SCC0265 – Sistemas Interativos Web

A Linguagem XML
DTD x XSD

Renata Pontin M. Fortes

(renata@icmc.usp.br)

PAE: Willian Watanabe (watinha@gmail.com)

Instituto de Ciências Matemáticas e de Computação

ICMC-USP S.Carlos, 2010

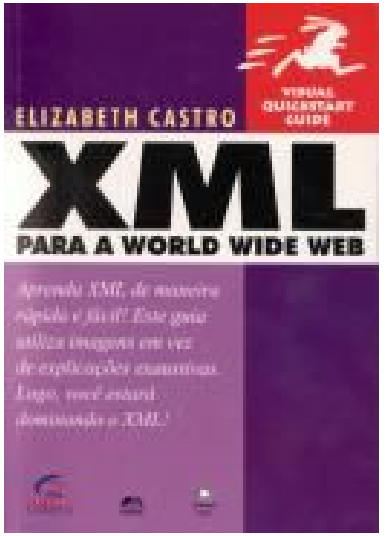
Resumo

- XML
  - O que é XML e como ele ajuda
  - Elementos XML
  - Documentos bem-formados
  - Exercício - Criando um documento XML (2 Passos)
  - Documentos válidos
- DTD
  - Definição
  - Tipos de dados
  - Exemplos
- XML Schema (XSD)
  - Definição
  - Estrutura
  - Transformação
  - Exemplos
- Pontos mais importantes

Linguagem de Marcação

- Imagine um livro com capítulos, seções, títulos etc.

- Para cada um desses itens existe um estilo que defino como cada um deve ser apresentado.



Linguagem de Marcação

- O conjunto desses estilos formam as folhas de estilo.

- Isso facilita a edição pois quando um autor precisa distinguir o título do livro do resto do conteúdo, basta indicar o estilo do título

Ao usar um indicador ou marca, para referenciar o título, ele está usando linguagem de marcação

Linguagem de Marcação

- Usuários de LaTeX e de HTML, usam marcações (Tags) constantemente. Ex:

- \LaTex = 

- <B>Parágrafo</B> = **Parágrafo**

XML – eXtensible Markup Language

- Especificação técnica desenvolvida pela W3C (World Wide Web Consortium) para superar as limitações do HTML, que é o padrão das páginas da Web.

XML x HTML

- Embora ambas as linguagens utilizem *tags*, um documento HTML funciona caso uma *tag* seja aberta mas não fechada. Isso não ocorre no XML. O documento simplesmente não funciona.
- HTML possui *tags* pré-definidas, o XML não, a menos que você utilize e indique que as *tags* utilizadas foram definidas em outro documento.

Exemplo de um documento XML

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

<Receita nome="pão" tempo_de_preparo="5 minutos" tempo_de_cozimento="1 hora">
  <título>Pão simples</título>
  <ingrediente quantidade="3" unidade="xícaras">
     Farinha
  </ingrediente>
  <ingrediente quantidade="7" unidade="gramas">
     Fermento
  </ingrediente>
  <ingrediente quantidade="1.5" unidade="xícaras" estado="morna">
     Água
  </ingrediente>
  <ingrediente quantidade="1" unidade="colheres de chá">
     Sal
  </ingrediente>
  <Instruções>
   <passo>Misture todos os ingredientes, e dissolva bem.</passo>
   <passo>Cubra com um pano e deixe por uma hora em um local morno.</passo>
   <passo>Misture novamente, coloque numa bandeja e asse num forno.</passo>
  </Instruções>
</Receita>
```

Entendendo o XML Passo a Passo

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

indica que o documento é um XML

Entendendo o XML Passo a Passo

```
<DOCUMENTOSIMPLES>
  <CABECALHO>Esse e o cabecalho do documento
  </CABECALHO>
  <CORPO>
      <PRIMEIROPARAGRAFO>Este é  1. paragrafo.
          </PRIMEIROPARAGRAFO>
      <SEGUNDOPARAGRAFO>Este é o segundo paragrafo.
           </SEGUNDOPARAGRAFO>
  </CORPO>
  <RODAPE>Essas informacoes se encontram no
  rodape.</RODAPE>

</DOCUMENTOSIMPLES>
```

Documentos Bem-formados

- São aqueles que atendem à sintaxe XML usada dentro do documento, sem qualquer erro.
- Para cada elemento aberto um outro é fechado.

<elemento>

</elemento>

- Sem caracteres que não podem ser analisados Ñ, ¢,Æ,å...

Documentos Válidos

- Documento Bem-Formado não é válido. A menos que ele contenha ou faça referência a um DTD.

- Verificar se as informações contidas no XML são consistentes.

DTD – *Document Type Definition*

- O DTD é um documento que define regras aos elementos do documento XML.

Exemplo de uma DTD

