Esquemas para XML

Vanessa Braganholo
vanessa@ic.uff.br

Curso baseado em mini-cursos apresentados no SBBD. Autores: Carlos Heuser, Carina Dorneles e Vanessa Braganholo

Esquema XML

- Esquema XML: um vocabulário específico
- Pode ser associado a um documento XML
  - O documento tem que seguir as “regras” do vocabulário/esquema associado

Esquema XML × Esquema Relacional

- SGBD Relacional
  - Esquema: estrutura das tabelas
  - Instância: tuplas
- XML
  - Esquema: vocabulário que define regras que os documentos XML devem seguir
  - Instâncias: documentos XML

Doc. XML Bem-formado x Válido

- Documento XML **bem-formado**: documento que segue as regras de formação do XML (raiz única, bem aninhado, etc.)

- Documento XML **válido**: documento **bem-formado** que segue as regras de um **esquema** associado

Como seria este esquema?

- DTD - Document Type Definition
  - Gramática Regular
- XML Schema
  - Esquema escrito em XML

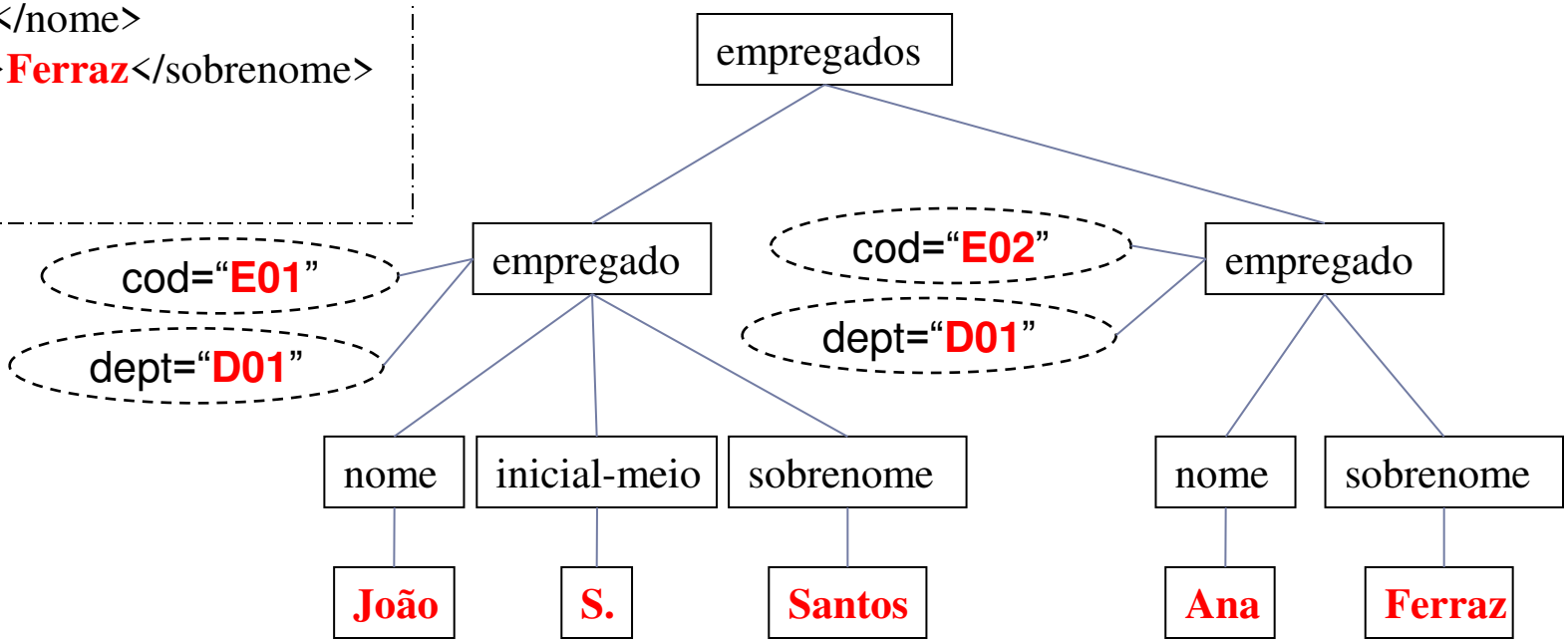
DTD - Document Type Definition

DTD (http://www.w3.org/TR/2008/REC-xml-20081126/#dt-markupdecl)

