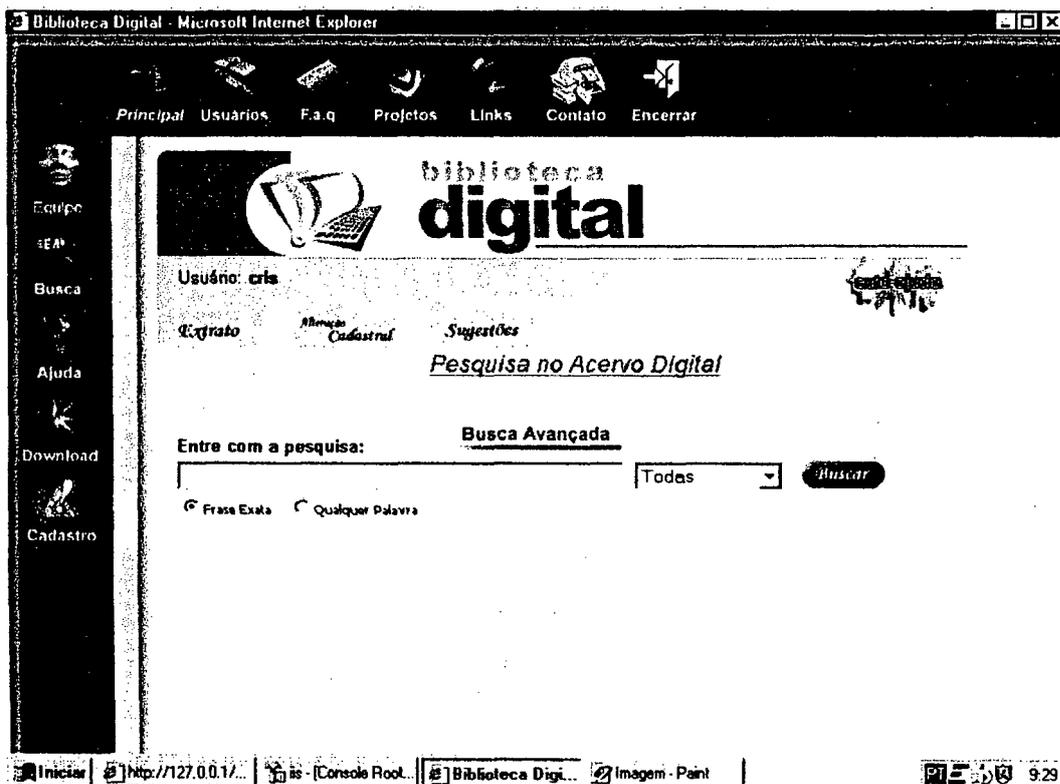


Figura 25 . Opções de acesso do usuário

Neste módulo, o usuário poderá acessar os seguintes itens:

- Extrato – nesta opção, o usuário consegue controlar todo o seu acesso à biblioteca digital, desde os dias e a hora da conexão e desconexão, até o total de tempo de acesso.
- Alteração cadastral – caso o usuário queira alterar sua senha.
- Sugestões – através desta opção pode-se enviar um *e-mail* para reclamações, sugestões, etc.

Uma vez autorizado a utilizar o sistema, todos os níveis de usuário poderão realizar consultas parametrizadas sobre a base de dados.

Os procedimentos iniciais para um consulta simples na base são expostos na Figura 26:

Pesquisa no Acervo Digital

Entre com a pesquisa: **Busca Avançada**

Todas Buscar

Frase Exata  Qualquer Palavra

Figura 26 . Pesquisa geral na base de dados multimídia.

- Para fazer uma consulta simples basta digitar as palavras a serem procuradas e especificar se a busca deve ocorrer através de "frase exata" ou "qualquer palavra";
- Selecionar a área de pesquisa ou não; e
- Clicar no botão "Buscar".

Desse modo, será feita uma procura nas bases de dados que estejam de acordo com o critério de pesquisa e o indexador da pesquisa será por padrão "o título" e, como resultado, veremos uma lista de itens ordenados pela sua relevância. Caso o usuário deseje fazer uma pesquisa avançada, conforme Figura 28, basta selecionar o botão de busca avançada. Esta pesquisa consiste de vários parâmetros, onde cada um deles pode interferir no resultado da pesquisa. Para fazer uma busca avançada é necessário proceder da seguinte maneira:

Pesquisa no Acervo Digital (Busca Avançada)

Entre com a pesquisa:

Frase Exata Buscar

Título Resultado/Página: 5

<p><b>Material:</b></p> <p><input checked="" type="checkbox"/> Todos</p> <p><input type="checkbox"/> Apresentação <input type="checkbox"/> Áudio</p> <p><input type="checkbox"/> Imagem <input type="checkbox"/> Livro</p> <p><input type="checkbox"/> Vídeo</p>	<p><b>Bibliotecas:</b></p> <p><input checked="" type="checkbox"/> Todas</p> <p><input type="checkbox"/> Biblioteca Geral</p>	<p><b>Áreas:</b></p> <p><input checked="" type="checkbox"/> Todas</p> <p><input type="checkbox"/> Administração</p> <p><input type="checkbox"/> Informática</p>
--	--	---

Figura 27 . Pesquisa avançada na base de dados multimídia

Segue abaixo os parâmetros gerais para se fazer uma busca avançada na biblioteca digital:

