



MODELAGEM E DESENVOLVIMENTO DE BANCO DE DADOS

Ramiro Córdova Júnior

Conceitos básicos e componentes de um documento XML

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Identificar os conceitos básicos e as características sobre o padrão XML.
- Construir estruturas lógicas e físicas em documentos XML.
- Propor mecanismos de estilos XML.

Introdução

XML é um formato de arquivos universalmente utilizado para compartilhamento de dados entre aplicações que rodam em diferentes plataformas de hardware. Apesar da linguagem XML ser relativamente simples, as oportunidades são inúmeras devido ao fato de permitir que se trabalhe com bancos de dados de diferentes fabricantes, facilitando, assim, a provisão de soluções de transações comerciais, catálogo de produtos, gráficos vetoriais, equações matemáticas, fórmulas químicas, relatórios financeiros, anúncios publicitários, entre outras.

Desde a sua criação, a linguagem XML faz muito sucesso em aplicações voltadas para Internet. A facilidade de intergração entre soluções está seguramente entre as principais potencialidades do XML.

Conceitos básicos e características sobre o padrão XML

A XML – *eXtensible Markup Language* – é uma linguagem de marcadores, como a HTML, e foi desenhada para descrever dados. A sua grande vantagem é que ela é extensível, ou seja, você não está limitado a certo número de **tags** e pode criar as suas próprias. Assim sendo, ela é uma linguagem autodefinível.

XML descreve uma classe de objetos de dados chamados de documentos XML e, parcialmente, o comportamento de softwares que os processam. O XML é um perfil de aplicativo ou uma forma restrita de SGML, o *Standard Generalized Markup Language* [ISO 8879]. Os documentos XML são constituídos por unidades de armazenamento, chamadas entidades, que contêm dados analisados ou não analisados. Os dados analisados são compostos de caracteres, alguns dos quais formam dados de caracteres e alguns dos quais formam marcação.

Existe certa confusão que induz as pessoas a acharem que o XML surgiu com o objetivo de substituir o HTML, mas, na realidade, o XML tem como objetivo principal a descrição de dados, enquanto que o HTML tem seu foco na apresentação de dados. Para descrever os dados, a XML usa a DTD – *Document Type Definition*. As marcações usadas em documentos HTML e a estrutura dos documentos HTML são predefinidas, e o autor de um documento HTML pode usar somente as marcações que estão definidas no HTML padrão. Já o XML permite que o autor defina as suas próprias marcações e a própria estrutura do seu documento. Apesar do crescimento da utilização de XML, não significa que ela será a substituta da HTML — ela será um complemento e usada para descrever e estruturar os dados, enquanto HTML será usada para formatar e exibir os mesmos dados.

Compartilhamento de dados



O XML possui uma sintaxe básica que pode ser utilizada no compartilhamento de informações entre diferentes soluções computacionais. Quando combinado com outros padrões, torna possível definir o conteúdo de um documento separadamente de seu formato, o que facilita a reutilização do código em outras aplicações. Sendo assim, uma das suas principais características é a portabilidade: um exemplo desta característica é um banco de dados que pode escrever um arquivo XML para que outro banco consiga lê-lo.

Muitas áreas de conhecimento, atualmente, necessitam aproveitar melhor a grande quantidade de dados disponíveis na Internet, transformando-os em dados mais gerenciáveis e úteis na resolução de problemas. Como o XML

permite a definição de suas próprias marcas, é possível melhorar significativamente os processos de recuperação e disseminação da informação. Algumas aplicações onde o XML é bastante útil são na manutenção de grandes sites, na troca de informação entre organizações, no gerenciamento do conteúdo de sites, em aplicações de comércio eletrônico, etc.



Saiba mais

Um fato importante para o crescimento da utilização da XML é o suporte garantido por grandes empresas do mercado de software, como Microsoft, Oracle, SAP, IBM e SUN. O XML vem se tornando um formato universal para o compartilhamento de dados entre aplicações, como transações comerciais, catálogos de produtos, relatórios financeiros e estatísticos, enfim, para todos os dados que necessitam de uma apresentação estruturada.