```
<?xml version"1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE DOCUMENTO [
<!ELEMENT DOCUMENTO (ASSUNTO, DATA, ENDERECO, MEMORANDO)>
<!ELEMENT ASSUNTO (#PCDATA)>
<!ELEMENT DATA (#PCDATA)>
<!ELEMENT ENDERECO (#PCDATA)>
<!ELEMENT MEMORANDO (#PCDATA)>
<!ENTITY EDITORA "Markron Books">
]>
<DOCUMENTO>
  <ASSUNTO>Memorando de hoje</DOCUMENTO>
  <DATA>1 de Julho de 2007</DATA>
  <ENDERECO>200 West 34th Suite 953, Anchorage, AK</ENDERECO>
  <MEMORANDO>Este memorando avisa que o XML  Book novo
   está sendo impresso. Publicado pela &EDITORA;, ele ensina
   tudo sobre XML.
  </MEMORANDO>
</DOCUMENTO>
```

Saída do Exemplo

Memorando de hoje

Assunto

1 de Julho de 2007

Data

200 West 34th Suite 953, Anchorage, AK

Endereco

Este memorando avisa que o XML Book novo está sendo impresso. Publicado pela Makron Books, ele ensina tudo sobre XML.

Memorando

Entendendo o Exemplo

- **<?XML version =``1.0'' encoding=``UTF-8'' standalone=``no''?>** - Essa declaração indica que se trata de um documento XML da versão 1.0, que utiliza o esquema de codificação *Unicode* de 8 bits e que não (no) depende de qualquer fonte externa.
- **<!DOCTYPE DOCUMENTO[** - A declaração de tipo de documento especifica onde se localiza a DTD. Neste caso, ela está dentro do documento em si.
- **<!ELEMENT DOCUMENTO (ASSUNTO, DATA, ENDERECO, MEMORANDO)>** - Define a lista de elementos para o elemento-raiz DOCUMENTO e a ordem que devem aparecer.
- **<!ELEMENT ASSUNTO (\#PCDATA)>** - Define o elemento ASSUNTO e especifica que ele conterá dados de caracteres analisados sintaticamente. O mesmo é valido para DATA, ENDERECO, MEMORANDO.
- **<!ENTITY EDITORA ``Makron Books''‘>** - Define uma entidade simples, EDITORA, e especifica que o valor para essa entidade é ``Makron Books''.
- **]** - Indica o final da DTD.

Elementos da DTD

- Declaração da DTD interna ao xml:
  <!DOCTYPE elemento-raiz [declaraçao de elementos]>
- Declaração da DTD externa ao xml:
  <!DOCTYPE elemento-raiz SYSTEM "NomeArquivo.dtd">
- Declaração de elementos raiz:
  <!ELEMENT elemento-raiz (declaraçao de elementos filhos)>
- Declaração de elementos filhos:
  <!ELEMENT elemento (#PCDATA)>
- Declaração de entidades internas:
  <!ENTITY nome-entidade "valor-entidade">
- Declaração de entidades externas:
  <!ENTITY nome-entidade SYSTEM "URI/URL">
- Declaração de atributos:
  <!ATTLIST nome-elemento nome-atributo tipo-atributo valor-default>

Definindo Atributos

<!ELEMENT ingrediente (#PCDATA)>
<!ATTLIST ingrediente quantidade CDATA #REQUIRED>
<!ATTLIST ingrediente unidade CDATA #REQUIRED>

Definindo tipos-atributos na DTD

- **CDATA** é um valor de caracteres
- (en1—en2—..) o valor pertence a uma lista de valores
- **ID** o valor é um id único
- **IDREF** o valor é id de outro elemento
- **IDREFS** o valor é uma lista de outros ids
- **NMTOKEN** o valor é um nome de um xml válido
- **NMTOKENS** o valor são nomes de um xml válido
- **ENTITY** o valor é uma entidade
- **ENTITIES** o valor é uma lista de entidades
- **NOTATION** o valor é o nome de uma notação
- **xml:** define que o valor é um valor de xml predefinido

Definindo valores *default* dos atributos

- **value** o valor *default*
- **#REQUIRED** o valor deve ser colocado
- **#IMPLIED** o valor não necessariamente deve ser colocado
- **#FIXED value** o valor do atributo é fixo

DTD – Exemplo Final