- Define as regras de formação dos elementos e atributos
  - Quais os elementos que podem aparecer em um documento
  - Em que ordem eles podem aparecer
  - Qual a hierarquia permitida para os elementos
  - Quais os atributos que um elemento pode conter

Exemplo que será utilizado na aula

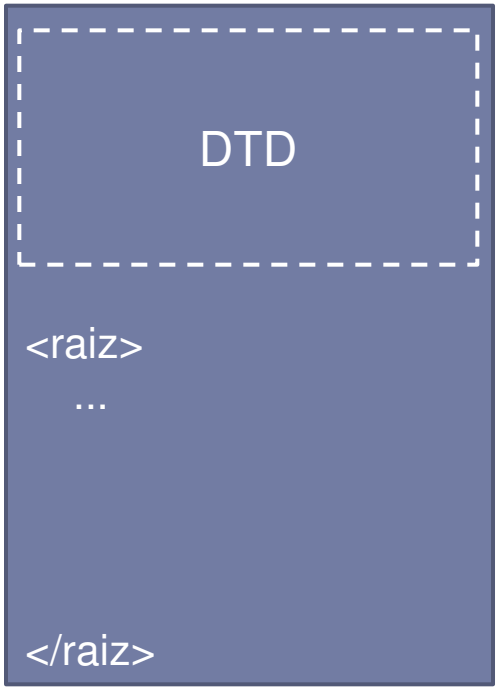
```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
     <nome>João</nome>
     <inicial-meio>S.</inicial-meio>
     <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
     <nome>Ana</nome>
     <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```



Local da Declaração

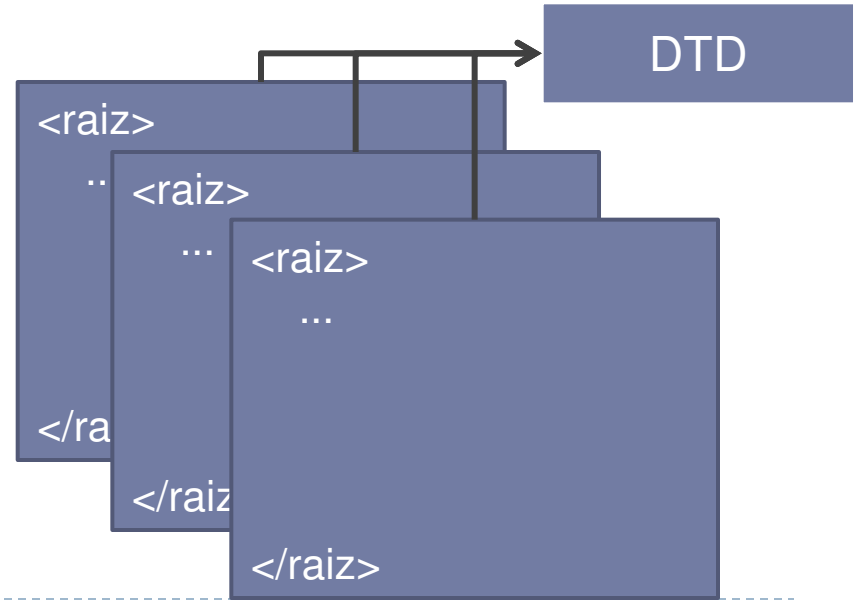
DTD Interna

- Definida dentro do documento XML



DTD Externa

- Definida fora do documento XML, em um arquivo fisicamente separado (.dtd)



DTD interna – no arquivo xml

```
<? xml version="1.0" ?>
<!DOCTYPE empregados [
<!ELEMENT empregados (empregado+)>
<!ELEMENT empregado (nome, inicial-
meio?, sobrenome)>
<!ATTLIST empregado
            cod CDATA #REQUIRED
            dept CDATA #REQUIRED
>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT inicial-meio (#PCDATA)>
<!ELEMENT sobrenome (#PCDATA)>
]>
<empregados>
  <empregado cod="E01" dept="D01">
     <nome>João</nome>
     <inicial-meio>S.</inicial-meio>
     <sobrenome>Santos</sobrenome>
  </empregado>
 <empregado cod="E02" dept="D01">
     <nome>Ana</nome>
     <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

DTD externa – arquivos físicos separados

```
<!ELEMENT empregados (empregado+)>
<!ELEMENT empregado (nome, inicial-meio?, sobrenome)>
<!ATTLIST empregado
                    cod CDATA #REQUIRED
                    dept CDATA #REQUIRED
>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT inicial-meio (#PCDATA
<!ELEMENT sobrenome (#PCDATA
```

Arquivo emp.dtd

Arquivo emp.xml

```
<? xml version="1.0" ?>
<!DOCTYPE empregados SYSTEM "emp.dtd">
<empregados>
  <empregado cod="E01" dept="D01">
     <nome>João</nome>
     <inicial-meio>S.</inicial-meio>
     <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
     <nome>Ana</nome>
     <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

Declaração da DTD

```
<? xml version="1.0" ?>
<!DOCTYPE empregados SYSTEM "emp.dtd">
<empregados>
  <empregado cod="E01" dept="D01">
      <nome>João</nome>
      <inicial-meio>S.</inicial-meio>
      <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
      <nome>Ana</nome>
      <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

Nome do tipo de documento
e do elemento raiz
devem ser **IGUAIS**

Declaração de Elemento

- <!ELEMENT empregados (empregado+)>

  Existe um elemento **empregados**, cujo conteúdo é constituído de um ou mais elementos **empregado**

- No documento XML

```
<empregados>
        <empregado>...</empregado>
        <empregado>...</empregado>
        <empregado>...</empregado>
</empregados>
```

Cardinalidade

- Cardinalidade: + * ?
  - + um ou mais
  - * zero ou mais
  - ? zero ou um
  - Se não houver símbolo de cardinalidade ao lado do elemento, ele é obrigatório

Exemplo

- <!ELEMENT empregado (nome, inicial-meio?, sobrenome)>

  O elemento empregado é composto de 3 sub-elementos: **nome** (obrigatório), **inicial-meio** (opcional), e **sobrenome** (obrigatório)

- No documento XML

```
<empregado>
        <nome>João</nome>
        <sobrenome>Santos</sobrenome>
</empregado>
```

Sequência e Escolha

- <!ELEMENT empregado (nome**,** inicial-meio**,** sobrenome)>

  O elemento empregado é composto de 3 sub-elementos: nome, inicial-meio, e sobrenome, **nessa ordem**

- <!ELEMENT empregado (nome **|** inicial-meio **|** sobrenome)>

  O elemento empregado é composto de 1 sub-elemento: **ou** nome **ou** inicial-meio **ou** sobrenome

Sequência e Escolha

- <!ELEMENT empregado (nome, inicial-meio, sobrenome)>

Doc. XML

```
<empregado>
        <nome>João</nome>
        <inicial-meio>S.</inicial-meio>
        <sobrenome>Santos</sobrenome>
</empregado>
```

- <!ELEMENT empregado (nome | inicial-meio | sobrenome)>

Doc. XML

```
<empregado>
        <nome>João</nome>
</empregado>
```

Elemento Textual

- <!ELEMENT nome (#PCDATA)>

  #PCDATA significa *Parsable Character Data*, ou seja, o conteúdo do elemento será analisado pelo processador que está lendo o arquivo XML

- No documento XML:

```
<nome>João da Silva</nome>
```

Elemento Vazio

- <!ELEMENT estudante EMPTY>

  O elemento estudante não possui conteúdo

- No documento XML:

```
<estudante/>
```

Elemento Misto

- <!ELEMENT endereco (#PCDATA | cidade)*>

Elemento endereço possui texto, e subelemento(s) cidade

- No documento XML:

```
<endereco>Rua das Flores, 45
            <cidade>Rio de Janeiro</cidade>
            CEP 24220-260
</endereco>
```

Conteúdo Misto

- Existe uma regra importante que deve ser seguida quando existe escolha entre texto e elementos filho.
  - **#PCDATA** deve ser obrigatoriamente
    - o primeiro *token* no grupo
    - o grupo deve ser de **escolha, opcional e com repetição.**
  - Exemplo

```
<!ELEMENT endereco (#PCDATA | cidade)*>
```