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
Olá Mundo!!!
</doc>
```

O caractere utilizado para determinar o início de uma anotação é “<”, e o final de uma anotação é determinado por “>”. Sendo assim, no exemplo, pode-se perceber que existem três anotações. A primeira linha (<?xml version="1.0" encoding="UTF-8"?>) é uma declaração XML que especifica a versão do XML. Um documento XML sempre deve iniciar com uma declaração XML. Caso um documento possua algo antes da declaração, ou se ela não existir, será acusado um erro ao tentar apresentar o documento. As declarações XML podem possuir três atributos e o seguinte formato:

```
<?xml
  version="1.0"
  standalone="yes"
  encoding="iso-8859-1" ?>
```

O atributo version indica a versão de XML e é um atributo obrigatório em todas as declarações. O atributo standalone é opcional e pode possuir dois valores. Se estiver marcado com “yes”, significa que o documento não possui referências externas, e o valor “no” significa que o documento contém referências a entidades externas. O atributo encoding também é opcional e

indica o tipo de codificação utilizada no documento. O valor padrão (caso não seja declarado) é UTF-8, mas, no caso do exemplo, foi declarado o padrão “iso-8859-1”.

Documentos XML também aceitam a utilização de comentários no seu código. Os comentários devem iniciar pela marcação “<!--” e terminar com a marcação “-->”. Além de permitir a documentação do código, os comentários podem ser utilizados para remover temporariamente trechos do código do documento, pois os trechos comentados não são considerados para fins de interpretação do código.

Existem três restrições básicas para utilização de comentários em documentos XML: (i) os comentários não podem aparecer antes da declaração; (ii) os comentários não podem aparecer dentro de uma anotação; e (iii) não é permitido usar a sequência de caracteres “--” dentro de um comentário.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!--Aqui é inserido o texto do comentário-->
<doc>OláMundo!!!</doc>
```

As instruções de processamento não fazem parte do conteúdo do documento, pois são apenas uma indicação direta ao processador do documento de que aquele trecho deve ser executado. As instruções de processamento começam com “<id-processador” e terminam com “?>”. O item id-processador deve indicar o tipo de processamento a ser realizado. Segue um exemplo de código XML com duas instruções de processamento.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<agenda>
<?htmlaction="hr"?>
<entrada id="01" tipo="pessoa">
<nome>Pedro Paulo</nome>
<email>pp@pp.com.br</email>
<telefone>51 9253 6044</telefone>
</entrada>
</agenda>
<?htmlaction="hr"?>
```

Nesse exemplo, temos, em dois pontos do documento, a instrução de processamento “<?htmlaction=”hr”?>”. Esta instrução de processamento indica que, quando o processamento do documento XML coincidir com a geração da sua versão HTML, devem ser geradas duas marcações “hr” nas posições onde se encontram as instruções de processamento.

Construção de documentos XML

Um documento XML é formado por dados e anotações — os dados são blocos de texto, e as anotações descrevem a estrutura do documento, além de induzirem uma interpretação do conteúdo. A anotação é composta por marcações de início e fim de elementos, marcações de elementos vazios, referências a entidades, comentários, declarações de tipo de documento e instruções de processamento. Segue, a seguir, um exemplo introdutório de código XML, que tradicionalmente é utilizado no aprendizado de outras linguagens de programação.

O conjunto estrutural de um documento XML é composto por elementos, que são os blocos lógicos que possuem o texto global do documento ou sua decomposição. Cada elemento é composto por sua marcação de início, seu conteúdo e tem uma marcação de fim. Todas as marcações começam com o caractere “<” e encerram-se com “>”. As marcações de fim do bloco começam com “</” e encerram com o caractere “<”. É convencionalizado que, em um documento XML, a marcação de fim deve ter o mesmo nome dado à marcação de início anterior. Todos os elementos de um documento XML são oriundos do elemento raiz, ou seja, os documentos XML podem ter uma estrutura hierárquica de elementos.

Existem três regras básicas para definição do nome de um elemento em um documento XML:

1. O nome deve iniciar com uma letra, um underscore ou um sinal de dois pontos “:”.
2. Os caracteres que compõem o nome (exceto o primeiro) podem ser letras, dígitos, um underscore, hífen, pontos e dois pontos.
3. Não é permitido espaço em branco no nome de um elemento.

Outra característica importante, que deve ser observada na confecção de um documento XML, é que existe distinção entre caracteres maiúsculos e minúsculos, ou seja, a palavra “teste” é diferente de “TESTE” para o interpretador do documento. Na estrutura hierárquica de marcações, um elemento pode ter outros elementos (elementos filho), além de texto.

Cada elemento de um documento XML pode ter um ou mais atributos, que podem ser opcionais ou obrigatórios. Os atributos têm como objetivo qualificar o elemento ao qual estão associados, ou seja, os atributos podem ser considerados como os adjetivos dos elementos. As regras para os nomes de atributos são as mesmas para os nomes de elementos, e o valor dos atributos

aparecem entre aspas. Segue um exemplo de um trecho de documento XML, onde podem ser observados os elementos e seus atributos.

