

# Módulo II

## XML Processing: XSLT, SAX e DOM

*Prof. Ismael H F Santos*

## Ementa

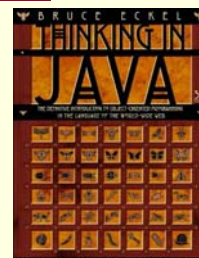
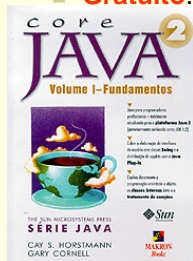
- Modulo II – XML Processing
  - XML Processing - XSLT

## Bibliografia

- *Linguagem de Programação JAVA*
  - *Ismael H. F. Santos, Apostila UniverCidade, 2002*

## Livros

- **Core Java 2**, Cay S. Horstmann, Gary Cornell
  - Volume 1 (Fundamentos)
  - Volume 2 (Características Avançadas)
- **Java: Como Programar**, Deitel & Deitel
- **Thinking in Patterns with JAVA**, Bruce Eckel
  - **Gratuito.** <http://www.mindview.net/Books/TIJ/>



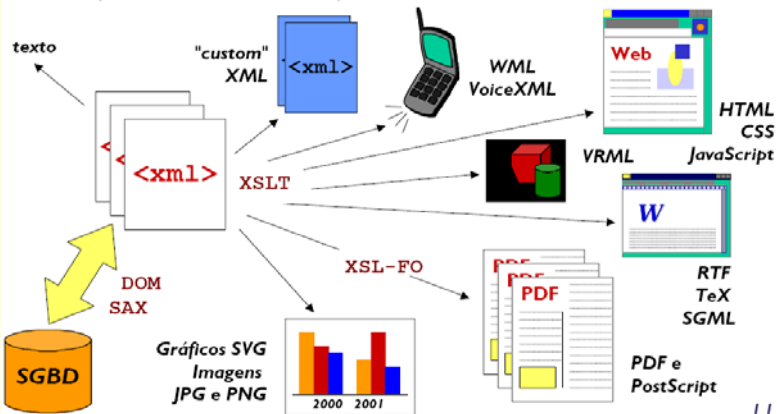
# SOA

XML Processing



# Onde usar XML?

- Dados armazenados em XML podem ser facilmente transformados em outros formatos



## Visualização em um browser

- **Folha de estilo: conjunto de regras para formatar ou transformar as informações de um documento XML**
- **CSS - Cascading Style Sheets**
  - Transformação visando apresentação visual
  - Aplicação do estilo em tempo de execução no cliente

## Visualização em um browser

- **XSLT - eXtensible Stylesheet Language**
  - Transformação em texto, HTML ou outro formato
  - Aplicação em tempo real ou prévia (no servidor)
- **Se não estiver associado a uma folha de estilo, o documento XML não tem uma "aparência" definida**
  - Internet Explorer e outros mostram a árvore-fonte XML
  - Netscape mostra apenas os nós de texto

# XLink

- **Namespace:** `http://www.w3.org/1999/xlink`
- **Sete atributos:**
  - `type = simple | extended | locator | arc | title | resource`
  - `href = a URI destino do vínculo`
  - `show = new | replace | embed | other | none`
  - `actuate = onLoad | onRequest | other | none`
  - `title = descrição detalhada`
  - `role = papel / contexto do vínculo`
  - `label = descrição sucinta (para exibição)`
- **Exemplo de uso típico**

```
<elemento xlink:href="http://lugar.com" />
```

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

9

# Exemplos de XLink

```
<p xmlns:xlink="http://www.w3.org/1999/xlink"
  xlink:type="simple"
  xlink:href="#end"
  xlink:show="replace"
  xlink:actuate="onRequest">link</p>
```

Conteúdo será substituído...  
.. quando o usuário solicitar.

```
<img xmlns:xlink="http://www.w3.org/1999/xlink"
  xlink:type="simple"
  xlink:href="12345.jpg"
  xlink:show="embed"
  xlink:actuate="onLoad" />
```

Conteúdo será embutido no documento...  
.. quando o documento for carregado.

