



DAS



Universidade
Federal de
Santa Catarina

Introdução a XML

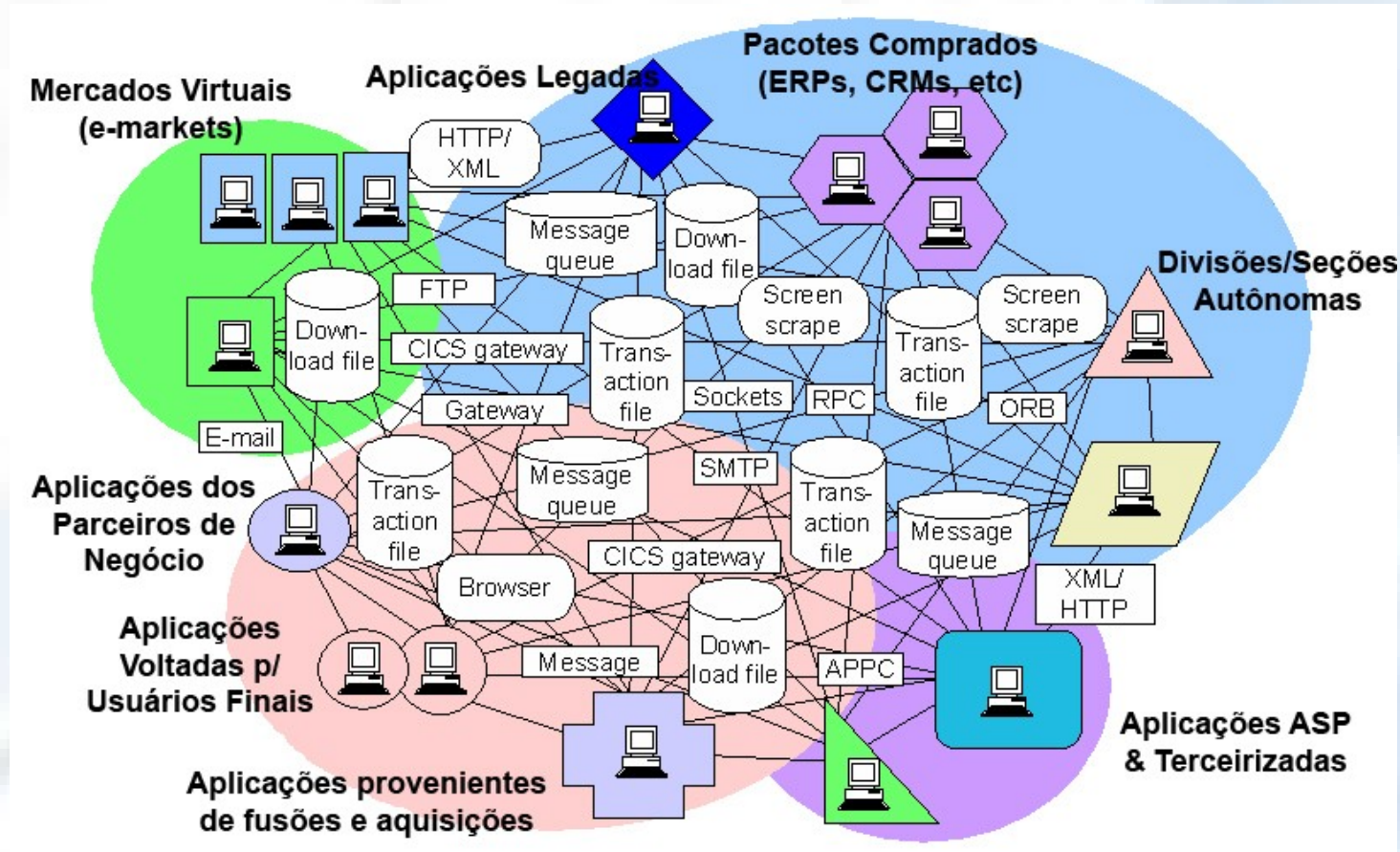
Saulo Popov Zambiasi

Roteiro

- ▶ Introdução / Motivação
- ▶ Definição
- ▶ Áreas de Aplicação
- ▶ Tecnologias Relacionadas
- ▶ Processando Documentos XML
- ▶ Componentes de um documento XML
- ▶ Regras Sintáticas
- ▶ Avaliação de Documentos XML
- ▶ Exemplos

Introdução / Motivação

Motivação: Interoperabilidade



- **Situação Atual:** Diversas organizações c/ diversas plataformas c/ diversas necessidades, etc...

Motivação: Interoperabilidade

► Objetivos da Interoperabilidade:

- Fazer com que sistemas heterogêneos trabalhem em conjunto;
- Maximizar a eficiência das pessoas e dos negócios, independente do ambiente;
- Melhorar as comunicações inter-organizacionais.

*“Para se atingir a **interoperabilidade**, é necessário que o foco seja dado em **padrões abertos**”*

(IDA-2004)

Padrões Abertos

“Conjunto de especificações disponível publicamente, que descreve as características de um hardware ou um software.

Padrões abertos são publicados para encorajar a **interoperabilidade** e conseqüentemente ajudar a popularizar novas tecnologias”

(<http://computing-dictionary.thefreedictionary.com>)

- ▶ Vantagens do uso de padrões abertos:
 - **Interoperabilidade:** implementações de diferentes fornecedores interoperam entre si.
 - ▶ Maior independência de fornecedores;
 - ▶ Definidos por uma comunidade;
 - ▶ Gratuitos (ou com eventuais taxas administrativas);
 - ▶ Sem restrições para o seu re-uso.

Padrões Abertos

► Exemplos de Padrões Abertos:

- Comunicação: **TCP** e **IP** (especificação da IETF para transporte de dados em rede), **GSM** (Sistema de comunicação móvel).
- Sistema Operacional: **POSIX**.
- Hardware: **ISA**, **PCI**, **AGP** (Padrão de barramento para computadores, padronizado pela IEEE).
- Documentos: **HTML** (W3C), **PDF** (Adobe), **ODF** (OASIS).
- Intercâmbio de dados: **EDIFACT** (ONU) e **XML** (W3C).

Padrões para Intercâmbio de Dados

- ▶ A padronização no intercâmbio eletrônico de dados (EDI) teve um grande impacto no mundo corporativo, visando reduzir os tempos e custos envolvidos nas relações entre empresas, tornando-as assim mais competitivas.
- Foram desenvolvidos diversos padrões para EDI, com destaque para o **EDIFACT** (*Electronic Data Interchange for Administration, Commerce and Transport*) criado pela ONU.
- ▶ Apesar de ter muitas vantagens, a utilização da EDI possui alto custo para implantação e manutenção, o que implica em dificuldades para pequenas e médias empresas.
- Havia, portanto, a necessidade de se resolver estas limitações, onde o padrão **XML** exerceu um papel fundamental.