```
<agenda>
<entrada id="01" tipo="pessoa" nome="Pedro Paulo"
email="pp@xyz.com.br" telefone="51 9253 60447"/>
<entrada id="02" tipo="empresa" nome="JK Ltda"
email="jk@xyz.com.br" telefone="51 3253 9090"/>

...
</agenda>
```

Nesse exemplo, o elemento raiz **agenda** pode ser composto por um ou mais elementos, chamados de **entrada**. Existem os subelementos chamados de **nome**, **e-mail** e **telefone**, que possuem os atributos entre aspas. Os atributos descrevem os elementos, mas os mesmos também podem ser descritos a partir de elementos. A seguir, segue o mesmo exemplo, porém com os elementos descritos por elementos.

```
<agenda>
<entrada>
  <id>01 </id>
  <tipo>pessoa</tipo>
  <nome>Pedro Paulo</nome>
  <email>pp@xyz.com.br</email>
  <telefone>51 9253 60447</telefone>
</entrada>
<entrada>
  <id>02</id>
  <tipo>empresa</tipo>
  <nome>JK Ltda</nome>
  <email>jk@xyz.com.br</email>
  <telefone>51 3253 9090</telefone>
</entrada>
```

Para que um documento XML seja um arquivo válido, é necessário utilizar um recurso chamado de Definição do Tipo do Documento (DTD). O propósito da DTD é definir uma construção de blocos válida para um documento XML, e ela define a estrutura do documento usando uma lista de elementos válidos. A DTD pode ser declarada dentro de um documento XML ou num arquivo à parte. A DTD permite descrever cada marcação e fornecer regras para interpretar cada informação usada em um arquivo XML. Quando usada em um arquivo XML a DTD, aparece entre colchetes “[” e termina com um colchete “]”, seguido de um sinal de maior “>”. Segue um exemplo de utilização da DTD.

No exemplo, temos, inicialmente, a declaração da versão de XML. Logo após, são definidos o elemento agenda com os subelementos id, tipo, nome e-mail e telefone. Nas linhas abaixo, foi definido que cada um desses elementos possui texto comum e não será interpretado como uma marcação XML.

```
<<?xmlversion="1.0"?>

<!DOCTYPE note [
  <!ELEMENT agenda (id,tipo,nome,email, telefone)>
  <!ELEMENT id (#PCDATA)>
  <!ELEMENT tipo (#PCDATA)>
  <!ELEMENT nome (#PCDATA)>
  <!ELEMENT email (#PCDATA)>
  <!ELEMENT telefone (#PCDATA)>
]>

<agenda>
  <entrada>
    <id>01 </id>
    <tipo>pessoa</tipo>
    <nome>Pedro Paulo</nome>
    <email>pp@xyz.com.br</email>
    <telefone>51 9253 60447</telefone>
  </entrada>
  <entrada>
    <id>02</id>
    <tipo>empresa</tipo>
    <nome>JK Ltda</nome>
    <email>jk@xyz.com.br</email>
    <telefone>51 3253 9090</telefone>
  </entrada>
```

Mecanismos de estilos XSL

Uma necessidade recorrente, em se tratando de criação de documentos XML, é fazer com que o navegador interprete as marcações criadas pelo usuário. Para isso, é necessário criar um mecanismo que descreva como o documento será exibido, e esse mecanismo chama-se XSL (*Extensible Stylesheet Language*). Este mecanismo tem o propósito de transformar XML em HTML.

A XSL pode ser encarada como uma linguagem que pode transformar XML em HTML, filtrar e ordenar dados em documentos XML e formatar dados XML. Ela pode ser usada para definir como um arquivo XML será exibido pela transformação em um arquivo reconhecido pelo navegador do usuário. Geralmente, esse serviço é feito pela transformação de cada elemento XML em um elemento HTML, sendo que a XSL pode incluir novos elementos, remover, rearranjar e ordenar elementos e, ainda, testar e tomar decisões.

A linguagem XSL é composta por três partes com funções distintas:

- XSLT - linguagem para transformar documentos XML;
- XPath - linguagem para navegar em documentos XML;
- XSL-FO - linguagem de formatação de documentos XML.

O reconhecimento do código XML por navegadores HTML dá-se devido ao processo de transformação realizado pela XLST. O XPath é utilizado para definir os padrões de combinação entre o documento de origem e o documento final. Nesse caso, as partes do documento de origem que não combinam permanecerão sem modificações no documento final.

Pode-se dizer que a utilização de folhas de estilos XSL proporciona duas grandes vantagens: uma delas é a independência da programação, pois as transformações são escritas em um arquivo separado; e a outra grande vantagem é que os desenvolvedores WEB podem editar rapidamente a folha de estilos XSL e verificar a mudança na saída rapidamente.

Vejamos um exemplo prático: vamos transformar o arquivo XML a seguir (estudantes.xml) em um documento HTML formatado.