- Alguns dos parâmetros seguem a consulta geral; subentende-se então que o usuário já efetuou a referida pesquisa.
- É possível pesquisar por vários tipos de mídias, por exemplo, imagens, áudio, vídeos e outros. Vale salientar que, caso não sejam escolhidas uma ou mais opções, a pesquisa será realizada em todas as mídias.
- O usuário poderá também selecionar em qual biblioteca fará a consulta, sendo que a opção padrão sempre será a pesquisa em todas bibliotecas.
- O sistema também disponibiliza em qual área será feita a pesquisa. Como padrão estarão selecionadas todas as áreas.
- Também é permitido que o usuário escolha o indexador para a pesquisa. O sistema permitirá os seguinte indexadores:
  - Título
  - Localização
  - Assunto
  - Descrição
  - Autor
  - Todos
- Será feito o controle de número de itens por página, onde o usuário poderá escolher entre, 5,10,20 e 50. O número de itens padrão por página é 10.

Após feita a pesquisa, o usuário receberá o resultado nesta forma:

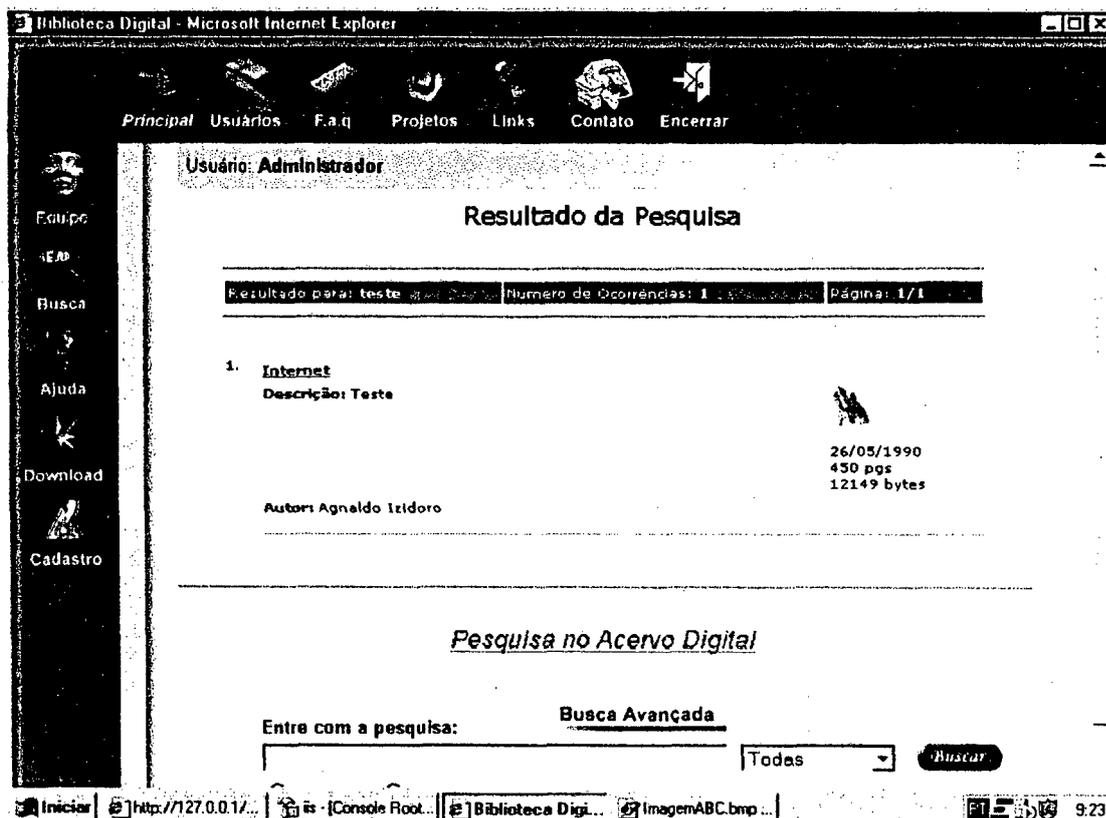


Figura 28 . Resultado da Pesquisa

A pesquisa terá como resultado os campos, título, descrição, autor, data da digitalização, tamanho do arquivo em bytes e o tipo de software que abre esta mídia. Caso o usuário deseje conhecer mais detalhes da pesquisa basta clicar no campo "título", o qual possui um *link* para detalhes desta consulta, conforme a Figura 29. Caso ele deseje ver o conteúdo<sup>53</sup> é necessário que se clique na imagem, conforme a Figura 30.

<sup>53</sup> Estamos falando do conteúdo interno de, por exemplo, um livro, que constará de sumário ou de um vídeo, que mostrará sua execução.