O Padrão XML

- **XML** (*Extensible Markup Language*) é um padrão que define um formato de texto para representação estruturada de dados.
- ▶ **Apesar de não ter surgido com a finalidade de EDI, XML apresenta algumas características que, comparadas com os padrões anteriores, levaram a sua adoção para este fim:**
 - ▶ Legibilidade
 - ▶ Flexibilidade
 - ▶ Baixo custo
- Assim, XML se estabeleceu como o padrão *de facto* para representar informações intercambiadas entre sistemas, não apenas na Internet, mas também como suporte à **interoperabilidade** entre aplicações **heterogêneas**.
- Além disso, é XML é usada como base para os mais diversos padrões para intercâmbio de dados: **ebXML**, **XML/EDI**, Web Services, etc...

XML: Definição

O que é XML?

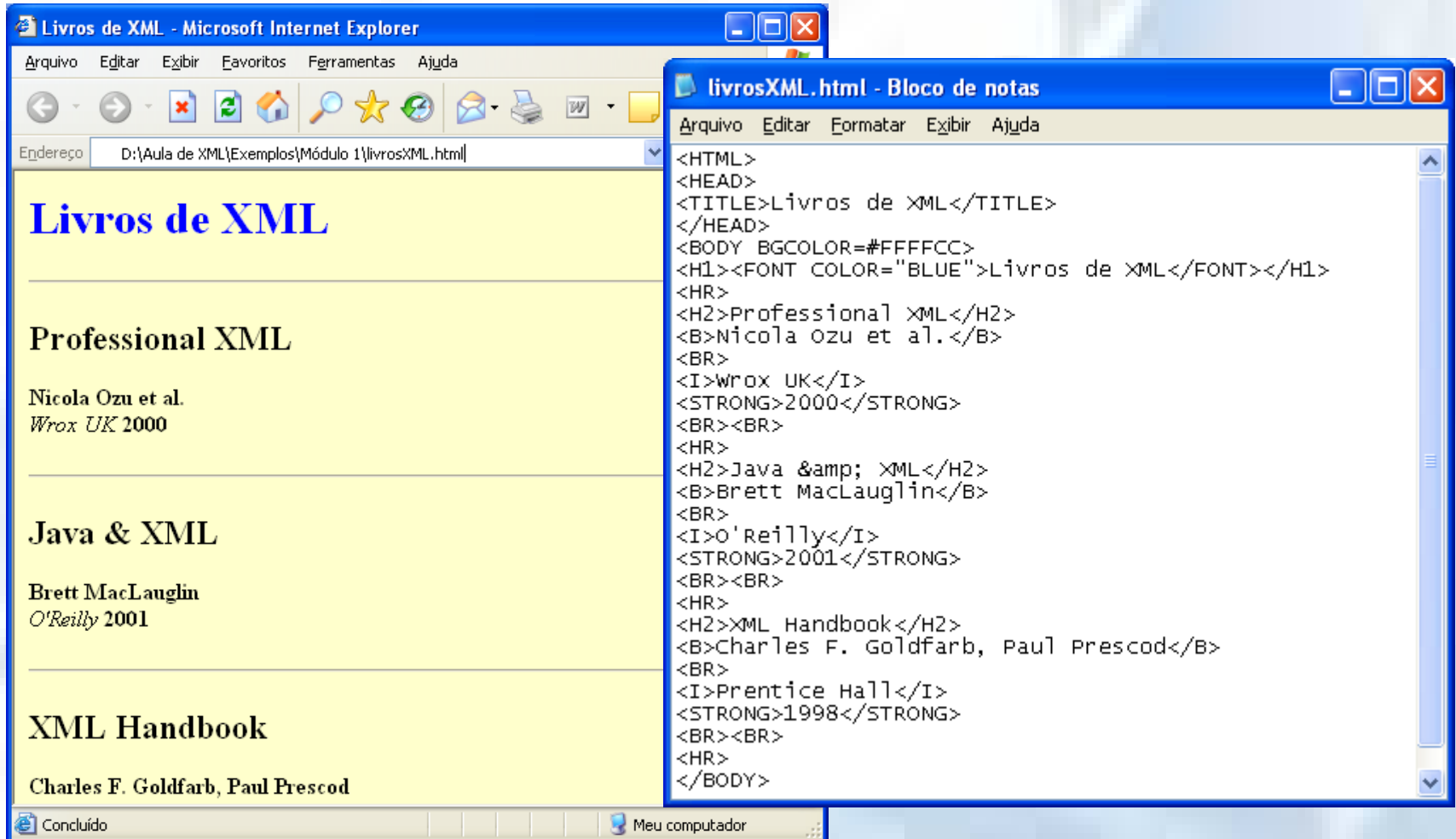
- ▶ Formato para descrever dados estruturados (documentos XML).
- Metalinguagem de marcadores, utilizada para definir novas linguagens de marcadores (*markup languages*).
- ▶ Padrão adotado pelo W3C (World Wide Web Consortium) desde fevereiro de 1998.
- Especificação é suportada por uma série de padrões (XSL, DOM, SAX, XML *Schema*, etc.), possuindo implementações para diversas plataformas.
- ▶ XML se tornou o padrão para intercâmbio e publicação de dados, e para integração entre aplicações.

XML vs. HTML

- ▶ XML possui notação semelhante a HTML, mas os conceitos envolvidos são completamente diferentes.
- HTML é uma linguagem de formatação de páginas Web, sendo portanto específica a um propósito particular e sem flexibilidade.
- ▶ XML é flexível, pois permite definir novas linguagens de marcadores.
- ▶ XML é uma linguagem que representa dados e sua estrutura (metadados). A apresentação (visualização) é definida externamente.