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

10

# XPointer

- **Função que recebe uma expressão XPath**
  - Serve para apontar para um recurso ou parte do documento
- Para apontar para um recurso descoberto por uma expressão XPath  
xpointer(expressão)
- Quando há campos marcados com ID no documento destino, pode-se usar um ponteiro  
#ponteiro  
xpointer(expressão)#ponteiro

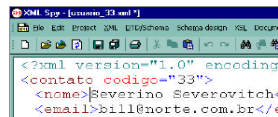
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

11

# Como produzir XML

- **Criando** um documento de texto Unicode a partir de qualquer editor de textos



```
<?xml version="1.0" encoding="UTF-8" ?>
<contato codigo="33">
  <nome>Severino Severovitch</nome>
  <email>bill@norte.com.br</email>
  <telefone tipo="celular">
    <area>11</area>
    <numero>9999 4321</numero>
  </telefone>
</contato>
```



```
<contato codigo="33">
  <nome>Severino Severovitch</nome>
  <email>bill@norte.com.br</email>
  <telefone tipo="celular">
    <area>11</area>
    <numero>9999 4321</numero>
  </telefone>
</contato>
```

- **Gerando** um documento a partir de uma árvore montada dinamicamente



```
<contato codigo="33">
  <nome>Severino Severovitch</nome>
  <email>bill@norte.com.br</email>
  <telefone tipo="celular">
    <area>11</area>
    <numero>9999 4321</numero>
  </telefone>
</contato>
```

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

12

## Como manipular XML?

- **Há duas APIs padrão para manipular (interpretar, gerar, extrair dados e tratar eventos) arquivos XML:**
  - **W3C Document Object Model (W3C DOM)**
  - **Simple API for XML (SAX)**
- **Servem a finalidades diferentes**
- **Implementações disponíveis em várias linguagens**
- **SAX oferece métodos que respondem a eventos produzidos durante a leitura do documento**
  - **notifica quando um elemento abre, quando fecha, etc.**
- **DOM monta uma árvore, que permite a navegação na estrutura do documento**
  - **propriedades dos objetos podem ser manipuladas**

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

13

## Leitura de XML com SAX

- Se um processador SAX receber o documento ...

```
<carta>
  <mensagem id="1">Bom dia!</mensagem>
</carta>
```
- ... ele irá disparar a seguinte seqüência de eventos:
  - ➔ `startDocument()`
  - ➔ `startElement("carta", [])`
  - ➔ `startElement("mensagem", [Attribute("id", "1")])`
    - ➔ `characters("Bom dia!")`
    - ➔ `endElement("mensagem")`
  - ➔ `endElement("carta")`
  - ➔ `endDocument()`
- Programador deve implementar um objeto "ouvinte" para capturar os eventos e extrair as informações desejadas

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

14

## Criação de documentos com DOM (1)

### • Criação dos elementos

/	Document	Obter objeto do tipo <b>Document</b> (raiz) (dependente de processador): <code>doc</code>
<carta>	Element	<code>carta = doc.createElement("carta")</code>
<mensagem>	Element	<code>mens = doc.createElement("mensagem")</code>
Bom dia!	String	<code>texto = doc.createTextNode("Bom dia!")</code>

### • Atributos

`<mensagem id="1">`      `mens.setAttribute("id", "1")`

April 05

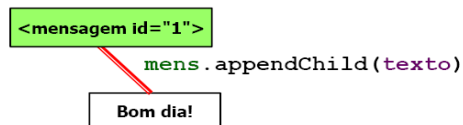
Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

15

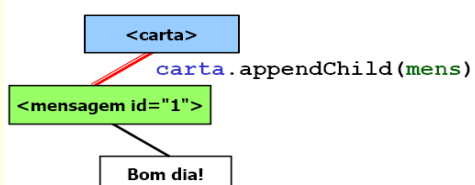
## Criação de documentos com DOM (2)

### Montagem da árvore passo-a-passo

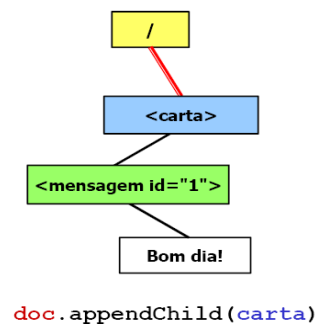
#### 1. Sub-árvore <mensagem>



#### 2. Sub-árvore <carta>



#### 3. Árvore completa



April 05

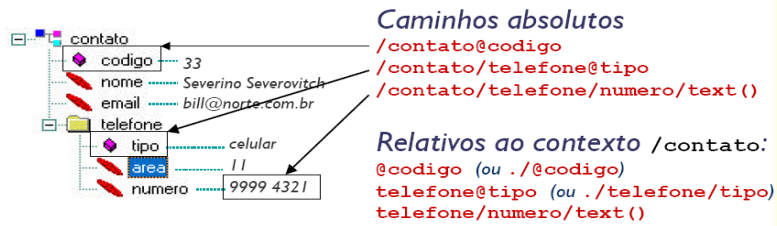
Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

16



# XPath

- Linguagem usada para **navegar** na árvore XML
  - Uma **expressão XPath** é um caminho\* na árvore que resulta em um valor (número, texto, booleano), objeto (elemento, atributo, nó de texto) ou conjunto de objetos



