



# Gerenciamento de Dados XML

\* Vanessa P. Braganholo

O surgimento de XML [1], em 1996, revolucionou a Web e a forma como as aplicações trocam e representam dados. Apesar de seu grande sucesso, é muito comum que pessoas não iniciadas tenham uma idéia errada sobre XML. Afinal, o que é XML? Uma linguagem de programação? Um substituto do HTML? Nada disso. XML é uma linguagem de representação de dados cujo foco é a semântica dos dados representados e não sua forma de apresentação.

XML é uma linguagem de marcação extensível (**eXtensible Markup Language**), derivada da SGML, de onde também veio o HTML (daí a confusão). Ao contrário do HTML, em XML as *tags* não são pré-definidas. Podem se definir *tags* de acordo com o significado do dado que se quer representar. Por exemplo, para representar o nome de uma pessoa, poderíamos usar a seguinte marcação: `<nome>João</nome>` ou então `<name>João</name>`. Documentos XML são organizados em elementos hierarquicamente dispostos, o que permite definir estruturas complexas. Como um exemplo, suponha que João é um autor e sua instituição é a UFRJ. Para representar tal informação, poderíamos definir uma estrutura hierárquica da seguinte forma:

```
<autor>
  <nome>João</nome>
  <instituição>UFRJ</instituição>
</autor>
```

Tal estrutura pode ser vista como uma árvore, onde **autor** é a raiz e nome e **instituição** são seus filhos. É possível aumentar mais essa hierarquia em quantos níveis forem necessários. Cada marcação `<>` e `</>` delimita um elemento em XML. Portanto, nome é um elemento e seu conteúdo é João. Do mesmo modo, autor também é um elemento e seu conteúdo (complexo) são os elementos nome e instituição. Note que as *tags* devem seguir uma estrutura “bem formada”. Toda *tag* que abre tem que ser fechada e não é possível fechar a *tag* do elemento pai sem antes fechar todas as *tags* de seus filhos.

## Aplicando XML na prática

**Apresentação de Dados.** O fato de XML não se preocupar com a apresentação dos dados traz uma grande vantagem. Os dados ficam separados da forma de apresentação, que pode ser definida por protocolos de estilos como CSS ou XSLT. Toda vez que se deseja alterar a forma de apresentação, basta mudar o estilo. Os dados permanecem inalterados. Por exemplo, um portal Web pode desejar tro-

car o *layout* de suas páginas dependendo da estação do ano. Para isso, basta que ele utilize quatro estilos diferentes, um para cada estação do ano.

**Troca de Dados.** Grandes empresas trocam dados o tempo todo (pedidos, ordens de compra, ordens de pagamento, etc.). XML facilita enormemente essa tarefa, uma vez que é possível definir “classes de documentos” usando um esquema (DTD ou XML Schema). Desse modo, se a empresa A quer comprar da empresa B, ela deve enviar pedidos que sigam a “classe de documentos” determinada pela empresa B. Desse modo, os aplicativos da empresa B saberão como processar o pedido. Por exemplo, a empresa B pode determinar que os pedidos que ela aceita devem respeitar a estrutura mostrada na Figura ao lado.

Antigamente, isso era feito através de um padrão chamado EDI, onde os dados eram enviados em pacotes e cada campo do pacote tinha um significado. O processamento de tais dados dependia de softwares específicos, o que não acontece mais hoje em dia. Existe inclusive um padrão para substituir o EDI, chamado XML/EDI [2], que fornece um *framework* baseado em XML, EDI, *templates*, agentes e repositórios para a construção de *e-business*.

O processamento de documentos XML pode ser feito usando uma das APIs padrão: DOM [1] (API que transforma o documento XML em uma árvore na memória) ou SAX [3] (API que dispara eventos à medida em que vai lendo os dados do documento – tais eventos devem ser tratados pela aplicação).

**Representação de dados.** XML é um formato baseado em texto e, por isso, mais verboso. O espaço ocupado por um arquivo XML é sem dúvida compensado pela semântica que ele implicitamente atribui aos dados. Relacionamentos em XML podem ser expressos por aninhamentos (relações pai/filho entre elementos) ou por referência (usando XLink e XPointer [1]). Um problema, no entanto, ainda é o tamanho dos documentos. Como o uso de DOM exige que o documento caiba todo na memória, existe uma limitação física para o tamanho dos arquivos XML. Cabe ressaltar, porém, que XML ainda vem sendo utilizado como um formato intermediário de dados, ou seja, ➤