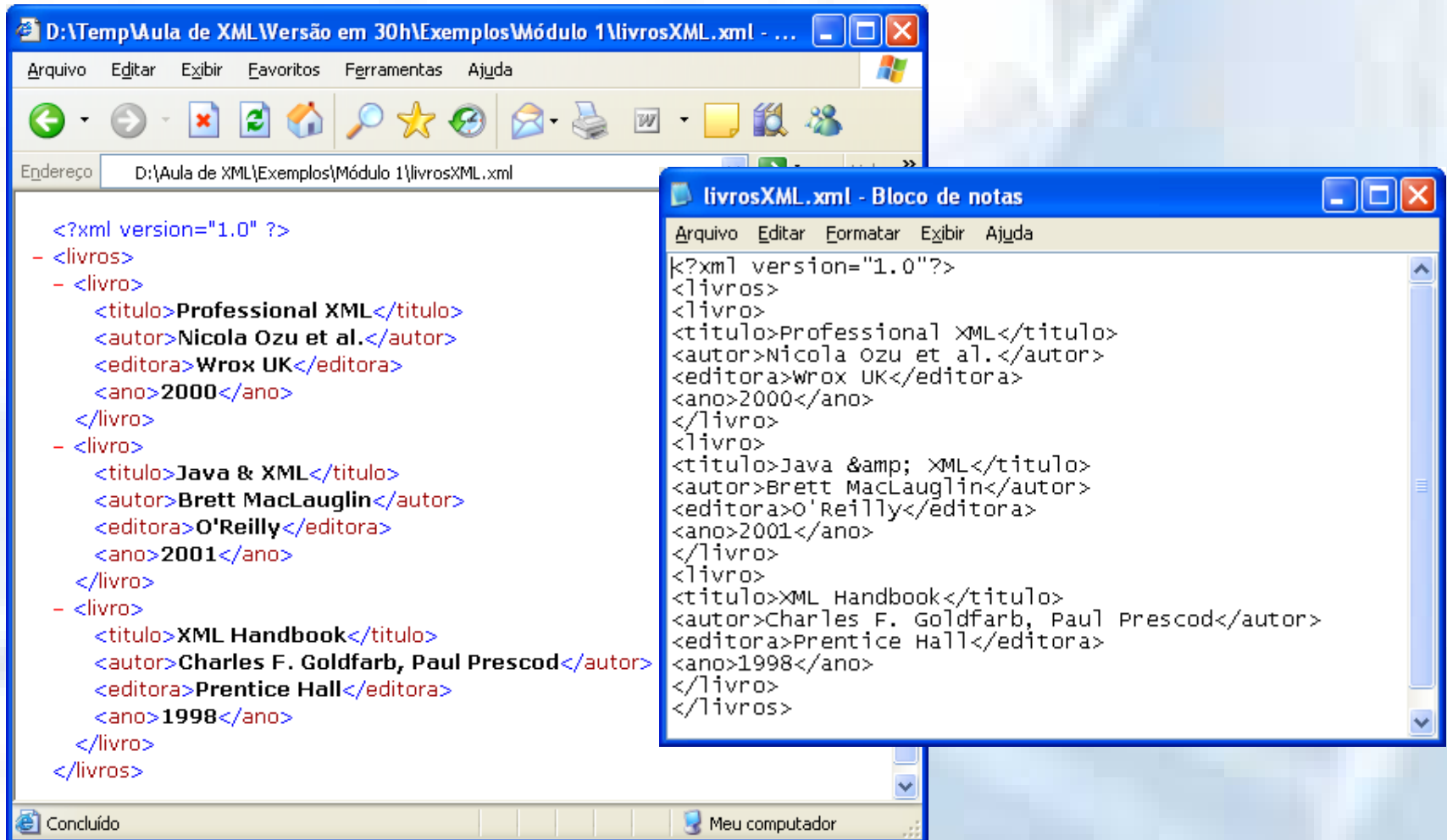
XML vs. HTML

HTML:

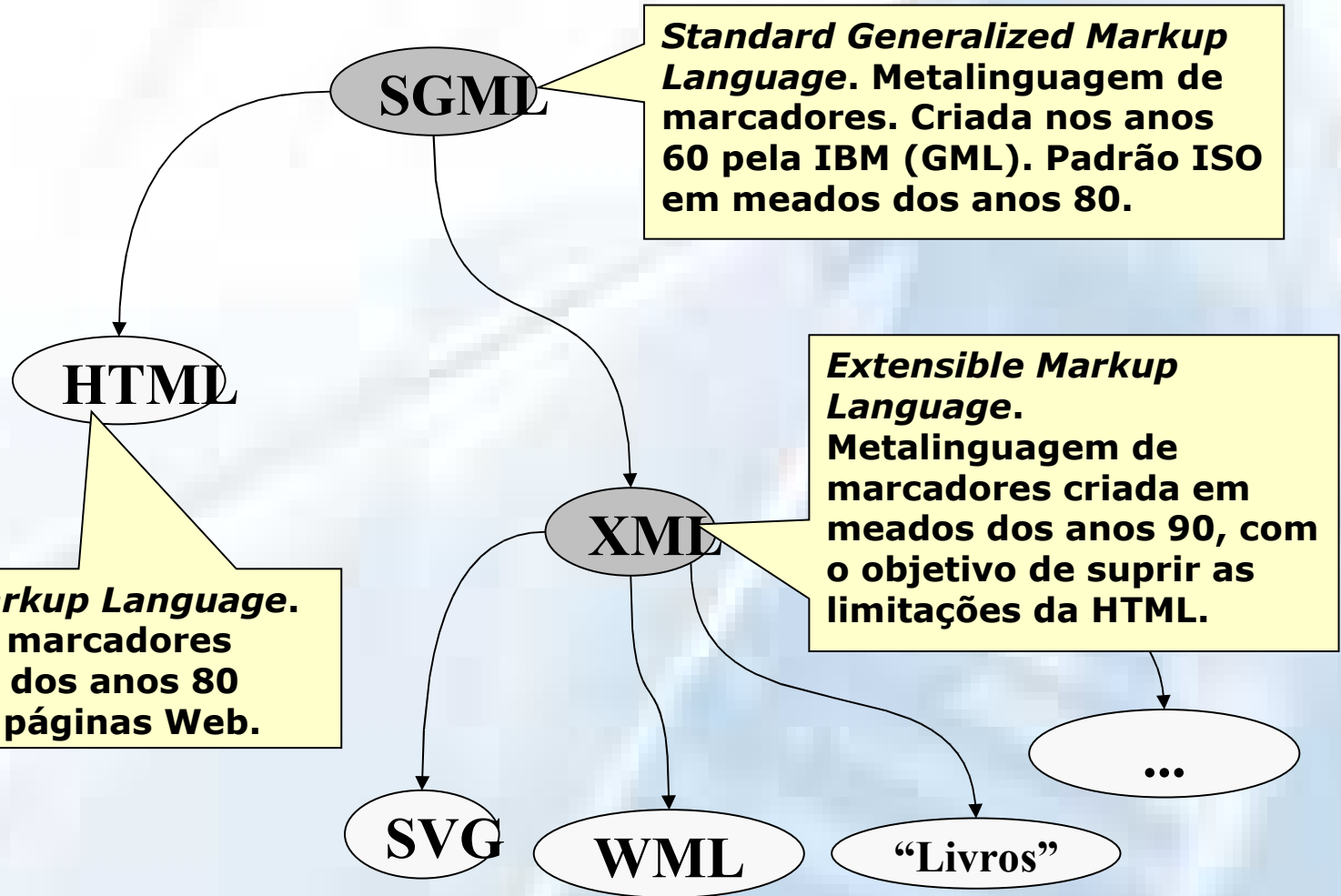


XML vs. HTML

XML:



Histórico



Características

- **Legível** (formato texto)
- **Independente de plataforma** (formato texto)
- **Autodescritiva** (estrutura - metadados)
- **Aberta e extensível** (esquemas XML)
- **Flexível** (conteúdo irrelevante pode ser ignorado)
- **Internacional** (multi-lingual, Unicode)
- **Padronizada** (W3C)
- **Validação automática** (*Parser XML*)

Desvantagens

- ▶ As características da XML, apesar de apresentarem vantagens para alguns fins, podem ser desvantajosas para outros:
 - **Formato texto:**
 - ▶ Processamento mais custoso do que um formato binário.
 - ▶ Não adequado para representar dados binários, tais como dados multimídia.
 - **Autodescritiva:**
 - ▶ Metadados podem ser redundantes, se o processador já sabe o que esperar.
 - ▶ Levam a um volume de dados muito maior.

Áreas de Aplicação

Áreas de Aplicação

- ▶ Gerenciamento de Documentos e de Conteúdo
- ▶ Formato particular de arquivo
 - Publicação de dados (XHTML, SVG, WML, etc.)
 - Intercâmbio de dados (XML/EDI, ebXML, XMI, etc.)
 - Web Services (SOAP, WDSL, UDDI)
- ▶ ***Integração entre aplicações heterogêneas***

Web Services

▶ Exemplo de mensagens SOAP:

```
<SOAP-ENV:Envelope>
  <SOAP-ENV:Body>
    <getPrice>
      <isbn type="string">1743006850</isbn>
    </getPrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Requisição

```
<SOAP-ENV:Envelope>
  <SOAP-ENV:Body>
    <getPriceResponse>
      <return type="float">72,50</return>
    </getPriceResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

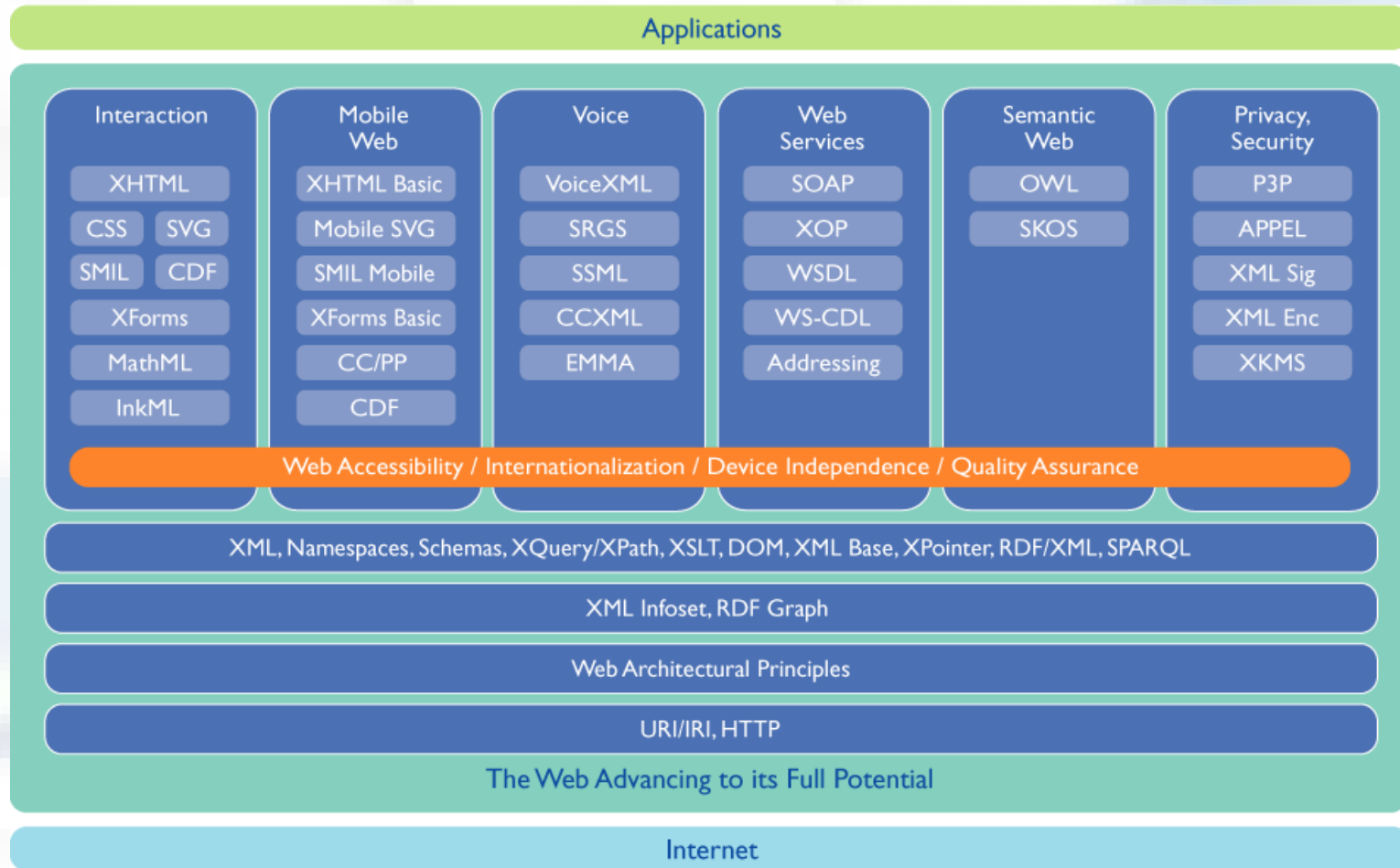
Resposta

Tecnologias Relacionadas

Tecnologias Relacionadas

- **DTD (Document Type Definition):** usado para definir a estrutura de documentos XML
- **XSD (XML Schema):** baseada em XML, vem substituindo o DTD.
- **XML DOM (XML Document Object Model):** define uma forma padronizada para acessar e manipular documentos XML.
- **XSL (Extensible Style Sheet Language):** consiste de três partes: **XSLT**, linguagem para transformar documentos XML; **XPath**, linguagem para navegação em documentos XML; e **XSL-FO**, linguagem para formatar documentos XML.
- **XLink (XML Linking Language) e XPointer (XML Pointer Language):** linguagens para criação de hyperlinks em documentos XML.
- **XQuery (XML Query Language):** linguagem para consulta em dados XML.
- **RDF (Resource Description Framework):** linguagem baseada em XML para descrever recursos na Web.
- **RSS (Really Simple Syndication):** formato para publicação de notícias em Web sites.