```
<!DOCTYPE NEWSPAPER [

<!ELEMENT NEWSPAPER (ARTICLE+)>
<!ELEMENT ARTICLE (HEADLINE,BYLINE,LEAD,BODY,NOTES)>
<!ELEMENT HEADLINE (#PCDATA)>
<!ELEMENT BYLINE (#PCDATA)>
<!ELEMENT LEAD (#PCDATA)>
<!ELEMENT BODY (#PCDATA)>
<!ELEMENT NOTES (#PCDATA)>

<!ATTLIST ARTICLE AUTHOR CDATA #REQUIRED>
<!ATTLIST ARTICLE EDITOR CDATA #IMPLIED>
<!ATTLIST ARTICLE DATE CDATA #IMPLIED>
<!ATTLIST ARTICLE EDITION CDATA #IMPLIED>

<!ENTITY NEWSPAPER "Vervet Logic Times">
<!ENTITY PUBLISHER "Vervet Logic Press">
<!ENTITY COPYRIGHT "Copyright 1998 Vervet Logic Press">

]>
```

XML Schema

Para que serve?

Descrever a estrutura do documento XML.

XML Schema

É uma alternativa baseada em XML para as DTDs e pode definir :

- elementos que podem aparecer no documento
- atributos que podem aparecer no documento
- quais elementos são elementos filhos
- a ordem dos elementos filhos
- o número de elementos filhos
- se um elemento é vazio ou se inclui texto
- tipos de dado, para os elementos e atributos
- valores fixos e default para elementos e atributos

Vantagens

- extensível para adições futuras
- são mais ricos e poderosos do que as DTDs
- são escritos em linguagem XML
- possuem suporte a vários tipos de dados

Dado um XML (nota.xml)

```
<?xml version="1.0"?>
<nota>
   <Para>Filho</Para>
   <De>mãe</De>
   <Cabecalho>Lembrete</Cabecalho>
   <Corpo>Não me esqueça!</Corpo>
</nota>
```

Sua DTD (nota.dtd)

```
<!ELEMENT nota (Para, De, Cabecalho,
  Corpo)>
<!ELEMENT Para (#PCDATA)>
<!ELEMENT De (#PCDATA)>
<!ELEMENT Cabecalho (#PCDATA)>
<!ELEMENT Corpo (#PCDATA)>
```

Seu XML Schema (nota.xsd)

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">

<xs:element name="nota">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Para" type="xs:string"/>
        <xs:element name="De" type="xs:string"/>
        <xs:element name="Cabecalho" type="xs:string"/>
        <xs:element name="Corpo" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>
```

Complicado ?

Vamos por partes...

XML Schema Definition Language

- Status junto ao W3C: ...
  - **XML Schema Part 0: Primer**
  - **XML Schema Part 1: Structures**
  - **W3C Recommendation 02 May 2001**

W3C Recommendation, 2 May 2001

XML Schema Definition Language

- Superconjunto dos recursos disponibilizados por *DTDs*

- Um XML schema é um documento XML cujos elementos especificam com rigor o conteúdo permitido nas instâncias dos documentos válidos correspondentes.

Características de um XML Schema

- ***Tipos simples de dados*** podem ser definidos por elementos que contêm apenas conteúdo. Os elementos são qualificados de tipo simples.
- ***Tipos complexos de dados*** requerem o uso de elementos com atributos e/ou subelementos. Os elementos são qualificados de tipo complexo.

- *Como elementos, atributos também são tipados, entretanto podem ser apenas de tipo simples.*

Em um XML Schema...

- a definição de tipos de elementos utiliza tipos definidos pelo autor do esquema ou os ***tipos simples pré-definidos*** de XML Schema (ex: `string`, `decimal, long, int, short, byte, bynary, date,time, year, positiveInteger, etc.`)
- o número de instâncias de um elemento pode ser rigorosamente especificado com `minOccurs` e `maxOccurs` (num DTD as opções são *nome, nome?, nome+, nome*)*
- novos tipos simples podem ser derivados a partir de outro tipos simples existentes -- por exemplo para especificar que um inteiro deve possuir valores entre “1” e “99” ou que uma *string* tenha a forma “768-KLM”
- o ***padrão de cadeias de caracteres*** pode ser especificado com o uso de expressões regulares de XML como em "`\d{3}-[A-Z]{2}`” que significa: “três dígitos seguido de um hífen seguido de duas letras maiúsculas.