- Expressões XPath são usadas dentro de **atributos XML**
  - Usadas em XSLT, XLink, XQuery e XPointer

\* pode também ser padrão de busca

# XSLT

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
>
<xsl:template match="livro/titulo">
<td><xsl:value-of select="*" /></td>

```

- **XSL Transformations**
  - Linguagem (XML) para criação de documentos que contêm regras de transformação para documentos XML
  - Documentos escritos em XSLT são chamados de **folhas de estilo** e contêm
    - **Elementos XSLT:** <template>, <if>, <foreach>, ...
    - **Expressões XPath** para localizar nós da árvore-fonte
    - **Texto ou XML** a ser gerado no documento-resultado
  - **Processador XSLT**

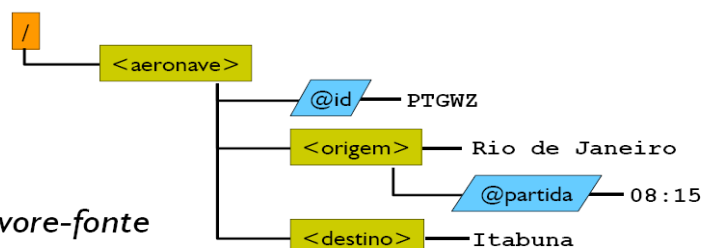


## XSLT: documento-fonte (1)

- Considere o seguinte documento-fonte:

```
<aeronave id="PTGWZ">
  <origem partida="08:15">Rio de
    Janeiro</origem>
  <destino>Itabuna</destino>
</aeronave>
```

- *Árvore-fonte*



## XSLT: folha de estilos (2)

- O seguinte **template** (parte de uma folha de estilos XSLT) pode extrair os dados do documento-fonte

```
<xsl:template match="aeronave">
  <p>A aeronave de prefixo
  <xsl:value-of select="@id" /> decolou
  de <xsl:value-of select="origem" /> às
  <xsl:value-of select="origem@partida" />
  tendo como destino o aeroporto de
  <xsl:value-of select="destino" />. </p>
</xsl:template>
```

documento resultado (em preto)

documento-fonte (em azul - XPath)

elementos XSLT (em vermelho, com prefixo xsl)

- Elementos XSLT geralmente são usados com um **prefixo** associado ao seu **namespace**: `<xsl:elemento>` para evitar conflitos com o documento-resultado.

## XSLT: documento-resultado (3)

- Após a transformação, o resultado será

```
<p>A aeronave de prefixo  
PTGWZ decolou  
de Rio de Janeiro às  
8:15  
tendo como destino o aeroporto de  
Itabuna.</p>
```

- Para obter outros resultados e gerar outros formatos com os mesmos dados, deve-se criar folhas de estilo adicionais

## XLink, XPointer e XQuery

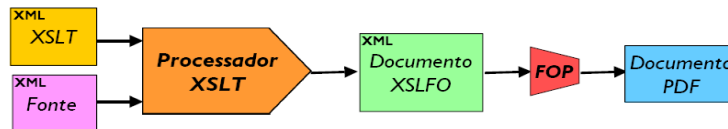
- **XLink**: é uma especificação W3C que permite definir vínculos entre documentos XML
  - Funcionalidade mínima é igual ao `<a href>` do HTML
  - Funcionalidade estendida permite vínculos bidirecionais, arcos, vários níveis de semântica, etc.
  - É uma coleção de atributos, com namespace próprio, que podem ser usados em elementos de qualquer linguagem XML.
- **XPointer**: aponta para partes de documentos XML
  - Identificador (ID) colocado no destino, acessível através de fragmento de URL: `xlink:href="#identificador"`
  - Caminho resultante de expressão XPath: `xpointer(/livro/id)`
- **XQuery**: linguagem para pesquisar documentos XML
  - Exemplo: 

```
FOR $b IN document("usuario_33.xml")/contato
WHERE nome="Severino Severovitch"
RETURN $b
```

# XSL-FO

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">  
  <fo:layout-master-set>  
    <fo:simple-page-master master-name="p1">  
      <fo:region-body/>  
    </fo:simple-page-master>  
  </fo:layout-master-set>  
</fo:root>
```

- **XSL Formatting Objects**
  - Linguagem XML de **descrição de página** com os mesmos recursos que PostScript ou PDF
  - Descreve o **layout preciso** de texto e imagens
  - Possui centenas de elementos, atributos e propriedades (que são semelhantes às propriedades do CSS)
  - Páginas são facilmente convertidas para PDF e PostScript
  - Ideal para gerar documentos para impressão (livros, etc.)
- Normalmente **gerada** via XSLT