A Pilha de Tenologias da W3C



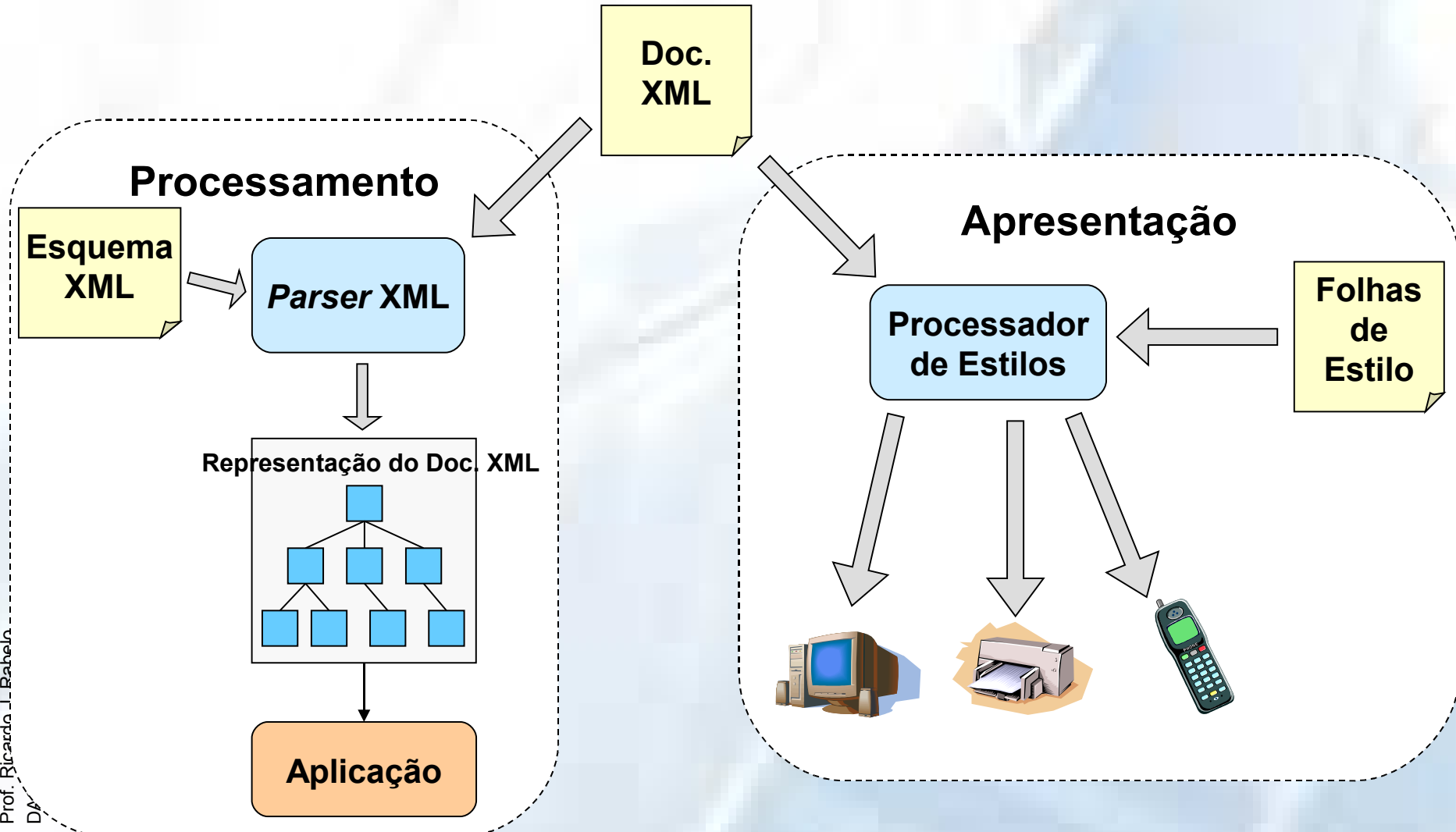
<http://www.w3.org/Consortium/technology>

Processando Dados XML

Flexibilidade da XML

- ▶ Um dos conceitos básicos da XML é a separação do conteúdo, descrição estrutural, e apresentação.
- Conteúdo: o documento XML contém dados e marcadores que identificam cada elemento de dados.
- Estrutura: para garantir a consistência de documentos de mesmo tipo, regras estruturais para este tipo de documento podem ser armazenadas em um *esquema XML*: Document Type Definition (DTD) ou *XML Schema*.
- Apresentação: informações referentes à apresentação dos dados são armazenadas em folhas de estilo.

Processando Dados XML



Prof. Ricardo J. Rabelin
DA

Componentes de um documento XML

Componentes de um documento XML

- ▶ Um documento XML contém os seguintes componentes:
 - **Elementos** (*Tags*)
 - ▶ **Atributos**
 - ▶ Referências a Entidades
 - ▶ Comentários
 - ▶ Instruções de processamento
 - ▶ *Document Type Declaration*

Exemplo

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mail_box SYSTEM "mbox.dtd">
<!-- Mail box do Leandro -->
<mail_box>
  <mensagem data="12/06/2005">
    <de>Leandro</de>
    <para>Fabiano</para>
    <cc>Carlos</cc>
    <cc>Leandro</cc>
    <anexo nome="mapa.gif"/>
    <assunto>Festa!!</assunto>
    <corpo>Não esqueça da festa na sexta!!</corpo>
  </mensagem>
  <mensagem data="13/06/2005">
    <de>Fabiano</de>
    <para>Leandro</para>
    <assunto>Re: Festa!!</assunto>
    <corpo>Festa?? Que Festa?</corpo>
  </mensagem>
</mail_box>
```

Componentes de um documento XML

Elementos

- ▶ São os principais componentes de documentos XML.
- ▶ Podem conter texto, outros elementos, ou ser vazios.
- ▶ Sintaxe:

`<nome_elemento> conteúdo </nome_elemento>`

Componentes de um documento XML

Elementos

Exemplos:

- Elemento XML "altura": `<altura>88</altura>`
- Elemento XML "nome": `<nome>João da Silva</nome>`
- Elemento XML "vazio": `<vazio/>` ou `<vazio></vazio>`
- Elemento XML "mensagem": `<mensagem>`
`<de>Leandro</de>`

`<para>Fabiano</para>`

`<assunto>F..</assunto>`

`<corpo>Não..</corpo>`

GSIG
`</mensagem>`

Exemplo (elementos)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mail_box SYSTEM "mbox.dtd">
<!-- Mail box do Leandro -->
<mail_box>
  <mensagem data="12/06/2005">
    <de>Leandro</de>
    <para>Fabiano</para>
    <cc>Carlos</cc>
    <cc>Leandro</cc>
    <anexo nome="mapa.gif"/>
    <assunto>Festa!!</assunto>
    <corpo>Não esqueça da festa na sexta!!</corpo>
  </mensagem>
  <mensagem data="13/06/2005">
    <de>Fabiano</de>
    <para>Leandro</para>
    <assunto>Re: Festa!!</assunto>
    <corpo>Festa?? Que Festa?</corpo>
  </mensagem>
</mail_box>
```


Componentes de um documento XML

Atributos

- São pares nome/valor e servem para oferecer informações adicionais sobre elementos.

Exemplos:

- HTML: ``
- XML: `<produto código="123">camisa</produto>`
- XML: `<altura unidade="cm">88</altura>`

Exemplo (atributos)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mail_box SYSTEM "mbox.dtd">
<!-- Mail box do Leandro -->
<mail_box>
  <mensagem data="12/06/2005">
    <de>Leandro</de>
    <para>Fabiano</para>
    <cc>Carlos</cc>
    <cc>Leandro</cc>
    <anexo nome="mapa.gif"/>
    <assunto>Festa!!</assunto>
    <corpo>Não esqueça da festa na sexta!!</corpo>
  </mensagem>
  <mensagem data="13/06/2005">
    <de>Fabiano</de>
    <para>Leandro</para>
    <assunto>Re: Festa!!</assunto>
    <corpo>Festa?? Que Festa?</corpo>
  </mensagem>
</mail_box>
```

Componentes de um documento XML

Referências a Entidades

- ▶ Entidades são nomes únicos associados a blocos de dados em XML.
- ▶ São usadas normalmente para referenciar dados repetidos ou que constantemente variam.
- ▶ São semelhantes a constantes em linguagens de programação: sendo expandidas (pré-processamento) antes da avaliação do processador.
- Sintaxe: `&nomeDaEntidade;`

Componentes de um documento XML

Referências a Entidades

Exemplo:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mail_box SYSTEM "mbox.dtd" [
  <!ENTITY assinatura "Leandro Silva">
]>
<mail_box>
  <mensagem data="12/06/2005">
    <de>Leandro</de>
    <para>Fabiano</para>
    <assunto>Festa!!</assunto>
    <corpo>Não esqueça da festa na sexta!!
      &assinatura;
    </corpo>
  </mensagem>
</mail_box>
```

Componentes de um documento XML

Referências a Entidades

- ▶ Outra função do uso de entidades é no caso de se usar caracteres especiais (reservados) como texto normal.

Caractere	Entidade
<	<
>	>
&	&
'	'
"	"

Componentes de um documento XML

Referências a Entidades

Exemplos:

`<expressão> x < 2 </expressão>` **ERRO!!**

`<expressão> x < 2 </expressão>` **CORRETO.**

`<marca>Johnson & Johnson</marca>` **ERRO!!**

`<marca>Johnson & Johnson</marca>` **CORRETO.**

Componentes de um documento XML

Referências a Entidades

- Quando um caractere especial não pode ser digitado usando o teclado, o código deste caractere pode ser usado como *referência a caractere*, de acordo com uma tabela específica.
- Tabelas de codificação de caracteres podem ser encontradas em: **http://en.wikipedia.org/wiki/Character_encoding**.

Exemplo:

Diferentes notações para o número "1":

<code><quantidade>&#49;</quantidade></code>	(ASCII)
<code><quantidade>&#x0967;</quantidade></code>	(árabe)
<code><quantidade>&#x0d67;</quantidade></code>	(malaio)

Componentes de um documento XML

Comentários

- ▶ Comentários são ignorados, ou seja, suas informações não fazem parte do conteúdo propriamente dito do documento XML.

- ▶ Sintaxe:

```
<!-- texto do comentário -->
```


Exemplo (comentário)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mail_box SYSTEM "mbox.dtd">
<!-- Mail box do Leandro -->
<mail_box>
  <mensagem data="12/06/2005">
    <de>Leandro</de>
    <para>Fabiano</para>
    <cc>Carlos</cc>
    <cc>Leandro</cc>
    <anexo nome="mapa.gif"/>
    <assunto>Festa!!</assunto>
    <corpo>Não esqueça da festa na sexta!!</corpo>
  </mensagem>
  <mensagem data="13/06/2005">
    <de>Fabiano</de>
    <para>Leandro</para>
    <assunto>Re: Festa!!</assunto>
    <corpo>Festa?? Que Festa?</corpo>
  </mensagem>
</mail_box>
```

Componentes de um documento XML

Instruções de Processamento

- Contêm comandos que são interpretados por um dado programa (*parser* XML ou processador de estilos)

▶ Sintaxe:

```
<?nomeDaInstrução listaDeAtributos?>
```

Exemplos:

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

```
<?xml-stylesheet type="text/css" href="mbox.css"?>
```

Componentes de um documento XML

Instruções de Processamento

XML: Indica a versão da XML utilizada no documento.

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

Obs: esta instrução tem um parâmetro opcional que define a codificação (*encoding*) de caracteres.

Exemplos:

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

(Valor padrão: Unicode codificado em UTF-8)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

(Caso o editor não suporte Unicode, pode-se usar codificações mais específicas, como para caracteres latinos)

Exemplo (instrução de processamento)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mail_box SYSTEM "mbox.dtd">
<!-- Mail box do Leandro -->
<mail_box>
  <mensagem data="12/06/2005">
    <de>Leandro</de>
    <para>Fabiano</para>
    <cc>Carlos</cc>
    <cc>Leandro</cc>
    <anexo nome="mapa.gif"/>
    <assunto>Festa!!</assunto>
    <corpo>Não esqueça da festa na sexta!!</corpo>
  </mensagem>
  <mensagem data="13/06/2005">
    <de>Fabiano</de>
    <para>Leandro</para>
    <assunto>Re: Festa!!</assunto>
    <corpo>Festa?? Que Festa?</corpo>
  </mensagem>
</mail_box>
```

Componentes de um documento XML

Document Type Declaration

- Faz referência a um DTD (*Document Type Definition*).

▶ Sintaxe:

```
<!DOCTYPE elemento_raiz SYSTEM "nome_do_dtd">
```

Exemplo (document type declaration)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mail_box SYSTEM "mbox.dtd">
<!-- Mail box do Leandro -->
<mail_box>
  <mensagem data="12/06/2005">
    <de>Leandro</de>
    <para>Fabiano</para>
    <cc>Carlos</cc>
    <cc>Leandro</cc>
    <anexo nome="mapa.gif"/>
    <assunto>Festa!!</assunto>
    <corpo>Não esqueça da festa na sexta!!</corpo>
  </mensagem>
  <mensagem data="13/06/2005">
    <de>Fabiano</de>
    <para>Leandro</para>
    <assunto>Re: Festa!!</assunto>
    <corpo>Festa?? Que Festa?</corpo>
  </mensagem>
</mail_box>
```

Exemplo (document type declaration)

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<!DOCTYPE mensagem SYSTEM "msg.dtd">  
<mensagem>  
  <de>Leandro</de>  
  <para>Fabiano</para>  
  <assunto>Festa!!</assunto>  
  <corpo>Não esqueça da festa na sexta!!</corpo>  
</mensagem>
```

Referência ao DTD é feita na Document Type Declaration

Arquivo "msg.dtd":

```
<!ELEMENT mensagem (de, para, assunto, corpo)>  
<!ELEMENT de (#PCDATA)>  
<!ELEMENT para (#PCDATA)>  
<!ELEMENT assunto (#PCDATA)>  
<!ELEMENT corpo (#PCDATA)>
```

Regras Sintáticas

Regras Sintáticas

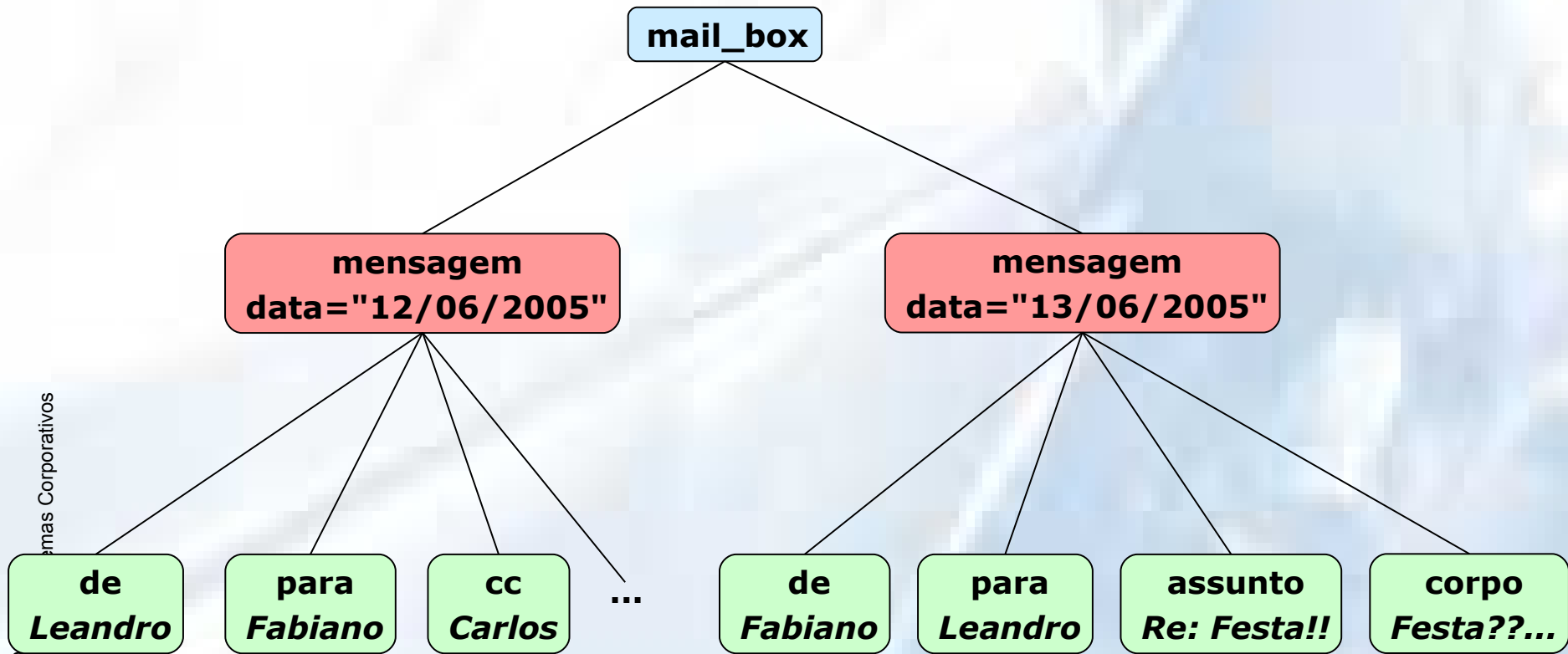
1) Estrutura Hierárquica: o primeiro elemento de um documento XML é chamado ***elemento raiz***. Todos os outros elementos devem estar corretamente ***aninhados*** dentro do elemento raiz.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mensagem data="12/11/99"> <!-- elemento raiz -->
  <de>Leandro</de>
  <para>Fabiano</para>
  <assunto>Festa!!</assunto>
  <corpo>Não esqueça da festa na sexta!!</corpo>
</mensagem>
```

Exemplo (estrutura hierárquica)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mail_box SYSTEM "mbox.dtd">
<!-- Mail box do Leandro -->
<mail_box> <!-- elemento raiz -->
  <mensagem data="12/06/2005">
    <de>Leandro</de>
    <para>Fabiano</para>
    <cc>Carlos</cc>
    <cc>Leandro</cc>
    <anexo nome="mapa.gif"/>
    <assunto>Festa!!</assunto>
    <corpo>Não esqueça da festa na sexta!!</corpo>
  </mensagem>
  <mensagem data="13/06/2005">
    <de>Fabiano</de>
    <para>Leandro</para>
    <assunto>Re: Festa!!</assunto>
    <corpo>Festa?? Que Festa?</corpo>
  </mensagem>
</mail_box>
```

Exemplo (estrutura hierárquica)



Regras Sintáticas

2) Instruções de processamento e a *document type declaration*, quando presentes, devem ser declaradas antes do elemento raiz.

Exemplo:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mail_box SYSTEM "mbox.dtd">
<!-- Mail box do Leandro -->
<mail_box>
  <mensagem data="12/06/2005">
    <de>Leandro</de>
    <para>Fabiano</para>
    <cc>Carlos</cc>
    <anexo nome="mapa.gif"/>
    <assunto>Festa!!</assunto>
    <corpo>Não esqueça da festa na sexta!!</corpo>
  </mensagem>
</mail_box>
```

Regras Sintáticas

3) Todo elemento XML deve ter um *tag* inicial e final;

`<p>Isto é um parágrafo</p>`

`<p>Isto é outro parágrafo</p>`

`<p/>` (Exceção!)

4) *Tags* são *case sensitive*. O tag `<Mensagem>` é diferente do tag `<mensagem>`.

`<Mensagem>Isto está incorreto</mensagem>`

`<mensagem>Isto está correto</mensagem>`

Regras Sintáticas

5) Todo valor de atributo deve estar entre aspas.

Correto:

```
<IMG SRC="computer.gif" />  
<mensagem data="12/11/2000" />
```

Incorreto:

```
<IMG SRC=computer.gif />  
<mensagem data=12/11/2000 />
```

Obs: Não há regra definida para quando usar elementos e quando usar atributos. Qualquer caso, respeitando-se a sintaxe, estará correto.

Elementos vs. Atributos

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<mensagem data="12/11/99">  
  <de>Leandro</de>  
  <para>Fabiano</para>  
  <assunto>Festa!!</assunto>  
  <corpo>Não esqueça da festa na sexta!!</corpo>  
</mensagem>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<mensagem>  
  <data>12/11/99</data>  
  <de>Leandro</de>  
  <para>Fabiano</para>  
  <assunto>Festa!!</assunto>  
  <corpo>Não esqueça da festa na sexta!!</corpo>  
</mensagem>
```

Avaliação de Documentos XML

Avaliação de Documentos XML

- **Documento XML Bem-formatado:** está de acordo com as regras sintáticas da linguagem XML.
- **Documento XML Válido:**
 - Se for *bem-formatado*, e ;
 - Estiver de acordo com a estrutura definida no ***esquema*** correspondente.
- Em XML, um ***esquema*** define formalmente a estrutura que os documentos devem ter, ou seja, a ordem e aninhamento dos elementos. Um esquema pode ser:
 - **DTD (*Document Type Definition*)**
 - **XML Schema**

Exemplos

Exemplo 1: Mail Box

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mail_box SYSTEM "mbox.dtd">
<!-- Mail box do Leandro -->
<mail_box>
  <mensagem data="12/06/2005">
    <de>Leandro</de>
    <para>Fabiano</para>
    <cc>Carlos</cc>
    <cc>Leandro</cc>
    <anexo nome="mapa.gif"/>
    <assunto>Festa!!</assunto>
    <corpo>Não esqueça da festa na sexta!!</corpo>
  </mensagem>
  <mensagem data="13/06/2005">
    <de>Fabiano</de>
    <para>Leandro</para>
    <assunto>Re: Festa!!</assunto>
    <corpo>Festa?? Que Festa?</corpo>
  </mensagem>
</mail_box>
```

Exemplo 2: Livros

```
<?xml version="1.0"?>
<!DOCTYPE livros SYSTEM "livros.dtd">
<livros>
  <livro>
    <titulo>Professional XML</titulo>
    <autor>Nicola Ozu et al.</autor>
    <editora>Wrox UK</editora>
    <ano>2000</ano>
  </livro>
  <livro>
    <titulo>Java & XML</titulo>
    <autor>Brett MacLaughlin</autor>
    <editora>O'Reilly</editora>
    <ano>2001</ano>
  </livro>
  <livro>
    <titulo>XML Handbook</titulo>
    <autor>Charles F. Goldfarb, Paul Prescod</autor>
    <editora>Prentice Hall</editora>
    <ano>1998</ano>
  </livro>
</livros>
```

Exemplo 3: XML News

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<nitf>
  <head>
    <title>Colombia Earthquake</title>
  </head>
  <body>
    <body.head>
      <headline>
        <h1>143 Dead in Colombia Earthquake</h1>
      </headline>
      <byline>
        <bytag>By Jared Kotler, Associated Press Writer</bytag>
      </byline>
      <dateline>
        <location>Bogota, Colombia</location>
        <story.date>Monday January 25 1999 7:28 ET</story.date>
      </dateline>
    </body.head>
  </body>
</nitf>
```

Exemplo 4: WML

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="card1" title="Tutorial">
    <do type="accept" label="Answer">
      <go href="#card2"/>
    </do>
    <p>
      <select name="name">
        <option value="HTML">HTML Tutorial</option>
        <option value="XML">XML Tutorial</option>
        <option value="WAP">WAP Tutorial</option>
      </select>
    </p>
  </card>
  <card id="card2" title="Answer">
    <p>You selected: $(name)</p>
  </card>
</wml>
```

card1

----- Tutorial -----

HTML Tutorial

XML Tutorial

WAP Tutorial

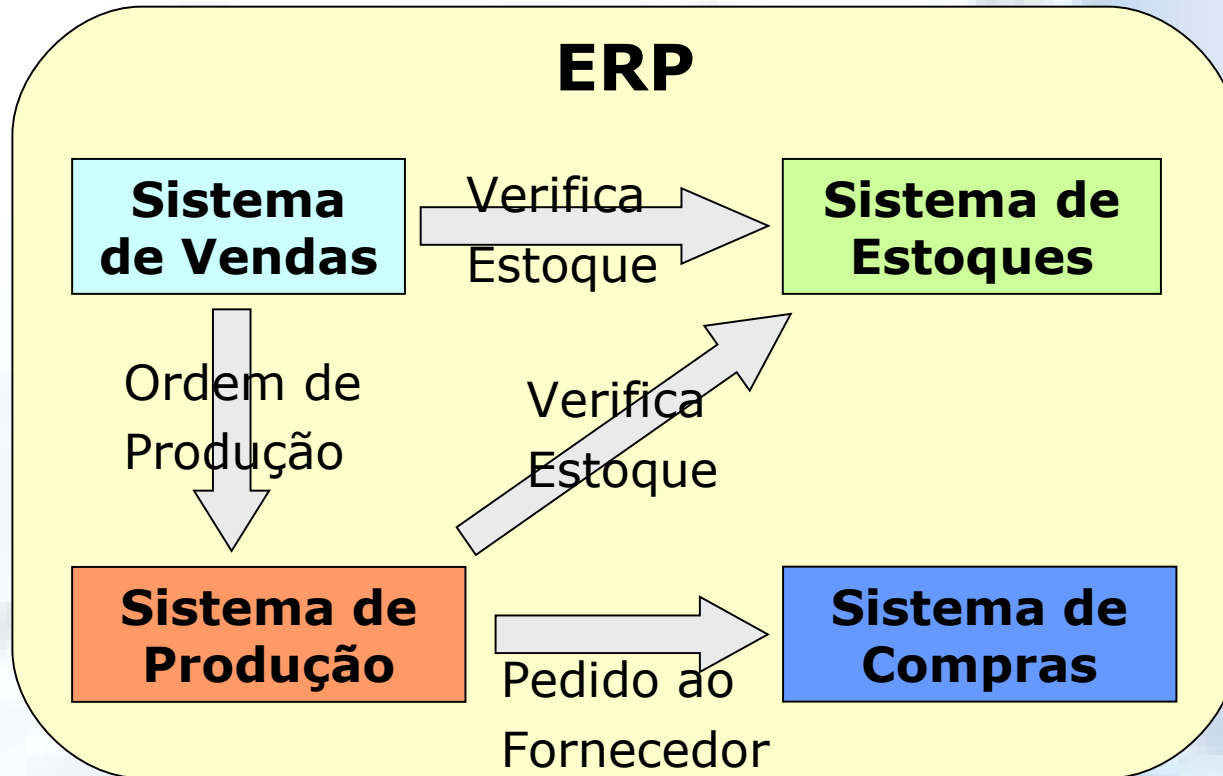
Answer

card2

----- Answer -----

You Selected: HTML

Exemplo “real”: Empresa



Exemplo “real”

- **Sistema de Vendas** verifica no **estoque** a existência dos produtos pedidos;
- **Sistema de Vendas** envia ordens de produção ao **Sistema de Produção**;
- **Sistema de Produção** verifica no **estoque** a existência de matéria-prima/componentes para a produção;
- **Sistema de Produção** envia pedido de matéria-prima/componentes ao **Sistema de Compras**;

Exemplo “real”: Verifica Estoque

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE verificaEstoque SYSTEM
"verifica_estoque.dtd">

<verificaEstoque>
  <produto código="XPTO" />
  <quantidade>20</quantidade>
</verificaEstoque>
```

Exemplo “real”: Ordem de Produção

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE ordemProdução SYSTEM "ordem_prod.dtd">

<ordemProdução>
  <número>OP_2532</número>
  <data>2001-09-22</data>
  <dataEntrega>2001-09-25</dataEntrega>
  <item id="1">
    <quantidade>20</quantidade>
    <produto código="XPTO" />
  </item>
</ordemProdução>
```

Exemplo “real”: Pedido de Material

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE pedidoMaterial SYSTEM "pedido_material.dtd">
<pedidoMaterial>
  <número>P_763</número>
  <códigoFornecedor>P_763</códigoFornecedor>
  <data>2001-09-22</data>
  <dataEntrega>2001-09-23</dataEntrega>
  <item id="1">
    <quantidade>30</quantidade>
    <produto código="XYZ" />
  </item>
  <item id="2">
    <quantidade>10</quantidade>
    <produto código="ABCD" />
  </item>
</pedidoMaterial>
```

Referências

- **W3C** – www.w3.org/XML
- **W3 Schools** – www.w3schools.com
- **Editix (Editor XML)** - www.editix.com
- **XML Software** – www.xmlsoftware.com
- **The XML Industry Portal** – www.xml.org
- **Microsoft** – msdn.microsoft.com/xml
- **W3C Technology Stack** – www.w3.org/Consortium/technology