

Fernanda Possato Ribeiro

**UM GERADOR AUTOMÁTICO DE PÁGINAS NA INTERNET PARA PROFESSORES DA
UNIVERSIDADE FEDERAL DE LAVRAS USANDO XML**

Monografia de Graduação apresentada ao Departamento de
Ciência da Computação da Universidade Federal de Lavras
como parte das exigências do curso de Ciência da
Computação para obtenção do título de Bacharel.

LAVRAS
MINAS GERAIS - BRASIL

2005

Fernanda Possato Ribeiro

**UM GERADOR AUTOMÁTICO DE PÁGINAS NA INTERNET PARA PROFESSORES DA
UNIVERSIDADE FEDERAL DE LAVRAS USANDO XML**

Monografia de Graduação apresentada ao Departamento de
Ciência da Computação da Universidade Federal de Lavras
como parte das exigências do curso de Ciência da
Computação para obtenção do título de Bacharel.

Área de Concentração:

Banco de Dados

Orientador:

Profa. Olinda Nogueira Paes Cardoso

LAVRAS
MINAS GERAIS - BRASIL

Ficha Catalográfica

Ribeiro, Fernanda Possato

Um Gerador Automático de Páginas na internet para os professores da Universidade Federal de Lavras usando XML / Fernanda Possato Ribeiro. Lavras – Minas Gerais, 2005. 41p : il.

Monografia de Graduação – Universidade Federal de Lavras. Departamento de Ciência da Computação.

1. Banco de Dados. 2. Página Pessoal. 3. Gerador Automático. 4. XML. I. RIBEIRO, F. P. II. Universidade Federal de Lavras. III. Um Gerador Automático de Páginas na internet para os professores da Universidade Federal de Lavras usando XML.

Fernanda Possato Ribeiro

**UM GERADOR AUTOMÁTICO DE PÁGINAS NA INTERNET PARA PROFESSORES
DA UNIVERSIDADE FEDERAL DE LAVRAS USANDO XML**

Monografia de Graduação apresentada ao Departamento de
Ciência da Computação da Universidade Federal de Lavras
como parte das exigências do curso de Ciência da
Computação para obtenção do título de Bacharel.

Aprovada em 27 de junho de 2005.

Prof. Heitor Augustus Xavier Costa

Prof. Reginaldo Ferreira de Souza

Profa. Olinda Nogueira Paes Cardoso
(Orientadora)

LAVRAS
MINAS GERAIS - BRASIL

Dedico este trabalho a meus pais, que sempre me conduziram por bons caminhos e sempre me apoiaram em momentos difíceis.

Agradecimentos

Agradeço a Deus por mais essa experiência de vida, a meus pais, meu irmão, meus amigos, meus colegas de sala, meus professores (em especial a Olinda, pela ajuda no desenvolvimento deste trabalho) e a todos que estiveram comigo nesta caminhada.

Resumo

Um gerador automático de Páginas na internet para professores da Universidade Federal de Lavras usando XML

O uso da *Web* vem crescendo muito nos últimos anos, pois oferece aos seus autores a facilidade de distribuir seus documentos eletrônicos para um grupo de usuários cada vez maior e com baixo custo. Os documentos da *Web* estão se tornando cada vez mais complexos e os usuários começam a sentir uma grande dificuldade na manipulação desses documentos e a falta de conhecimento suficiente para sua criação. O objetivo deste trabalho é preencher a lacuna entre os recursos que a universidade oferece tais como internet e e-mail, e a não utilização destes recursos pelos professores, ou seja, o trabalho busca sanar o problema dos professores que não possuem uma Página Pessoal, que acham de grande importância o uso da mesma, mas que não possuem conhecimento ou tempo suficiente para o seu desenvolvimento. A contribuição foi criar um Gerador Automático de Páginas na internet para os professores da Universidade Federal de Lavras, onde o único trabalho do professor é preencher um formulário com os dados que gostaria de disponibilizar na Página e fazer o cadastro. Esses dados são armazenados em um Banco de Dados dos professores; a partir destes dados e da utilização da linguagem XML, as Páginas Pessoais de cada professor foi gerada.

Palavras-chaves: Banco de Dados, Página Pessoal, Gerador Automático, XML.

Abstract

An automatic generator of web pages for professors of the Federal University of Lavras using XML

The use of the Web has been increasing in the last years, because it offers to its authors the easiness to distribute its electronic documents for a group of users in constant growth, and with low cost. The documents of the Web are each time more complex and the users start to feel a great limitation in the manipulation of these documents and the lack of enough knowledge for their creation. The main problem to be decided is to fill the gap between the resources that the university offers and the lack of their use by the professors, in other words, the work tries to solve the problem of the professors who do not possess a Personal Home Page, even though they find its use as of great importance, but they do not possess knowledge or enough time for its development. The contribution was to create an Automatic Generator of Pages in the internet for the professors of the Federal University of Lavras, where the only work for the professors is to fill in a form with the data that will be put available in the Page, and then, to make a register in cadastre. These data are stored in a Data base for the professors; out of these datas and the use of language XML, the Personal Pages of each professor were mounted.

Key Words: Data base, Personal Home Page, Generating Automatic, XML.

SUMÁRIO

LISTA DE FIGURAS	i
LISTA DE TABELAS	i
RESUMO ESTENDIDO.....	ii
1 INTRODUÇÃO	1
1.1 Considerações Iniciais	1
1.2 Objetivos.....	2
1.3 Escopo do Trabalho.....	3
2 REFERENCIAL TEÓRICO	4
2.1 Internet e a World Wide Web.....	4
2.2 Banco de Dados na Web.....	6
2.3 Dados Semi-Estruturados	7
2.4 XML - <i>eXtensible Markup Language</i>	9
2.4.1 HTML vs. XML.....	11
2.4.2 Principais benefícios da Linguagem XML	14
2.4.3 XSL - <i>eXtensible Stylesheet Language</i>	16
2.4.4 Aplicações da XML	17
2.4.5 XML e Bancos de Dados	18
3 METODOLOGIA	20
3.1 Procedimento Metodológico.....	20
3.2 PHP e MySQL	21
3.3 Desenvolvimento	23
4 RESULTADOS E DISCUSSÕES	25
5 CONCLUSÕES	30
6 REFERÊNCIAS BIBLIOGRÁFICAS	31

LISTA DE FIGURAS

Figura 2.1: Documento HTML versus documentos XML	14
Figura 2.2: Múltiplas apresentações a partir de um mesmo documento XML.....	16
Figura 2.3: Usando XSL para exibição de dados XML	27
Figura 4.1: Formulário.....	26
Figura 4.2: Página Principal.....	27
Figura 4.3: Página de Disciplinas.....	28
Figura 4.4: Página de <i>Links</i>	28.

LISTA DE TABELAS

Tabela 2.1: Diferenças entre dados tradicionais e dados semi-estruturados	9
---	---

RESUMO ESTENDIDO

É de grande importância para um professor universitário possuir uma Página Pessoal, tanto para uso dos alunos como para sua divulgação na internet. O uso da *Web* vem crescendo muito nos últimos anos, pois oferece aos seus autores a facilidade de distribuir seus documentos eletrônicos para um grupo de usuários cada vez maior e com baixo custo. Os documentos da *Web* estão cada vez mais complexos e os usuários começam a sentir uma grande dificuldade na manipulação desses documentos e a falta de conhecimento suficiente para sua criação. O objetivo deste trabalho é preencher a lacuna entre os recursos que a universidade oferece e a não utilização pelos professores, ou seja, o trabalho busca sanar o problema dos professores que não possuem uma Página Pessoal, que acham de grande importância o uso da mesma, mas que não possuem conhecimento ou tempo suficiente para o seu desenvolvimento. O objetivo é facilitar o trabalho do professor ao construir sua Página Pessoal através de um Gerador Automático de Páginas na internet, onde o único trabalho do professor é preencher um formulário com os dados que gostaria de disponibilizar na Página e se cadastrar. O referencial teórico relaciona-se com os conceitos básicos utilizados para construção do Gerador Automático de Páginas, como por exemplo, as linguagens XML, PHP e o Banco de dados MySQL. A metodologia utilizada envolveu a pesquisa bibliográfica, a pesquisa documental e o método do estudo de caso, uma vez que foram utilizados dados referentes aos professores da Universidade Federal de Lavras. Esses dados foram armazenados em um Banco de Dados dos professores; a partir destes dados e da utilização da linguagem XML, as Páginas Pessoais de cada professor foram geradas. O resultado obtido foi satisfatório, tendo o projeto sido concluído com sucesso. A contribuição foi criar o Gerador Automático para facilitar para os professores a construção e manutenção de sua Página Pessoal. São propostos como trabalhos futuros os detalhes finais do projeto, tais como a melhoria da parte estética e da interface, a construção de um sistema de recuperação de informação, dentre outros.

1 INTRODUÇÃO

1.1 Considerações Iniciais

A utilização do ambiente *World Wide Web*, ou simplesmente *Web*, vem crescendo consideravelmente nos últimos anos, devido principalmente à habilidade que oferece aos seus autores de distribuir, facilmente e com baixo custo, seus documentos eletrônicos para um grupo de usuários cada vez maior. Os documentos da *Web* estão se tornando cada vez mais complexos e os usuários começam a sentir uma grande limitação na manipulação desses documentos e a falta de extensibilidade e intercâmbio com outras aplicações e plataformas.

Atualmente, a *Web* tem se caracterizado como um dos maiores mecanismos de disseminação de informações no mundo. A crescente utilização deste veículo de comunicação e o conseqüente crescimento do volume de informações disponíveis tornam necessária a aplicação de conceitos da tecnologia de Banco de Dados para tornar possíveis os processos de gerenciamento e utilização destas informações.

De acordo com Buyens (2001), o aumento do volume e da estrutura dos dados apresentados na *Web* justificam seu armazenamento e organização em um banco de dados, para depois então gerar as páginas *Web* com base nesses bancos de dados. É grande a quantidade de *sites* da *Web* que podem se beneficiar dessa metodologia. Qualquer *site* da *Web* que apresente informações sobre uma coleção de itens semelhantes é candidato ao uso de um banco de dados *Web*. Esses itens semelhantes poderiam ser produtos para venda, dados pessoais, *hyperlinks*, dentre outros. Em vez de criar e manter uma página *Web* individual para cada pessoa, lugar ou coisa, o objetivo é colocar a parte variável em um Banco de Dados e codificar uma única página *Web* para exibir o que se quer.

Se os seus dados puderem residir em um banco de dados, podem-se escrever páginas *Web* para inserí-los, mantê-los e distribuí-los.

Segundo Silveira (2004), um dos principais pontos a serem observados é que os dados contidos em páginas da *Web*, em sua maioria, são formados por textos, figuras, arquivos de áudio e vídeo, sendo estes dados considerados semi-estruturados. Dados semi-estruturados são aqueles que não possuem uma estrutura fixa e pré-definida, como nos bancos de dados estruturados tradicionais. Nestes dados existe alguma estrutura implícita, que não é evidente, mas que pode ser extraída. Desta forma, torna-se mais complexo o

processo de gerenciamento de dados semi-estruturados em bancos de dados convencionais. Juntamente com as técnicas de banco de dados, outras tecnologias visam igualmente ajudar na tarefa de gerenciamento de dados semi-estruturados, entre as quais podem ser citadas a inteligência artificial e a hipermídia.

Segundo Mello (2003), o amplo uso da XML (*eXtensible Markup Language*) como um formato de representação e transferência de dados entre computadores justifica as pesquisas atuais da comunidade de banco de dados nesta área.

O atual cenário na Universidade Federal de Lavras (UFLA) demonstra que existe uma grande utilização dos recursos da *Web* de maneira isolada, ou seja, tanto os professores quanto os alunos utilizam a *Web* para realizar pesquisas relacionadas ao ensino, pesquisa e extensão na UFLA, porém não existem páginas próprias dos professores para este fim. O problema que pode ser observado é que os recursos relacionados à internet na UFLA não estão sendo bem aproveitados.

1.2 Objetivos

O objetivo geral deste trabalho é realizar a modelagem e o desenvolvimento de um banco de dados dos professores da Universidade Federal de Lavras, que será utilizado por um gerador automático de páginas para internet, utilizando XML, para posterior desenvolvimento de uma página pessoal para cada professor.

Para tanto, tem-se os seguintes objetivos específicos:

- 1) levantamento das necessidades dos professores e informações que justificassem o desenvolvimento do trabalho;
- 2) modelagem e implementação do banco de dados dos professores da UFLA;
- 3) transformação dos dados contidos no banco de dados dos professores em documentos XML;
- 4) geração, a partir dos documentos XML, de diversas páginas para os professores;

O problema resolvido foi preencher a lacuna entre os recursos que a Universidade Federal de Lavras oferece, tais como, uso de internet e e-mail, e a não utilização destes recursos pelos professores. Foram desenvolvidas páginas na *Web* para professores da Universidade Federal de Lavras, usando XML.

1.3 Estrutura do Trabalho

O Capítulo 2 deste trabalho apresenta o Referencial Teórico, para que haja um melhor entendimento dos conceitos deste projeto, como: Internet e a *World Wide Web*, Banco de Dados na *Web*, Dados Semi-Estruturados, XML - *eXtensible Markup Language*, HTML vs. XML, Principais benefícios da Linguagem XML, XSL - *eXtensible Stylesheet Language*, Aplicações da XML, XML e Bancos de Dados.

O Capítulo 3 apresenta a Metodologia, ou seja, a forma como o trabalho foi desenvolvido, dividido em etapas que foram seguidas para sua conclusão.

O Capítulo 4 apresenta os Resultados e Discussões acerca do trabalho realizado. É mostrada a solução final apresentada e como foi desenvolvido.

No Capítulo 5, é feita a Conclusão.

2 REFERENCIAL TEÓRICO

2.1 Internet e a World Wide Web

A internet está mudando o mundo visivelmente. Nos últimos anos, a internet passou a ser essencial e a desempenhar um papel central em vários ramos da atividade humana. Atualmente, as páginas da *Web*, que segundo Silberschatz *et al.* (1999), é um sistema de informação distribuído baseado em hipertextos, são um elemento necessário da infraestrutura empresarial e também costumam fazer parte da vida privada das pessoas, tornando-se praticamente incontável a quantidade de informações disponíveis na internet (Tolentino, 2004).

Segundo Date (2003), a internet evoluiu a partir da *Arpanet*, que foi um projeto do final da década de 1960 sob o patrocínio da Agência de Projetos de Pesquisa Avançada do Departamento de Defesa dos EUA (DARPA), para conectar todas as diversas redes do governo e acadêmicas nos Estados Unidos, em uma única e consistente “sub-rede”, com um protocolo de comunicação comum, chamado TCP/IP (*Transmission Control Protocol/Internet Protocol*).

Segundo Tolentino (2004), com o crescimento extremamente expressivo da internet como meio de troca de dados e exibição de informações, instituições de todos os portes têm procurado deixar disponíveis na rede mundial sistemas de informação através dos quais possam interagir com seus parceiros comerciais, clientes e/ou fornecedores.

Segundo Date (2003), os termos *Web* e internet são usados como se fossem intercambiáveis, mas eles têm significados diferentes. Pode-se caracterizar a diferença da seguinte forma: a *Web* é um banco de dados gigantesco (embora não criado com os princípios convencionais de banco de dados); a internet é igualmente uma rede gigantesca sobre a qual esse banco de dados é distribuído. O acesso a *Web* não é o único serviço fornecido pela internet - outros incluem leitores de notícias (*newsreaders*), mensagens instantâneas, correio eletrônico (*e-mail*), *ftp*, *telnet* etc. A internet sem a *Web*, ainda não era tão integrada como poderia ser; os usuários ainda tinham de usar vários mecanismos diferentes, como diferentes tipos de correio eletrônico, para acessar informações. A *Web* foi inventada por Tim Berners-Lee em 1989-1990 como uma base para resolver toda essa complexidade. A noção central é a do hipertexto, inventado vários anos antes por Ted Nelson.

O usuário de um sistema *Web* enxerga textos formatados ao lado de imagens. O texto formatado com imagens é visualmente mais atrativo do que o texto puro. Além disso, o documento é mostrado como um documento hipertexto, que de acordo com Date (2003), é um meio de estruturar informações, permitindo que documentos de texto referenciem outros documentos e arquivos, ou componentes de outros documentos e arquivos, por meio de *links* embutidos, que com um programa navegador apropriado, o usuário pode dar um clique em uma região que possui um *link* associado e o documento apontado pelo *link* será mostrado. Assim, a interface hipertexto da *Web* é uma interface de navegação poderosa e visualmente atrativa, como afirma Silberschatz *et al.* (1999).

De acordo com Date (2003), a grande contribuição de Berners-Lee foi implementar essas ligações em um *browser* (navegador) gráfico que pudesse integrar todos os diferentes tipos de informações em uma única janela; o resultado foi que os usuários poderiam acessar e exibir qualquer coisa que quisessem por meio de um único clique do *mouse*, em vez de ter que usar todos os comandos e procedimentos separados que precisavam usar antes. Ele conseguiu essa simplificação notável definindo:

- um mecanismo, *Uniform Resource Locator* (URL, mais tarde generalizado para *Uniform Resource Identifier*, ou URI), para identificar e referenciar documentos e outros recursos;
- uma linguagem de marcação, HTML (*HyperText Markup Language*), para criar documentos que incluem instruções sobre como eles devem ser exibidos;
- um protocolo, *Hypertext Transfer Protocol* (HTTP), para transmitir esses documentos pela internet.

Segundo Date (2003), a *Web* é um banco de dados gigantesco. Os usuários percebem esse banco de dados, por meio de um *browser Web* como sendo distribuído por um conjunto de *sites* (*Web sites*), cada um com seu próprio servidor *Web* e identificado por seu próprio URL. Cada *site* contém um conjunto de páginas *Web* e cada página possui um documento raiz associado, que especifica como essa página deve ser exibida. Como todos os documentos, o documento raiz normalmente inclui *links* de URL para vários outros tipos de informações (texto, imagens, áudio, vídeo), em diversos *sites* diferentes; porém para o usuário, ele é percebido como um único documento integrado.

O usuário geralmente conhece o URL da página original, no máximo, e nada mais, mas, quando a página é exibida, os *links* também são exibidos, e se o usuário clicar em um

deles, o *browser* mostrará a informação correspondente na mesma janela ou em alguma janela adicional (Date, 2003).

Segundo Date (2003), as informações em determinado *site* podem ser armazenadas em arquivos do sistema operacional; porém, cada vez mais, elas são armazenadas em banco de dados e os servidores *Web*, portanto, precisam ser capazes de interagir com os Sistemas Gerenciadores de Banco de Dados (SGBDs).

2.2 Banco de Dados na Web

De acordo com Buyens (2001), o uso da *Web* para apresentar e acumular dados tem crescido muito além da simples exibição de uma ou outra página.

Segundo Date (2003), as vantagens adicionais de usar um banco de dados na *Web* são:

- muito mais fácil manter do que todas as páginas *Web* individuais;
- facilita a procura de itens desejados;
- facilita a apresentação dos mesmos dados de maneiras diferentes: por categorias, por descrição, por idade, ou com base em qualquer outro campo do banco de dados.

Date (2003) afirma que dispor de interfaces de banco de dados para a *Web* é importante por duas razões. Primeira, com o crescimento do comércio eletrônico na *Web*, os bancos de dados utilizados para processamento de transações precisam estar ligados a *Web*. A segunda razão é que as fontes de dados HTML fixas são limitadas em relação à apresentação aos usuários.

Conforme afirma Silveira (2004), a visão da *Web* como um grande *site* é de vital importância, não sendo apenas um banco de dados, mas sim um sistema de informações construído em volta de vários bancos de dados.

Segundo Buyens (2001), a primeira etapa para usar qualquer banco de dados é definir suas tabelas e campos. A segunda etapa é implementar essa estrutura usando um Sistema Gerenciador de Modelagem Conceitual e de Implementação de banco de dados. A terceira e quarta etapas destinam-se a adicionar e a consultar dados, que são totalmente dependentes das duas primeiras etapas.

De acordo com Date (2003), virtualmente todo sistema de banco de dados aceita comandos de aplicativos em um formato chamado de SQL (*Structured Query Language*). A maioria das páginas de banco de dados *Web* constrói suas próprias instruções SQL, freqüentemente incorporando opções ou valores de busca recebidos do visitante *Web*. A SQL é a linguagem padrão para lidar com banco de dados relacionais e é aceita por quase todos os produtos existentes no mercado.

Segundo Silveira (2004), do mesmo modo que os *Web sites* provêm informações, pode-se também aplicar técnicas de Sistemas de Banco de Dados no processo de construção e manutenção de tais *sites*.

2.3 Dados Semi-Estruturados

De acordo com Silveira (2004), várias são as diferenças entre os documentos encontrados na *Web* e um banco de dados. Na maioria dos documentos da *Web*, não há estruturas uniformes, modelos de dados, restrições de integridade, transações, nem mesmo uma linguagem padrão para consultas. Por estas razões, vários são os trabalhos desenvolvidos na área de banco de dados com o objetivo de possibilitar o tratamento de dados semi-estruturados.

Conforme Mello *et al.* (2000), os dados semi-estruturados apresentam uma representação estrutural heterogênea, não sendo nem completamente não-estruturados nem estritamente tipados.

Mello *et al.* (2000) afirma também que dados semi-estruturados são dados nos quais os esquemas de representação está presente (de forma explícita ou implícita) juntamente com o dado, ou seja, o mesmo é auto-descritivo. Isto significa que uma análise do dado deve ser feita para que a sua estrutura possa ser identificada e extraída.

As principais características dos dados semi-estruturados, de acordo com Abiteboul (1997), são:

- *definição à posteriori*: esquemas para dados semi-estruturados são usualmente definidos após a existência dos dados, com base em uma investigação de suas estruturas particulares e da análise de similaridades e diferenças. Isto não significa que sempre existe um esquema associado a um dado semi-estruturado;
- *estrutura irregular*: coleções extensas de dados semanticamente similares estão organizados de maneiras diferentes, podendo algumas ocorrências terem

informações incompletas ou adicionais em relação a outras. Em suma, não existe um esquema padrão para esses dados;

- estrutura implícita: muitas vezes existe uma estrutura básica para os dados, porém, essa estrutura está implícita na forma como os dados são apresentados. É necessário realizar uma computação para obter essa estrutura;
- estrutura parcial: apenas parte dos dados disponíveis pode ter alguma estrutura, seja implícita ou explícita. Como consequência, um esquema para estes dados nem sempre é completo do ponto de vista semântico e nem sempre todas as informações esperadas estão presentes;
- estrutura extensa: a ordem de magnitude de uma estrutura para estes dados é grande, uma vez que os mesmos são muito heterogêneos;
- estrutura evolucionária: a estrutura dos dados modifica-se tão frequentemente quanto os seus valores. Dados *Web* apresentam este comportamento, uma vez que existe o interesse em manter dados sempre atualizados;
- estrutura descritiva e não prescritiva: dado à natureza irregular e evolucionária dos dados semi-estruturados, as estruturas de representação implícitas ou explícitas normalmente se restringem a descrever o estado corrente de poucas ocorrências de dados similares;
- distinção entre estrutura e dados não é clara: como a estrutura está embutida na descrição dos dados, muitas vezes a distinção lógica entre estrutura e valor não é clara.

Segundo Mello *et al.* (2000), as características de dados semi-estruturados diferem bastante das características de dados mantidos em banco de dados tradicionais, como banco de dados relacionais. A tabela 2.1 apresenta estas diferenças.

Tabela 2.1: Diferenças entre dados tradicionais e dados semi-estruturados

Dados tradicionais	Dados semi-estruturados
Esquema predefinido	Nem sempre há um esquema predefinido
Estrutura regular	Estrutura irregular
Estrutura independente dos dados	Estrutura embutida nos dados
Estrutura reduzida	Estrutura extensa
Estrutura fracamente evolutiva	Estrutura fortemente evolutiva
Estrutura prescritiva	Estrutura descritiva
Distinção entre estrutura e dado é clara	Distinção entre estrutura e dado não é clara

Fonte: Mello *et al.* (2000).

Modelos de dados para banco de dados tradicionais não são adequados a dados semi-estruturados, uma vez que todas as ocorrências de dados devem apresentar a mesma estrutura. Assim, modelos de dados para dados semi-estruturados devem ser flexíveis no sentido de suportar representações heterogêneas de dados semanticamente iguais (Mello *et al.*, 2000).

A XML é uma linguagem de marcação para construção de páginas na *Web*, que, devido as suas características, facilita a extração de dados semi-estruturados.

2.4 XML - *eXtensible Markup Language*

De acordo com Date (2003), a XML foi desenvolvida originalmente em 1996 por um comitê de revisão da SGML sob o patrocínio do *World Wide Web Consortium*, W3C, que, segundo Deitel *et al.* (2003), foi fundado em 1994 para desenvolver protocolos comuns para a evolução da *Web*, com o objetivo de remediar certos problemas com a SGML (*Standard Generalized Markup Language*) e a HTML.

O problema com a SGML é que ela era simplesmente muito grande e complicada para ser admitida com facilidade na *Web*. Quanto a HTML, os problemas eram basicamente dois:

- ela não conseguia separar corretamente os dados estruturais, semânticos e de formatação;
- ela permitia que documentos violassem as regras de “boa formatação”, ou seja, ela permitia que eles desobedecessem a suas próprias regras de sintaxe.

XML é uma abreviação de *eXtensible Markup Language*. Segundo Ramalho (2002), a XML tem como finalidade marcar um determinado texto que sofrerá algum tipo de processamento. Em outras palavras, pode-se dizer que a XML é uma linguagem para criar dados estruturados baseados em um arquivo-texto. A estruturação dos dados deve-se ao fato de que ela permite identificar de forma inequívoca uma peça individual de informação.

XML é como afirma Mello *et al.* (2000), um padrão para publicação, combinação e intercâmbio de documentos multimídia. Assim, como outras linguagens de marcação, XML lida com instruções embutidas no corpo de documentos chamadas *tags* (palavras encapsuladas por sinais '<' e '>'), que permitem a descrição de dados. XML tem como base linguagens mais antigas como SGML e HTML, sendo atualmente empregada na representação de estruturas de dados estruturados e semi-estruturados e seu intercâmbio na *Web*.

O primeiro passo para entender XML é colocar de lado um preconceito comum: a crença de que a XML é algum tipo de super-HTML com novos recursos fantásticos de apresentação que todos os navegadores vão suportar de modo uniforme. Na verdade, os recursos de apresentação da XML são zero. A XML fornece apenas os dados; depois disso, é necessária outra tecnologia para ordenar, resumir, formatar, organizar ou fazer alguma coisa com os dados (Buyens, 2001).

De acordo com Buyens (2001), XML é um formato padrão para transferência de dados entre computadores. Diversos domínios de aplicação como comércio eletrônico e cadastro bibliográfico, vêm definindo protocolos XML para os seus dados e cada vez mais aplicações e pessoas disponibilizam e manipulam informações XML na *Web*. O uso cada vez mais intensivo de XML vem despertando o interesse de diversas áreas da ciência da computação, como Linguagens de Programação, Engenharia de Software e Bancos de Dados, todas elas preocupadas com a definição de mecanismos para o tratamento de dados neste formato.

Conforme Esteves *et al.* (2001), XML descreve o conteúdo e a estrutura lógica de como um conteúdo deve ser agregado, mas não a maneira como é representada e/ou deve ser apresentada no *browser*. Começou a ser usado para que documentos estruturados pudessem ser usados na internet.

A flexibilidade do XML provém da possibilidade de transportar qualquer tipo de dados, mantendo-os estruturalmente coesos e inteligíveis. Devido a esta estrutura, é também possível combinar em um mesmo documento, vários objetos com tipos de dados diferentes

(Esteves *et al.*, 2001).

Para a visualização e interpretação de documentos em XML é, por vezes, necessário utilizar a linguagem XSL (*eXtensible Stylesheet Language*). É essencial para quem pretende usar marcas nos documentos que disponibiliza na *Web* ou em uma rede interna (*Intranet*). É uma tecnologia única que agrupa a descrição de um documento e o seu conteúdo (Esteves *et al.*, 2001).

2.4.1 HTML vs. XML

Segundo Ramalho (2002), a XML foi criada para descrever dados, e o seu foco está na descrição do dado. A HTML, por sua vez, foi criada para exibir dados e seu foco é na aparência do dado. A linguagem XML não substitui a linguagem HTML, que continua sendo a principal linguagem usada pelos navegadores para exibirem informações na *Web*. A XML não é só o mais novo jargão da internet, como também é uma ferramenta que rapidamente ganhou maturidade neste domínio. É uma ferramenta extremamente poderosa no que diz respeito à gestão, organização e visualização de bases de dados textuais. A XML, usada em conjunto com a HTML, amplia as possibilidades de exibição de dados por intermédio de uma página *Web*, permitindo entre outras coisas:

- ordenar, filtrar, localizar e arranjar informações de maneira diferente;
- apresentar dados de forma estruturada.

Segundo Light (1999), HTML e XML operam em níveis diferentes de generalidade, então geralmente elas não estarão em competição direta. A HTML é uma aplicação da SGML, o que significa que ela fornece um grupo específico de tipos de elementos, com um objetivo particular: exibir páginas *Web on-line*, com *hiperlinks*. Por outro lado, a XML é um perfil SGML, o que significa que pode dar suporte a um intervalo ilimitado de aplicações, sendo algumas dessas aplicações do tipo HTML, mas a maioria terá objetivos e projetos muito diferentes.

Segundo Duarte *et al.* (2005), HTML e XML são primos. Eles derivam da mesma inspiração, o SGML. Ambos identificam elementos em uma página e ambos utilizam sintaxes similares. A grande diferença entre HTML e XML é que o HTML descreve a aparência e as ações em uma página na rede enquanto o XML não descreve nem aparência nem ações, mas sim o que cada trecho de dados é ou representa.

Segundo Silberschatz *et al.* (1999), a SGML fornece uma gramática para especificar

os formatos dos documentos baseados em marcações. A HTML é uma linguagem de apresentação de hipertexto de uso geral com base em SGML. Ela especifica um formato de documento que aceita comandos de formatação de textos, assim como comandos de *links* de hipertextos e comandos de apresentação de imagens. A HTML também fornece recursos limitados para entrada de dados.

Segundo Buyens (2001), a HTML foi projetada para controlar a aparência visual das informações, já a XML é total e exclusivamente projetada para transmitir dados estruturados de um computador para outro ou de um aplicativo para outro. A XML não propicia nenhuma informação visual. A apresentação visual, se necessária, fica totalmente sob controle e decisão do programa que recebe os dados. Com a XML você codifica os dados por atributos como estrutura, tipo da informação e formato de armazenamento. Uma vez os dados codificados no formato XML, um programa de formatação os converte em um documento legível. Resumindo, uma página *Web* torna-se um depósito de dados estruturados com uma camada de apresentação por cima.

Em traços muito gerais, o HTML é identificado por um conjunto de códigos que permitem representar graficamente um determinado documento, através de uma formatação baseada em um esquema de etiquetas (*tags*) que, posteriormente, são interpretadas pelos *browsers*. Tal como o HTML, o XML também utiliza etiquetas como `<title>` ou `<body>`. Na HTML é a própria linguagem que especifica o que cada etiqueta e atributos especificam; a XML usa as etiquetas apenas para delimitar blocos de dados e deixa a interpretação dos mesmos para a aplicação. Assim, em função do contexto programático, um `<p>` que em HTML significa sempre um novo parágrafo, pode em XML significar um preço, uma pessoa, ou o que se desejar (Esteves *et al.*, 2001).

Segundo Esteves *et al.* (2001), apesar de ser uma ferramenta fácil de usar e aprender, fato pelo qual se generalizou bastante o seu uso, verificam-se algumas desvantagens na utilização da HTML que se prendem com o fato de os dados estarem dependentes da sua apresentação gráfica, não fornecer conteúdo semântico e ter um número limitado de etiquetas.

Tolentino (2004) afirma que existem alguns problemas básicos com a própria HTML, tais como:

- há enorme quantidade de combinações possíveis delas, quase 100 marcações, e nem sempre produzem o mesmo resultado visual em cada diferente navegador;
- sempre surge a necessidade de uma nova marcação, dependendo do tipo de

aplicação;

- a despeito do crescimento de novas marcações na HTML, há uma tendência de se utilizar menos os navegadores e mais outros dispositivos, como telefones celulares e *palmtops*, para acessar os dados disponíveis na *Web*, e nem sempre esses dispositivos têm capacidade suficiente para interpretar uma linguagem que cresce tanto em complexidade;
- possui capacidade limitada para o intercâmbio de informações;
- não é capaz de distinguir entre dados computadorizados e a própria informação;
- pouca semântica pode ser extraída de um documento, levando à dificuldade de manutenção e compreensão.

Segundo Deitel (2003) a HTML não é uma linguagem procedural de programação como C, Fortran, Cobol ou Pascal. Diferentemente delas, é uma linguagem de marcação que serve para identificar os elementos de uma página de maneira que um navegador possa exibir aquela página na tela do computador.

Relativamente a HTML, a XML trás grandes vantagens na sua utilização. Além da possibilidade de definição de novas marcas e etiquetas à medida dos utilizadores e de um maior controle sobre a aplicação de formatações e da forma como estas aparecem nos *browsers*, o utilizador do XML fica com a certeza de publicar os seus documentos em um formato padrão (independente dos fabricantes de software), facilmente interpretável por múltiplas aplicações e, em certa medida, auto-explicativo (Esteves *et al.*, 2001).

Segundo Ramalho (2002), algumas diferenças e similaridades entre XML e HTML são:

- a XML é uma linguagem de marcação como a HTML;
- a XML tem como finalidade descrever os dados;
- a XML foi criada para transportar os dados;
- a XML não possui *tags* ou marcas predefinidas como HTML;
- enquanto a XML se preocupa em descrever os dados, a HTML se preocupa com a sua exibição.

Segundo Light (1999), a XML será mais interessante para pessoas e organizações que têm recursos de informação que não se encaixam no molde da HTML e recursos que eles queiram deixar disponíveis na *Web*. Uma possibilidade interessante é que as aplicações

XML podem ser usadas para melhorar as aplicações HTML.

A Figura 2.1 mostra diferenças de conteúdo, apresentação e estrutura entre documentos HTML e XML:

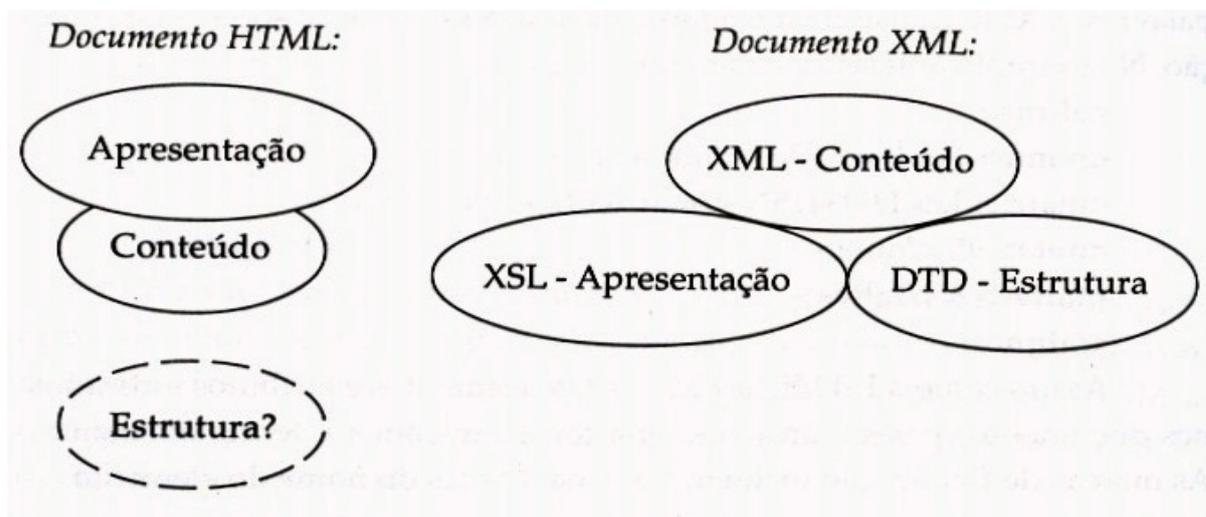


Figura 2.1

Documento HTML versus documentos XML

Fonte: Tolentino (2004).

2.4.2 Principais benefícios da Linguagem XML

Segundo Duarte *et al.* (2005), a XML tem por objetivo trazer flexibilidade e poder às aplicações *Web*. Dentre os benefícios para desenvolvedores e usuários têm-se:

- buscas mais eficientes: os dados em XML podem ser unicamente “etiquetados”, o que permite que, por exemplo, uma busca por livros seja feita em função do nome de autor;
- desenvolvimento de aplicações flexíveis para *Web*: o desenvolvimento de aplicações *Web* em três camadas é altamente factível com o XML. Os dados XML podem ser distribuídos para as aplicações, objetos ou servidores intermediários para processamento. Esses mesmos dados também podem ser distribuídos para o *desktop* para ser visualizado em um navegador;
- integração de dados de fontes diferentes: atualmente é praticamente impossível a procura em múltiplos bancos de dados e incompatíveis. O XML permite que tais dados possam ser facilmente combinados. Essa combinação seria feita via software em um servidor intermediário, estando os bancos de dados na extremidade da rede;

- computação e manipulação local: os dados XML recebidos por um cliente são analisados e podem ser editados e manipulados de acordo com o interesse do usuário. Ao contrário de somente visualizar os dados, os usuários podem manipulá-los de várias formas. Os recursos disponíveis no DOM permitem que os dados sejam manipulados via *scripts* ou outra linguagem de programação. A separação da interface visual dos dados propriamente ditos permite a criação de aplicações mais poderosas, simples e flexíveis;
- múltiplas formas de visualizar os dados: os dados recebidos por um usuário podem ser visualizados de diferentes formas uma vez que o XML define somente os dados e não o visual. A interpretação visual poderia ser dada de várias maneiras diferentes, de acordo com as aplicações. Os recursos de CSS (*Cascading Style Sheets*) e XSL (*Extensible Stylesheets Language*) permitem essas formas particulares de visualização;
- atualizações granulares dos documentos: os dados podem ser atualizados de forma granular, evitando que uma pequena modificação no conjunto de dados implique na busca do documento inteiro novamente. Dessa forma, somente os elementos modificados seriam enviados pelo servidor para o cliente. Atualmente, uma modificação em um item de dados acarreta na necessidade de atualização da página inteira. O XML também permite que novos dados sejam adicionados aos já existentes, sem a necessidade de reconstrução da página;
- fácil distribuição na *Web*: assim como o HTML, o XML, por ser um formato baseado em texto aberto, pode ser distribuído via HTTP sem necessidade de modificações nas redes existentes;
- escalabilidade: devido ao fato dos documentos XML separarem completamente os dados da forma com a qual são visualizados, autores de aplicações de visualização de dados podem torná-las muito poderosas e interativas, permitindo ao usuário visualizar os dados da forma que lhe agrada. Dessa forma, a interatividade, em termos, não dependeria tanto da comunicação cliente servidor, mas sim seria feita "*offline*", reduzindo o tráfego do *link* com o servidor;
- compressão: a compressão de documentos XML é fácil devido à natureza repetitiva das *tags* usadas para definir a estrutura dos dados. A necessidade de compressão é dependente da aplicação e da quantidade de dados a serem movidos entre clientes e servidores.

Segundo Tolentino (2004), uma grande virtude da XML está no fato de que ela separa o esquema de exibição do documento do seu conteúdo. Isso permite que um mesmo documento possa ser apresentado por meio de várias formas e dispositivos, como o monitor de um computador pessoal, a tela de um celular ou transformado em voz para utilização por deficientes visuais. A Figura 2.2 apresenta um exemplo:

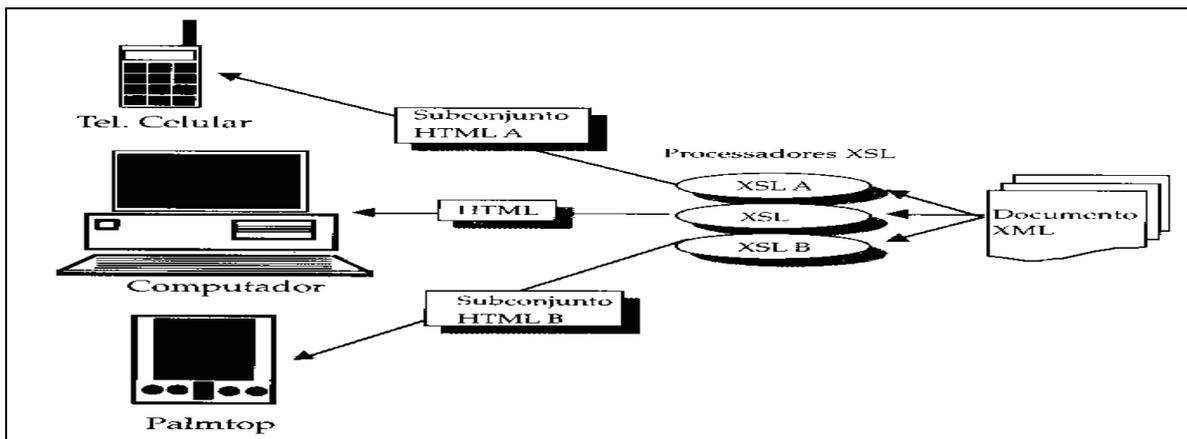


Figura 2.2: Múltiplas apresentações a partir de um mesmo documento XML

Fonte: Tolentino (2004).

2.4.3 XSL - *eXtensible Stylesheet Language*

Segundo Deitel *et al.* (2003), a XSL, *eXtensible Stylesheet Language*, é utilizada para formatar documentos XML para apresentação. A grande maioria dos recursos XSL é composta por objetos de formatação. Esses objetos são geralmente utilizados quando o resultado de uma transformação é para mídia impressa, como por exemplo, livros, revistas, etc. O documento XML é transformado em um documento XSL que marca os dados utilizando objetos de formatação. Esse documento XSL pode, então, ser transformado em outros formatos.

Como afirma Ramalho (2002), a XSL é uma linguagem de folhas de estilo, que são linguagens usadas dentro de um programa XML para definir o estilo, só que mais poderosa e mais difícil de aprender, mas com a mesma finalidade da CSS, que são folhas de estilo em cascata.

Segundo Tolentino (2004), a XSL é uma proposta em curso do W3C para uma linguagem extensível de folhas de estilo XML. Sua função principal é permitir aos usuários

escrever transformações de XML para HTML, descrevendo, assim, a apresentação do documento XML.

Um documento XSL, na verdade, é um documento XML que consegue transformar um documento XML em outros formatos de documentos, entre os quais o HTML. A linguagem XSL pode ser utilizada para acrescentar aspectos de apresentação aos elementos de um documento XML. Dessa forma, é possível criar múltiplas representações da mesma informação a partir de diferentes documentos XSL, aplicados a um único documento XML (Tolentino, 2004).

Segundo Tolentino (2004), a XSL oferece facilidades muito maiores para exibir um documento XML, por ter sido desenvolvida especificamente para a XML. Além de exibir essa flexibilidade, a XSL pode ser executada no servidor, ou seja, não exige que as transformações e todo o suporte necessário à linguagem XSL estejam implementados em um cliente. Assim, podem ser aplicadas transformações XSL complexas a um documento XML e o resultado ser uma página HTML simples, que possa ser exibida em um navegador de gerações anteriores, sem o suporte a XML. Neste caso, portanto, os dados são transferidos em formato HTML.

A figura 2.3 simboliza o processo:

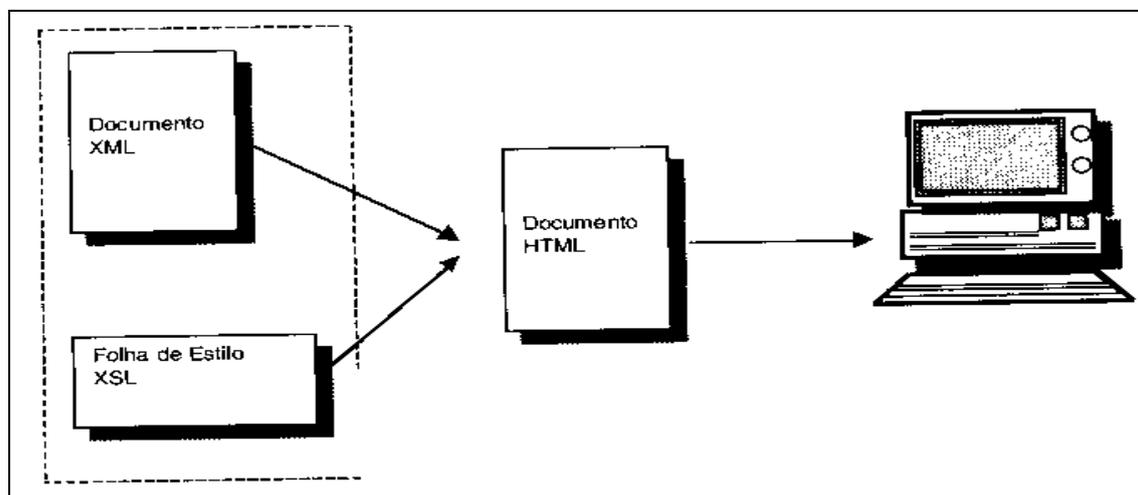


Figura 2.3: Usando XSL para exibição de dados XML

Fonte: Tolentino (2004).

2.4.4 Aplicações da XML

Segundo Esteves (2001), apesar de recente a XML tem grande implementação na área das

tecnologias de informação, marcando presença em grande parte dos sistemas baseados em tecnologias *Web* e que tenham sido implementados recentemente. Navegando pela *Web*, encontram-se freqüentemente exemplos de implementações de soluções XML, especialmente procurando-se por *sites* que realizem operações bancárias, leilões *on-line* ou pesquisa indexada de documentos.

Esteves (2001) afirma que um dos exemplos mais divulgados é o do comércio eletrônico. A maior parte das soluções de *E-Commerce* implementadas em ambientes *Web*, utilizam precisamente a XML. O objetivo destas aplicações é a integração de sistemas e tecnologias diversas, com apenas uma coisa em comum: a utilização da XML como linguagem de comunicação. Também no mundo empresarial esta tecnologia é utilizada, por exemplo, para realizar ações que se repetem periodicamente, como é o caso dos relatórios estatísticos, em que o relatório em si, só é produzido uma vez em XML, depois é visualizada no meio que for desejado, qualquer que ele seja.

No contexto das bibliotecas digitais, a utilização desta ferramenta é, seguramente, um passo importante em uma nova estruturação e organização do conhecimento e da difusão da informação (Esteves, 2001).

Diante das características apresentadas da linguagem XML, pode-se observar que esta é mais apropriada para o relacionamento entre bancos de dados e a internet, uma vez que esta, apesar de não ser completamente estruturada, apresenta um certo grau de estruturação que se aproxima dos bancos de dados tradicionais. Por este motivo optou-se por utilizar a linguagem XML na geração automática de páginas.

2.4.5 XML e Bancos de Dados

Buyens (2001) afirma que XML tem muito a ver com desenvolvimento de Banco de Dados na *Web*, considerando-se os seguintes fatos:

- a linguagem XML destina-se ao armazenamento e à transmissão de dados estruturados;
- os dados armazenados em banco de dados, por definição são estruturados;
- a *Web* propicia uma forma fantástica de armazenar e transmitir dados.

Existe um requisito de armazenar um documento XML em banco de dados e poder apanhar e atualizar esses dados conforme a demanda. E existe o caso contrário, apanhar

dados regulares, como o resultado de uma consulta, e transformá-los em formato XML (Buyens, 2001).

Conforme afirma Mello (2003), os protocolos XML são utilizados amplamente em aplicações desenvolvidas para a *Web* ou que utilizam a *Web* para a transferência de dados, no caso dos dados XML necessitarem ser armazenados em um banco de dados, existe a necessidade de gerenciar adequadamente estes dados. Devido a esta necessidade novos desafios surgem em decorrência, uma vez que a tecnologia de banco de dados precisa ser adaptada para garantir o armazenamento e a manipulação de dados XML de forma eficiente.

Um documento XML pode ser considerado uma coleção de dados, da mesma forma que um banco de dados. Porém, um dado XML possui uma natureza diferente de um dado convencional de um banco de dados, uma vez que um dado do banco de dados é um dado totalmente estruturado enquanto que um dado XML é um dado semi-estruturado. Em um dado tipicamente semi-estruturado, apenas parte de sua definição possui alguma estruturação e cada ocorrência de um mesmo dado possui um esquema particular e auto-descritivo. Desta forma, o esquema de representação destes dados é bastante irregular (Mello, 2003).

Segundo Mello (2003), um dado XML tem uma organização mais ou menos semi-estruturada definindo assim duas categorias de documentos:

- documento XML Orientado a Dados (*DOD*): documento que possui uma estrutura mais regular e repetitiva, sendo assim fracamente semi-estruturado. Documentos desta categoria são tipicamente usados para transferência de dados convencionais entre aplicações;
- documento XML Orientado a Documentos (*DODoc*): documento que possui uma estrutura mais irregular e particular, sendo assim fortemente semi-estruturado. Documentos desta categoria são tipicamente usados para o relato da linguagem natural, com destaque para alguns dados delimitados dentro do seu conteúdo.

3 METODOLOGIA

Este trabalho utiliza a metodologia de Pesquisa Bibliográfica, Pesquisa-Ação e é também um Estudo de Caso.

A Pesquisa Bibliográfica deu base para a aquisição de conhecimento acerca dos temas envolvidos no projeto. Envolveu, basicamente, consultas a livros de referências e a artigos científicos.

É uma Pesquisa-Ação, pois envolve os professores da Universidade Federal de Lavras (UFLA) e seus interesses. Segundo Lakatos (1983), a pesquisa-ação é realizada a partir do envolvimento dos pesquisadores e representantes da situação de maneira cooperativa e participativa na ação ou na resolução do problema. Esse envolvimento leva ao melhor conhecimento do objetivo da pesquisa e produz novas formas de conhecimento. Seu maior objetivo é solucionar problemas práticos, o que leva uma efetiva participação dos interessados.

É também um Estudo de Caso, pois se concentra nos professores da UFLA, fazendo uma exploração em profundidade. Conforme citado por Lakatos (1983), o objetivo do estudo de caso é estudar um ou poucos objetos, de maneira a permitir amplo e detalhado conhecimento. Sua vantagem é estimular novas descobertas, o estudo da multiplicidade das dimensões do problema e a simplicidade do procedimento.

3.1 Procedimento Metodológico

Foi realizada, inicialmente, uma pesquisa bibliográfica dos assuntos discutidos nas seções anteriores para adquirir um embasamento teórico. Foi necessário conhecer todas as ferramentas utilizadas no projeto, como por exemplo, a linguagem XML, HTML, PHP, o banco de dados MySQL, entre outros.

Além disso, foi necessário também, conhecer as necessidades e as dificuldades encontradas pelos professores da UFLA para construção da sua Página Pessoal.

Após a coleta de informações necessárias para o embasamento teórico, foi iniciada a parte prática do projeto realizando-se primeiramente a construção de formulários, utilizando a linguagem HTML e PHP, e posteriormente o armazenamento dos dados preenchidos no formulário em um banco de dados MySQL.

Depois dos dados armazenados no banco de dados, utilizou-se o PHP para gerar os

arquivos XML, que, com o auxílio de arquivos XSL, foram geradas as páginas dos professores da UFLA.

3.2 PHP e MySQL

PHP (*Personal Home Page*) é, segundo Vivas (2000), uma linguagem que permite criar páginas dinâmicas (que são aquelas geradas a partir da interação com bancos de dados ou outras fontes, permitindo que o conteúdo da página seja montado durante a execução da aplicação, podendo ser diferente a cada montagem, dependendo de opções definidas pelo usuário em outros *scripts* do *site*), tornando possível uma interação com os usuários através dos formulários, dos parâmetros da URL e dos *links* entre as páginas.

Segundo Cisneiros (2003), a linguagem PHP é uma linguagem de programação criada especialmente para o uso em páginas *Web*, mas nem por isso ela pode deixar de ser usada em ambientes *desktop*, aplicações servidoras, aplicações de rede, entre outros. Mas o principal escopo da linguagem é justamente trabalhar com o ambiente *Web*. O uso do PHP já é muito difundido na internet, uma boa quantidade de páginas dinâmicas é feita utilizando essa linguagem. Uma das melhores habilidades do PHP é lidar com bancos de dados de uma forma fácil. Hoje em dia, os sistemas para *Web* estão cada vez mais usufruindo das capacidades magníficas que os bancos de dados podem oferecer.

Segundo Vivas (2000), pode-se utilizar o PHP para diversas finalidades, como processar dados originados de um formulário, gerar páginas dinamicamente, etc. Além destas possibilidades, o PHP possui suporte nativo a diversos bancos de dados, além de interagir com esses bancos de dados para gerar páginas dinâmicas, tornando esse processo extremamente simples e rápido.

A escolha pelo PHP deve-se ao fato de, segundo Vivas (2000), ser uma das linguagens de programação mais utilizadas na *Web* para criação de páginas dinâmicas, além de possuir algumas características viáveis ao trabalho, tais como:

- gratuito e com código aberto;
- embutido no HTML;
- baseado no servidor;
- suporte a diversos bancos de dados;
- portátil.

Segundo Soares (2001), o MySQL é um banco de dados gratuito e muito poderoso, que nasceu da insatisfação de algumas pessoas com o gerenciador anterior.

Segundo Aroca (2001), hoje em dia, bancos de dados é parte fundamental da vida de quase todos os seres humanos. Viver sem sistemas de bancos de dados em nosso mundo hoje é impossível. Sem nem mesmo perceber, a todo o momento estamos usando um banco de dados, mesmo nas mais triviais tarefas. Normalmente, bases de dados com muitas informações são armazenadas em computadores de grande porte, chamados de servidores, e que permitem o uso das mesmas informações, através de uma rede, por um número ilimitado de usuários.

Aroca (2001) afirma que um dos mais rápidos programas para servidores de SQL, hoje no mercado, é o MySQL, desenvolvido pela T.c.X. DataKonsultAB. Além de oferecer vários recursos não existentes em outros servidores, o MySQL tem a vantagem de ser totalmente gratuito para uso tanto comercial, quanto privado. Alguns exemplos de organizações que dependem totalmente de banco de dados, como bancos, universidades e bibliotecas. A própria internet, usa um sistema de banco de dados para controle e funcionamento dos sites.

Para Stoco (2000), no mercado existem diversas tecnologias especializadas na integração de banco de dados com a *Web*. É muito difícil dizer qual das tecnologias seria a melhor. O grande anseio dos programadores está sendo atendido pela integração entre PHP e MySQL.

Segundo Stoco (2000), algumas vantagens do PHP e do MySQL são apresentadas abaixo:

PHP

- linguagem de script de fácil utilização e com grande poder de iteratividade;
- excelente velocidade de execução;
- ligação nativa com diversos bancos de dados.

MySQL

- número ilimitado de utilização por usuários simultâneos;
- capacidade de manipulação de tabelas com mais de 50.000.000 registros;
- velocidade de execução de comandos muito rápida, sendo este um dos mais rápidos do mercado;
- fácil e eficiente controle de privilégios de usuários.

Como afirma Stoco (2000), todas essas vantagens de ambas as tecnologias tornam-se, hoje em dia, uma das mais utilizadas para o desenvolvimento de páginas *Web* interativas, criação de *e-commerce*, criação de grandes portais, entre outras aplicações.

3.3 Desenvolvimento

Para alcançar os objetivos deste trabalho, as seguintes etapas da metodologia foram realizadas:

Etapa 1: realizou-se entrevistas com os professores levantando as suas necessidades e obtendo informações que justificassem a construção do gerador automático de páginas através de questionários respondidos pelos mesmos.

Através da entrevista foi possível detectar a carência de informações e conhecimento suficiente para que os próprios professores criassem suas páginas pessoais, além da falta de tempo para desenvolver a página e para poder aprender alguma ferramenta para construção da mesma.

Esses fatos justificam o desenvolvimento do gerador automático de páginas para os professores da UFLA, tendo essa idéia uma grande aceitação entre os mesmos.

Etapa 2: realizou-se a modelagem e implementação do formulário para que os professores preenchessem de acordo com suas necessidades.

O primeiro passo foi a construção de páginas de formulários na *Web* para que os professores fornecessem os dados que estariam disponíveis na página pessoal. Para a construção dos formulários foi utilizada a linguagem HTML e PHP, pois o HTML e o PHP podem ser intercalados dentro de um mesmo código.

Etapa 3: realizou-se o armazenamento dos dados preenchidos no formulário no banco de dados.

Para realização desta Etapa, foram utilizadas duas funções em PHP, uma para fazer a conexão com o banco de dados e a outra para fazer a inserção dos dados no banco de dados.

O banco de dados escolhido foi o MySQL pois é um banco de dados gratuito e

muito poderoso, além de ser um dos mais rápidos programas para servidores de SQL.

O banco de dados possui as seguintes tabelas:

PROFESSORES				
Código	Nome	Telefone	E-mail	Currículo

DISCIPLINAS						
Código	Nome	Objetivos	ConteúdoProgramático	Bibliografia	Horário	Avaliações

LINKS	
CódigoProfessor	Link

DISCPROFESSORES	
CódigoProfessor	CódigoDisciplina

MATERIALDISC	
CódigoDisciplina	MaterialDisciplina

Etapa 4: realizou-se a transformação dos dados contidos no banco de dados dos professores em diversos documentos XML.

Para realização desta Etapa, foi utilizado um arquivo em PHP que faz as consultas SQL no banco de dados transformando-as em arquivos XML.

Etapa 5: gerou-se a partir dos documentos XML, diversas páginas para os professores da UFLA.

Para realização desta Etapa, foi utilizado arquivos XSL para exibir a página XML do professor.

Etapa 6: realizou-se a análise dos resultados obtidos.

4 RESULTADOS E DISCUSSÕES

Neste Capítulo serão apresentados os resultados obtidos neste trabalho, mostrando passo a passo o desenvolvimento realizado, descrevendo suas Etapas.

Na Etapa 1, onde foi realizada a entrevista com os professores (o roteiro da entrevista encontra-se no Anexo A), pode-se observar que:

- todos os professores utilizam computador no seu dia a dia, além de utilizarem recursos como internet e e-mail;
- todos os professores consideram o uso dos mesmos de grande importância para a vida acadêmica;
- considerando os professores entrevistados do Departamento de Ciência da Computação, todos os professores possuem página pessoal. Considerando os professores entrevistados dos demais Departamentos da UFLA, poucos possuem página pessoal;
- os principais motivos alegados pelos professores para não possuírem uma página é a falta de tempo para a construção da página, falta de conhecimento suficiente e falta de tempo para aprenderem uma ferramenta para a construção da página;
- todos os professores acreditam que uma página pessoal facilita a comunicação entre o professor e o aluno, acham importante o professor ter uma página e que também é importante o uso da mesma pelos alunos;
- todos os professores consideram importante a construção de um gerador automático de páginas, acreditando assim que isso pode facilitar muito o trabalho do professor.

Na Etapa 2, onde foi realizada a modelagem e implementação do formulário, foi utilizado um código (apresentado no Anexo B) para a geração do mesmo, onde se pode observar que os cadastros dos professores são feito da seguinte forma: primeiro o professor preenche os campos referentes aos dados pessoais, depois os campos referentes às disciplinas ministradas e por último os *links* que gostaria de disponibilizar na sua página.

A Figura 4.1 apresenta o formulário como forma de ilustração.

The image shows a screenshot of a Microsoft Internet Explorer browser window. The title bar reads 'Cadastro de Professores - Microsoft Internet Explorer'. The address bar shows 'http://localhost/cadastro.php'. The page content is titled 'Cadastro de Professores' and is divided into two sections: 'Dados Pessoais' and 'Disciplinas'. The 'Dados Pessoais' section contains five text input fields: 'Código:', 'Nome:', 'Telefone:', 'E-mail:', and 'Currículo:'. The 'Disciplinas' section contains four input fields: 'Código da Disciplina:', 'Nome da Disciplina:', 'Objetivo:' (with a scrollable area), and 'Conteúdo Programático:' (with a scrollable area). The browser's taskbar at the bottom shows several open applications, including 'Intranet local', 'C:\apach...', and 'Cadastr...'. The system tray shows the time as 15:44.

Figura 4.1: Formulário.

Fonte: Elaborado pelo autor.

Na Etapa 3, onde realizou-se o armazenamento dos dados preenchidos no formulário no banco de dados, foram utilizadas duas funções em PHP, uma para fazer a conexão com o banco de dados e outra para fazer a inserção dos dados no banco de dados.

Na Etapa 4, onde realizou-se a transformação dos dados contidos no banco de dados dos professores em diversos documentos XML, foi utilizado um arquivo PHP para fazer as consultas SQL no banco de dados e retornar os arquivos XML.

Na Etapa 5, gerou-se a partir dos documentos XML as páginas correspondentes ao professor, utilizando para isso um arquivo XSL.

A Figura 4.2, ilustra a Página Principal correspondente ao professor:

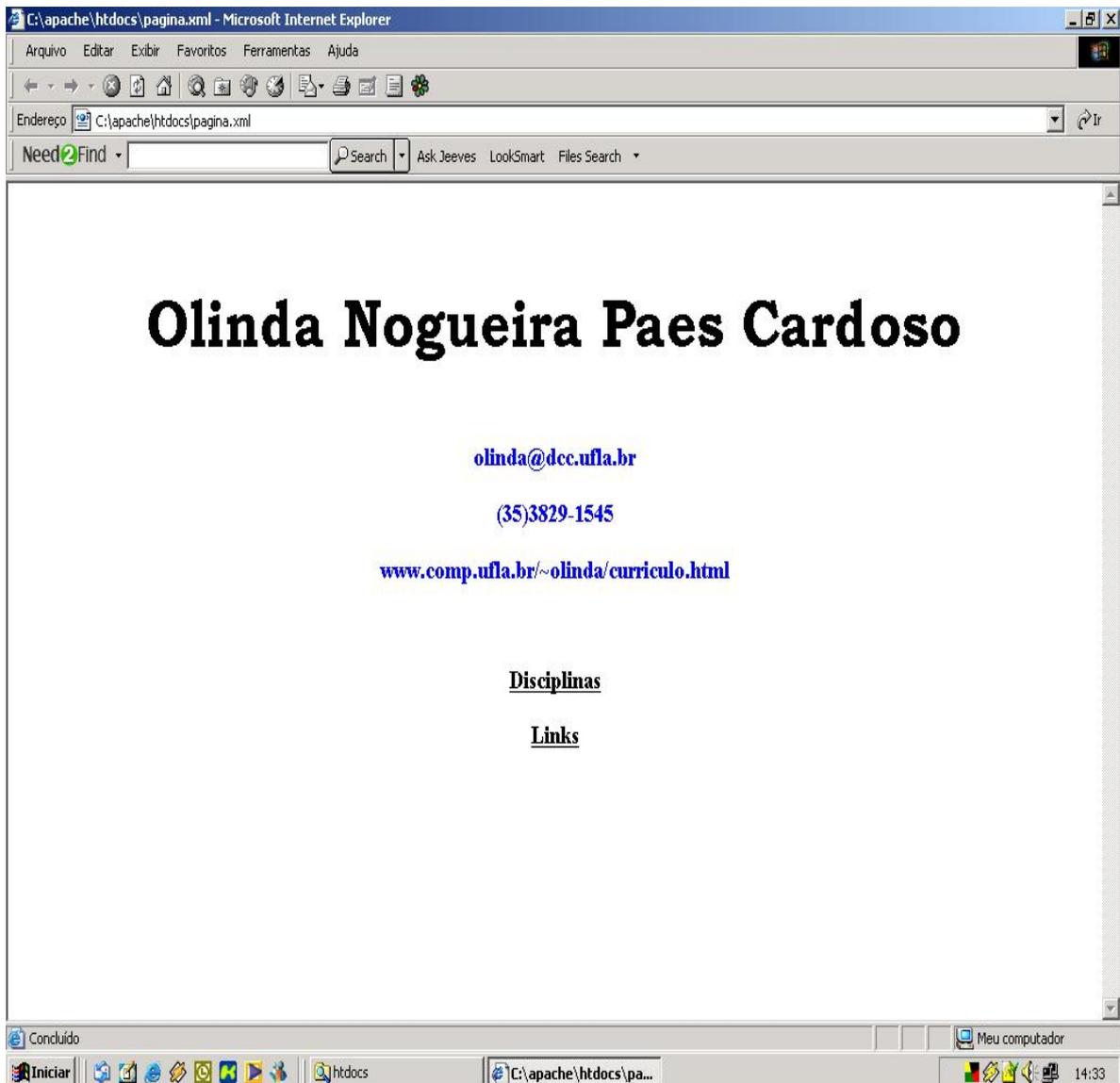


Figura 4.2: Página Principal.

Fonte: Elaborado pelo autor.

A Figura 4.3 ilustra a Página de Disciplinas correspondente ao professor:

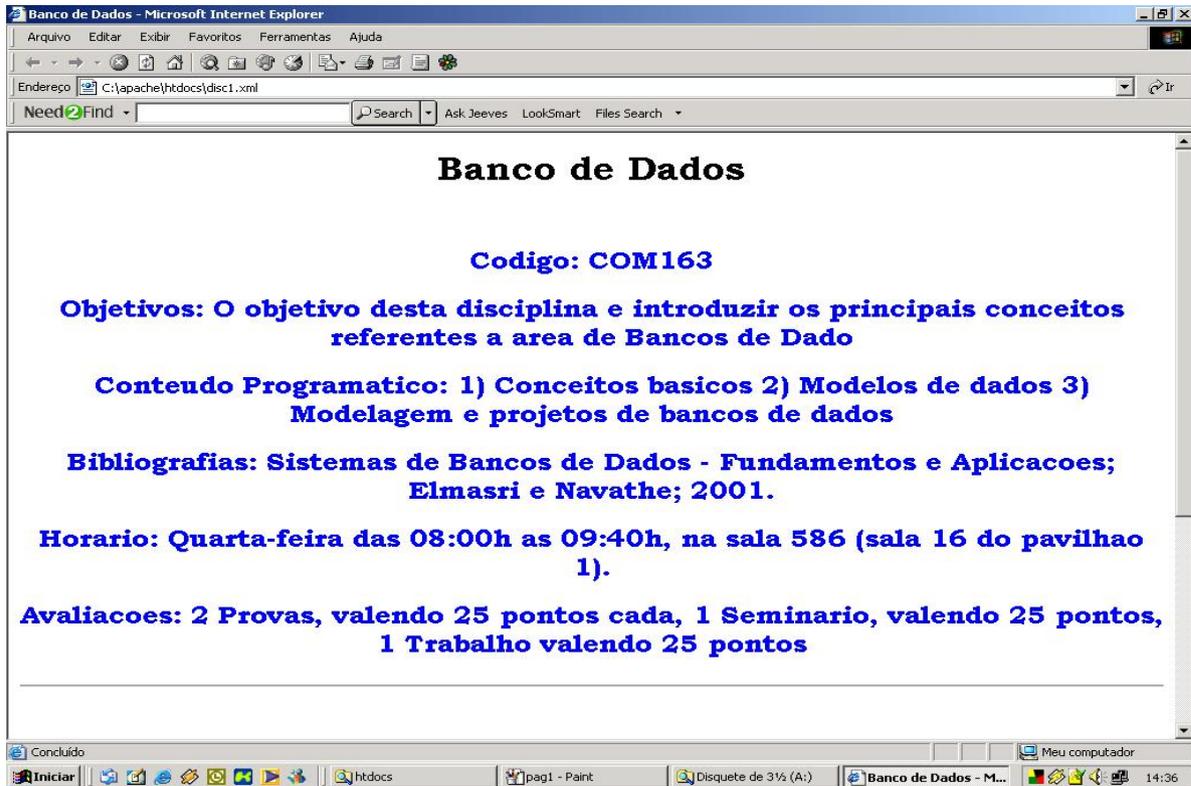


Figura 4.3: Página de Disciplinas.

Fonte: Elaborado pelo autor.

A Figura 4.4 ilustra a Página de *Links* correspondente ao professor:

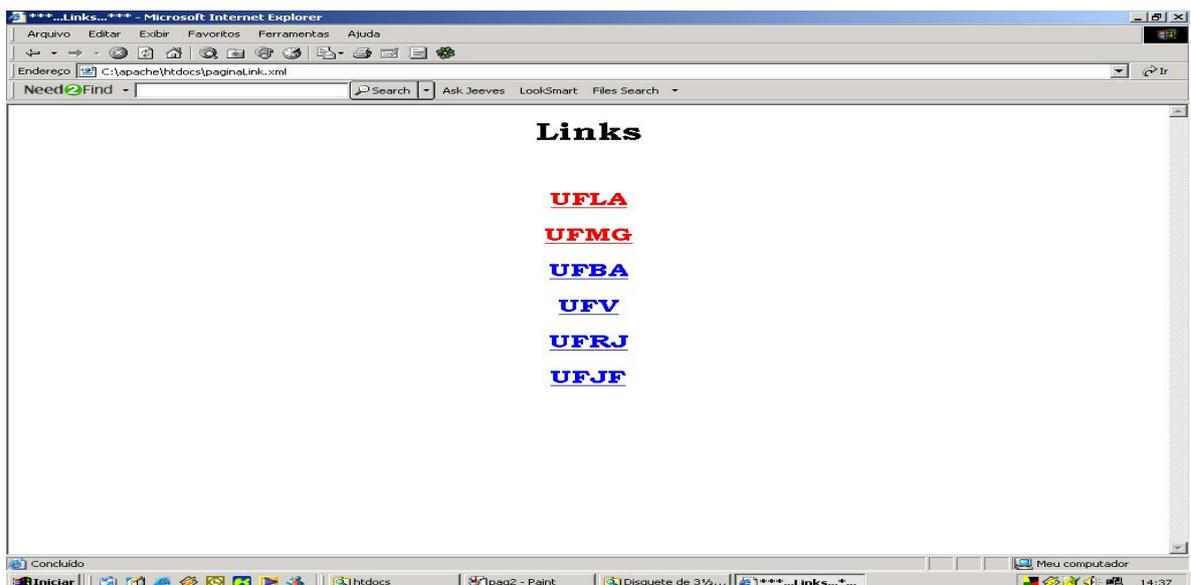


Figura 4.4: Página de *Links*.

Fonte: Elaborado pelo autor.

Na Etapa 6 realizou-se a análise dos resultados obtidos.

Os resultados foram satisfatórios, tendo em vista a necessidade dos professores da UFLA, embora o Gerador Automático de Páginas tenha sido construído de maneira simplificada, não tratando detalhes como:

- a parte estética, não disponibilizando para usuários a escolha das cores, das fontes, do tamanho da letras, etc;
- a Interface, não levando em consideração o tipo de usuário da página;
- entre outros detalhes.

Como trabalhos futuros, sugere-se a implementação da parte estética e da interface, além da criação de um sistema de recuperação de informação.

5 CONCLUSÕES

Pode-se observar que os objetivos inicialmente propostos neste trabalho foram alcançados, pois foi desenvolvida uma ferramenta para auxiliar os professores da UFLA na construção de suas Páginas Pessoais. Para tanto foi construído um gerador automático de páginas para os professores da UFLA.

O principal problema resolvido foi preencher a lacuna entre os recursos que a UFLA oferece tais como a Internet e e-mail, e a não utilização desses recursos pelos professores, ou seja, o trabalho busca sanar o problema dos professores que não possuem uma Página Pessoal, que acham de grande importância o uso da mesma, mas que não possuem conhecimento ou tempo suficiente para o seu desenvolvimento.

Vale ressaltar que este gerador automático de páginas foi testado em simulações, de onde pôde-se observar seu correto funcionamento. Porém este ainda não está sendo utilizado pelos professores. Desta forma, conclusões a cerca de sua real utilização ficam limitadas.

Acredita-se que os resultados que seriam obtidos por meio da utilização deste gerador automático de páginas seriam satisfatórios, tendo em vista a necessidade dos professores da UFLA. Acredita-se também que o trabalho, uma vez que este seja implementado junto aos professores da UFLA, vindo a ser utilizado no dia-a-dia de seu trabalho de maneira simples, será de grande aceitação, pois estará facilitando o acesso a documentos, ampliando a comunicação entre o professor e seus alunos, dentre outras funcionalidades.

Como sugestões para trabalhos futuros têm-se: a implementação da parte estética, da parte da Interface e a implementação de um sistema de Recuperação de Informação, além da apresentação e utilização do gerador automático de páginas aos professores da UFLA.

6 REFERÊNCIAS BIBLIOGRÁFICAS

[Abiteboul (1997)] ABITEBOUL, S.. **Querying Semi-Structure Data**, in Proc. of International Conference on Data Base Theory (ICDT97). Springer, Verlag, 1997.

[Aroca (2001)] AROCA, R. V.. **Tutorial MySQL**. Disponível em: <http://apostilando.com>, atualizada em 18 set. 2001.

[Buyens (2001)] BUYENS, J.. **Desenvolvendo Banco de Dados na Web**, São Paulo: Makron Books, 2001.

[Cisneiros (2003)] CISNEIROS, H.. **Trabalhando com PHP e MySQL: Uma Introdução**, disponível em http://www.devin.com.br/eitch/intro_php, consultada em 07 abr. 2005.

[Date (2003)] DATE, C. J.. **Introdução a Sistemas de Banco de Dados**, Tradução de Daniel Vieira, Rio de Janeiro: Campus, 2003.

[Deitel *et al.* (2003)] DEITEL, H. M.; DEITEL, P. J.; NIETO, T. R.; LIN, T. M.; SADHU, P.. **XML, Como Programar**, tradução de Luiz Augusto Salgado e Edson Furmankiewicz, Porto Alegre: Bookman, 2003.

[Duarte *et al.* (2005)] DUARTE, O. C. M. B.; FURTADO, M. B. Jr.. **XML -Extensible Markup Language**, Universidade Federal do Rio de Janeiro. Disponível em: http://www.gta.ufrj.br/grad/00_1/miguel, consultado em 27 abr. 2005.

[Esteves *et al.* (2001)] ESTEVES, A.; SANTOS, L.; GUIMARÃES, P.. **XML nas Bibliotecas Digitais**. Disponível em www.bibliosoft.pt/projectoxml, atualizada em 2001, consultada em 08 nov. 2004.

[Fischer (2000)] FISCHER, H. G.. **PHP – Guia de Consulta Rápida**, São Paulo : Novatec Editora Ltda, 2000.

[Lakatos (1983)] LAKATOS, E. M.. **Metodologia do Trabalho Científico**, São Paulo: Atlas, 1983.

[Light (1999)] LIGHT, R.. **Iniciando em XML**. São Paulo: Makron Books, 1999.

[Lozano (2001)] LOZANO, F.. **Apache + PHP + MySQL**. Faculdade Metodista Bennett, SEPAI 2001. Disponível em: www.lozano.eti.br/palestras/apache-mysql-php.pdf, consultado em: 06 maio 2005.

[Mello (2003)] MELLO, R. S.. **Gerenciamento de Dados XML**, Departamento de Informática e Estatística (INE), Centro Tecnológico (CTC) da Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC, 2003. Disponível em <http://www.ulbra-to.br/ensino/43020/artigos/anais2003/anais/ein/minicursoEIN-1>, consultada em 08 nov. 2004.

[Mello *et al.* (2000)] MELLO, R. S.; DORNELES, C.; F., KADE, A.; BRAGANHOLO, V. P.; HEUSER, C. A.. **Dados Semi-Estruturados**. Universidade Federal do Rio Grande do Sul (UFRGS). Disponível em: http://metropole.inf.ufrgs.br/semistructured/grupo_heuser/ronaldo/tut3sbbd.pdf, consultado em 27 abr. 2005.

[Navathe *et al.* (2000)] NAVATHE, S. B.; ELMARSI, R.. **Sistemas de Banco de Dados – Fundamentos e Aplicações**. Tradução de **Fundamentals of Database Systems**, terceira edição, LTC, 2000.

[Ramalho (2002)] RAMALHO, J. A.. **XML Teoria e Prática**, São Paulo: Berkeley, 2002.

[Silberschatz *et al.* (1999)] SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S.. **Sistemas de Banco de Dados**, 3ª Edição. São Paulo: Makron Books, 1999.

[Silveira (2004)] SILVEIRA, F. F.. **Técnicas de Banco de Dados para a World Wide Web** - Universidade da Região da Campanha, CCEI - Centro de Ciências da Economia e Informática, 2004.

[Soares (2001)] SOARES, W.. **Crie um Site B2C – Business to Consumer: com PHP 4 e MySQL**. 2ª Edição, São Paulo: Érica, 2001.

[Stoco (2000)] STOCO, L. M.. **Integrando PHP com MySQL – Guia de Consulta Rápida**, São Paulo: Novatec Editora Ltda, 2000.

[Tolentino (2004)] TOLENTINO, R. J. V.. **Aplicações Web em XML: estágio atual e tendências futuras**. Belo Horizonte: C/Arte, 2004.

[Vivas (2000)] VIVAS, M.. **Aplicações Web em PHP**, disponível em: <http://www.cipsga.org.br>, consultada em 24 maio 2005.

ANEXOS

ANEXO A	34
ANEXO B	35
Figura 1: Código referente ao cabeçalho do Formulário e ao cadastramento dos dados pessoais...	35
Figura 2: Código referente ao cadastramento das disciplinas.....	36
Figura 3: Código referente ao cadastramento dos <i>links</i> e ao fim do formulário.....	37
ANEXO C	38
Figura 1: Arquivo conexão.php.....	38
Figura 2: Arquivo Inserir.php.....	39
Figura 3: Arquivo Config.php.....	39
ANEXO D	40
Figura 1: Arquivo gera_xml.php.....	40
ANEXO E	41
Figura 1: Arquivo pagina.xml.....	41

ANEXO A

Questões da Entrevista

1. Você tem ou trabalha com computador?
2. Acha importante o uso do mesmo?
3. Você considera importante o uso de recursos tais como Internet no seu dia a dia?
4. Você usa e-mail como forma de se comunicar com seus alunos ou colegas?
5. Você tem uma Página Pessoal disponível para seus alunos?
 - Caso afirmativo:
 - Acha que isso facilita a comunicação entre aluno e professor?
 - Tudo que você considera importante para o aluno está presente nesta página?
 - Caso contrário:
 - Acha importante ter uma página pessoal?
 - Acha importante o uso da mesma pelos alunos e até outros professores?
 - Acha que uma página pode facilitar tanto o aluno como o próprio professor?
6. Você considera importante a idéia de se construir um gerador automático de páginas?
7. Considera que isso facilitaria seu trabalho?

ANEXO B

O código utilizado para gerar o formulário é apresentado em três figuras, uma para cada etapa do preenchimento. A Figura 1 apresenta o código referente ao cabeçalho do formulário e ao cadastramento dos dados pessoais, a Figura 2 apresenta o código referente ao cadastramento das disciplinas e a Figura 3 apresenta o código referente ao cadastramento dos *links* e ao fim do formulário.

```
<html>
<head>
  <title>Cadastro de Professores</title>
</head>
<body>
  <p>Dados Pessoais</p>
  <form method = "post" action = "inserir.php">
    <p><label>Código:
      <input name = "codigo" type = "text" size = "10">
    </label></p>
    <p><label>Nome:
      <input name = "nome" type = "text" size = "25">
    </label></p>
    <p><label>Telefone:
      <input name = "telefone" type = "text" size = "10">
    </label></p>
    <p><label>E-mail:
      <input name = "email" type = "text" size = "25">
    </label></p>
    <p><label>Currículo:
      <input name = "curriculo" type = "text" size = "35 ">
    </label></p>
```

Figura 1: Código referente ao cabeçalho do Formulário e ao cadastramento dos dados pessoais.

Fonte: Elaborado pelo autor.

```
<p>Disciplinas</p>

<p><label>Código da Disciplina:
  <input name = "codDis" type = "text" size = "10">
</label></p>

<p><label>Nome da Disciplina:
  <input name = "nomeDis" type = "text" size = "25">
</label></p>

<p><label>Objetivo:
  <textarea name = "objetivo" row ="4" cols = "36"></textarea>
</label></p>

<p><label>Conteúdo Programático:
  <textarea name="contProg" row ="4" cols = "36"></textarea>
</label></p>

<p><label>Bibliografia:
  <textarea name="bibliografia" row="4" cols= "36"></textarea>
</label></p>

<p><label>Horário e Local:
  <textarea name = "horario" row ="4" cols = "36"></textarea>
</label></p>

<p><label>Avaliações:
  <textarea name="avaliacoes" row="4" cols="36"></textarea> </label></p>
```

Figura 2: Código referente ao cadastramento das disciplinas.

Fonte: Elaborado pelo autor.

```
<p>Links</p>

<p>Digite todos os links que deseja na caixa abaixo: </p>

<p><label>Links:
      <textarea name = "link" row = "50" cols = "56"></textarea>
</label></p>

<p>
      <input type = "submit" value = "Cadastrar">
</p>
</form>
</body>
</html>
```

Figura 3: Código referente ao cadastramento dos *links* e ao fim do formulário.

Fonte: Elaborado pelo autor.

ANEXO C

A Figura 1 e a Figura 2 mostram as funções de conexão e de inserção respectivamente. A Figura 1 possui um arquivo denominado “config.php”, que é um arquivo que contém os dados para fazer a configuração do banco de dados. Esse código é mostrado na Figura 3.

```
<?    if(!defined("BD_PHP")){
        define("BD_PHP", 1);
        class BD{ // Atributos da classe
            var $usuario; // Usuario do BD
            var $bd;     // BD a ser utilizado
            var $senha; // Senha para conexão
            var $host;  // Host onde está sendo executado o SGBD
            var $porta; // Porta a ser utilizada para conexão
            function BD(){ //arquivo contendo as configurações do bd
                include("config.php");
                $this->usuario = $usuarioBD;
                $this->senha = $senhaBD;
                $this->bd = $nomeBD;
                $this->host = $hostBD;
                $this->porta = $portaBD; }
            // Conecta-se ao servidor para futuro envio de consultas
            function conectar(){if(mysql_connect($this->host.".".$this->porta,$this->usuario,$this->senha)){
                mysql_select_db($this->bd);
                return 1; // Conexão realizada com êxito
                } else    return 0; // Erro de conexão  }
            function desconectar(){ // Fecha a conexão com o servidor
                if (@mysql_close()) {
                    return 1; // Conexão encerrada com êxito.
                } else    return 0; // Ocorreu algum erro ao se fazer a desconexão }
            } // classe
        } // defined
?>
```

Figura 1: Arquivo conexão.php.

Fonte: Elaborado pelo autor.

```

<?php
    include("index.php");
    $result = $bd->executarSQL("insert into professores(codigo, nome, telefone, email,
curriculo)
    values('$codigo', '$nome', '$telefone', '$email', '$curriculo)");

    $result = $bd->executarSQL("insert into disciplinas(codigo, nome, objetivos, conteudoP,
bibliografia, horario,avaliacoes)
    values('$codDis', '$nomeDis', '$objetivo', '$contProg', '$bibliografia', '$horario',
'$avaliacoes)");

    $result = $bd->executarSQL("insert into links(cod_Prof, link)
    values('$codigo', '$link)");

    $result = $bd -> executarSQL("insert into discprofessores(codProf,codDisc)
    values('$codigo', '$codDis)");

?>

```

Figura 2: Arquivo Inserir.php

Fonte: Elaborado pelo autor.

```

<?php
    // variaveis para configuracao do bd
    $usuarioBD = "fernanda";
    $senhaBD = "senha";
    $nomeBD = "fernanda";
    $hostBD = "estelle.comp.ufla.br";
    $portaBD = 3306;

?>

```

Figura 3: Arquivo Config.php

Fonte: Elaborado pelo autor.

ANEXO D

A Figura 1 mostra o arquivo gera_xml.php que foi utilizado para gerar os documentos XML. Esse código retorna o arquivo XML referente somente aos dados pessoais do professor, sendo esse código referente à página principal, pois é a partir dela que se chegam às outras.

```
<html> <body> <?php
    $arquivo = "pagina.xml";
    $servidor = "estelle.comp.ufla.br";
    usuario = "fernanda";
    $senha = "senha";
    $banco = "fernanda";
    $con=mysql_connect($servidor, $usuario, $senha);
    $sql="select nome, telefone, email, curriculo from professores where codigo = '001'";
    $res=mysql_db_query($banco, $sql);
    $total=mysql_numrows($res);
    if($total==1){ $fd = fopen ($arquivo, "w+");
    fwrite($fd,"<?xmlversion='1.0'?><?xml:stylesheettype='text/xml'href='pagina.xml' ?>\n");
    $nome=mysql_result($res,0, "nome");
    $telefone=mysql_result($res,0,"telefone");
    $email=mysql_result($res,0,"email");
    $curriculo=mysql_result($res,0,"curriculo");
    $xml="<paginas>\n";
        $xml.="<nome>$nome</nome>\n";
        $xml.="<telefone>$telefone</telefone>\n";
        $xml.="<email>$email</email>\n";
        $xml.="<curriculo>$curriculo</curriculo>\n";
    $xml.="</paginas>\n";
    fwrite($fd, $xml);
        fclose($fd);
    }mysql_close($con);
?>
<p align="center">Arquivo pagina.xml gerado.</p>
</body>
</html>
```

Figura 1: Arquivo gera_xml.php

Fonte: Elaborado pelo autor.

ANEXO E

Esse código que é apresentado na Figura 1 possui um arquivo denominado pagina.xml, que foi utilizado para exibir os arquivo XML que são retornados.

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <HTML>
      <BODY BGCOLOR="white" TEXT ="blue" BACKGROUND="" LINK="blue"
VLINK="black" ALINK="magenta">
        </BODY>
          <H1 ALIGN="center"><font face="Bookman Old Style" size="+4"
color="black"><br><xsl:value-of select="paginas/nome"/></br> </font>          </H1> <br>
</br>
          <H3 ALIGN="center"><font size="+1" color="blue"> <xsl:value-of
select="paginas/email"/><br></br></font></H3>
          <H3 ALIGN="center"><font size="+1" color="blue"> <xsl:value-of
select="paginas/telefone"/><br></br></font></H3>
          <H3 ALIGN="center"><font size="+1" color="blue"> <xsl:value-of
select="paginas/curriculo"/><br></br></font></H3>
          <H3 ALIGN="center" size="+3"><a href=
"C:\apache\htdocs\paginaDisc.xml"> Disciplinas</a></H3>
          <H3 ALIGN="center" size="+3"><a href=
"C:\apache\htdocs\paginaLink.xml">Links</a></H3>
        </HTML>
      </xsl:template>
    </xsl:stylesheet>
```

Figura 1: Arquivo pagina.xml

Fonte: Elaborado pelo autor.