# XSL-FO: menor documento

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">  
  <fo:layout-master-set>  
    <fo:simple-page-master master-name="p1">  
      <fo:region-body/>  
    </fo:simple-page-master>  
  </fo:layout-master-set>  
  
  <fo:page-sequence master-name="p1">  
    <fo:flow flow-name="xsl-region-body">  
      <fo:block color="blue" font-size="20pt">  
        Hello PDF!  
      </fo:block>  
    </fo:flow>  
  </fo:page-sequence>  
</fo:root>
```

Annotations:

- Blue arrow: "Este é o '<head>' do XSL-FO" (points to the layout-master-set)
- Black arrow: "Ligação entre as regras de layout e o conteúdo afetado" (points to the simple-page-master and page-sequence)
- Green arrow: "Este é o '<body>' do XSL-FO" (points to the page-sequence)

# XHTML

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Página XHTML</title></head>
<body>
<h1>Página XHTML</h1>
</body>
</html>
```

## • eXtensible HTML

- Linguagem XML de **descrição de página Web**
- Mesmos elementos do HTML 4.0 Strict
- Elementos descrevem **somente a estrutura** dos componentes da página.
  - A **forma** precisa ser especificada usando CSS: não há elementos/atributos para mudar cor, alinhamento, etc.
- Pode ser misturada (estendida) com outras linguagens XML (MathML, SVG, linguagens proprietárias)

## • Normalmente gerada via XSLT



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

25

# SVG

```
<svg>
<circle style="fill: red" cx="3cm" cy="3cm" r="2.5cm" />
<rect style="fill: blue" x="6cm" y="6cm"
height="2.5cm" width="1.5cm" />
</svg>
```

## • W3C Scalable Vector Graphics

- Gráficos vetoriais em XML
- Plug-ins para principais browsers: concorre com Flash
- Suporta animações, links, JavaScript, CSS
- Produzido por ferramentas como Adobe Illustrator
- Pode ser embutido no código XHTML e XSL-FO



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

26

# Exemplo SVG

**Exemplo de SVG**

```

<svg width="10cm" height="10cm">
  <g onclick="alert('Clicou no grupo 1!')">
    <circle style="fill: red"
      cx="3cm" cy="3cm" r="2.5cm" />
    <rect style="fill: blue" x="6cm" y="6cm"
      height="2.5cm" width="1.5cm" /></g>
    <g onclick="alert('Clicou no grupo 2!')">
      <circle style="fill: green; opacity: 0.5"
        cx="5cm" cy="5cm" r="2cm" /></g>
    <a xmlns:xlink="http://www.w3.org/1999/xlink"
      xlink:href="http://www.w3.org/Graphics/SVG">
      <text style="color: black; font-family: tahoma;
        font-size: 12pt" x="3cm" y="8cm">
        SVG é XML</text></a>
  </svg>
  
```

Annotations: JavaScript (onclick), CSS (style), XLink (xlink:href).

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

27

# Algumas outras linguagens XML

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

**MathML**

**WML**  
**VoiceXML**

**Web Services**  
SOAP  
WSDL  
UDDI  
XML-RPC  
ebXML

**XMI**

**RDF**

**SMIL 2.0**

**CML**

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

28

## Ferramentas XML

- **Para programação**
  - **Parsers-validadores:** Xerces, Crimson, MSXML 4.0
  - **Validadores:** MSV (Sun)
  - **Transformadores XSL:** TrAX, Xalan, Xt, Saxon
  - **APIs:** JDOM, JAX (P, B, M, R, RPC), SAX e DOM
  - **Veja mais em** [xml.apache.org](http://xml.apache.org) e [www.alphaworks.ibm.com](http://www.alphaworks.ibm.com)
- **Para edição (de XML genérico)**
  - XML Spy Suite
  - Framemaker / ArborText
  - JEdit com plug-ins para XML, XSLT e XPath
  - **Veja mais em** [www.w3.org/XML/](http://www.w3.org/XML/)

## Conclusões

- XML é uma ótima solução para **compartilhar** dados
- Para **implementar** soluções em gestão de informações usando XML, pode-se usar
  - **DTD** ou **XSchema** para especificar o modelo de dados e validar as informações
  - As APIs **DOM** ou **SAX** para extrair dados dos documentos, gerar documentos, ler e gravar em bancos de dados
  - **XSLT** e **XPath** para transformar os dados em outros formatos
  - **XLink**, **XPointer** e **XQuery** para criar vínculos lógicos entre os documentos e localizar seus componentes
  - **XSL-FO** ou **XHTML** para formatar os dados para impressão ou visualização na tela (PDF, Word ou Web)
  - **SVG** para gerar informações em forma de gráfico vetorial