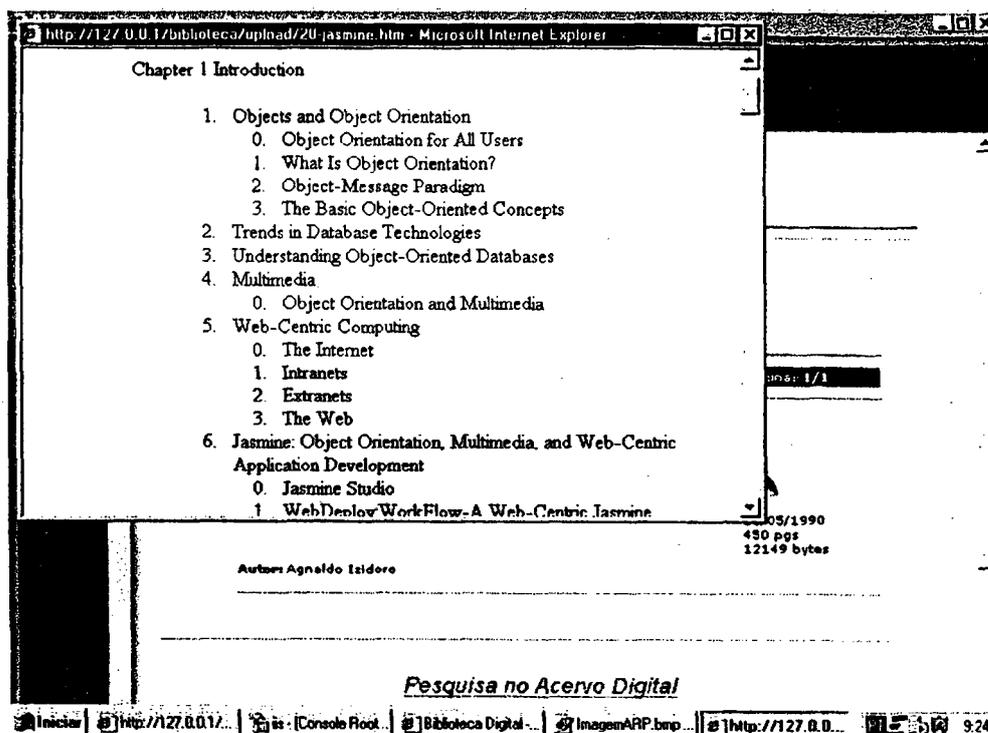


Figura 30 . Verificando o conteúdo da mídia.

Todas as páginas da aplicação, com exceção das páginas estáticas de ajuda *on-line* são geradas dinamicamente pelos programas correspondentes. Cada programa recebe um conjunto de parâmetros inseridos pelo usuário em formulários HTML via ASP.

A aplicação está disponível na *Internet* e pode ser usada a partir de qualquer *browser* que suporte HTML versão 2.0 ou superior e a linguagem *JavaScript*.

### 7.3 Desenvolvimento

A aplicação foi desenvolvida então em um ambiente ASP, cujo objetivo principal é o de possibilitar ao usuário fazer a consulta na base da biblioteca digital. Podemos destacar as seguintes considerações durante o desenvolvimento:

- A implementação da aplicação foi mais complexa do que em aplicações convencionais de Banco de Dados. Isto se deve ao fato de que, além da lógica da aplicação, adicionou-se a lógica da apresentação da interface (HTML) e a lógica de validação de dados.

- A transferência de variáveis de entrada do cliente *Web* por formulários para as consultas em SQL no servidor de Banco de Dados não é simples. Este é um ponto particularmente importante, em virtude do fato de que a transferência de dados de um formulário HTML para a aplicação ASP via ODBC é bastante complexa.

As seguintes considerações para a fase de manutenção também merecem ser destacadas:

- Qualquer manutenção na aplicação vai exigir que o desenvolvedor conheça detalhes de como funcionam as tecnologias envolvidas: a linguagem HTML (inclusive formulários), o ambiente ASP e a conexão com o Banco de Dados.
- Qualquer modificação na interface ou na validação dos dados implica em mudanças no código do programa respectivo, mesmo que a lógica do programa correspondente permaneça inalterada.

#### 7.4 Segurança

Esta aplicação só pode, em princípio, ser usada por usuários autorizados. O mecanismo utilizado para garantir este nível de segurança baseou-se na verificação do *login* e senha, solicitados na primeira página da aplicação, pois é comparado com uma tabela de usuários do banco de dados. No entanto, podemos apontar várias falhas de segurança da aplicação como as descritas a seguir:

- O *login* e senha solicitados na página inicial da aplicação são enviados sem proteção pela rede até o servidor *Web*;
- O usuário não poderá digitar a senha mais que cinco vezes errada, senão mecanismos do sistema travarão o acesso somente liberando-o após o contato com o funcionário da biblioteca.
- Todos os demais dados transmitidos na rede podem ser interceptados por usuários não autorizados, já que não foi utilizado nenhum mecanismo seguro para a comunicação cliente *Web* e servidor *Web*.

- Pessoas que não são usuárias da aplicação poderão executá-la se conhecer a URL com os parâmetros que devem ser passados ao programa que pretendem executar. A partir daí, a aplicação torna-se ativa a partir das opções presentes nas páginas da aplicação.

Os problemas de segurança descritos anteriormente poderão ser contornados utilizando outros mecanismos de segurança, conforme descrito no capítulo 5. De uma maneira geral, tentou-se controlar estas falhas, por exemplo, controlando o acesso à URL do usuário sempre redirecionando para a página principal.

### 7.5 Concorrência

Todo banco de dados tem que garantir a concorrência da informação, por exemplo, quando dois usuários distintos solicitam uma informação ao *SGBDs*, duas conexões distintas são estabelecidas e o *SGBDs* isoladamente garante a concorrência ao acesso dos dados durante a execução das operações solicitadas pelos programas.

No entanto, alguns problemas de concorrência, particularmente de isolamento dos dados podem ocorrer:

- Usuários com acesso a cadastramento ou exclusão de informações estão atualizando ou excluindo dados da base e simultaneamente outros estão tentando fazer uma consulta.
- Existe o problema de que o usuário que está utilizando o sistema sempre o faz sob um único *login*. Isto implica que existe somente um usuário, que concorre consigo mesmo o tempo todo no *SGBDs*. Esta opção só é interessante quando todos os usuários têm o mesmo nível de segurança aos objetos do *SGBDs* e não há concorrência direta entre os dados armazenados no Banco de Dados.

### 7.6 Consistência

Toda transação deve deixar o *SGBDs* num estado consistente ao final de sua execução. No entanto, devido à possibilidade do usuário voltar a uma página de interação anterior via navegação *back/forward* dos *browsers*, a aplicação pode apresentar um comportamento não desejado.

O principal problema foi encontrado nas páginas que geravam atualizações no Banco de Dados. Isto porque nas interações que geram consultas a informações no Banco de Dados não há problemas, como perda de consistência das informações armazenada decorrentes da navegação *back/forward*. No máximo, a consulta é re-submetida ao Banco de Dados, o que pode gerar uma carga maior de trabalho no sistema como um todo.

No caso de atualizações, o problema pode ser muito sério. Por exemplo, em algumas páginas de inclusão de novas informações os usuários poderiam submeter a página e, logo em seguida, retornar à mesma página de inclusão e re-submetê-la, duplicando a informação no Banco de Dados. Para evitar esta inconsistência, o programa que trata a página submetida sempre verifica, antes da inclusão, se a informação já existe no Banco de Dados com base em cláusulas SQL. Esta solução evitou alguns estados indesejados passíveis de ocorrer em virtude da navegação *back/forward*.

O outro problema de inconsistência não resolvido decorrente da navegação dos *browsers* está relacionado à opção *stop* dos *browsers*, que permite cancelar uma solicitação enviada ao servidor *Web* ainda em andamento. Assim, mesmo que o usuário submeta operações ao *SGBDs*, ele pode em seguida selecionar a opção *stop* dos *browsers*. Este cancelamento, no entanto, não se reflete no *SGBDs* já que, uma vez submetida a página da operação, todas as transações associadas serão efetivadas pela aplicação no *SGBDs*. Se o usuário seleciona a opção *stop* durante a execução de uma solicitação, ele não vai receber a resposta de que a operação solicitada foi ou não realizada com sucesso, já que o *browser* sempre ignora toda resposta do servidor *Web* após a seleção da opção *stop*.

Uma maneira de tentar contornar essa situação foi tentando deixar a biblioteca digital em tela cheia, tirando esta barra de ferramenta onde estariam estes botões.

## 7.7 Recuperação

Outro problema de difícil solução na aplicação implementada está relacionado com as possíveis falhas de sistema que podem ocorrer e como tratá-las. De

fato, se o usuário submete uma página da aplicação e ocorre uma falha no *browser*, na rede, no servidor *Web*, na aplicação ou no *SGBDs*, ele não terá condições de saber se sua operação foi ou não bem sucedida. É retornada ao cliente *Web* uma mensagem genérica do ambiente *Web* que não especifica onde ocorreu a falha e se a falha ocorreu antes ou depois da operação ter sido submetida.

## 7.8 Portabilidade

A aplicação está implementada utilizando a plataforma *Windows*, portanto, só funcionará utilizando um servidor *Web* que implemente esta tecnologia conforme já explanado neste trabalho.

## 7.9 Problemas estruturais

Os principais problemas estruturais que foram encontrados para a construção desta biblioteca são os seguintes:

- Acesso e Recursos: nem todos possuem acesso às facilidades da *Web*.
- Custo: dentro da instituição de ensino talvez o estudante possa ter acesso liberado à Internet. Entretanto, o acesso de casa deve ser feito através de conexões a um provedor de serviços e de ligações telefônicas, cobradas em função do tempo de duração.
- Necessidade de Treinamento: embora a *Web* apresente interface bastante familiar, ainda assim alguns alunos e professores necessitarão de treinamento, quer seja nos recursos básicos da *Web* como em interfaces específicas do curso.
- Infra-estrutura, Suporte e Administração: bibliotecas digitais requerem infra-estrutura, suporte e procedimentos administrativos. A instituição mantenedora da biblioteca pode enfrentar problemas se não conseguir providenciar tal infra-estrutura.
- Estágio Tecnológico: a tecnologia *Web* é bastante nova, não estando ainda totalmente estabilizada. A banda passante das redes é ainda limitada, sendo em geral insuficiente para aplicações que envolvam vídeo. Não existem ainda mecanismos reguladores que permitam

confiar em *sites* de terceiros, o que significa que todo o material essencial do curso deve ser (eventualmente replicado) armazenado em *site* próprio. Documentos produzidos com facilidades específicas de um *browser* podem ser impedidos de manipulação por outros *browser*.

- *Copyright*: Se para mídias tradicionais os direitos autorais têm sido difíceis de ser garantidos, para a *Web*, que é extremamente dinâmica, o problema certamente é bem maior.

## Conclusão

Esta arquitetura em questão apresenta sérias deficiências no que diz respeito à manutenção das propriedades ACID no ambiente de integração, além de não permitir o gerenciamento eficiente do estado da aplicação e de não ser segura. Neste sentido, aplicações *Internet* que requeiram atualizações no Banco de Dados devem possuir outros mecanismos de segurança tanto em nível de comunicação, quanto em nível de atualização e recuperação de informações.

## CONCLUSÕES FINAIS

A integração WWW e SGBDs visando o desenvolvimento de Bibliotecas Digitais, são áreas de pesquisa que ainda estão em pleno desenvolvimento. A diversidade de arquiteturas para integrar e a difícil resolução dos problemas, aliada aos problemas de direitos autorais e à dificuldade de acesso, têm dificultado a construção de sistemas de bibliotecas digitais capazes de atender aos principais requisitos e funcionalidades a que a mesma se propõe.

O estado atual das pesquisas indica não haver um conhecimento suficiente sobre determinados aspectos para a sua construção, devido à complexidade de alguns pontos (alguns discutidos neste trabalho e outros não), dentre os quais podemos destacar: digitalização, acesso, armazenamento, transação, recuperação de falhas, atomicidade, concorrência, segurança, portabilidade, interface e outros.

A contribuição desta dissertação para elucidar uma pequena parte destes problemas foi a de fazer uma abordagem genérica utilizando os conceitos de *frameworks*, delimitando as questões que abrangem as etapas para a construção de uma biblioteca digital, pois esta aplicação tem algo razoavelmente comum: pertencem a um mesmo domínio do problema, conforme já explanado nesta dissertação. Para tanto, foram levantadas as tecnologias que possibilitam esta construção, salientado as vantagens e desvantagens de cada uma delas, além de seu funcionamento. As questões referentes à banco de dados multimídia foram tratadas de maneira específica tentando levantar um banco que consiga amenizar os problemas que ocorrem em bancos de dados comuns.

Pudemos ainda levantar algumas vantagens em relação às bibliotecas digitais:

- Dados digitais são fáceis de armazenar e distribuir;
- Grande quantidade de dados pode ser facilmente pesquisada e analisada;

- A velocidade com que a informação digital é transmitida faz com que diminua a importância de onde ela está armazenada.

Por fim, o estudo de caso analisado no capítulo 7 possibilitou identificar elementos que levam a um conflito com a tendência no desenvolvimento de aplicações *Internet* baseadas em Banco de Dados: a arquitetura na qual estas aplicações têm se baseado apresenta diversos problemas funcionais e não soluciona os principais problemas decorrentes da integração das tecnologias.

### **Contribuições deste trabalho**

#### **As principais contribuições desta dissertação são:**

- Levantamento dos principais problemas referentes à biblioteca digital multimídia, banco de dados e tecnologia, seus problemas e soluções para sua implementação.
- Apresentação de um conjunto de requisitos funcionais de tecnologia que fazem a integração *Web* Banco de Dados, dando ênfase às que rodam do lado do servidor. Os poucos trabalhos que abordam requisitos para o desenvolvimento de aplicações *Web* Banco de Dados não tratam a questão sob o ponto de vista funcional, como proposto aqui.
- Elaboração de uma análise funcional qualitativa das principais arquiteturas de integração *SGBDs* e *Web* existentes atualmente.
- Descrição dos fundamentos básicos relacionados à *Web* e ao ambiente de integração segundo uma intensa pesquisa de trabalhos relacionados no que diz respeito ao desenvolvimento de aplicações *Internet*. O levantamento bibliográfico sobre o assunto é extenso e é ainda muito difícil encontrar artigos técnicos sobre o assunto.
- Modelagem de uma proposta para implementação de uma construção de uma biblioteca digital satisfazendo os conceitos de distribuição, acesso, segurança, dentre outros.

## Trabalhos Futuros

Considerando que a área de pesquisa envolvendo a construção de bibliotecas digitais juntamente com a integração *Web* e Banco de Dados é relativamente nova, muitas questões técnicas ainda devem ser estudadas para que essa aplicação se torne funcional e operacional, a fim de que possa ser utilizada de forma intensa. Em função disso, vários outros trabalhos podem ser desenvolvidos nesta área com o objetivo de qualificar e disponibilizar Bibliotecas Digitais em aplicações *Internet* para o seu funcionamento. Neste sentido, os trabalhos futuros decorrentes diretamente desta dissertação poderão ser os seguintes:

- O estudo de desempenho quantitativo das arquiteturas de integração existentes. Sabe-se que o desempenho da arquitetura é dependente de vários fatores como tráfego na rede, capacidade do servidor *Web* para atender solicitações, capacidade para gerenciar a aplicação, funcionalidades como abrir e fechar a conexão com o Banco de Dados e otimização de consultas sob o Banco de Dados, entre outros. Entretanto, não se sabe ainda qual é de fato o impacto individual de cada um desses fatores no tempo de resposta final da consulta. Alguns estudos já procuram estabelecer métricas que indicam quais são, na prática, os fatores que afetam mais significativamente o desempenho no ambiente de integração.
- Especificação e desenvolvimento de uma linguagem própria para o desenvolvimento de bibliotecas digitais utilizando todo o poder dos *SGBDs* que possibilitem realizar consultas e atualizações de dados via *Web* de forma independente do esquema do Banco de Dados e de mecanismos tradicionais.
- Os sérios problemas da interface HTML (para modelagem, impressão e formatação de tabelas de dados) ainda são um fator limitante para que a *Web* se torne de fato uma plataforma viável para desenvolvimento de diversas aplicações de Banco de Dados. Pesquisas devem ser direcionadas a resolver estes problemas de interface.

- Requisitos de segurança especialmente direcionados para aplicações *Web* Banco de Dados ainda não estão claros. Questões como o quê, por que e como controlar devem ser melhor direcionadas. Em particular, devem ser direcionados requisitos de segurança no ambiente *Web* para melhor garantir o acesso, manutenção e distribuição das informações armazenadas em *SGBDs*.
- Seguindo uma linha de trabalho de ponta identificada em diversas pesquisas de Banco de Dados atuais, parece ser promissor a investigação de um modelo de transações mais adequado especificamente ao ambiente de integração *Web* e Banco de Dados, no que diz respeito ao desenvolvimento de aplicações.
- Criar mecanismos de busca que pesquisem em dados multimídia, utilizando os campos *Blobs*, que já estão sendo utilizados neste trabalho para armazenar imagens.

### **Alguns limites deste trabalho**

Alguns dos principais limites deste trabalho já foram destacados no decorrer desta dissertação, contudo vale ainda salientar os seguintes aspectos:

- Por ser um sistema totalmente genérico, ou seja, todas as inclusões na base são tratadas dinamicamente, o sistema não tem como prover uma interface com outras bases de dados.
- Por ser uma questão altamente complexa o sistema não está tratando a questão da pesquisa no objeto multimídia, deixando esta para trabalhos futuros, pois todas as consultas estão sendo feitas apenas em metadados.

Para finalizar, se deve salientar que, embora todas as tecnologias envolvidas neste trabalho sejam independentes e, de certa forma, tenham sido projetadas em momentos diferentes para atuarem em áreas distintas, os frutos decorrentes dos experimentos que as integram já apresentam resultados promissores, o que justifica a continuidade do estudo e da pesquisa no tema desenvolvido nesta dissertação.

## REFERÊNCIAS BIBLIOGRÁFICAS

ADJEROH, Donald A. e NWOSU, Kingsley C., "*Multimedia Database Management – Requirements and Issues*". IEEE Multimedia, v. 4, n.3, Jul-Set. 1997, p.25-33.

ALLAIRE Corporation: "ColdFusion user's guide", 1997.  
<http://www.allaire.com/products/coldfusion/20/userguide/index.htm>

APACHE Corporation: "Apache - HTTP Server Project", 1997.  
<http://www.apache.org/docs/>

ARMS, Willian. J. *Library Digital*. D-lib Magazine. Cambridge, 2000.  
<http://www.cnri.restpm.va.us/home/dlib/>

BERGHEL.H, *The client's side of the World-Wide Web*", Communications of the ACM, Vol. 39(1), 1996, pp. 30-40. pesquisado em 15/11/2000

BERNERS..L.Les, "Uniform Resource Locators (URL)", RFC 1738, University of Minnesota, 1997.

BERNSTEIN, Terry. *Segurança na Internet*. Editora Campos, Rio de Janeiro, p.148 – 173. 1997.

BEZY, Michel. *Digital libraries for large multimedia collections*.1997  
<ftp://www.software.ibm.com/is/dig-lib/dlip.htm>

BORGES, Karem Selbach; OLIVEIRA, João Batista. *Ensino à distância, Bibliotecas Digitais e os Direitos Autorais*. Congresso da SBC. Rio de Janeiro. Jul. 1999.

BRAD, Baker. *MultiMedia and Imaging Databases*, Morgan Kaufmann Publishers, Editor. 1999.

BREMMER .Lynn, IASI.Antonio F. , *A biblia da Internet*, Makron Books, 1997.

BUYENS, Jim. *Web Database Development: Step by Step*. Microsoft Press, Bk&Cd edition. Mar, 2000.

BUZATO L. E. e RUBIRA C. M. F., **Construção de Sistemas Orientados a Objetos Confiáveis**, DCC/IM, COPPE Sistemas, NCE/UFRJ, Rio de Janeiro, 1998.

CRUMIS, Cristian, *Explorando a Internet*, Makron Books, 1999.

DATE, C.J. *Introdução a Sistemas de Bancos de Dados*. Campus, Rio de Janeiro, 1990  
4ª. edição.

DAVID, Michael M., - "*Multimedia Databases Throug the Looking Glass*". Database Programming and Design Magazine, Mai 1997.

DRUCKER, Peter. *Os novos paradigmas da administração*. Exame. São Paulo : Editora Abril, ano 32, n. 4, p.34-64, fev. 1999.

ELMASRI.R e NAVATHE S. B., "*Fundamentals of Database Systems*", Tree Edition. The Benjamin/Cummings Publishing Company, Inc. 2000.

FILHO,O .Marques. NETO,H.Vieira. *Processamento digital de Imagens* , Rio Janeiro. Brasport, 1999.

FURHT,B. *Multimedia Systems: Na Overview*, IEEE Multimedia, pp. 47-59, Spring 1994.

GONÇALVES, Marcus. *Segurança na Internet*, Axcel Books, 1997.

GONCALVES, Marcus. *Segurança na Internet*. Axcel Books, 1997.

GRAHAM, Peter S. *Requirements for the Digital Research Library*.\_College & Research Libraries, p. 331-339, Jul. 1995.

GRAY. J , *Principles of Multimedia Databases System*, Morgan Kaufmann Publishers, Editor. 1998.

GRAY. J , REUTER.A, "*Transaction Processing: Concepts and Techniques*", Morgan Kaufmann Publishers, Inc. 1993.

IBM Corporation: "**DB2 World Wide Web Connection**", 1997.  
<http://www.software.ibm.com/data/db2/db2wgafs.html>

KAHN, Robert; WILENSHY, Robert. *A framework for distributed digital object services*. 1995. <http://www.cnri.reston.va.us/home/cstr/arch/k-w.html>

KHOSHAFIAN. Setrag, *Banco de dados Orientado a Objetos*, Rio Janeiro, Infobooks, 1994.

KORTH. H. F, SILBERCHATZ.A, *"Database System Concepts"*, Second Edition, McGraw-Hill, 1997.