```
<?xmlversion = "1.0"?>
<cadastro>
  <estudante id = "001">
    <PrimeiroNome>Jose</PrimeiroNome>
    <UltimoNome>Silva</UltimoNome>
    <Apelido>Zé</Apelido>
    <Nota>85</Nota>
  </ estudante >
  < estudante id = "002">
    <PrimeiroNome>Carlos</PrimeiroNome>
    <UltimoNome>Meira</UltimoNome>
    <Apelido>Carlito</Apelido>
    <Nota>95</Nota>
  </ estudante >
  < estudante id = "003">
    <firstname>Abel</firstname>
    <UltimoNome>Guimaraes</UltimoNome>
    <Apelido>Abelito</Apelido>
    <Nota>90</Nota>
  </ estudante >
</cadastro>
```

Nesse caso, é necessário definir uma folha de estilos XSLT para o arquivo estudantes.xml. A página deverá conter um título e apresentar uma tabela de alunos com os cabeçalhos e seus respectivos dados extraídos do arquivo XML. Para tanto, é possível utilizar um arquivo xsl com o mesmo nome

(estudantes.xml) que deverá ser salvo no mesmo local do arquivo XML de origem. Segue o arquivo xsl com os comentários em cinza.

```
<?xml version = "1.0" encoding = "UTF-8"?>

<xsl:stylesheet version = "1.0"
xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">

<!-- Declaração de templatexsl: Informa ao interpretador qual a seção xml a ser formatada. No caso do exemplo, o
documento raiz é combinada com este documento -->
<xsl:template match = "/">

<!-- Marcações HTML: Seção utilizada para fins de formatação. O interpretador xml ignora e o navegador as interpreta. -->

<html>
<body>
<h2>Students</h2>

<table border = "1">
<tr bgcolor = "#9acd32">
<th>id</th>
<th>PrimeiroNome</th>
<th>UltimoNome</th>
<th>Apelido</th>
<th>Nota</th>
</tr>

<!-- for-each: para cada instrução processada busca um
elemento para ser combinado -->

<xsl:for-eachselect="cadastro/estudante">
<tr>
<td>
<!-- valor do elemento correspondente a instrução
aserprocessada -->

<xsl:value-of select = "@id"/>
</td>

<td><xsl:value-of select = "PrimeiroNome"/></td>
<td><xsl:value-of select = "UltimoNome"/></td>
<td><xsl:value-of select = "Apelido"/></td>
<td><xsl:value-of select = "nota"/></td>

</tr>
</xsl:for-each>

</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Após a criação do arquivo XSL, é necessário atualizar o documento de origem (estudantes.xml) com a marcação xml-stylesheet, fazendo referência ao arquivo estudantes.xsl. A seguir, segue o código inicial a ser modificado no arquivo xml. Após essa etapa, basta abrir o arquivo estudantes.xml no navegador de Internet.

```
<?xml version = "1.0"?>
<?xml-stylesheet type = "text/xsl" href = "estudantes.xsl"?>
<cadastro>
...
</cadastro>
```



Referência

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *ISO 8879: information processing - text and office systems - Standard Generalized Markup Language (SGML)*. Genebra, 1986. Disponível em: <http://www.iso.org/iso/catalogue_detail.htm?csnumber=16387>. Acesso em: 02 mar 2018.

Leituras recomendadas

ALMEIDA, M. B. Uma introdução ao XML, sua utilização na Internet e alguns conceitos complementares. *Ciência da informação*, Brasília, v. 31, n. 2, p. 5-13, 2002.

HEITLINGER, P. *O guia prático da XML: conceito, exemplos, prática e aplicações da linguagem universal*. Lisboa (PO): Centro Atlântico, 2001. Disponível em: <<http://www.centroatl.pt/titulos/tecnologias/imagens/oguiapratricoda-xml-excerto.pdf>>. Acesso em: 02 mar 2018.

HOMER, A.; HORNER, A. *IE5, XML and XSL programmer's reference*. Birmingham (UK): Wrox Press, 1999.

LIMA, C. E.; SCHMITZ, L. R. B. Introdução XML. In: Congresso de Computação e Sistemas de Informação do CEULP/ULBRA – Ecoinfo, 2003, Tocantins. *Anais...* Tocantins, CEULP/ULBRA, 2003. Disponível em <<http://arquivo.ulbra-to.br/ensino/43020/artigos/anais2003/anais/MiniCursos/MinicursoEncoinfo2003v22.pdf>>. Acesso em: 02 mar 2018.

MACORATTI, J. C. *XML – introdução e conceitos básicos*. Disponível em: <<http://www.macoratti.net/xml.htm>>. Acesso em: 02 mar 2018.

RAMALHO, J. C.; HENRIQUES, P. *XML & XSL: da teoria à prática*. Lisboa (PO): FCA, 2002.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.



Conteúdo:



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS