

**UNIVERSIDADE DO VALE DO RIO DOS SINOS  
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS  
CURSO DE INFORMÁTICA**

**GERENCIAMENTO E ACESSO A  
DOCUMENTOS NA INTERNET  
ATRAVÉS DE XML, RDF E  
ONTOLOGIA**

**Marcirio Silveira Chaves**

**Prof. PhD. Renata Vieira  
Orientadora**

*Monografia submetida como requisito  
parcial para a obtenção do título de  
Bacharel em Informática.*

São Leopoldo, novembro de 2001.

## **AGRADECIMENTO**

Agradeço em primeiro lugar a Deus por permitir a execução deste trabalho. Em segundo lugar, aos meus pais e ao meu avô, Alcides Rodrigues da Silva (*in memoriam*), por proporcionarem o acesso à educação. E finalmente, aos professores Sandro Rigo e Sérgio Crespo por participarem com críticas relevantes no desenvolvimento desta monografia.

Um agradecimento especial para a professora Renata Vieira pela orientação e por nortear as idéias expostas neste documento.

"A grandeza não consiste em receber honras, mas em merecê-las."  
Aristóteles 384 - 322 a.C.

## SUMÁRIO

<b>LISTA DE ABREVIATURAS.....</b>	<b>8</b>
<b>LISTA DE FIGURAS.....</b>	<b>9</b>
<b>RESUMO.....</b>	<b>10</b>
<b>ABSTRACT .....</b>	<b>11</b>
I. INTRODUÇÃO .....	12
II. A LINGUAGEM XML .....	14
2.1 Conceito.....	14
2.2 O uso.....	14
2.3 As diferenças entre HTML e XML.....	15
2.3.1 Comparação entre códigos de documentos HTML e XML .....	16
2.4 A estrutura em forma de árvore dos documentos XML.....	17
2.5 DTD – Document Type Definition (Definições de tipo de documento).....	18
2.6 Documentos bem-formatados .....	18
2.7 Documentos válidos.....	18
2.8 Como é um documento XML? .....	19
2.8.1 Declarações.....	19
2.8.2 Elementos .....	19
2.8.3 Comentários.....	19
2.8.4 Referências a caracteres.....	20
2.8.5 Instruções de processamento .....	20
III. VISUALIZAÇÃO DE DOCUMENTOS XML.....	21
3.1. XSL - <i>eXtensible Style Language (linguagem de estilo extensível)</i> .....	21
3.2 Termos XSL importantes .....	21
3.3 Funcionamento da linguagem .....	22
3.4 A XSLT - <i>eXtensible Style Language Transformation</i> .....	22
3.5 Algumas funções da linguagem XSLT aplicadas ao estudo de caso.....	23
3.5.1 Templates .....	23
3.5.2 Cláusula <code>&lt;xsl:for-each select&gt;</code> .....	23
3.6 Exibição de informações antes da implementação em XML.....	24
3.7 As folhas de estilo do estudo de caso .....	25
IV. BUSCA DE INFORMAÇÃO.....	28
4.1 Consulta aos documentos XML.....	28
4.1.1 XQuery .....	28
4.2 Semantic Web .....	33
4.2.1 Ontologias.....	33
4.2.1.1 Tipos de Ontologia .....	34
4.2.1.2 Benefícios da Ontologia .....	35
4.2.1.3 Construção da Ontologia .....	35
4.2.1.4 Ontologia e RDF schema.....	36
4.2.2 XML e RDF - <i>Resource Description Framework</i> (estrutura de descrição de recursos) .....	36

4.2.2.1 Terminologia e conceitos importantes.....	37
4.2.2.2 Declarações RDF.....	38
4.2.2.3 A utilização do RDF.....	39
4.2.2.4 RDF(S).....	39
4.2.2.5 Consulta em RDF.....	41
4.2.2.6 A linguagem RQL.....	41
V. CONCLUSÃO.....	43
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>44</b>
<b>BIBLIOGRAFIA COMPLEMENTAR.....</b>	<b>47</b>
<b>ANEXOS.....</b>	<b>49</b>

## **LISTA DE ABREVIATURAS**

DTD - Document Type Definition  
E-commerce - Electronic-commerce  
EDI – Eletronic Data Interchange  
HTML - Hipertext Markup Language  
ISO - International Standars Organization  
Lorel - Lighweight Object Repository  
PIs – Processing Instructions  
RDF - Resource Description Framework  
RDFS - Resource Description Framework Schema  
RTF - Rich Format Text  
SGML - Standard Generalized Markup Language  
UCS - Universal Character Set  
URI - Uniform Resource Identifier  
URLs - Uniform Resource Locators  
XML - eXtensible Markup Language  
XML-GL - XML - Graphic Language  
XML-QL - eXtensible Markup Language - Query Language  
XQL - XML Query Language  
XSL - eXtensible Style Language  
XSLT - eXtensible Style Language Transformation  
W3C - World Wide Web Consortium

## LISTA DE FIGURAS

Figura 2.1 - Quando usar o XML ou o HTML.....	15
Figura 2.2 - Benefícios do XML em relação ao HTML.....	16
Figura 2.3 – Exemplo de código HTML. ....	16
Figura 2.4 - Exemplo de código XML. ....	17
Figura 2.5 - Exemplo de como a biblioteca é estruturada. ....	18
Figura 3.1 - Referência bibliográfica marcada em HTML.....	24
Figura 3.2 - Exibição da referência bibliográfica marcada em HTML .....	24
Figura 3.3 - Exemplo de uma folha de estilos XSL para exibir os elementos de uma bibliografia .....	25
Figura 3.4 – Referência bibliográfica marcada em XML.....	26
Figura 3.5 – Exibição da referência bibliográfica marcada em XML (caso I).....	26
Figura 3.6 – Exibição da referência bibliográfica marcada em XML (caso II).....	27
Figura 4.1 – Exemplo de uma consulta utilizando a linguagem XQuery.....	29
Figura 4.2 – Resultado de uma consulta utilizando a linguagem XQuery .....	29
Figura 4.3 – Consulta em XQuery utilizando o <i>corpus</i> do estudo de caso.....	30
Figura 4.4 – Resultado da consulta em XQuery utilizando o <i>corpus</i> do estudo de caso.....	30
Figura 4.5 – Consulta em XQuery para verificar as disciplinas do <i>corpus</i> que possuem a maior parte das aulas práticas .....	31
Figura 4.6 – Disciplinas do <i>corpus</i> que possuem a maior parte das aulas práticas .....	31
Figura 4.7 – Consulta em XQuery relacionando professores e disciplinas do <i>corpus</i> . ....	32
Figura 4.8 – Resultado da consulta em XQuery relacionando professores e disciplinas do <i>corpus</i> . ....	32
Figura 4.9 – O modelo de linguagem em camada para a Web.....	34
Figura 4.10 - Tipos de ontologias e seus relacionamentos.....	35
Figura 4.11 – Representação de uma declaração RDF.....	38
Figura 4.12 - Declaração RDF marcada no formato XML.....	38
Figura 4.13 – Modelo de dados RDF. ....	39
Figura 4.14 – Modelo de dados RDF incorporado em um Schema. ....	40
Figura 4.15 – Exemplo de uma consulta em RQL (Caso I) .....	41
Figura 4.16 – Resultado da consulta em RQL apresentada na Figura 4.15.....	41
Figura 4.17 – Exemplo de uma consulta em RQL (Caso II) .....	42
Figura 4.18 – Resultado da consulta em RQL apresentada na Figura 4.17.....	42

## RESUMO

A *World Wide Web* foi originalmente construída para ser usada por humanos, entretanto, ela deve ser legível para as máquinas também. A maior parte dos documentos disponíveis na Web hoje está identificado apenas com informações de exibição. Diante da necessidade de prover meios mais adequados para possibilitar acesso automático ao repositório de informações disponível na Internet, é necessário a utilização da tecnologia XML (*eXtensible Markup Language*), que possibilita a exploração da semântica dos conteúdos desses documentos.

Este trabalho apresenta um estudo sobre técnicas para organização, apresentação e busca de informações na Web. A tecnologia XML foi utilizada para etiquetar os dados do documento de estudo de caso, caracterização das disciplinas da UNISINOS, com o objetivo de identificar semanticamente as partes do mesmo.

Para associar e representar as informações dos professores e das disciplinas na Internet, foi utilizado o aplicativo XML RDF (*Resource Description Framework*), que permite adicionar semântica formal para a Web.

A definição dos principais conceitos, relações e restrições representados no RDF está formalmente descrita em uma ontologia para o domínio universidade. A ontologia e o RDF facilitam a estruturação e o acesso a documentos na Web, tanto por máquinas, quanto por pessoas, permitindo uma melhor identificação do significado do conteúdo dos mesmos em um domínio.

Baseado na tecnologia XML, no aplicativo RDF e na ontologia, que constituem o termo *Semantic Web*, procurou-se adicionar semântica aos dados disponíveis na Internet de modo a facilitar a busca e exibição de informações.

Palavras-chave: XML, Ontologia e RDF



## ABSTRACT

The *World Wide Web* was built to be used by people, however, it must be also legible to machines. Most of the documents nowadays available in the Web is identified only by layout information. Facing the need of providing better automatic access to the amount of information available in the Web, it is necessary to use the XML (eXtensible Markup Language) technology, which allows the semantic exploration of the contents of these documents.

This work presents a study on the techniques for organization, presentation and search of information through the Web. The XML technology has been used to mark the data from case study document – UNISINOS' subject characterization, willing to identify semantically its parts.

In order to associate and represent information about teachers and subjects on the Internet, RDF (Resource Description Framework), an application XML, was used, allowing adding formal semantics to the Web.

The definition of the main concepts, relations and restrictions represented on the RDF is formally described in an ontology for domain university. The ontology and the RDF easy the framework and the access to documents in the Web, as by machines, as by people, allowing a better identification of the meaning of its content in a domain.

Based on XML technology, on application RDF and on the ontology, which form the "Semantic Web" term, we try to add semantics to available data on the Internet in order to improve information search and exhibition.

Key Words: XML, Ontology and RDF

# I. INTRODUÇÃO

Este trabalho é um estudo sobre a aplicação de técnicas para organização, apresentação e busca de informações através da Internet. Foi realizado um estudo de caso com o documento de caracterização das disciplinas que é constituído de informações sobre as disciplinas de graduação da Universidade do Vale do Rio dos Sinos (UNISINOS), tais como: nome da disciplina, conteúdo, bibliografia, entre outras.

A tecnologia estudada é XML (*eXtensible Markup Language*), que está rapidamente tornado-se uma das tecnologias mais amplamente adotadas para intercâmbio e representação de informações na *World Wide Web*. Visto que o XML está emergindo como o formato padrão de dados da era Internet, tem havido um aumento substancial da quantidade de dados codificados no formato XML.

A exibição dos documentos é tratada com folhas de estilo XSL (*eXtensible Style Language*) que permite a apresentação das partes de um documento que interesse a determinados usuários.

As consultas aos documentos marcados em XML são feitas com a linguagem XQuery que proporciona acesso aos dados através dos elementos identificados semanticamente possibilitando a elaboração de consultas específicas para determinadas partes dos documentos.

Para organizar e produzir um acesso inteligente às informações é apresentada uma ontologia com os termos mais relevantes para o domínio universidade e para a aplicação do documento de estudo de caso. Além da ontologia, o mesmo documento é representado na forma de RDF (*Resource Description Framework*) para permitir a exploração de consultas através de metadados.

O documento de caracterização das disciplinas da UNISINOS (ver Anexo 1) apresenta as informações relacionadas a cada disciplina dos cursos de graduação oferecidos pela universidade. Para este trabalho, foi selecionado um *corpus* (conjunto de documentos) de trinta disciplinas dos cursos de Informática do Centro de Ciências Exatas e Tecnológicas. O conjunto completo das disciplinas está disponibilizado na Internet na URL <http://inf.unisinos.br/cursos/graduacao/as/mostra.php?curso=6074>.

Esses documentos estão codificados em HTML (*Hipertext Markup Language*) com informações de exibição através de etiquetas pré-definidas. Os Centros de Ensino da UNISINOS, muitas vezes, necessitam receber informações de apenas uma parte do documento de caracterização das disciplinas, objetivos ou conteúdos programáticos, por exemplo. A visualização das informações referentes às disciplinas se dá pela exibição de todo o documento, sendo bastante difícil direcionar a exibição de informação relativa a uma determinada parte do documento devido à limitação das marcações HTML.

A fim de possibilitar o acesso às partes específicas de qualquer disciplina, foi utilizada a tecnologia XML que possibilita a identificação semântica e estruturada das informações facilitando a extração de dados relacionados, ou seja, busca-se permitir consultas do tipo: em quais disciplinas o conteúdo *qualidade de projetos* é indicado com a bibliografia [ROC00]? O conteúdo *ergonomia* faz parte de quais disciplinas? Quais disciplinas têm como objetivo

*apresentar ao aluno os principais conceitos e métodos utilizados em projetos de interface humano-computador?*

Este trabalho está organizado do seguinte modo: O capítulo II introduz a linguagem XML, apresenta as diferenças entre XML e HTML, expõe a gramática DTD e descreve os componentes de um documento marcado em XML. No capítulo III são apresentadas as folhas de estilo XSL e alguns exemplos de aplicação das mesmas no documento de estudo de caso. O capítulo IV descreve alguns exemplos de consultas utilizando a linguagem XQuery e a apresenta dois componentes da *Semantic Web*, ontologia e RDF. No fim deste capítulo são apresentados dois exemplos de consulta utilizando a sintaxe RQL. No último capítulo a conclusão expõe o aprendizado obtido no desenvolvimento deste trabalho, a experiência adquirida e introduz algumas idéias para trabalhos futuros.

## II. A LINGUAGEM XML

Desde o surgimento da Internet, a imensa maioria das informações está codificada com as marcações pré-definidas pela HTML, que fornecem apenas informação de exibição. Diante dessas limitações e da necessidade de se identificar semanticamente os conteúdos descritos nos documentos surgiu a XML.

Neste capítulo são apresentados alguns conceitos sobre a linguagem XML e as diferenças entre HTML e XML. A seção 2.5 introduz a gramática DTD que especifica um conjunto de regras para um documento XML. Documentos bem formados e válidos são descritos nas seções 2.6 e 2.7. Finalmente, na seção 2.8 e nas suas subseções, é abordada a constituição de um documento XML.

### 2.1 Conceito

Segundo [HAR99], XML acrônimo para *eXtensible Markup Language* (linguagem de marcação estendida), é um conjunto de regras para definir etiquetas semânticas que divide um documento em partes e identifica as diferentes partes deste documento.

[PIM00] também define XML como sendo uma linguagem de marcação apropriada à representação de dados, documentos e demais entidades, cuja essência, fundamenta-se na capacidade de agregar informações.

Além das duas definições acima, a introdução da especificação XML elaborada pelo *W3Consortium*<sup>1</sup>, [BRA00] diz que “XML descreve uma classe de dados chamada documentos XML e descreve parcialmente o comportamento de programas de computador que os processa. O XML é um perfil de aplicativo ou uma forma restrita do SGML *Standard Generalized Markup Language* (linguagem padrão de marcação generalizada). Em termos construtivos, os documentos XML estão em consonância com os documentos SGML.”

Um dos principais motivos que levaram a criação do XML foi a necessidade de uma linguagem de marcação que suportasse tanto marcação baseada em apresentação como em conteúdo.

### 2.2 O uso

O XML que é extensível e flexível atraiu um enorme interesse e vem sendo adotado desde a primeira vez em que foi formulado, ou seja, em 1996.

Atualmente, ele vem sendo usado em algumas áreas genéricas, tais como: órgãos governamentais norte-americanos, fabricantes de equipamentos para telecomunicações, assistência médica, sistemas especializados para recursos humanos, ferramentas, recursos relacionados a processos de engenharia de linguagem natural, entre outros. Além disso, pode-se citar segmentos específicos, por exemplo, a DHL (maior empresa mundial de entregas), a edição interativa do Wall Street Journal e o serviço on-line ZDNet da Ziff-Davis (reconhecida empresa do mercado editorial – revistas e livros - de informática).

---

<sup>1</sup> Órgão que desenvolve tecnologias interoperáveis (especificações, diretrizes, programas e ferramentas) para conduzir a Internet ao seu potencial máximo como um fórum para informação, comércio, comunicação e entendimento coletivo. <http://www.w3c.org>

Empresas como a Cisco Systems já usam o XML para distribuir notícias e informações dentro de redes corporativas, coletando essas informações de várias fontes, tanto internas como externas à empresa. O Citibank está trabalhando atualmente em um sistema de cobrança/pagamento baseado fundamentalmente no XML. Ademais, a maioria, senão todos os importantes sites de busca, tais como, Google, Yahoo e Alta Vista por exemplo, utilizam a tecnologia XML porque ela permite otimizar as respostas para os usuários.

Além das empresas citadas acima, “o XML também irá acelerar a aceitação do EDI – *Electronic Data Interchange* (intercâmbio eletrônico de dados), bastante comum entre empresas. É importante enfatizar que com a utilização do EDI o envio e o processamento de documentos é feito de forma automática sem intervenção manual. Isso porque o EDI utiliza documentos existentes em papel e os transforma em documentos que podem ser verificados eletronicamente. Por meio de marcas XML padronizadas para o EDI, hospitais, companhias de seguro e consultórios podem trocar informações entre si sobre um determinado paciente, e os formulários adequados podem ser facilmente transferidos através da Internet ou através de uma Intranet segura distinta” [PIT00].

### 2.3 As diferenças entre HTML e XML

O HTML, *Hipertext Markup Language* (linguagem de marcação de hipertexto) é usado basicamente para a apresentação de conteúdo e o XML para estruturar dados. Neste contexto, é importante ficar explícito que o XML não é um substituto do HTML. O XML é, na realidade, uma maneira de solucionar determinados problemas que são encontrados ao usar-se o HTML.

Para elucidar esse raciocínio, [PIT00] apresenta a Figura 2.1 que mostra alguns problemas e a solução mais indicada para solucioná-los.

Número	Problema	Solução
1	Os dados são constituídos de algumas poucas páginas e imagens, de forma muito semelhante àquelas encontradas em folhetos?	Usar HTML
2	Os dados requerem mecanismos avançados de pesquisa de forma que os usuários possam obter aquilo de que precisam do site?	Usar XML
3	Os dados serão vistos em vários tipos de ambientes e de dispositivos?	Usar XML
4	O conteúdo mantém-se relativamente estático?	Usar HTML
5	É necessário interfacear os dados entre máquinas ou mecanismos de bancos de dados diferentes?	Usar XML
6	Você está tendo problemas com seu site em termos dos usuários se perderem ou de não serem capazes de encontrar aquilo que desejam?	Usar XML
7	O site conterá basicamente recursos multimídia como clips de som ou de vídeo?	Usar HTML, possivelmente com plug-ins

Figura 2.1 - Quando usar o XML ou o HTML

Os problemas 2, 3, 5 e 6 apresentados na Figura 2.1 serão explicados no decorrer desse trabalho.

[PIT00] ainda destaca que “com o XML tem-se um melhor controle em relação ao layout, um menor esforço no servidor Web devido à capacidade de acessar informações do lado do

cliente, a capacidade de publicar qualquer tipo de informação tanto na Web quanto em intranets e um número menor de problemas ao exibir páginas longas.”

Com linguagens de marcação estruturadas, como XML, tem-se também maior flexibilidade de pesquisa. Pode-se não somente procurar qualquer palavra, como é possível em qualquer página Web padrão, mas também pesquisar dentro de vários elementos de um documento. É possível, por exemplo, pesquisar apenas informações de cabeçalho ou então pesquisar em cabeçalhos específicos como os cabeçalhos de um livro. Não será possível fazer isso se o livro foi criado como um documento HTML. Se o livro inteiro fosse um único documento, seria necessário pesquisar o arquivo inteiro; se fosse um arquivo individual, seria preciso um aplicativo de servidor Web separado para pesquisar todos ao mesmo tempo.

Além disso, o XML fornece uma ampla gama de recursos que não são encontrados no HTML, incluindo uma linguagem extensível que fornece a capacidade de definir suas próprias marcas e atributos. Esses elementos e suas marcas de início e de fim juntamente com seus atributos o ajudam a definir os elementos estruturais do documento, de forma muito semelhante aos elementos do SGML. A capacidade de aninhar estruturas de documentos dentro das estruturas de outros documentos, para criar documentos complexos é outro recurso a ser citado. Por fim, ele é capaz de verificar a validade de estruturas de documentos durante o processamento [PIT00].

Na Figura 2.2 [MAD01] apresenta alguns dos benefícios do XML em relação ao HTML através de cinco importantes características.

Característica	HTML	XML
Extensibilidade	Conjunto de tags fixo	Conjunto de tags extensível
Propósito das tags	Tags descrevem apresentação	Tags descrevem conteúdo do dado
Visão	Apresentação única	Múltiplas visões do mesmo documento (por XSL)
Orientação	Documentos	Documentos mais dados semi-estruturados
Pesquisa	Somente pesquisa palavra-chave	Palavra-chave mais consultas campo case-sensitive

Figura 2.2 - Benefícios do XML em relação ao HTML

### 2.3.1 Comparação entre códigos de documentos HTML e XML

Tanto o HTML quanto o XML usam ou são constituídos por elementos. Entretanto, diferentemente do HTML, o XML não descreve como o conteúdo deve ser exibido. Em vez disso, ele descreve o que é o conteúdo. Para iniciar a explicação de uma maneira simples de como se parece um documento HTML e um documento XML, é mostrado a seguir o processo de conversão de um código em HTML para XML.

```
<H1>Biblioteca do Marcirio</H1>
<TABLE><TBODY>
  <TR><TD>XML Black Book – Solução e Poder</TD> <TD> Learning XML </TD>
  <TD> XML Bible </TD></TR>
  <TR><TD> português </TD><TD> inglês </TD> <TD> inglês </TD> </TR>
  <TR><TD>2000</TD> <TD> 2001</TD> <TD> 1999</TD></TR>
  <TR><TD>10</TD> <TD>15</TD> <TD>2</TD></TR>
</TBODY></TABLE>
```

Figura 2.3 – Exemplo de código HTML.

Na Figura 2.3, os títulos dos livros simplesmente foram colocados dentro de uma tabela. Se alguém quisesse pesquisar esse documento em busca de um determinado nome de livro, o recurso de pesquisa do navegador iria percorrer o documento, parando em cada ocorrência do nome que o visitante da página Web estivesse interessado. Se o visitante estivesse procurando o livro *XML Avançado* e o Marcirio tivesse incluído uma referência a um outro livro com a palavra *Avançado* em seu título, por exemplo, *Linux Avançado*, o recurso de pesquisa retornaria para o usuário este livro também.

Se analisarmos o documento mais profundamente, não saberemos exatamente o que os números 10,15 e 2 significam. Sabe-se apenas que eles serão exibidos em forma de tabela, isto é, tem-se apenas informação de layout. Mas para uma máquina de busca esses conteúdos necessitam uma semântica que pode ser contemplada com a utilização da flexibilidade que o XML oferece para criar os próprios elementos para fins específicos.

```

<BIBLIOTECA>
<PROPRIETARIO>Marcirio Silveira Chaves</PROPRIETARIO>
<LIVROS>
  <LIVRO versao="português" ano="2000" nro_copias = "10"> XML Black Book –
  Solução e Poder </LIVRO>
  <LIVRO versao ="inglês" ano="2001" nro_copias = "15"> Learning XML
  </LIVRO>
  <LIVRO versao =" inglês" ano="1999" nro_copias = "2"> XML Bible </LIVRO >
</LIVROS>
</BIBLIOTECA>

```

Figura 2.4 - Exemplo de código XML.

Na Figura 2.4 foi criado o elemento LIVRO. Os atributos especificam a versão que o livro está, em que ano foi lançado e o número de cópias disponíveis do livro de forma semântica, pois os atributos *versao*, *ano* e *nro\_copias* dão sentido ao conteúdo inserido no elemento LIVRO, uma vez que dão um significado para os números 10, 15 e 2.

## 2.4 A estrutura em forma de árvore dos documentos XML

Ao criar um documento XML, o que realmente está se fazendo, é criando uma estrutura em forma de árvore, diferentemente do HTML. Devido ao método não estruturado no qual o HTML opera, é possível inserir um elemento <H2> antes de um elemento <H1>, e a estrutura do documento não seria afetada. pelo fato de o HTML somente exibir – e não interpretar realmente – os elementos como estruturas, o conteúdo dentro desses elementos ainda será exibido, independentemente de serem usados ou não na posição correta.

Porém, no XML isso é diferente. Ao criar um documento XML o que realmente está se fazendo é definindo uma hierarquia e especificando elementos, subelementos e assim por diante. Um diagrama do documento da biblioteca do Marcirio ficaria conforme a Figura 2.5.

Cada documento XML tem um único elemento-raiz, e a partir dele, derivam todos os demais elementos do documento. Isso é diferente no HTML no qual há, na realidade, apenas duas seções distintas em um documento: a seção <HEAD> e a seção <BODY>.

No exemplo da biblioteca, o elemento <BIBLIOTECA> é o elemento-raiz no documento XML. A partir desse elemento raiz, todos os demais – normalmente chamados descendentes – se ramificam.

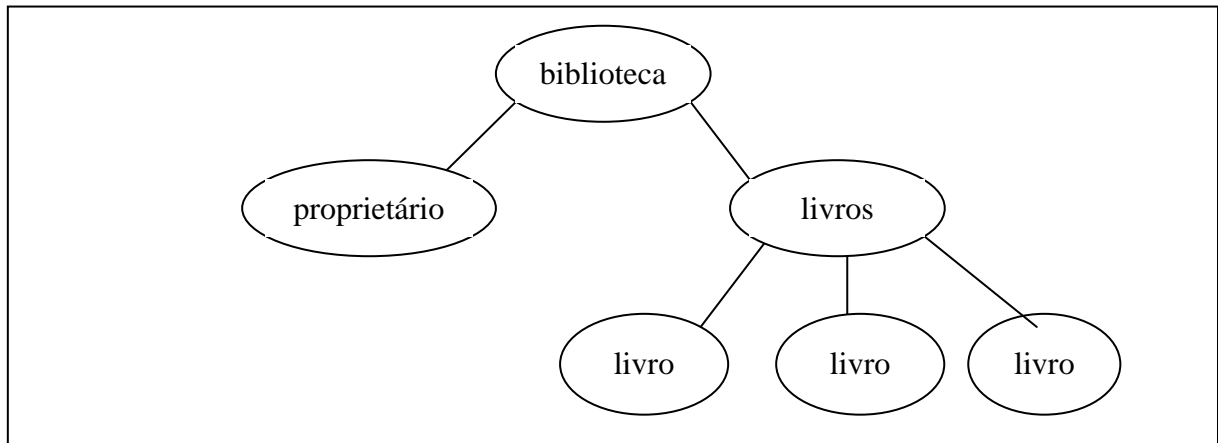


Figura 2.5 - Exemplo de como a biblioteca é estruturada.

Segundo [BON01], um documento XML é modelado em forma de árvore que contém sete tipos de nodos (nodo raiz, nodo elemento, nodo texto, nodo atributo, nodo namespace, nodo processamento de instrução e nodo comentário). Na Figura 2.5, estão representados apenas os nodos elementos. Pode-se observar relacionamentos semânticos entre os nodos elementos, por exemplo, o relacionamento hierárquico (pai/filho) entre o elemento `<livros>` e o elemento `<livro>`.

## 2.5 DTD – Document Type Definition (Definições de tipo de documento)

Segundo [PIT00], DTD é um arquivo que é separado do restante do documento XML principal e que fornece um conjunto de regras para o documento XML ao qual ele é anexado. DTDs são o que realmente distingue tanto o XML quanto o SGML das demais linguagens de marcação.

Uma DTD fornece uma lista de elementos, atributos, notações e entidades contidas em um documento, além dos relacionamentos entre eles. DTDs especificam um conjunto de regras para a estrutura de um documento. Por exemplo, uma DTD pode definir que um elemento LIVRO tenha exatamente um elemento filho ISBN, exatamente um elemento filho TÍTULO e um ou mais elementos filhos AUTORES, e ele pode ou não conter um SUBTÍTULO. A DTD realiza isso com uma lista de declarações de marcação para elementos particulares, entidades, atributos e notações [HAR99].

Assim, pode-se imaginar a DTD como um guia de diretrizes para o documento.

## 2.6 Documentos bem-formados

Um documento bem-formado é aquele que atende à sintaxe XML usada dentro do documento. Por exemplo, se o programador não incluir marcas de fechamento ao inserir elementos no documento, se ele esquecer de incluir a declaração de documento XML no início do documento ou se o documento incluir caracteres que não possam ser analisados sintaticamente ou sejam inválidos, o programador não possuirá um documento XML bem-formado [PIT00].

É importante lembrar que se um documento XML é bem-formado, ele também é um documento SGML bem-formado.

## 2.7 Documentos válidos

Um documento bem-formado não é válido a menos que ele contenha uma DTD apropriada. O documento também precisa obedecer às restrições dessa declaração.



Cada documento XML válido deve iniciar com um cabeçalho que deve conter as seguintes informações, segundo [PIT00]:

- Uma descrição das regras estruturais que o documento deve seguir;
  - Uma lista de recursos externos ou entidades externas que criam qualquer parte específica do documento;
  - Quaisquer declarações dos recursos internos ou entidades internas;
  - Quaisquer notações ou recursos não-XML que devem ser enumerados no documento.
- Essas notações ou recursos não-XML também devem especificar quaisquer aplicativos complementares necessários;
- Listas de qualquer recurso não-XML (como, por exemplo, entidades binárias) que possam ser encontradas no documento.

## 2.8 Como é um documento XML?

Segundo [PIT00], todo o documento XML possui os seguintes componentes:

- Declarações
- Elementos
- Comentários
- Referências a caracteres
- Instruções de processamento

A seguir, é mostrada uma breve descrição de cada componente acompanhada de um exemplo para elucidar a definição.

### 2.8.1 Declarações

Se um documento for válido, ele possui uma DTD associada que é declarada como um conjunto de regras para o documento. Uma declaração é uma marcação que fornece ao processador XML instruções especiais sobre como processar o documento.

Exemplo:

```
<!ELEMENT area_conhec (id_area, nome_area)*>
```

Neste exemplo, a declaração informa ao processador que existe um elemento *area\_conhec* que possui dois elementos filhos chamados *id\_area*, *nome\_area* e pode aparecer mais de uma vez no documento XML.

### 2.8.2 Elementos

Elementos de documentos consistem de marcação e texto contido pela marcação. O miolo de um documento XML é a combinação de elementos conforme definido pela DTD associada ao documento. Os elementos podem ter atributos e aninharem-se entre si. Alguns elementos são obrigatórios ao passo que outros são opcionais. Todas as regras que governam os elementos são definidas na DTD.

Exemplo:

```
<!ELEMENT ano (#PCDATA)>
```

A DTD informa ao processador através da declaração acima que existe um elemento *ano* do tipo *PCDATA*, que é caracter de dado a ser interpretado.

### 2.8.3 Comentários

Os comentários do XML são notas em um documento XML que são ignoradas por um processador XML. Os mecanismos de comentários do XML são idênticos aos comentários do HTML.

Exemplo:

<!--Biblioteca do Marcirio marcada em XML é um comentário-->

#### 2.8.4 Referências a caracteres

As referências a caracteres se referem a todo o texto usado no documento para criar declarações, marcações e texto dentro de documentos XML. O XML usa o conjunto de caracteres ISO 10646, também conhecido como Unicode, que usa padrões de 16 bits para representar caracteres. Em última instância, esse conjunto de caracteres pode representar 65.000 caracteres diferentes, inclusive caracteres gregos e de pontuação especial.

Exemplo:

&#960 – referência ao caracter grego  $\Pi$  (pi).

#### 2.8.5 Instruções de processamento

Freqüentemente, instruções precisam ser passadas do documento XML por meio do analisador sintático para o software, para informar ao software como processar todo o documento ou parte dele. Essas instruções são chamadas instruções de processamento (IP). Uma IP pode ser incluída em qualquer ponto dentro de um documento XML. Normalmente, elas são inseridas no prólogo de modo que elas configurem um ambiente de processamento global para todo o documento.

Exemplo:

<?XML version= “1.0”>

O objetivo da instrução de processamento é o valor especificado após a palavra *version*. Por exemplo, *1.0* especifica que este documento está de acordo com a versão 1 da especificação XML.

Após a descrição da linguagem XML, tendo visto que a mesma apresenta o tratamento de conteúdo e exibição em documentos separados permitindo uma maior flexibilidade para os desenvolvedores, será abordado no próximo capítulo o aspecto da exibição de informações utilizando folhas de estilo para definir a forma como os dados serão apresentados.

## III. VISUALIZAÇÃO DE DOCUMENTOS XML

Este capítulo introduz o tratamento de informações sob o aspecto da visualização dos documentos de caracterização das disciplinas. A linguagem XSL é utilizada para dar suporte a construção das folhas de estilo, e o processador XSL para permitir a interpretação dos elementos XML.

### 3.1. XSL - *eXtensible Style Language* (linguagem de estilo extensível)

A XSL é um mecanismo de folhas de estilo que é personalizado para o XML. Segundo [HAR99], XSL é a linguagem de folha de estilo mais avançada especificamente projetada para ser usada com documentos XML. Além disso, documentos XSL são documentos XML bem-formatados.

Uma folha de estilo contém instruções que passam para um processador (como um *browser*, por exemplo) como traduzir a estrutura lógica de um documento-fonte em uma estrutura de apresentação.

Com a XSL é possível converter um documento XML em HTML, RTF – *Rich Text Format* e outros vocabulários XML ou informar a um navegador ou processador XML exatamente como os documentos XML devem ser exibidos. A XSL tem por objetivo habilitar o desenvolvimento de interfaces amigáveis ao usuário com a criação e modificação de *stylesheets* em XSL. Para [PIT00], a XSL é, em última instância, a linguagem de formatação para o XML.

Baseando-se em folhas de estilo para apresentação, os dados de fonte XML podem ser escritos uma única vez e exibidos de várias formas possíveis. Por exemplo, para criar dados que poderiam ser vistos por um grande número de pessoas, o projetista poderia usar tipos de hardware diferentes e folhas de estilo distintas para acomodar diferentes plataformas de visualização. Isso significa que os mesmos dados poderiam ser exibidos em uma tela de computador, no visor de um telefone celular ou através de dispositivos de gerenciamento de informações pessoais portáteis. Os dados permanecem os mesmos, apenas a folha de estilos muda. Cada dispositivo usaria as folhas de estilos apropriadas para exibir os mesmos dados em seu formato de tela particular [PIT00].

### 3.2 Termos XSL importantes

Qualquer discussão sobre o XSL incluirá, segundo [HAR99], o uso repetido de vários termos-chave. Entre eles estão:

**Regra de construção** – Ela contém instruções de formatação para qualquer elemento do documento.

**Padrão** – É a parte de uma regra de construção que identifica o elemento do documento que recebe a formatação.

**Ação** – É a parte de uma regra de construção que descreve como o elemento do documento (padrão) deve ser formatado.

**Objeto de fluxo** – Estruturas usadas para descrever como o conteúdo de um documento deve ser formatado. Objetos de fluxo são associados a elementos por meio de regras de construção.

**Regra-raiz** - A regra de construção que descreve como um elemento de um documento deve ser formatado.

**Regra-padrão** - A regra de construção que descreve como todas as regras não administradas por outras regras de construção devem ser formatadas.

Exemplo de uma parte de uma folha de estilo utilizando alguns termos-chave:

```
<regra>
  <target-element type= "LIVRO" />
  <P font-size="12pt" fontstyle="italic">
    <children/>
  </P>
</regra>
```

No exemplo acima, o conteúdo do elemento LIVRO será apresentado no *browser*/navegador em forma de parágrafo, pois está marcado com a etiqueta `<P>`, que neste caso é o objeto de fluxo para o elemento "LIVRO", o tamanho da letra será 12 e a aparência será em itálico. Essas informações constituem a parte da ação de uma regra. Além disso, a marcação `<children/>` diz ao analisador sintático que todos os subelementos ou elementos filhos do elemento bibliografia deverão ser exibidos no mesmo formato do elemento "LIVRO".

Assim como em outros tipos de mecanismos de estilos, podem ser criadas mais do que uma folha de estilos XSL para um único documento XML para fornecer várias possibilidades de exibição final. Por exemplo, para o mesmo documento poder ser exibido em um navegador da Web e ser impresso, duas folhas de estilo distintas seriam utilizadas.

### 3.3 Funcionamento da linguagem

Antes de discutir XSL em maiores detalhes, é necessário considerar o modelo de processamento da XSL. Um processador XSL começa com uma folha de estilo e uma "árvore-fonte". A árvore-fonte é a representação de árvore do código-fonte do documento XML, analisado gramaticalmente. Todos os documentos XML podem ser representados como uma árvore.

Conceitualmente, o processador XSL inicia a verificação do nodo raiz da árvore-fonte e processa achando o modelo na folha de estilos que descreve como aquele elemento deveria ser exibido. Enquanto cada nodo é processado, nenhum outro nodo pode ser utilizado. (De fato, o processo é um pouco mais complicado, pois cada modelo pode especificar quais nodos processar; assim alguns nodos podem ser processados mais de uma vez e alguns podem não ser processados).

O produto de todo esse processo é uma "árvore de resultado". Se a árvore de resultado está composta de objetos no formato XSL, então descreve como apresentar o código-fonte do documento.

A linguagem XSL trabalha em dois documentos separados. O primeiro aborda a sintaxe e a semântica para XSL, aplicando "folha de estilo" para transformar um documento. O segundo, XSLT, trata a XSL que formata os objetos, os seus atributos e como eles podem ser combinados.

### 3.4 A XSLT - eXtensible Style Language Transformation

Uma das poucas limitações existentes quanto à apresentação de documentos XML diz respeito à questão dos *browsers*. Alguns *browsers* ainda não suportam a linguagem XML, colocando assim um empecilho na apresentação desses documentos. Para acabar com tais

limitações a linguagem XSLT foi construída, possibilitando a conversão de documentos XML, em muitos outros tipos de documentos, por exemplo HTML, que é acessível para qualquer *browser* [SOO00].

XSL define o formato e a apresentação de documentos XML, e esta apresentação está dividida em dois processos: o primeiro é onde o documento passa por uma transformação estrutural, em que os elementos são selecionados, agrupados e reordenados; o segundo é o processo no qual os elementos resultantes do primeiro processo são apresentados em um *browser*, por exemplo [KAY00]. Esta divisão mostra que os processos são bastante independentes. Assim, XSL foi dividida em duas partes: a primeira, estipulada XSLT, para definir as transformações, e a segunda ficou ainda sendo chamada de XSL e trata das apresentações dos documentos.

XSLT foi projetada para ser usada como parte da linguagem XSL. Além de XSLT, XSL inclui um conjunto de regras em XML para especificar a formatação e o estilo de um documento XML, usando XSLT para descrever como o documento é transformado em outro documento [W3C99].

Em uma ação de XSLT, o processador XSL lê os dados do documento XML e os dados do arquivo XSL referenciado pelo documento XML. Baseado nas instruções que o processador XSL encontra no arquivo XML, ele produz um novo documento XML ou em outro formato.

### 3.5 Algumas funções da linguagem XSLT aplicadas ao estudo de caso

A linguagem XSLT possui inúmeras funções que são usadas para prover uma melhor apresentação dos documentos XML e que auxiliam no controle e estruturação dos documentos apresentados. Abaixo, de acordo com [HAR99] são descritas algumas delas.

#### 3.5.1 Templates

O uso de *templates* permite especificar como um determinado bloco de informações do documento XML será formatado, por exemplo: pode-se definir que o conteúdo de um elemento chamado "LIVRO" deve ser apresentado na forma de parágrafo toda vez que ele é encontrado pelo processador XSL, então se define um *template* para tal função:

Exemplo:

```
<xsl:template match="LIVRO">
  <p> <xsl:apply-templates/></p>
</xsl:template>
```

Isso significa que no documento XML, toda vez que o processador XSL encontrar um elemento "LIVRO", ele vai apresentar seu conteúdo em uma formatação definida pelo *template* "LIVRO". Isto é, o *template* associa o elemento "LIVRO" a um elemento <p>.

#### 3.5.2 Cláusula <xsl:for-each select>

Essa opção permite especificar transformações para uma determinada seleção obtida pela cláusula acima. A seleção é feita pela atribuição de um parâmetro que indica com quais elementos a cláusula vai interagir. Depois de especificar o parâmetro para a cláusula, pode-se informar como os atributos selecionados serão apresentados.

Exemplo:

```
<xsl:for-each select="BIB_BAS/LIVRO | BIB_COMP/LIVRO">
  <tr>
    <td><xsl:value-of select="AUTOR"/></td>
```

```

<td><b><xsl:value-of select="TITULO"/></b></td>
<td><xsl:value-of select="ANO_PUBLIC"/></td>
</tr>
</xsl:for-each>

```

O exemplo acima demonstra como fazer uma seleção de elementos que estão contidos dentro dos elementos BIB\_BAS/LIVRO e BIB\_COMP/LIVRO. No exemplo, os elementos AUTOR, TITULO, ANO\_PUBLIC serão selecionados e serão exibidos em forma de tabela.

### 3.6 Exibição de informações antes da implementação em XML

O documento de caracterização de disciplinas está armazenado em um banco de dados com marcações HTML que implicam em algumas limitações, tanto para consultas quanto para exibição do mesmo. Nesse item, será abordada a situação da exibição das informações na qual a tecnologia XML faz uma importante contribuição no tratamento da informação quando a mesma é exibida em um *browser*, por exemplo.

A Figura 3.1 apresenta uma parte da bibliografia do documento de caracterização das disciplinas da UNISINOS, marcada em HTML. A Figura 3.2 mostra a exibição da mesma em um *browser*.

```

<strong>Bibliografia básica</strong><br>
<P class=Biblio>BERNSTEIN, P. A. <B>Concurrency control and recovery in database
systems</B>. Reading: Addison-Wesley, 1987. 370 p.</P>
<P class=Biblio>CONNOLLY, Thomas M.; BEGG, Carolyn E. (Adapt.).<B> Database
systems: a practical approach to desing, implementation, and management. </B> 2. ed.
England: Addison-Wesley, 1999. 1094 p.</P>
<P class=Biblio>DATE, C. J. <B>Introducao a sistemas de bancos de dados</B>. 2. ed.
Rio de Janeiro: Campus, 1993. 513 p.</P>
...

```

Figura 3.1 - Referência bibliográfica marcada em HTML

```

Bibliografia básica
BERNSTEIN, P. A. Concurrency control and recovery in database systems. Reading:
Addison-Wesley, 1987. 370 p.
CONNOLLY, Thomas M.; BEGG, Carolyn E. (Adapt.). Database systems: a practical
approach to desing, implementation, and management. 2. ed. England: Addison-
Wesley, 1999. 1094 p.
DATE, C. J. Introdução a sistemas de bancos de dados. 4. ed. Rio de Janeiro: Campus,
1993. 513 p.
...

```

Figura 3.2 - Exibição da referência bibliográfica marcada em HTML

Na marcação apresentada na Figura 3.1 pode-se observar que não existe identificação semântica das informações, tem-se apenas informação de exibição misturada com o conteúdo, o que dificulta a leitura e o trabalho de uma máquina de consulta. Além disso, uma vez que usuário desejasse visualizar apenas as informações *autor, título da obra e ano de publicação*,

isso não seria possível pois a marcação HTML não oferece meios de atender a diferentes formas de visualização, dificultando o trabalho do desenvolvedor.

### 3.7 As folhas de estilo do estudo de caso

Para possibilitar a exibição dos documentos do estudo de caso, foi construída uma folha de estilos XSL, que diz ao interpretador como os elementos XML serão exibidos. A Figura 3.3 mostra uma folha de estilos para a bibliografia do documento de caracterização das disciplinas.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <xsl:template match="/">
    <html>
      <head>
        <title>Bibliografia do documento de caracterização das disciplinas da
UNISINOS.</title>
      </head>
      <body>
        <xsl:apply-templates select="BIBLIOGRAFIA"/>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="BIBLIOGRAFIA">
    <table border="2" bgcolor="yellow">
      <!--cabeçalho da tabela-->
      <th align="center"> Nome do(s) autor(es)</th>
      <th align="center"> Título do Livro</th>
      <th align="center">Ano de Publicação</th>
      <xsl:apply-templates select="BIB_BAS/LIVRO | BIB_COMP/LIVRO"/>
    </table>
  </xsl:template>
  <xsl:template match="BIB_BAS/LIVRO | BIB_COMP/LIVRO">
    <tr>
      <td><xsl:value-of select="AUTOR"/></td>
      <td><b> <xsl:value-of select="TITULO"/></b></td>
      <td><xsl:value-of select="ANO_PUBLIC"/></td>
    </tr>
  </xsl:template>
</xsl:stylesheet>
```

Figura 3.3 - Exemplo de uma folha de estilos XSL para exibir os elementos de uma bibliografia

A Figura 3.3 apresenta uma das contribuições deste trabalho, uma vez que possibilita a apresentação de apenas uma parte do conteúdo de uma bibliografia. Com a marcação do documento com tags HTML, caso o usuário desejasse visualizar apenas alguns itens de uma bibliografia, sua necessidade poderia não ser atendida. Entretanto, a marcação XML permite

uma flexibilidade capaz de satisfazer a necessidade apresentada, pois a folha de estilo está separada do documento XML e permite apresentar apenas aqueles elementos que o usuário solicitar.

Em um exemplo prático para visualização de uma bibliografia, o Centro de Ciências da Saúde poderia solicitar a apresentação de apenas o nome dos autores, o título do livro e o ano de publicação. O Centro de Ciências Exatas e Tecnológicas poderia solicitar a exibição do nome dos autores, do título do livro, e do número de páginas existentes na obra.

Essa flexibilidade é possível devido à possibilidade do desenvolvedor poder definir seus próprios elementos de acordo com a necessidade do seu domínio. A Figura 3.4 apresenta uma parte da bibliografia de uma disciplina marcada em XML e em seguida a Figura 3.5 exibe a mesma conforme poderia ser solicitada pelo Centro de Ciências da Saúde, por exemplo.

```

<BIBLIOGRAFIA>
<BIB_BAS>Bibliografia básica
<LIVRO>
  <AUTOR>BERNSTEIN, P. A. </AUTOR>
  <TITULO>Concurrency control and recovery in database systems</TITULO>
  <LOCAL> Reading</LOCAL>
  <EDITORA> Addison-Wesley</EDITORA>
  <ANO_PUBLIC>1987</ANO_PUBLIC>
  <NRO_PAGINAS>370 </NRO_PAGINAS>
</LIVRO>

<LIVRO>
  <AUTOR>DATE, C. J. </AUTOR>
  <TITULO>Introducao a sistemas de bancos de dados</TITULO>
  <NRO_EDICAO> 2</NRO_EDICAO>
  <LOCAL> Rio de Janeiro</LOCAL>
  <EDITORA> Campus</EDITORA>
  <ANO_PUBLIC>1993</ANO_PUBLIC>
  <NRO_PAGINAS>513</NRO_PAGINAS>
</LIVRO>
...

```

Figura 3.4 – Referência bibliográfica marcada em XML

| Nome do(s) autor(es) | Título do Livro   | Ano de Publicação |
|----------------------|---|-------------------|
| BERNSTEIN, P. A.     | <b>Concurrency control and recovery in database systems</b> | 1987              |
| DATE, C. J.          | <b>Introdução a sistemas de bancos de dados</b>             | 1993              |
| ...                  |   |                   |

Figura 3.5 – Exibição da referência bibliográfica marcada em XML (caso I)

É importante observar na Figura 3.5 que foram eliminadas as seguintes informações: local de publicação, nome da editora, número da edição e número de páginas da obra, restando apenas o conteúdo solicitado pelo usuário do Centro de Ciências da Saúde (caso I).



| Nome do(s) autor(es) | Título do Livro   | Nro. de Páginas |
|----------------------|---|-----------------|
| BERNSTEIN, P. A.     | <b>Concurrency control and recovery in database systems</b> | 370             |
| DATE, C. J.          | <b>Introdução a sistemas de bancos de dados</b>             | 513             |
| ...                  |   |                 |

Figura 3.6 – Exibição da referência bibliográfica marcada em XML (caso II)

Na Figura 3.6 são apresentadas as informações para atender um pedido do Centro de Ciências Exatas e Tecnológicas (caso II). Essa figura é bastante semelhante a Figura 3.5, entretanto ela exibe a informação número de páginas ao invés de editora. Os exemplos acima, caracterizam a flexibilidade permitida pelas folhas de estilo XSL, permitindo a exibição de partes de um documento conforme seja a necessidade do usuário.

Como foi apresentado neste capítulo, a tecnologia XML permite uma visualização das informações de um documento de diversas formas. Além disso, as informações de exibição estão em um arquivo separado do conteúdo, o que facilita a manutenção e a reutilização das mesmas.

O próximo capítulo apresentará o aspecto de busca de informações na Internet e a descrição de alguns exemplos de consultas a documentos XML utilizando a linguagem XQuery. Em seguida, para facilitar a representação computacional do conhecimento, será descrito o conceito de ontologia. Pelo fato das informações das disciplinas estarem disponíveis na Web, também será introduzido o aplicativo XML RDF.

## IV. BUSCA DE INFORMAÇÃO

Este capítulo descreve alternativas para consultas em documentos XML. São apresentados alguns exemplos de busca de informações em documentos etiquetados em XML utilizando a linguagem XQuery.

O conceito de *Semantic Web*, segundo [BER01], facilitar a estruturação e o acesso a documentos na Web, tanto por máquinas, quanto por pessoas, permitindo uma melhor identificação do significado do conteúdo dos mesmos em um domínio, também é apresentado neste capítulo. Os componentes da *Semantic Web*, ontologia e RDF, serão descritos nas próximas subseções. No final do capítulo são apresentadas algumas consultas em RDF utilizando a linguagem RQL.

### 4.1 Consulta aos documentos XML

Segundo [ROB00] e [CHA01], XML é uma linguagem de marcação extremamente versátil, capaz de etiquetar o conteúdo da informação de diversas fontes de dados incluindo documentos estruturados e semi-estruturados, banco de dados relacional e repositórios de objetos. Com essa versatilidade, e com o crescente número de informações armazenadas em XML, torna-se bastante importante a habilidade de consultar os dados marcados em XML. [BON01] fez um estudo comparativo entre seis linguagens de consulta, a saber: (Lorel, Quilt, XML-QL, XML-GL, XQL e XSLT) onde as mesmas são examinadas através das características de uma linguagem de consulta a banco de dados.

Entretanto, [CHA01] está fazendo um estudo sobre a linguagem XQuery (ver seção 4.1.1) que não é citada em [BON01] mas é derivada da linguagem de consulta XML chamada Quilt, que integra as características das várias outras linguagens citadas em [BON01].

Este trabalho faz uso da linguagem XQuery, apresentada como um *working draft* pelo W3C, para consultar os documentos marcados em XML.

#### 4.1.1 Xquery

Segundo [CHA01], XQuery é desenvolvida para ser uma linguagem pequena e facilmente implementável em que as consultas são concisas e facilmente interpretáveis. Ela também é flexível o suficiente para consultar um amplo espectro de fontes de informações XML, incluindo banco de dados e documentos.

A Figura 4.1 apresenta um exemplo de consulta utilizando XQuery. O objetivo da mesma é percorrer o documento de caracterização das disciplinas, e encontrar os livros que são indicados na bibliografia das disciplinas Banco de Dados I e Banco de Dados II, cujo autor possua o em seu nome a string *Date*. Além disso, o usuário pode desejar visualizar os títulos dos livros selecionados em ordem descendente de ano de publicação, sendo necessário a exibição do mesmo no resultado.

```

<results>
  {FOR $L IN document("banco1.xml")//LIVRO UNION
    document("banco2.xml")//LIVRO
    WHERE contains($L/AUTOR/text(), "DATE")}
  RETURN
    <TITANO>
      {$L /TITULO}
      {$L /ANO_PUBLIC}
    </TITANO>
  SORTBY (ANO_PUBLIC DESCENDING)}
</results>

```

Figura 4.1 – Exemplo de uma consulta utilizando a linguagem XQuery

O resultado da consulta é apresentado na Figura 4.2, onde pode-se constatar que existem três livros, cujo autor é *Date*, que são indicados na bibliografia das disciplinas Banco de Dados I e Banco de Dados II.

```

<?xml version="1.0"?>
<results xmlns:ino="http://namespaces.softwareag.com/tamino/response2"
xmlns:xql="http://metalab.unc.edu/xql">

  <TITANO>
    <TITULO>An introduction to database systems</TITULO>
    <ANO_PUBLIC>2000</ANO_PUBLIC>
  </TITANO>
  <TITANO>
    <TITULO>Relational database writings 1994-1997</TITULO>
    <ANO_PUBLIC>1998</ANO_PUBLIC>
  </TITANO>
  <TITANO>
    <TITULO>Introdução a sistemas de banco de dados</TITULO>
    <ANO_PUBLIC>1993</ANO_PUBLIC>
  </TITANO>

```

Figura 4.2 – Resultado de uma consulta utilizando a linguagem XQuery

O resultado da Figura 4.2 apresenta mais uma vantagem em relação a marcação HTML, pois com o conteúdo melhor identificado, a consulta ao documento pode ser mais flexível e apresentar resultados mais personalizados. A marcação HTML não permitia uma consulta sobre o ano de publicação de um livro em uma bibliografia de qualquer disciplina, uma vez que a tag *<P>* que identifica o livro acrescenta apenas informação de exibição informando o *browser* para exibir o conteúdo do livro em forma de parágrafo.

A consulta descrita na Figura 4.1 pode ser estendida de diversas maneiras, entre elas, poderia se acrescentar a editora e/ou o local de publicação dos livros aumentando o conjunto de informações para um usuário que tivesse interesse em adquirir algum livro publicado pelo autor *Date*.

Além da consulta relacionando duas disciplinas, é possível pesquisar em um conjunto maior de informações. Na Figura 4.3, foi utilizado o *corpus* deste trabalho com o objetivo de selecionar todas as disciplinas que contenham a palavra “lógica” no conteúdo programático.

```
<result>
{
  FOR $b1 IN document("disp/orgbanco.xml")//.
    union document("disp/arqmicroproc.xml")//.
    union document("disp/banco1.xml")//.
    union document("disp/banco2.xml")//.
    ...
    ...
  WHERE contains($b1//CONTEUDO_PROG_ITEM, "lógica")
  RETURN
    $b1/NOME_DISCIPLINA
}
```

Figura 4.3 – Consulta em XQuery utilizando o *corpus* do estudo de caso

Conforme a implementação em HTML, não seria possível direcionar a pesquisa para a parte de conteúdo programático e a máquina de busca retornaria todos os nomes de disciplinas com ocorrências da palavra “lógica” encontradas no documento inteiro e não somente no conteúdo programático. Entretanto, o resultado da Figura 4.4, que está baseado na marcação XML, apresenta somente as disciplinas que possuem a palavra “lógica” no conteúdo programático.

```
<?xml version="1.0"?>

<result xmlns:ino="http://namespaces.softwareag.com/tamino/response2"
xmlns:xql="http://metalab.unc.edu/xql/">

  <NOME_DISCIPLINA>Conceitos Básicos de Informática</NOME_DISCIPLINA>
  <NOME_DISCIPLINA>Inteligencia Artificial</NOME_DISCIPLINA>
  <NOME_DISCIPLINA>Organização de Computadores</NOME_DISCIPLINA>
  <NOME_DISCIPLINA>Paradigmas de Programação</NOME_DISCIPLINA>
  <NOME_DISCIPLINA>Programação em Lógica</NOME_DISCIPLINA>
</result>
```

Figura 4.4 – Resultado da consulta em XQuery utilizando o *corpus* do estudo de caso

A implementação do documento de caracterização das disciplinas em XML permite atender diversos tipos de consultas. Outro exemplo, está na Figura 4.5, que permite encontrar quais disciplinas tem a maior parte de suas aulas práticas. Para atender essa consulta, a máquina de busca acessa somente o conteúdo do elemento “NRO\_HORAS\_AULA\_PRA” de cada disciplina. Se o número for superior a 30, então é selecionado o conteúdo do elemento “NOME\_DISCIPLINA” para ser exibido ao usuário.

```

<result>
{
  FOR $d IN document("disp/orgbanco.xml")//.
    union document("disp/arqmicroproc.xml")//.
    union document("disp/banco1.xml")//.
    union document("disp/banco2.xml")//.
    ...
    ...
  WHERE $d//NRO_HORAS_AULA_PRA > 30
  RETURN
    $d//NOME_DISCIPLINA
}

```

Figura 4.5 – Consulta em XQuery para verificar as disciplinas do *corpus* que possuem a maior parte das aulas práticas

```

<?xml version="1.0"?>

<result xmlns:ino="http://namespaces.softwareag.com/tamino/response2"
xmlns:xql="http://metalab.unc.edu/xql/">

  <NOME_DISCIPLINA>Conceitos Básicos de Informática</NOME_DISCIPLINA>
  <NOME_DISCIPLINA>Empreendimentos em Informática</NOME_DISCIPLINA>
  <NOME_DISCIPLINA>Paradigmas de Programação</NOME_DISCIPLINA>
  <NOME_DISCIPLINA>Seminários de Informática</NOME_DISCIPLINA>

```

Figura 4.6 – Disciplinas do *corpus* que possuem a maior parte das aulas práticas

A Figura 4.6 apresenta o resultado da consulta, e permite ao usuário concluir que das trinta disciplinas selecionadas para compor o *corpus* do trabalho, existem quatro disciplinas com mais de trinta horas-aula práticas durante um semestre.

Outra vantagem dessa abordagem é possibilitar consultas entre professores e disciplinas, ou seja, permitir ao usuário consultar as disciplinas ministradas por um professor ou saber quais professores ministram quais disciplinas.

A Figura 4.7 apresenta uma consulta que torna possível o objetivo acima, pois a máquina de busca percorre todas as disciplinas selecionando apenas o conteúdo do elemento “NOME\_DISCIPLINA”<sup>2</sup>.

<sup>2</sup> No “FOR” mais interno, a busca é feita sobre um documento que contém informações de professores e disciplinas e cada ocorrência em que o nome da disciplina for igual ao conteúdo do elemento “NOME\_DISCIPLINA” a pesquisa retornará um resultado com o nome do professor e o nome da disciplina ministrada pelo mesmo, conforme Figura 4.8.

```

FOR $d IN document("orgbanco.xml")//NOME_DISCIPLINA
  union document("arqmicroproc.xml")//NOME_DISCIPLINA
  union document("banco1.xml")//NOME_DISCIPLINA
  union document("banco2.xml")//NOME_DISCIPLINA

RETURN
  FOR $p IN document("denise.xml")//[NOME_DISCIPLINA= $d]
  RETURN
    <PROFESSOR_DISCIPLINA>
      {$p //NOME_PROF}
      {$d}
    </PROFESSOR_DISCIPLINA>

```

Figura 4.7 – Consulta em XQuery relacionando professores e disciplinas do *corpus*.

```

<PROFESSOR_DISCIPLINA>
  <NOME_PROF>DENISE BANDEIRA DA SILVA</NOME_PROF>
  <NOME_DISCIPLINA>Organização de Banco de Dados</NOME_DISCIPLINA>
</PROFESSOR_DISCIPLINA>

<PROFESSOR_DISCIPLINA>
  <NOME_PROF>SANDRO RIGO</NOME_PROF>
  <NOME_DISCIPLINA>Arquitetura de Microprocessadores</NOME_DISCIPLINA>
</PROFESSOR_DISCIPLINA>

<PROFESSOR_DISCIPLINA>
  <NOME_PROF>DENISE BANDEIRA DA SILVA</NOME_PROF>
  <NOME_DISCIPLINA>Banco de Dados I</NOME_DISCIPLINA>
</PROFESSOR_DISCIPLINA>

```

Figura 4.8 – Resultado da consulta em XQuery relacionando professores e disciplinas do *corpus*.

A possibilidade de combinação ilustrada pelo resultado apresentado na Figura 4.8, é uma das contribuições que este trabalho faz para auxiliar a busca de informações na Web, uma vez que é possível acessar as partes de um documento identificadas semanticamente com as etiquetas XML.

Esta seção apresentou uma alternativa de busca de informações que otimiza as respostas a consultas a documentos disponíveis na Internet, através da tecnologia XML associada à linguagem de consulta XQuery.

Apesar das consultas acima citadas gerarem resultados satisfatórios para tornar a busca mais inteligente<sup>3</sup>, faz-se necessário que a marcação de documentos XML esteja baseada em

<sup>3</sup> Busca de informações baseada no contexto de um determinado domínio.

uma ontologia que defina os conceitos relevantes de um determinado domínio, neste caso, universidade.

Devido a isso, a próxima seção apresentará os conceitos de ontologia e RDF que definem alguns metadados inerentes ao contexto deste trabalho. No final do capítulo, serão apresentados alguns exemplos de consulta em RDF utilizando a linguagem RQL que permite pesquisas sobre metadados do domínio.

## 4.2 Semantic Web

A *World Wide Web* foi originalmente construída para concepção humana, entretanto ela deve ser legível para as máquinas também. É muito difícil automatizar qualquer coisa na *Web*, e pelo volume de informações que ela contém (mais de um bilhão de documentos) não é possível gerenciá-la manualmente.

Esta seção apresenta dois componentes, Ontologias e RDF, que facilitam a estruturação e o acesso a documentos na *Web*, tanto por máquinas, quanto por pessoas, permitindo uma melhor identificação do significado do conteúdo dos mesmos em um domínio.

[BER01] afirma que a *Semantic Web* é composta por três componentes, a saber: XML, Ontologias e RDF. A tecnologia XML já foi apresentada no capítulo II. Ontologias e RDF serão descritas nas próximas subseções.

### 4.2.1 Ontologias

Na Ciência da Computação, o termo "ontologia", vem sendo aplicado desde o início da década de 1990 na área de inteligência artificial para representação computacional de conhecimento em áreas como engenharia de conhecimento e processamento de linguagem natural. [NOY01] diz que o desenvolvimento de uma ontologia é similar à definição de um conjunto de dados e da estrutura deles para outros programas utilizarem.

Para [GRU93] uma ontologia é uma especificação formal e explícita de uma conceitualização compartilhada. [FEN00b] descreve esse conceito em partes, afirmando que uma "conceitualização" refere-se para um modelo abstrato de algum fenômeno no mundo que identifica conceitos relevantes daquele fenômeno. [GUA97] ainda comenta que uma "conceitualização" explica o significado pretendido dos termos usados para indicar relações relevantes. "Explícito" significa que os tipos de conceitos usados e as restrições para esses conceitos são definidas explicitamente. "Formal" refere-se ao fato de que uma ontologia deve ser legível para as máquinas. "Compartilhada" reflete a noção de que uma ontologia captura o conhecimento consensual, isto é, o conhecimento não é restrito a algum indivíduo, mas aceito por um grupo.

Para a Inteligência Artificial e os pesquisadores *Web*, uma ontologia é um documento ou arquivo que formalmente define as relações entre os termos. O tipo mais típico de ontologia para a Internet tem uma taxonomia e um conjunto de regras de inferência [BER01]. A taxonomia define as classes de objetos e as relações entre eles. As regras de inferência fornecem um poder maior a ontologia, por exemplo, uma ontologia pode expressar a regra "se o código de uma cidade está associado ao código de um estado, e um endereço usa aquele código da cidade, então aquele endereço tem o código do estado associado". Pelo fato deste trabalho propor um estudo sobre organização, apresentação e busca de informações através da Internet, é essencial a utilização de uma ontologia.

Ontologias vêm sendo aplicadas no gerenciamento de dados semi-estruturados como um suporte semântico para o acesso a determinadas informações de interesse presentes em um conjunto de fontes semi-estruturadas. Por exemplo, [MEL00b] apresenta uma solução para acessar dados marcados com XML usando ontologias como suporte para a integração e a

consulta a fontes de dados XML. Uma importante vantagem neste contexto é que a ontologia provê uma interpretação semântica unificada para diferentes representações de dados semi-estruturados referentes a um mesmo domínio.

Uma vez definido um nível conceitual ontológico, outra vantagem a salientar é que consultas que levam em conta a semântica do domínio podem ser formuladas. Esta vantagem é significativa, pois grande parte das linguagens de consulta para dados semi-estruturados baseiam-se na especificação de padrões a serem percorridos em uma estrutura de grafo, considerando apenas a organização hierárquica dos objetos e não a sua semântica. Por exemplo, no relacionamento entre dois recursos *Web*, tais como uma URL de um professor e a URL de uma disciplina, tem-se uma propriedade *ministra* que relaciona semanticamente os dois recursos permitindo a linguagem de consulta explorar essa semântica.

A ontologia abstrai a organização hierárquica dos objetos, permitindo que a intenção de consulta do usuário esteja concentrada nos conceitos do domínio e seus relacionamentos e não nas estruturas lógicas de representação de dados semi-estruturados. [STA01] conclui essa idéia afirmando que a ontologia permite o acesso inteligente aos documentos na *Web* e infere ou deduz o conhecimento implícito das regras e fatos declarados explicitamente na ontologia.

[MAE01] afirma que o sucesso da *Semantic Web* depende fortemente da proliferação de ontologias, que requer uma construção rápida e fácil das mesmas. Entretanto, [CUI00] comenta que o desenvolvimento de uma ontologia processável por máquinas é intrinsecamente difícil, pois a semântica de um termo varia de um contexto para outro.

A Figura 4.9 proposta por [FEN00] estrutura em camadas os componentes da *Semantic Web*.

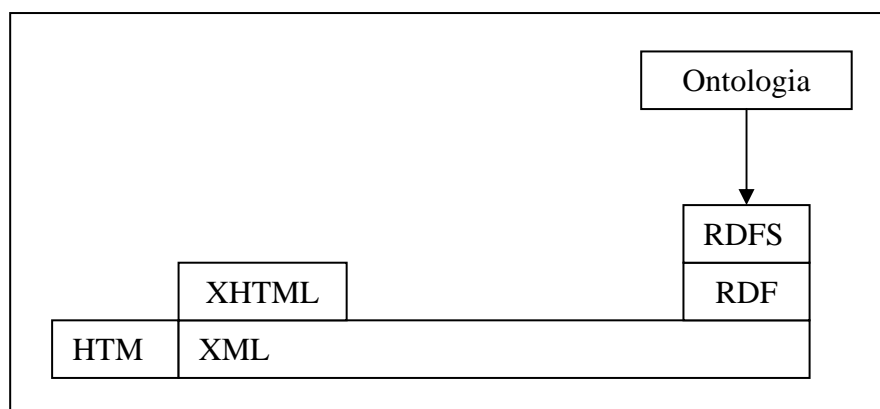


Figura 4.9 – O modelo de linguagem em camada para a *Web*

Na Figura 4.9, fica claro que a partir da ontologia pode-se gerar um RDFS (ver seção 4.2.2.5) que será instanciado pelo RDF. Utiliza-se a tecnologia XML para codificar o aplicativo RDF (ver seção 4.2.2).

[FEN99] afirma que o uso de ontologias torna explícita a semântica de páginas *Web*. Assim, a ontologia construída para este trabalho pretende prover uma melhor identificação para o conteúdo presente no documento de caracterização de disciplinas da UNISINOS.

#### 4.2.1.1 Tipos de Ontologia

Ontologias são usualmente classificadas em quatro tipos, de acordo com a Figura 4.10[GUA98]. Uma ontologia de nível superior descreve conceitos genéricos, como espaço, tempo, objeto, ação, entre outros, sendo utilizada na definição de meta-domínios. Uma ontologia de domínio e uma ontologia de tarefa descrevem uma conceitualização para,



respectivamente, um domínio genérico (universidades) ou uma tarefa genérica (especificação de currículo base, por exemplo, MEC), especializando conceitos da ontologia de nível superior. Uma ontologia de aplicação, específica para uma certa atividade dentro de um domínio, define regras a serem seguidas por conceitos do domínio quando uma certa tarefa é realizada (marcação do documento de caracterização das disciplinas da UNISINOS).

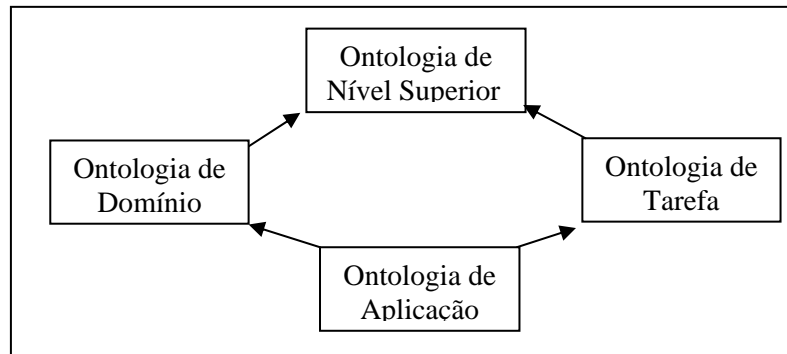


Figura 4.10 - Tipos de ontologias e seus relacionamentos.

#### 4.2.1.2 Benefícios da Ontologia

[JAS99] afirma que as ontologias oferecem um benefício unânime, que é a reutilização do conhecimento. Além disso, uma ontologia provê vantagens de documentação, manutenção e confiabilidade, pois as definições dos termos relevantes a um determinado domínio estarão localizados em apenas um local de maneira bem definida.

No contexto deste trabalho, o uso de uma ontologia agregará todos os benefícios citados acima, uma vez que serão documentados os conceitos relevantes referentes ao domínio universidade, tais como: área de conhecimento, disciplinas e professores. As restrições aos conceitos também serão explicitadas na ontologia, por exemplo, a propriedade “ministra” é aplicada apenas a professores e disciplinas tendo a propriedade inversa “é ministrada por”.

#### 4.2.1.3 Construção da Ontologia

Em [MAE01], os autores discorrem sobre três problemas que devem ser levados em consideração na elaboração de ontologias, a saber: tempo, dificuldade e confiança. Em outras palavras, é importante responder as seguintes questões:

- É possível desenvolver uma ontologia rapidamente?
- Como o engenheiro sabe que a ontologia está correta?
- É difícil construir uma ontologia?

No desenvolvimento da ontologia para este trabalho, foram encontrados os três problemas.

Idealmente o desenvolvimento de uma ontologia é um processo coletivo que deve levar em consideração as mais diversas visões de um domínio. No caso de uma universidade, a visão dos administradores, dos professores e dos demais empregados é importante para o desenvolvimento da mesma. Por isso, a construção de uma ontologia é uma tarefa que consome bastante tempo.

Quanto à ontologia estar correta, foram estudados os principais termos relevantes do documento de caracterização das disciplinas, e em seguida, houve um consenso entre dois professores e uma bolsista de iniciação científica da área de lingüística, mais o autor deste trabalho, sobre a definição dos termos e relacionamentos. [NOY01] afirma que não existe um modo correto para modelar um domínio - existe sempre alternativas viáveis. A melhor

solução quase sempre depende da aplicação que se tem em mente e as extensões que a mesma poderá ter.

Para este trabalho foram estudadas duas ferramentas que permitem a elaboração de ontologias: *Prótege*<sup>4</sup> e OilEd<sup>5</sup>. Inicialmente a ontologia foi construída com a ferramenta OilEd que supriu as necessidades da aplicação. Em seguida, optou-se por experimentar o software *Prótege*, largamente utilizado na comunidade científica, que também apresentou um bom desempenho. Uma das facilidades que ambas ferramentas apresentam é a possibilidade de exportar a ontologia em diversos formatos, tais como: rdf *schema*, html, texto, entre outros.

Pelo grande número de pessoas com diversas formações e pela quantidade de informações envolvidas, a construção de uma ontologia é uma tarefa complexa. Dado o escopo de um trabalho de conclusão, a ontologia construída foi para apenas uma pequena parte, a caracterização das disciplinas da UNISINOS, do domínio universidade.

O Anexo 2, apresenta a ontologia para o contexto das disciplinas da UNISINOS.

#### 4.2.1.4 Ontologia e RDF schema

A ferramenta *Prótege*, utilizada para o desenvolvimento da ontologia apresentada no Anexo 2, possibilita a exportação dos dados para o formato RDFS. Assim, foram eliminados alguns *slots* (relações) da ontologia descrita no anexo 2 para facilitar a representação da Figura 4.14<sup>6</sup>, restando as relações *ministra*, *cont\_prog* e *metodologia*.

[BRA01] afirma que RDF *Schema* é um sistema de classes extensível e genérico que pode ser utilizado como base para esquemas de um domínio específico. Esses esquemas podem ser compartilhados e estendidos através de refinamento de subclasses. Além disso, definições de metadados podem ser reutilizadas através do compartilhamento de esquemas.

Para [BRI00], o papel do RDF *Schema* é prover um mecanismo para definir os relacionamentos entre propriedades e recursos declarados no RDF. Comunidades que descrevem recursos requerem a habilidade de dizer certas coisas sobre certos tipos de recursos. Para descrever recurso sobre uma bibliografia, por exemplo, atributos descritivos incluindo *autor*, *título* e *ano de publicação* são comuns. A declaração destas propriedades (atributos) e a semântica correspondente delas são definidas no contexto do RDF como um RDF *Schema*. O *schema* define não somente as propriedades do recurso (*autor*, *título*, *ano de publicação*, entre outros), mas pode também definir os tipos de recursos que estão sendo descritos (*livros*, *pessoas*, *disciplinas*, por exemplo).

#### 4.2.2 XML e RDF - *Resource Description Framework* (estrutura de descrição de recursos)

O RDF é um aplicativo XML recomendado pelo W3C (*World Wide Web Consortium*) que usa notação XML como sintaxe de codificação para descrição de metadados (ver seção 4.2.2.1). Os elementos construtivos do XML são usados para descrever o modelo de dados do RDF. Isso permite que as descrições sejam analisadas sintaticamente por analisadores sintáticos XML e passadas para aplicativos que entendam o XML.

Segundo [PIT00], o principal objetivo do RDF, é o de facilitar o intercâmbio de informações (que podem ser interpretadas por máquinas) entre aplicativos via *Web*. [FEN00a] também acrescenta mais um objetivo do RDF que é adicionar semântica formal para a *Web*.

<sup>4</sup> Disponível para download em <http://www.smi.stanford.edu/projects/protege/>

<sup>5</sup> Disponível para download em <http://img.cs.man.ac.uk/oil/>

<sup>6</sup> O schema apresentado na Figura 4.14 é descrito no Anexo 3 e validado no Anexo 4 pelo parser RDF que está disponível em <http://www.w3c.org>.

Um dos alicerces importantes do RDF é sua habilidade de automatizar o processamento de recursos *Web*.

#### 4.2.2.1 Terminologia e conceitos importantes

[PIT00] apresenta a terminologia e conceitos relacionados a RDF que serão abordados nesta seção.

**Metadados** são dados sobre dados, informações sobre informações. Fichas de bibliotecas contém entradas de metadados para livros e outros recursos encontrados em uma biblioteca.

Por exemplo, eis alguns metadados referentes ao conteúdo deste capítulo:

- Título – Busca de Informação
- Autor – Marcirio Silveira Chaves
- Descrição – Uma visão sobre consultas XML

Agora, eis alguns metadados sobre o trabalho em si:

- Título – Gerenciamento e Acesso a Documentos na Internet através de XML, RDF e Ontologia
- Autor – Marcirio Silveira Chaves
- Assunto – XML, RDF e Ontologia

O primeiro conjunto de metadados é específico para este capítulo, ao passo que, o segundo, descreve o trabalho que contém o capítulo. Quando aplicados especificamente no contexto do RDF, os metadados são dados sobre recursos da *Web*. O grande objetivo do RDF é descrever metadados de forma consistente. Especificamente, em uma página *Web* as imagens podem ter seus próprios metadados, seguidos da página em si, o conjunto de páginas ao qual esta página pertence, o site inteiro e a *World Wide Web*.

Por exemplo, a página *Web* que tem informações sobre a disciplina Trabalho de Conclusão tem metadados como a página do Centro de Ciências Exatas e Tecnológicas, seguida do site institucional da UNISINOS<sup>7</sup> e da *World Wide Web*.

**Grafos** - São usados no mundo RDF (na ciência da computação em geral) como um mapa físico de dados. Os grafos mostram os relacionamentos entre pequenas quantidades de dados por meio do uso de conectores.

**Nós** - São instâncias de dados em um grafo. Os grafos contêm pontos de dados (nós) e representam os relacionamentos entre eles.

**ISO 10646** – É o *International Standards Organization* (organização internacional de padrões) para o UCS - *Universal Character Set* (conjunto universal de caracteres), que é o equivalente ao Unicode.

As expressões RDF são apresentadas de forma diferente, independente da sintaxe usada, ou seja, transformadas em valores de string ISO 10646 e são esses valores de strings que são comparados. As expressões RDF são as mesmas se os valores de string são coincidentes.

**Espaço de nome** - É uma tecnologia XML avançada que usa PIs – *Processing Instructions* (instruções de processamento) para associar nomes exclusivos a URIs em um documento. Espaços de nomes permitem que seja usado um esquema de atribuição de nomes ao longo de um conjunto de documentos. Eles também permitem usar repetidamente partes de DTDs bem-elaboradas.

**Esquema** - É constituído de conceitos recolhidos de DTDs, programação orientada a objetos e bancos de dados relacionais. Os esquemas vão além das DTDs, já que fornecem informações adicionais sobre herança e tipos de dados em documentos. Pelo fato de usarem a mesma sintaxe que os tipos de dados, mas não serem limitados à criação de DTDs, um único

---

<sup>7</sup> <http://www.unisinos.br>

esquema pode ser utilizado ao longo de vários conjuntos de dados organizados. Os esquemas usam espaços de nomes para criar componentes reutilizáveis que podem ser transferidos de um esquema para outro.

#### 4.2.2.2 Declarações RDF

Um documento ou elemento RDF faz declarações sobre recursos. Uma declaração diz que um certo recurso tem uma ou mais propriedades. Cada propriedade tem um tipo (que é um nome) e um valor. O valor de uma propriedade pode ser um literal, tal como uma string, um número, ou o valor pode ser outro recurso.

Segundo [LAS99] uma declaração é composta de três partes:

1 - Recurso – Qualquer objeto que é identificável unicamente por um URI *Uniform Resource Identifier* (Identificador Uniforme de Recurso). URIs são um superconjunto das URLs *Uniform Resource Locators* (Localizador Uniforme de Recurso), mas os URIs podem também identificar livros, elementos em uma página, shows de televisão e pessoas individuais, por exemplo.

2 - Propriedade – É uma característica, um atributo ou um relacionamento específico de um recurso, isto é, uma propriedade expressa os relacionamentos dos valores associados com os recursos.

3 - Valores – Devem ser atômicos na natureza (string de texto ou números, por exemplo) ou outros recursos, os quais podem ter suas próprias propriedades. Uma coleção destas propriedades que se referem ao mesmo atributo é chamada de descrição.

Essas três partes da declaração são chamadas respectivamente, sujeito, predicado e objeto. O recurso sendo descrito é o sujeito. A propriedade usada para descrever o recurso é o predicado, e o valor da propriedade, é o objeto da declaração.

Por exemplo, uma declaração RDF diz: A página *Web* <http://inf.unisinos.br/cursos/graduacao/as/mostra.php?curso=6074> tem a disciplina Trabalho de Conclusão. Aqui o *recurso* é a página *Web* “<http://inf.unisinos.br/cursos/graduacao/as/mostra.php?curso=6074>” e a *propriedade* disciplina deste recurso tem o *valor* “Trabalho de Conclusão”. A Figura 4.11 demonstra um modo comum de representar essa declaração RDF.

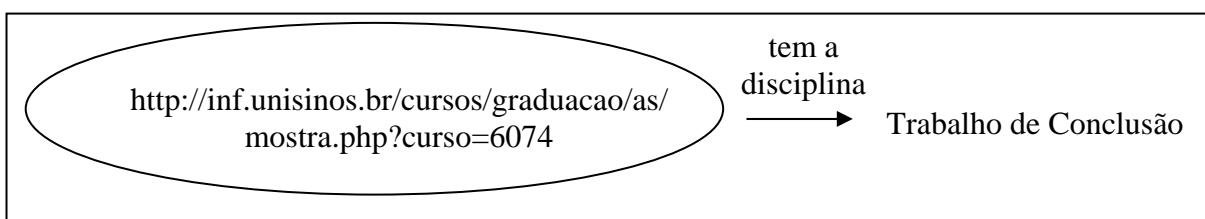


Figura 4.11 – Representação de uma declaração RDF.

```

<rdf:RDF xmlns:rdf="http://www.inf.unisinos.br">
  <rdf:Description about = "http://inf.unisinos.br/cursos/graduacao/as/mostra.php?curso=6074">
    <disciplina> Trabalho de Conclusão </disciplina >
  </rdf:Description>
</rdf:RDF>
  
```

Figura 4.12 - Declaração RDF marcada no formato XML.

Na Figura 4.12, a declaração RDF é decomposta em código XML como um elemento *Description*. Cada *propriedade* do recurso sendo descrito é um elemento filho do elemento *Description*. O conteúdo do elemento filho é o *valor da propriedade*.

#### 4.2.2.3 A utilização do RDF

Os mecanismos de busca e recuperação textual, de acordo com [PIT00], indexam apenas uma fração do número total de documentos na *Web*; a cobertura de qualquer mecanismo é significativamente limitada. A *Web* está em busca de uma maneira de descrever recursos, e o XML é extensível o bastante, tanto é, que foi usado para criar um aplicativo (RDF) que pudesse fazer isso.

O RDF, segundo [PIT00], funciona com todos os tipos de dados eletrônicos, e ele possui uma variedade de aplicações, entre as quais, podemos destacar: a identificação consistente de direitos autorais, marcas registradas e outras informações de propriedade intelectual em recursos da *Web*; a catalogação avançada de recursos e seus relacionamentos dentro de um único sistema ou entre vários sistemas; e a pesquisa mais acurada de arquivos de dados, pois os dados em si estão melhor identificados.

Além dessas aplicações, o RDF também permite a filtragem mais precisa de dados para sistemas de avaliação de conteúdo mais viáveis, a capacidade de representar conjuntos inteiros de documentos e como um único e grande documento, quando apropriado, e o estabelecimento de relacionamentos seguros entre documentos e computadores para facilitar o intercâmbio de idéias e de recursos, bem como de *E-commerce* [PIT00].

#### 4.2.2.4 RDF(S)

RDF(S) é um modelo abstrato de dados que define relacionamentos entre recursos. Para exemplificar um modelo de dados incorporado em um RDF(S), é necessário visualizarmos o mesmo em um RDF, conforme a Figura 4.13.

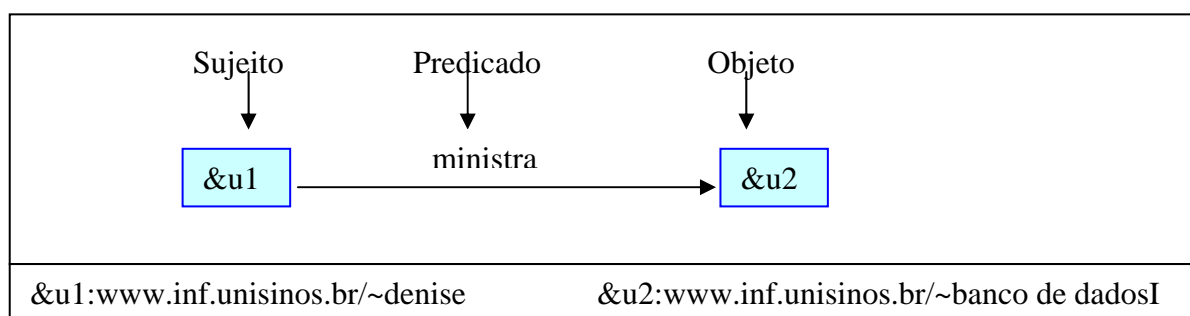


Figura 4.13 – Modelo de dados RDF.

A Figura 4.13, retoma os conceitos de recurso, propriedade e valor, ou melhor, sujeito, predicado e objeto, sendo o sujeito representado pelo recurso *&u1*, o predicado pela propriedade *ministra* e o objeto pelo recurso *&u2*.

Para facilitar a exploração das propriedades de orientação a objetos, tais como reuso e herança, é apresentado o mesmo modelo da Figura 4.14 ampliado e incorporado a um RDF(S) na Figura 4.13.

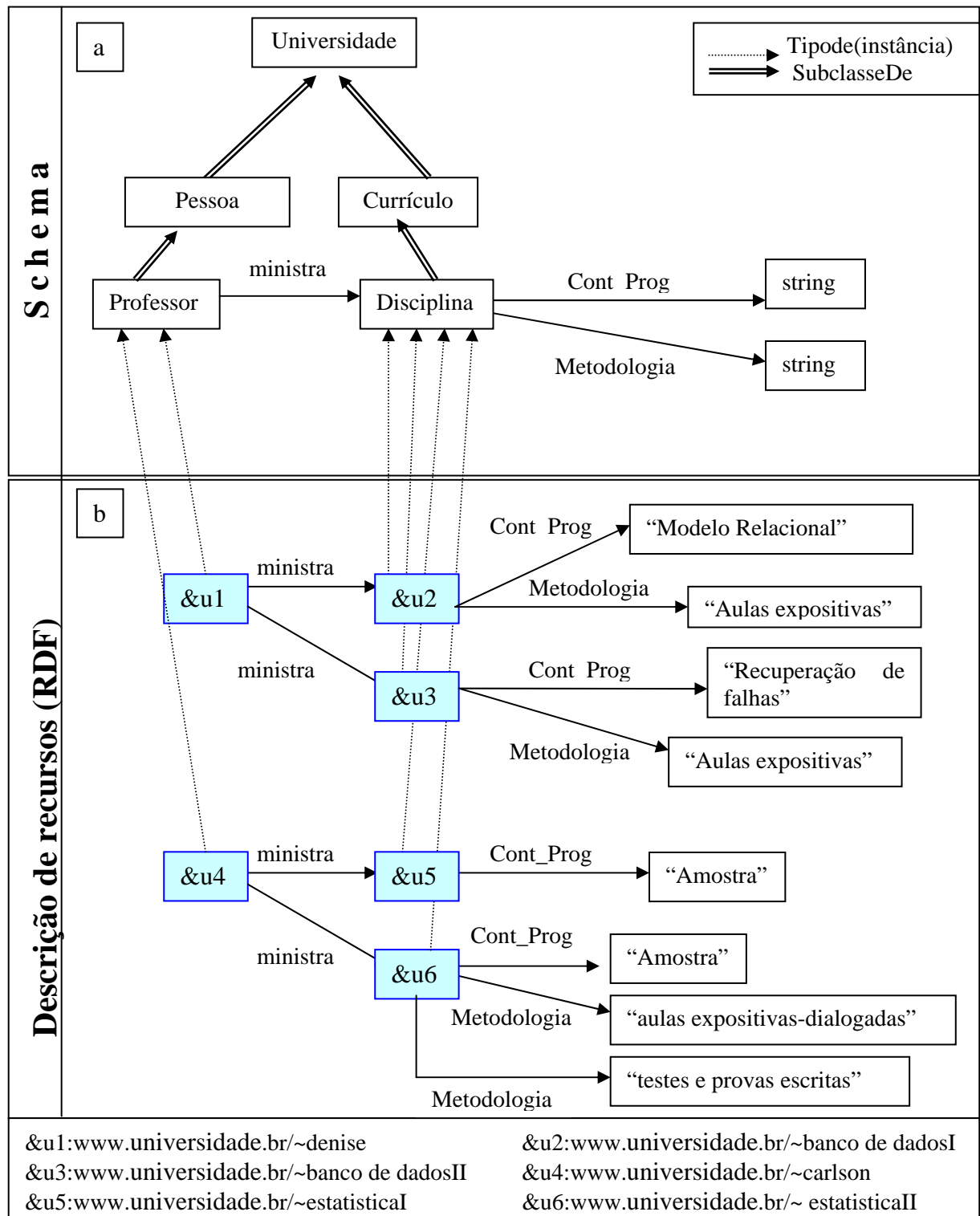


Figura 4.14 – Modelo de dados RDF incorporado em um *schema*.

Os modelos das Figuras 4.13 e 4.14 estão baseados em [STA00b] e [KAR00a], cuja proposta é apresentar uma abordagem para modelar ontologias em RDF(S). Em (a), é apresentada a camada *schema* que define algumas classes e relações relevantes para o contexto do documento de caracterização das disciplinas. Em (b), tem-se os dados atuais que

fazem parte do documento de estudo de caso. A seguir, é explicado em detalhes o modelo da Figura 4.14.

A classe mais geral no *schema* é *Universidade*. Ela tem duas subclasses *Pessoa* e *Currículo*. A classe *Pessoa*, denota o conjunto de todas as classes em um sentido orientado a objeto. Isso significa que a classe *Professor* é uma subclasse da meta-classe *Pessoa*.

O *schema* define uma propriedade especial *SubclasseDe*, que define o relacionamento das subclasses entre as classes. Visto que, *SubclasseDe* é transitiva, definições são herdadas pelas classes mais específicas das classes mais genéricas, e os recursos que são instâncias de uma classe, são automaticamente instâncias de todas as superclasses dessa classe. No *schema*, é proibido que qualquer classe seja uma *SubclasseDe* dela mesma ou uma de suas subclasses.

#### 4.2.2.5 Consulta em RDF

Após a descrição dos três componentes da *Semantic Web*, XML, RDF e ontologia, é importante explicitar que a mesma não é uma *Web* separada, mas uma extensão da *Web* corrente, em que as informações tem um significado bem definido, capacitando os computadores e as pessoas a trabalharem em cooperação [BER01]. [HAR01] acrescenta que a comunicação entre máquinas e pessoas será mais eficiente devido à definição de um domínio comum e compartilhado.

Uma maneira de possibilitar a comunicação entre máquinas é através da representação de conhecimento em RDF, que permite a exploração de consultas baseadas em metadados. A seção 4.2.2.6 introduz alguns exemplos de consulta utilizando a linguagem RQL.

#### 4.2.2.6 A linguagem RQL

RQL é uma linguagem de consulta para descrições e *schemas* RDF[KAR00a]. Segundo [ALE00], RQL permite-nos consultar descrições RDF semi-estruturadas, usando a taxonomia do nodo e as etiquetas dos arcos definidas em um RDF *Schema*. Além disso, [KAR00b] afirma que RQL é baseada em um modelo de dados em forma de grafo permitindo considerar instâncias RDF como dados semi-estruturados que podem ser (parcialmente) interpretados pelo significado de um ou mais RDF *Schemas*.

A Figura 4.15 apresenta um exemplo de consulta em RQL sobre o RDF descrito na Figura 4.14. O objetivo é selecionar todos os recursos (neste caso, URL das disciplinas) que tenham a string “amostra” citada no conteúdo programático do documento de caracterização da disciplina.

```
select X
from {X}Cont_Prog{ Y}
where Y like "amostra"
```

Figura 4.15 – Exemplo de uma consulta em RQL (Caso I)

O resultado da consulta da Figura 4.15 é apresentado na Figura 4.16.

```
<rdf:type="resource" rdf:resource="http://www.inf.unisinos.br/~estatisticaI " />
<rdf:type="resource" rdf:resource="http://www.inf.unisinos.br/~estatisticaII " />
```

Figura 4.16 – Resultado da consulta em RQL apresentada na Figura 4.15

Se o objetivo da coordenação de um curso fosse encontrar quais professores ministram disciplinas com a metodologia "aulas expositivas" a consulta seria representada em RQL conforme a Figura 4.17.

```
select X
from {X}ministra{Y}metodologia{Z}
where Z like "aulas expositivas"
```

Figura 4.17 – Exemplo de uma consulta em RQL (Caso II)

```
<rdf:type="resource" rdf:resource="http://www.inf.unisinos.br/~denise" />
<rdf:type="resource" rdf:resource="http://www.inf.unisinos.br/~carlson" />
```

Figura 4.18 – Resultado da consulta em RQL apresentada na Figura 4.17

A consulta em RQL possibilitou identificar o conteúdo de uma disciplina disponível em uma página *Web* e associá-lo a todas as outras disciplinas que trabalham o mesmo no decorrer de um semestre. Além disso, o modelo RDF permite encontrar as URLs dos professores que ministram determinada disciplina com determinado conteúdo programático ou metodologia.

É importante ficar explícito que o modelo RDF, apresentado na Figura 4.14, pode ser estendido para qualquer propriedade do documento de caracterização das disciplinas, tal como: área de conhecimento, objetivos, pré-requisitos, entre outros. Pois o mesmo foi gerado a partir da ontologia apresentada no Anexo 2.

Este capítulo permitiu a exploração de diversas consultas através das marcações XML, as quais estão descritas na ontologia, utilizando a linguagem XQuery e possibilitou uma filtragem mais precisa dos resultados pela marcação semântica dos documentos. O termo *Semantic Web* foi apresentado juntamente com seus componentes, ontologia e RDF, o primeiro, definindo explicitamente os conceitos, relações e restrições para o domínio universidade, e o segundo, possibilitando a representação de metadados na *Web*. Finalmente, foram descritos alguns exemplos de consulta em RDF utilizando a linguagem RQL.



## V. CONCLUSÃO

Este trabalho realizou a aplicação da linguagem XML para etiquetar semanticamente documentos *Web* e explorar as possibilidades de exibição e consulta dos mesmos com o objetivo de apontar as vantagens em relação à marcação anteriormente utilizada em HTML. Com isso, foi desenvolvido um programa que transforma a marcação HTML em XML e adiciona mais algumas marcações para melhor identificação semântica do *corpus* de estudo de caso.

A partir do estudo realizado com XSL, verificou-se a possibilidade de exibição das mesmas informações de um documento *Web* em diferentes formatos conforme a necessidade do usuário. Já tendo o documento marcado em XML e as folhas de estilo, foram estudadas várias linguagens de consulta XML para uma melhor extração de informações. A linguagem XQuery foi utilizada por estar em desenvolvimento pelo W3C e por suprir os objetivos das consultas elaboradas durante o trabalho.

Logo, surgiu a necessidade de relacionar as informações referentes a professores e disciplinas e foi desenvolvido um aplicativo que transforma a marcação HTML, com informações sobre professores e disciplinas, em XML. Devido a essas informações estarem disponíveis na Internet, utilizou-se o aplicativo XML RDF para representar os relacionamentos e metadados existentes. O mesmo foi construído baseado em uma ontologia que possibilitou a geração de um RDF *Schema*.

A partir da marcação XML RDF, foi possível a elaboração de consultas utilizando a linguagem de consulta RQL que forneceu as respostas buscadas no relacionamento de professores e das disciplinas ministradas pelos mesmos. Com os documentos de caracterização das disciplinas marcados em XML e representados em RDF, foi finalizada a construção da ontologia que constitui o terceiro elemento da *Semantic Web*.

Uma das contribuições desta monografia está direcionada para o tratamento inteligente e semântico de informações na Internet, uma vez que possibilita a extração de informação baseada em informações do contexto universidade. A aplicação deste trabalho se restringe as disciplinas dos cursos de graduação da UNISINOS, mas pode ser estendida para outras aplicações cujo domínio seja universidade, pois uma das vantagens que a ontologia oferece é a reutilização dos conceitos nela definidos o que proporciona uma economia de tempo, dinheiro e recursos.

Para este trabalho foi utilizado um *corpus* de trinta disciplinas. Entretanto, é possível estender o mesmo para todas as disciplinas ministradas na UNISINOS. Para isso, seria necessário a construção coletiva da ontologia e a marcação dos dados diretamente no banco de dados para otimizar as respostas às consultas.

Uma possibilidade para a futura aplicação desta monografia é na implementação de uma interface para as consultas aos documentos marcados em XML, sendo que, um dos objetivos seria possibilitar consultas em linguagem natural. Além disso, é possível também marcar os dados diretamente no banco e explorar as informações sobre o histórico dos documentos de estudo de caso. Outro trabalho futuro, seria a construção coletiva de uma ontologia, com a participação de administradores e funcionários da UNISINOS. Finalmente, uma outra alternativa, seria o desenvolvimento de um protótipo de busca de informação inteligente.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [ALE00] ALEXAKI, Sofia and et al. Managing RDF Metadata for Community Webs. **In Proc. of the 2nd International Workshop on the World Wide Web and Conceptual Modeling (WCM'2000)**, pp. 140-151., Salt Lake City, U. US, October 2000.. Disponível por <http://www.ics.forth.gr/~christop>.
- [BER01] BERNERS-LEE, Tim; HENDLER, James; LASSILA, Ora. **The Semantic Web. A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities.** Scientific American, Issue 01/05/2001. Disponível por <http://www.sciam.com/2001/0501issue/0501berners-lee.html>.
- [BON01] BONIFATI, A.; LEE, D. **Technical Survey of XML Schema and Query Languages.** Submitted for for journal publication, Jan. 2001. Disponível por <http://www.cs.ucla.edu/~dongwon/paper/by-year.html>.
- [BRA00] BRAY, T. et al. **Extensible Markup Language (XML) 1.0 (Second Edition) W3C Recommendation** (outubro 2000). Disponível por <http://www.w3c.org/TR/REC-xml>.
- [BRA01] BRAGANHOLO, Vanessa; HEUSER, Carlos. XML Schema, RDF(S) e UML: uma comparação. **In: IDEAS 2001 - 4th Iberoamerican Workshop on Requirements Engineering and Software Environments**, Santo Domingo, Heredia, Costa Rica, page 78-90. Disponível por <http://www.inf.ufrgs.br/~vanessa/artigos/ideas2001.pdf>.
- [BRI00] BRICKLEY, Dan and GUHA, R.V. **Resource Description Framework (RDF) Schema Specification 1.0.** W3C Candidate Recommendation 27 March 2000. Disponível por <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>.
- [CHA01] CHAMBERLIN, Don and et al. **XQuery 1.0: An XML Query Language.** W3C Working Draft 07 June 2001. Disponível por <http://www.w3.org/TR/2001/WD-xquery-20010607/>.
- [CUI00] CUI, Zhan and O'BRAIN, Paul. Domain Ontology Management Environment. **Proceedings of HICSS33: Hawaii International Conference on System Sciences**, 4-7 Jan. 2000, IEEE Comput. Soc, p 9 pp. vol.1. Disponível por <http://dlib.computer.org/conferen/hicss/0493/pdf/04938015.pdf>.
- [DEC00] DECKER, Stefan et al. The Semantic Web: Roles of XML and RDF. **IEEE Expert**, 15(3), October 2000. Disponível por <http://www.cs.vu.nl/~frankh/publications.html>.
- [ERD00] ERDMANN, M. e STUDER R. How to Structure and Access XML Documents With Ontologies. **Data and Knowledge Engineering**, Special Issue on Intelligent Information Integration, 2001. DKE 36(3): 317--335.
- [FEN99] FENSEL, Dieter et. al. On2broker: Semantic-Based Access to Information Sources at the WWW. **In: Proceedings of the Workshop on Intelligent Information Integration (III99)** during IJCAI-99, Stockholm, Sweden, August 1999. Disponível por <http://www.aifb.uni-karlsruhe.de/~mer/>.
- [FEN00a] FENSEL, Dieter. The semantic web and its languages. **IEEE Intelligent Systems** 2000 (November/December 2000). Disponível por <http://www.cs.man.ac.uk/~horrocks/Publications/download/2000/faqs-on-oil.pdf>.

- [FEN00b] FENSEL, Dieter. **Ontologies: Silver Bullet for knowledge Management and Eletronic Commerce**, Springer-Verlag, Berlin, 2001. 138 pp. Disponível por <http://www.cs.vu.nl/~dieter>.
- [GRU93] GRUBER, T. R.. A Translation Approach to Portable Ontology Specifications. **Knowledge Acquisition**, 5 199-200, 1993.
- [GUA97] GUARINO, Nicola. Understanding, Building and Using Ontologies. A commentary to "*Using Explicit Ontologies in KBS Development*", by van Heijst, Schreiber, and Wielinga. **International Journal of Human and Computer Studies** , 46, 293-310, 1997. Disponível por <http://ksi.cpsc.ucalgary.ca/KAW/KAW96/guarino/guarino.html>.
- [GUA98] GUARINO, Nicola. Formal Ontology and Information Systems. In N. Guarino, editor, **Proceedings of the 1st International Conference on Formal Ontologies in Information Systems**, FOIS'98, Trento, Italy, pages 3--15. IOS Press, June 1998.
- [HAR01] HARMELEN, Frank van. Ontology-based information Visualisation. **Proceedings of the Workshop on Visualisation of the Semantic Web (VSW'01)**, September, London, 2001. Disponível por <http://www.cs.vu.nl/~frankh/abstracts/VSW01.html>.
- [HAR99] HAROLD, Elliotte Rusty. **XML Bible**. - 1. ED.- Chicago: IDG Books, 1999.
- [JAS99] JASPER, Robert and USCHOLD, Mike. A Framework for understanding and classifying ontology applications. in KAW'99 **Twelfth Workshop on Knowledge Acquisition, Modeling and Management**. 1999. Voyager Inn, Banff, Alberta, Canada. Disponível por <http://sern.ucalgary.ca/KSI/KAW/KAW99/papers/Uschold2/final-ont-apn-fmk.pdf> .
- [KAR00a] KARVOUNARAKIS, Greg and et al. Querying RDF Descriptions for Community Web Portals. **17<sup>èmes</sup> Journées Bases de Donnees Avancees (BDA'01)**, pp. 133-144, Agadir, Maroc, 29 Octobre - 2 Novembre, 2001.. Disponível por <http://139.91.183.30:9090/RDF/publications/sigmod2000.html> .
- [KAR00b] KARVOUNARAKIS, Greg and et al. Querying Semicstructured (Meta)Data and Schemas on the Web: The Case of RDF & RDF(S). Technical Report 269, **ICS-FORTH**, 2000. Disponível por <http://www.ics.forth.gr/proj/isst/RDF/rdfquerying.pdf> .
- [KAY00] KAY, Michael. **The place of XSLT in the XML family**. Disponível por [http://www.abc2xml.com/wrox/chapters/3129\\_chapter01/31290103.asp](http://www.abc2xml.com/wrox/chapters/3129_chapter01/31290103.asp) . (Agosto, 2001)
- [LAS99] LASSILA, O. e SWICK, R.R. **Resource Description Framework (RDF) model and syntax specification, W3C Recommendation**. Disponível por <http://www.w3.org/TR/REC-rdf-syntax/> .
- [MAD01] MADNICK, Stuart E. The Misguided Silver Bullet: What XML Will and Will NOT Do to Help Information Integration. **Working Papers MIT**, August 2001. Disponível por <http://ebusiness.mit.edu/research/papers-number.html> .
- [MAE01] MAEDCHE, Alexander and STAAB, Steffen. Learning Ontologies for the Semantic Web. In: **Semantic Web 2001** (at WWW10). May 1, Hongkong, China. Disponível por <http://www.aifb.uni-karlsruhe.de/WBS/sst/Research/Publications/semweb2001.pdf> .
- [MEL00a] MELLO, Ronaldo dos Santos; DORNELES, Carina Friedrich et al. 2000. Dados Semi-Estruturados. **XV Simpósio Brasileiro de Banco de Dados**.
- [MEL00b] MELLO, Ronaldo dos Santos. **A Mediation Layer for Integration of XML Data Sources with Ontology Support** (June, 2000). Thesis Proposal. Disponível por <http://www.inf.ufrgs.br/~ronaldo/phdreports.html>.
- [NOY01] NOY, Natalya Fridman and MCGUINNESS, Deborah L.. Ontology Development 101: A Guide to Creating Your First Ontology. **Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880**, March 2001. Disponível por <http://protege.stanford.edu/publications/ontologydevelopment/ontology101.html>.

- [PIM00] PIMENTEL, M. G.C., TEIXEIRA, C. A. C., SANTANCHE, A. XML: Explorando suas Aplicações na Web. **Congresso da Sociedade Brasileira de Computação**, Curitiba 2000. Anais da XIX Jornada de Atualização em Informática.
- [PIT00] PITTS-MOULTIS, N., Kirk, C. **XML Black Book - Solução e Poder**. Makron Books, 2000. 627 p.
- [RAY01] RAY, Erik T.. **Learning XML**. 1. ED. O'Reilly, 2001. 350 p.
- [ROB00] ROBIE, Jonathan and et al. **Quilt: an XML Query Language**. 31 March 2000. Disponível por [http://www.almaden.ibm.com/cs/people/chamberlin/quilt\\_euro.html](http://www.almaden.ibm.com/cs/people/chamberlin/quilt_euro.html).
- [SOO00] SOOHOO, Bonnie. **XML Tutorial 3: XSL Transformations**. Disponível por [http://www.webreview.com/2000/09\\_01/index.shtml](http://www.webreview.com/2000/09_01/index.shtml) .
- [STA00b] STAAB, S., ERDMANN M., MAEDCHE, A. and DECKER, S. An extensible approach for modeling ontologies in RDF(S). **First Workshop on the Semantic Web at the Fourth European Conference International Workshop on Research and Advanced Technology for Digital Libraries**, Lisbon, Portugal 18-20 September, 2000.
- [STA01] STAAB, S., MAEDCHE, A.. **knowledge Portals - Ontologies at Work**. Institute AIFB, University of Karlsruhe, Ontoprise Gmb H. Disponível por <http://www.aifb.uni-karlsruhe.de/WBS/Publ/pub2001.html>.
- [W3C99] WORLD WIDE WEB CONSORTIUM. **Extensible Stylesheet Language (XSL)**. Disponível por <http://www.w3.org/Style/XSL/> . (Agosto, 2001)
- [WEN98] WEINSTEIN, Peter C. and BIRMINGHAM, William P.. Creating Ontological metadata for Digital Content and Services. **International Journal on Digital Libraries 2(1): 19-36**. Disponível por <http://reggae.rs.itd.umich.edu/~peterw/abstracts.html> .

## BIBLIOGRAFIA COMPLEMENTAR

### XML

- [AND00] ANDERSON, Richard et. al. **Professional XML**. - Chicago : Wrox, 2000. 1169 p. : il.
- [CON00] CONRAD, Rainer; SCHEFFNER, Dieter; FREYTAG, J. Christoph. XML Conceptual Modelling using UML. **Proceedings of the 19th International Conference on Conceptual Modeling**, ER 2000 Salt Lake City, Utah, USA. Disponível por <http://www.dbis.informatik.hu-berlin.de/publications/conferences.html> .
- [CRI99] CHRISTOPHER, A. e WELTY, Nancy Ide. Using the right tools: enhancing retrieval from marked-up documents. **J. Computers in the Humanities**. 33(10), 1999. Kluwer Academic Publishers. Disponível por <http://www.cs.vassar.edu/faculty/welty/papers/CHUM-99.ps> .
- [KUP01] KUPER, Gabriel M., SIMÉON, Jérôme. Subsumption for XML Types. Bell Laboratories. **In Proceedings of International Conference on Database Theory (ICDT)**, London, UK, Jan. 2001. Disponível por <http://www-db.research.bell-labs.com/users/simeon/subsumption.ps>. (junho 2001).

### DUBLIN CORE

- [COR95] Descrição e exemplos dos elementos que compõe o Dublin Core. Disponível por <http://dublincore.org/> (maio 2001).
- [DUC98] **Dublin Core Metadata Element Set, Version 1.0: Reference Description**. Disponível por <http://dublincore.org/documents/1998/09/dces/#> (março 2001).

### SEMANTIC WEB

- [LUK97] LUKE, Sean et. al. Ontology-based Web Agents. **Proceedings of the First International Conference on Autonomous Agents (Agents'97)**. Disponível por <http://www.cs.umd.edu/fs/www/projects/plus/papers.html> .

### ONTOLOGIA

- [BEM99] BENJAMINS, R.; FENSEL, D.; DECKER, S.. Building ontologies for the internet. a Mid Term Report. **International Journal of Human-Computer Studies**, 51:687-712, 1999. Disponível por <http://citeseer.nj.nec.com/276747.html> .
- [DUI99] DUINEVELD, A.J., STOTER, R., WEIDEN. M.R., KENEP, B., & BENJAMINS, V.R. (1999): WonderTools? A comparative study of ontological engineering tools. **International Journal of Human-Computer Studies** 52:1111-1133.
- [HWA00] HWANG, C. "Incompletely and Imprecisely Speaking: Using Dynamic Ontologies for Representing and Retrieval Information" **Technical Report of Microelectronics and Computer Technology Corp**, MCC, 2000.
- [MAE00] MAEDCHE, A. and STAAB, S.: Ontology Learning from Text. Poster Session at the **5th International Conference on Applications of Natural Language to Information Systems NLDB'2000**. Versailles - France June 28-30 2000. Disponível por <http://www.aifb.uni-karlsruhe.de/WBS/ama/publications> .

- [PRE99] PRETSCHNER, A., and GAUCH, S. 1999. Ontology based personalized search. In **Proc. 11th IEEE Intl. Conf. on Tools with Artificial Intelligence**, pp. 391--398.
- [STA00a] STAAB, S. and MAEDCHE, A.. Ontology Engineering beyond the Modeling of Concepts and Relations. **CAI'2000 Workshop on Applications of Ontologies and Problem-Solving Methods**, Berlin, 2000. Disponível por <http://www.aifb.uni-karlsruhe.de/WBS/ama/publications.html>.
- [USC96] USCHOLD M. and GRUNINGER M.. Ontologies: Principles, Methods and Applications. **Knowledge Engineering Review**, 11(2), (1996). Disponível por <http://citeseer.nj.nec.com/uschold96ontologie.html> .

#### RDF

- [DEC98] DECKER, Stefan et. al. A query and Inference Services for RDF. In **QL'98 - The Query Languages Workshop**. W3C, 1998. Disponível por <http://www.w3.org/TandS/QL/QL98/pp/queryservice.html> .
- [MIL98] MILLER, Eric. An Introduction to the Resource Description Framework. **D-Lib Magazine**, 1998. Disponível por <http://www.dlib.org/dlib/may98/miller/05miller.html> .

## ANEXOS

*Anexo 1* - Um documento do estudo de caso (Caracterização das disciplinas da UNISINOS)

**Disciplina:** 65082 - Trabalho de Conclusão

**Ano/Semestre:** 2001/2 **Créditos acadêmicos:** 8

**Horas aula:** 120 **Teóricas:** 0 **Práticas:** 0

**Área de conhecimento e aplicação:** 125 - Informática

### **Identificação sumária dos conteúdos**

- O Trabalho de Conclusão constitui-se numa atividade curricular, de caráter individual e de natureza científica, sobre um tema relacionado com as áreas de conhecimento vinculadas às habilitações dos Cursos de Informática.
- O Trabalho de Conclusão deverá ser desenvolvido pelo aluno sob orientação de um professor - o professor orientador - segundo as normas de desenvolvimento de um trabalho científico.

### **Objetivos da disciplina**

O objetivo do Trabalho de Conclusão (TC) é consolidar os conteúdos vistos ao longo do curso em um trabalho de pesquisa aplicada e/ou desenvolvimento de sistemas computacionais. O desenvolvimento deste trabalho deve possibilitar ao aluno a integração entre teoria e prática, verificando a capacidade de síntese das vivências do aprendizado adquiridas durante o curso. Serão avaliados o domínio do conteúdo e a capacidade crítica sobre o tema escolhido bem como a capacidade de situar o tema escolhido numa perspectiva globalizante.

### **Exigências prévias de conhecimentos e habilidades**

- Conhecer as normas para elaboração de trabalho científico e métodos para apresentação oral;
- Ter capacidade analítica e crítica em relação às diferentes visões tecnológicas;
- Habilidade em redação.

### **Padrões mínimos de desempenho**

- Conhecer as normas para elaboração de um trabalho científico;
- Ser capaz de elaborar um trabalho adequando teoria, métodos e objetivos;
- Ter análise crítica do problema e da solução apresentada em um trabalho científico;
- Preparar e efetuar apresentação oral de um trabalho.

### **Conteúdos programáticos**

A escolha do tema do trabalho deve ser o primeiro passo para o início da elaboração do Trabalho de Conclusão. Esta escolha deverá ser inteiramente baseada nos interesses pessoais e profissionais do aluno, desde que atendidos os objetivos do curso.

### **Metodologias, técnicas e recursos de ensino e de avaliação de aprendizagem**

Cumprir as determinações das "Normas para realização do Trabalho de Conclusão" dos Cursos de Informática.

- Reunião com os matriculados na disciplina;
- Escolha do tema do trabalho;
- Escolha do orientador;
- Apresentação da proposta;
- Acompanhamento e orientação;

- Relatório de Andamento - texto ou apresentação oral;
- Entrega do Trabalho de Conclusão - quatro exemplares encadernados do Trabalho de Conclusão desenvolvido; uma cópia do trabalho em meio eletrônico; um arquivo html contendo o título do trabalho, autor, orientador, data de conclusão, palavras-chave, resumo e *abstract*;
- Apresentação Oral - O trabalho deverá ser apresentado oralmente perante a banca de avaliação, composta por três professores.

**Bibliografia básica**

BECKER, Fernando; FARINA, Sérgio; SCHEID, Urbano. **Apresentação de trabalhos escolares**. Porto Alegre: Multilivros, 1997.

**Bibliografia complementar**

FRANÇA, Junia Lessa et al. **Manual para normalização**: de publicações técnico-científicas. 4. ed. Belo Horizonte: Universidade Federal de Minas Gerais, 1998.



Anexo 2 – Exemplo de uma ontologia para o domínio universidade no formato HTML.

### Class Hierarchy for TC Project

- :THING
  - :SYSTEM-CLASS
    - :CLASS
      - :STANDARD-CLASS
    - :SLOT
      - :STANDARD-SLOT
    - :FACET
      - :STANDARD-FACET
    - :CONSTRAINT
      - :PAL-CONSTRAINT
    - UNIVERSIDADE
      - PESSOA
        - FUNCIONÁRIO
        - PROFESSOR
      - CURRÍCULO
        - DISCIPLINA

#### Class DISCIPLINA

Abstract Class (Instance of :STANDARD-CLASS MetaClass) Extends  
CURRÍCULO

#### Direct Subclasses:

None

Template Slots					
Slot name	Documentation	Type	Allowed Values/Classes	Cardinality	Default
<i>créditos-acadêmicos</i>	Cada crédito corresponde a uma hora-aula (45 minutos).	Integer		1:1	4
<i>Bibliografia</i>	Referências bibliográficas da disciplina	String		1:1	
<i>cont_prog</i>	Descrição dos conteúdos que serão abordados na disciplina.	String		1:1	
<i>nome_da_disciplina</i>	Designação da disciplina.	String		1:1	
<i>horas_aula_práticas</i>	Duração de tempo das aulas práticas de uma disciplina durante um semestre.	Integer		1:1	
<i>Exigências_prev_conhec</i>	Conhecimentos prévios que um aluno deve ter para cursar uma disciplina.	String		1:1	

<i>horas_aula_</i> <i>teóricas</i>	Duração de tempo das aulas teóricas de uma disciplina durante um semestre.	Integer		1:1	
<i>objetivo</i>	Meta ou alvo que se quer atingir.	String		1:1	
<i>metodologia</i>	Teoria dos procedimentos de ensino, geral ou particular para cada disciplina; didática teórica.	String		1:1	
<i>padroes_min_</i> <i>desemp</i>	Requisitos mínimos para o aluno ser aprovado em uma disciplina.	String		1:1	
<i>número_da_</i> <i>disciplina</i>	Identificação numérica única para cada disciplina.	Integer		1:1	
<i>nome_do_</i> <i>centro</i>	Designação do centro de ensino ao qual a disciplina está vinculada.	Symbol	Centro de Ciências Humanas, Centro de Ciências da Saúde, Centro de Ciências Jurídicas, Centro de Ciências Exatas e Tecnológicas, Centro de Ciências da Comunicação, Centro de Ciências Econômicas	1:1	
<i>horas_aula</i>	Duração de tempo que a disciplina deve ser ministrada durante um semestre.	Integer		1:1	60
<i>numero_do_</i> <i>centro</i>	Número do centro de ensino ao qual a disciplina está vinculada.	Symbol	6, 5, 4, 3, 2, 1	1:1	
<i>área_conhec_</i> <i>aplicação</i>	Designação da área de conhecimento em que se aplica a disciplina.	String		1:1	
<i>número_do_</i> <i>curso</i>	Identificação numérica própria para cada curso.	Integer		1:1	

Anexo 3 - Serialização XML do RDF(S) apresentado na Figura 5.6.

```

<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdfs:Class rdf:ID="Universidade"/>
  <rdfs:Class rdf:ID="Pessoa">
    <rdfs:subClass rdf:resource="#Universidade"/>
  </rdfs:Class>
  <rdfs:Class rdf:ID="Curriculo">
    <rdfs:subClass rdf:resource="#Universidade"/>
  </rdfs:Class>
  <rdfs:Class rdf:ID="Professor">
    <rdfs:subClass rdf:resource="#Pessoa"/>
  </rdfs:Class>
  <rdfs:Class rdf:ID="Disciplina">
    <rdfs:subClass rdf:resource="#Curriculo"/>
  </rdfs:Class>
  <rdf:Property rdf:ID="ministra">
    <rdfs:domain rdf:resource="#Professor"/>
    <rdfs:range rdf:resource="#Disciplina"/>
  </rdf:Property>
  <rdf:Description ID="cont_prog">
    <rdfs:domain rdf:resource="#Disciplina"/>
    <rdfs:range rdf:resource="http://www.w3c.org/TR/xmlschema-2/#string"/>
  </rdf:Description>
  <rdf:Description ID="metodologia">
    <rdfs:domain rdf:resource="#Disciplina"/>
    <rdfs:range rdf:resource="http://www.w3c.org/TR/xmlschema-2/#string"/>
  </rdf:Description>
  <Professor rdf:about="http://www.inf.unisinos.br/~denise">
    <ministra>
      <Disciplina rdf:about="http://www.inf.unisinos.br/~banco de dadosI">
        <cont_prog> Modelo Relacional </cont_prog>
        <metodologia>Aulas expositivas</metodologia>
      </Disciplina>
    </ministra>
    <ministra>
      <Disciplina rdf:about="http://www.inf.unisinos.br/~banco de dadosII">
        <cont_prog>recuperação de falhas</cont_prog>
        <metodologia>Aulas expositivas</metodologia>
      </Disciplina>
    </ministra>
  </Professor>
  <Professor rdf:about="http://www.inf.unisinos.br/~carlson">
    <ministra>
      <Disciplina rdf:about="http://www.inf.unisinos.br/~estastisticaI">
        <cont_prog>amostra</cont_prog>

```

```
</Disciplina>
</ministra>
</Professor>
<Professor rdf:about="http://www.inf.unisinos.br/~carlson">
  <ministra>
    <Disciplina rdf:about="http://www.inf.unisinos.br/~estastisticaI">
      <cont_prog>amostra</cont_prog>
      <metodologia> aulas expositivas-dialogadas </metodologia>
      <metodologia> testes e provas escritos</metodologia>
    </Disciplina>
  </ministra>
</Professor>
</rdf:RDF>
```

## Anexo 4 - RDF interpretado pelo parser do W3C.