LESK, Michael. *Practical Digital Libraries: Books,Bytes, and Bucks*. Morgam Kaufmann.1997.

LEWIS, Ted. *Object-Oriented Application Frameworks*, Manning Publications, Greenwich, CT, 1995.

MARQUES, F.B.C. & Kelner, J. *Um Paradigma de Autoria para o Sistema de Autoria em Multimídia (SAM)*, [www.propesq.ufpe.br/anais/ccen/ccen65.htm](http://www.propesq.ufpe.br/anais/ccen/ccen65.htm).

MAYER. Marlyn, BABER,Robert. *Computer in your future*, Prentise Hall College, 1999

MICROSOFT Corporation: *"Internet Server API Reference"*, 1997. <http://microsoft.com/win32dev/apiext/isapiref.htm>.

MINOLI, Daniel; SCHIMIDT, Andrew. *Internet Architectures*. John Wiley & Sons, Jan. 1999.

NETSCAPE Communications Corporation: *"Netscape Communications Server. A WWW Server"*, 1997. [Http://www.netscape.com/comprod/server\\_central/index.html](http://www.netscape.com/comprod/server_central/index.html).

ORACLE Corporation: *"Web Request Broker API Oracle WebServer 3.0"*, 1996.<http://www.olab.com/datasheets/wrbapi.pdf>.

PAZANDAK, Paul e SRIVASTAVA, Jaideep. - *"Evaluating Object DBMSs for Multimedia"*. IEEE Multimedia, v. 4, n.3, Jul-Set. 1997, p.34-49.

PEREIRA, M. *Bibliotecas Virtuais: Realidade, possibilidade ou alvo de sonho*. Brasília. Abril, 1995, p.10 – 100.

PREE, Wolfgang. **Design Patters for Object - Oriented Software Development**. Addison - Wesley, 1995.

PRESMMAM. S. Roger. *Engenharia de Software*, Sao Paulo, Makron Books, 1995.

RAHMEL.D, "*Systems and Methodologies for Identifying and Protecting Weak Spots in Your Web-Enabled Database*", *Internet Systems*, April, 1997. <http://www.dbmsmag.com/9704i03.html>

REED. P., "*Using ODBC to access nontabular data*", *DBMS On-line Magazine*, April, 1996. <http://www.dbmsmag.com/9604d54.html> .

REICHARD. K, "*Web servers for database applications*", *Internet Systems*, October, 1996. <http://www.dbmsmag.com/9610i08.html>

ROGERS, Gregory F. **Framework - Based Software Development in C++**. New Jersey: Prentice-Hall, 1997.

ROSENKRANTZ, Daniel. **System Level Concurrency Control For Distributed Database System**, University of New York, ACM. 1998.

RUSS, John C. . *The image processing handbook*. 2 ed., Boca Raton, CRC Press, 1995.

SHERRY, Marcus. *Multimedia Database System*. University of Maryland, 1998.

SILBERCHATZ. Abrahan, SUDARSHAM. S, *Sistema de Banco de Dados*, São Paulo, Makron Books, 1999.

SMEULDERS, Arnold. JAIN, Ramesh. *Image Database na Multimedia Search*, 1998.

STERLING, Software. **Component Based Development and Object Modeling**. 1997 em hyperlink <http://www.cool.sterling.com/cbd/whitepaper.html>

SUB. Henrike, WEGENER. Klaus, *Interoperability Architectures for distributed Multimedia Database Systems*, Dresden, 1998.

TRANGER, Irving. **Transactions and Consistency in Distributed Database Systems**, IBM Research Laboratory, ACM. 1999.

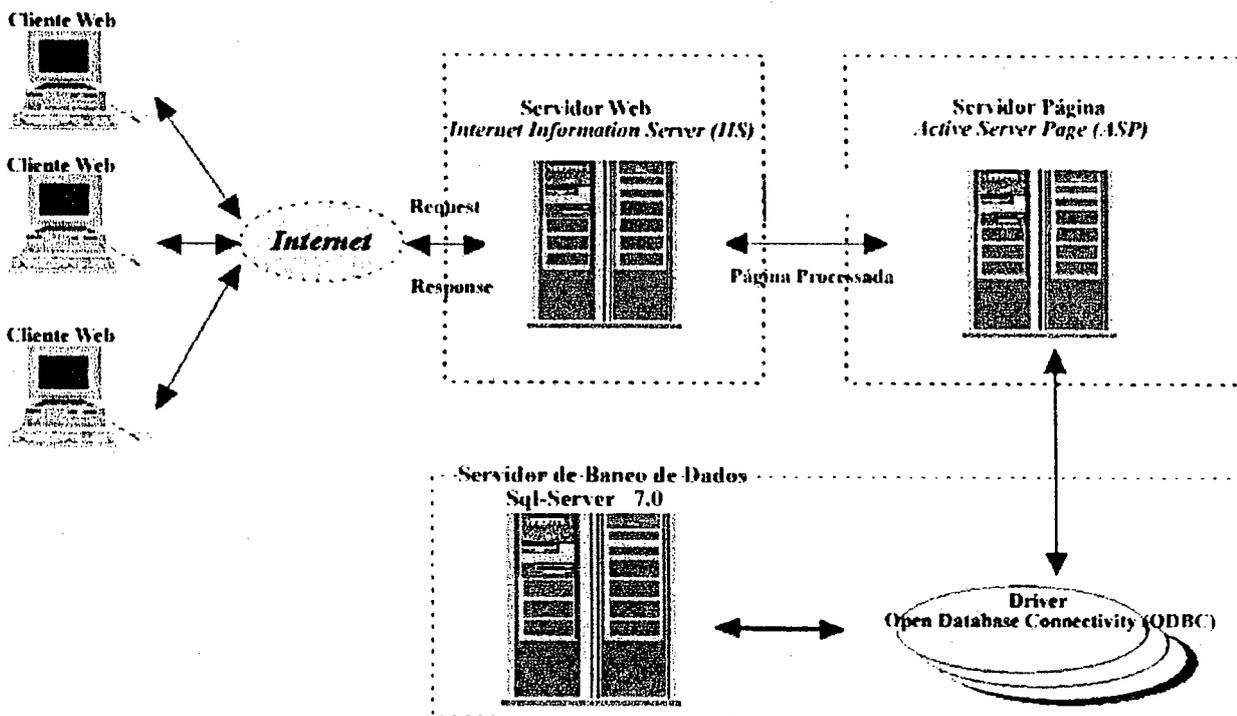
WEGENER, Klaus Meyer. *Multimedia Databases: Integrated Storage and Retrieval of Text, Imagens, Sound, and Video*, IEEE Multimedia, Ago-Set. 1997.

WOOD, Chuck. *Ole DB and Odbc developer*. IDG books Worldwide, Bk&Cd edition. Sep, 1999.

YEAGER.N, MCGRATH.R, "*Web Server Technology - The advanced guide for World Wide Web Information Providers*", Morgan Kaufmann Publishers, Inc., 1996.

# Apêndice A

## Estudo de caso - Ambiente para construção de bibliotecas digitais



# Apêndice B

## Diagrama de Entidade e Relacionamento

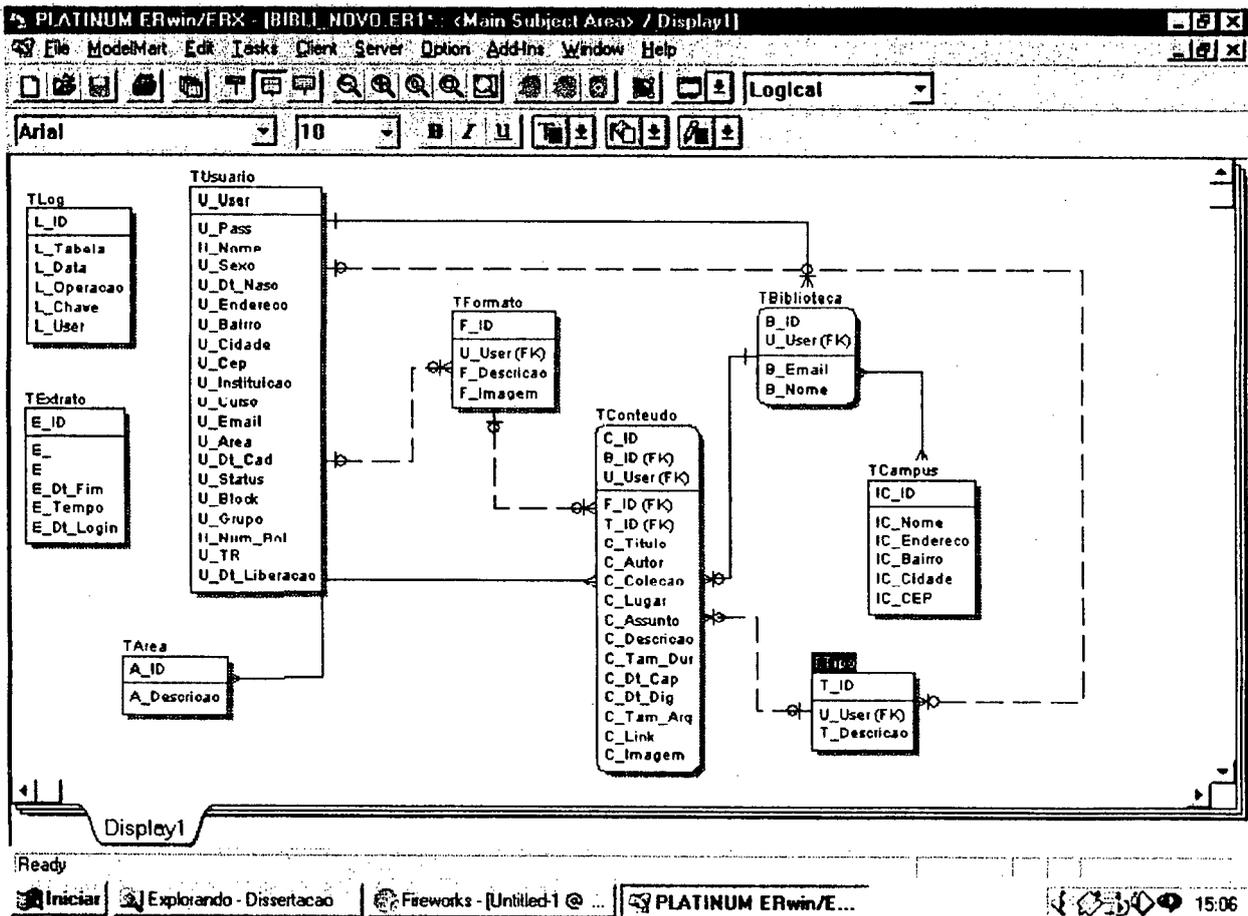


Diagrama de Use Case