Seu XML Schema (nota.xsd)

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">

<xs:element name="nota">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Para" type="xs:string"/>
        <xs:element name="De" type="xs:string"/>
        <xs:element name="Cabecalho" type="xs:string"/>
        <xs:element name="Corpo" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>
```

Como referenciar um XSD no XML?

O elemento <schema> é o elemento raiz de todo XML Schema!

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
...
</xs:schema>
```

```
<nota
xmlns="http://www.w3schools.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3schools.com nota.xsd">
```

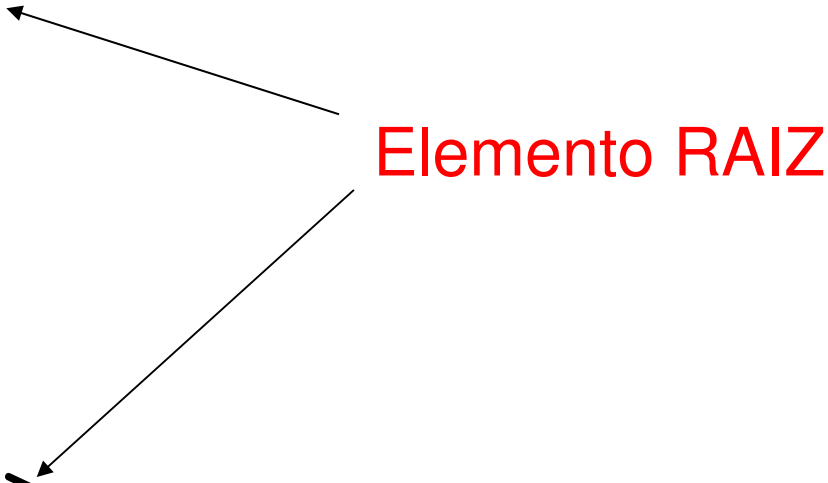
Onde tudo começa

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

<xs:schema>

...
...

</xs:schema>
```



- Também pode conter atributos

```
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

Os dois tipos de elementos

- Elemento Simples
- Elemento Complexo

Elemento simples

- É um elemento XML que pode conter somente **texto**; não pode conter outros elementos ou atributos.

O texto pode ser de vários tipos diferentes, incluindo tipos customizados.

- Sintaxe:

```
<xs:element name="title" type="xs:string"/>
```

Elemento Simples

```
<xs:element name="xxx" type="yyy"/>
```

Type:

- xs:string
- xs:decimal
- xs:integer
- xs:boolean
- xs:date
- xs:time

```
<Cabecalho>Lembrete</Cabecalho>

<XS:element name="Cabecalho" type="xs:string">
```

Elemento Simples

- Valores (Fixo e Default)

```
<xs:element name="cor"
  type="xs:string" fixed="vermelha"/>

<xs:element name="cor"
  type="xs:string"
  default="vermelha"/>
```

Atributos XSD

- O atributo é declarado como sendo um tipo simples.
- Sintaxe:

```
<xs:attribute name="id" type="xs:integer"/>
```

- Elemento XML com atributo:

```
<lastnane lang="EN">Smith</lastname>
```

- Definição em XML Schema:

```
<xs:attribute name="lang" type="xs:string"/>
```

Atributos – valores default e fixed

```
<xs:attribute name="lang"
  type="xs:string" default="EN"/>

<xs:attribute name="lang"
  type="xs:string" fixed="EN"/>
```

Atributos – obrigatórios e opcionais

```
<xs:attribute name="lang"
  type="xs:string" use="optional"/>

<xs:attribute name="lang"
  type="xs:string" use="required"/>
```

Restrições (1/5)

- Restrições são usadas para controlar valores aceitáveis para elementos XML ou atributos.

```
<xs:element name="age">
<xs:simpleType>
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="100"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
```

Restrições (2/5)

- Sobre um conjunto de valores – enumeration:

```
<xs:element name="car">
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:enumeration value="Audi"/>
    <xs:enumeration value="Ferrari"/>
    <xs:enumeration value="BMW"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
```

Restrições (3/5)

- O mesmo exemplo poderia ser:

```
<xs:element name="car“ type=“carType”/>

<xs:simpleType name=“carType”>
  <xs:restriction base=“xs:string”>
    <xs:enumeration value=“Audi”/>
    <xs:enumeration value=“Ferrari”/>
    <xs:enumeration value=“BMW”/>
  </xs:restriction>
</xs:simpleType>
```

Restrições (4/5)

- Sobre uma série de valores – pattern:

```
<xs:element name="letter">

<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9][0-9][0-9]"/>
  </xs:restriction>
</xs:simpleType>

</xs:element>
```

Restrições (5/5)

- Sobre tamanho – lenght:

```
<xs:element name="password">

<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:minLength value="5"/>
    <xs:maxLength value="8"/>
  </xs:restriction>
</xs:simpleType>

</xs:element>
```

Elemento Complexo

- 4 tipos :
  - Elemento vazio
  - Elemento somente com outros elementos
  - Elemento somente com texto
  - Elemento com texto e outros elementos

Elemento Complexo (Vazio)

Elemento vazio é aquele que não possui conteúdo algum, por exemplo:

```
<Produto prodid="1234">
```

```
<xs:element name="Produto">
  <xs:complexType>
    <xs:attribute name="prodid“
  type="xs:positiveInteger"/>
  </xs:complexType>
</xs:element>
```

OBS: Atributos são definidos como elementos simples

Elemento Complexo (Elementos filhos)

```
<Pessoa>
 <Nome>André</Nome>
 <Sobrenome>Silva</Sobrenome>
</Pessoa>

<xs:element name="Pessoa">
   <xs:complexType>
     <xs:sequence>
       <xs:element name="Nome" type="xs:string"/>
       <xs:element name="Sobrenome"
  type="xs:string"/>
     </xs:sequence>
  </xs:complexType>
</xs:element>
```

Elemento Complexo (Só texto)

Deve-se adicionar um elemento **<xs:simpleContent>** dentro do elemento *complexType*.

Ao utilizar o elemento <xs:simpleContent> deve-se por sua vez definir um elemento de extensão ou de restrição .

Exemplo: **<numerope pais="França">35</numerope>**

```
<xs:element name="numerope">
  <xs:complexType>
     <xs:simpleContent>
        <xs:extension base="xs:integer">
           <xs:attribute name="pais" type="xs:string" />
        </xs:extension>
     </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

Elemento Complexo (MIX)

```
<carta>
Caro Mr.<nome>John Smith</nome>.
Seu pedido <pedidoid>1032</pedidoid>
será enviado <data>2010-07-13</data>.
</carta>

<xs:element name="carta">
  <xs:complexType mixed="true">
    <xs:sequence>
      <xs:element name="nome" type="xs:string"/>
      <xs:element name="pedidoid"
 type="xs:positiveInteger"/>
      <xs:element name="data" type="xs:date"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Restrições

```
<xs:element name="idade">

<xs:simpleType>
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="120"/>
  </xs:restriction>
</xs:simpleType>

</xs:element>
```

Restrições

```
<xs:element name="carro">

<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:enumeration value="Audi"/>
    <xs:enumeration value="Golf"/>
    <xs:enumeration value="BMW"/>
  </xs:restriction>
</xs:simpleType>

</xs:element>
```

Restrições

- **enumeration** define uma lista de valores possíveis
- **fractionDigits** define um valor fracionário maior ou igual a zero
- **length** define o número exato de caracteres, desde que
- esse número seja maior que 0
- **maxExclusive** define um número máximo menor que o definido
- **maxInclusive** define um número máximo menor ou igual ao definido
- **maxLength** define um número máximo de caracteres
- **minExclusive** define um número mínimo menor que o definido
- **minInclusive** define um número mínimo menor ou igual ao definido
- **minLength** define um número mínimo de caracteres
- **pattern** define a seqüência exata de caracteres
- **totalDigits** define o número exato de dígitos permitido
- **whiteSpace** define como ajustar espaços em branco.

Restrições

```
<xs:element name="carta">

<xs:simpleType>
   <xs:restriction base="xs:string">
      <xs:pattern value="[a-z]"/>
   </xs:restriction>
</xs:simpleType>

</xs:element>
```

Revisão

- XML é uma linguagem de marcação que auxilia na estruturação e padronização de documentos.
- XML não possui restrição de tags, uma vez que qualquer coisa pode ser uma tag.
- Os documentos XML começam com <?XML version =``1.0''?> e são salvos com a extensão .xml.
- Documentos bem-formados são aqueles que atendem todas as regras da linguagem XML.
- Documentos válidos são documentos bem-formados definidos por regras de uma DTD ou de um XSD.

Revisão

- Document Type Definition (DTD) define a estrutura do XML e possui apenas um tipo de dado (#PCDATA)
- Os elementos que compõem a DTD são: CODTYPE, ELEMENT, ENTITY e ATTLIST
- XML Schema ou XSD é uma alternativa para DTDs
- Os XSDs são mais flexíveis e possem definições e restrições para vários tipos de dados.
- Os aquivos XSDs são salvos com a extensão **.xsd**.
- Existem dois tipos de elementos no XSD: elemento simples (só texto) e elemento complexo (elementos filhos e atributos)

Exercício prático (Individual)

- Defina o respectivo XSD o documento XML de sua grade horária.

- dica:

http://tools.decisionsoft.com/schemaValidate/

fim.