Qualquer conteúdo

- <!ELEMENT obs ANY>

- Neste exemplo, o elemento **obs** pode conter como conteúdo qualquer outro elemento que tenha sido declarado na DTD

Declaração de Atributos

- Declarados em uma declaração ATTLIST
  - Nome do elemento que contém o atributo
  - Lista de atributos
  - Para cada atributo da lista, devem ser declaradas as seguintes informações
    - nome do atributo
    - tipo do atributo (CDATA, ID, IDREF, etc.), ou lista de valores possíveis, separados por “|”
    - obrigatoriedade (#REQUIRED, #FIXED, #IMPLIED, ou um valor default para o atributo, informado entre aspas)

Exemplo

```
<!ATTLIST empregado
        cod CDATA #REQUIRED
        dept CDATA #REQUIRED
>
```

O elemento **empregado** possui dois atributos **cod**, e **dept**

O tipo de ambos é um texto (**CDATA**), e ambos são obrigatórios (**#REQUIRED**)

Exemplo

- <!ATTLIST empregado
  cod CDATA #REQUIRED
  dept CDATA #REQUIRED
  >

- No documento XML:

```
<empregado cod="E01" dept="D01">...</empregado>
```

Cardinalidade

- Atributos obrigatórios: utiliza-se a palavra reservada **#REQUIRED**

  <!ATTLIST empregado cod CDATA #REQUIRED>

- Atributos opcionais: utiliza-se a palavra reservada **#IMPLIED**

  <!ATTLIST empregado dept CDATA #IMPLIED>

- Atributos com valores fixos: **#FIXED** + valor

  <!ATTLIST empregado empresa CDATA #FIXED "ABC">

- Atributos podem ter valores *default*

  <!ATTLIST empregado sexo (F | M) "F" >

  <!ATTLIST empregado sexo CDATA "F" >

Neste caso, se o documento XML não fornecer o valor do atributo, o parser o adiciona automaticamente na hora do processamento

Declarações múltiplas

- Um elemento pode ter mais de uma declaração de lista de atributos. Exemplo:

```
<!ATTLIST livro id ID #REQUIRED
                    tipo (romance| policial) #REQUIRED>
..... ....
<!ATTLIST livro tipo (pocket| normal) #REQUIRED
                          autor CDATA #IMPLIED>
```

- A lista de atributos é combinada
- No caso de haver atributo com o mesmo nome, a **primeira declaração** é a que vale!

Tipos permitidos para atributos

- **CDATA** – texto

<ATTLIST documento versao CDATA #REQUIRED>
Uso: <documento versao="10.10.2001">…</documento>

- **NMTOKEN** – um name token

Não pode conter branco.
Aceita apenas *Letter | Digit | '.' | '-' | '_' | ':')*
<ATTLIST documento digito_verificacao NMTOKEN #IMPLIED>
Uso: <documento digito_verificacao="art1">…</documento>

- **NMTOKENS** – vários name tokens

<ATTLIST documento paginas NMTOKENS #IMPLIED>
Uso: <documento paginas="elem10 elem45 el77 ar102">…</documento>

Tipos permitidos para atributos

- **ENTITY** – entidade

<ATTLIST figura arquivo ENTITY #REQUIRED>

Uso: <figura arquivo="foto"/>

- **ENTITIES** – várias entidades

<ATTLIST figura arquivo ENTITIES #REQUIRED>

Uso: <figura arquivo="foto1ano foto10anos foto15anos"/>

Mais detalhes sobre entidades
podem ser encontrados no final deste material

Tipos permitidos para atributos

- **ID** – um identificador → deve ter valor único no documento

<ATTLIST capitulo Nr ID #REQUIRED>

Uso: <capitulo Nr="A1">…</capitulo>

Os valores para o tipo ID seguem as mesmas regras de construção de nomes de elementos (devem começar por letra, ":" ou "_")

- **IDREF** – referência a um ID

<ATTLIST referencia cap IDREF #REQUIRED>

Uso: O capítulo <referencia cap="A1"/> mostra as características de …

- **IDREFS** – referência a vários IDs

<ATTLIST referencia cap IDREFS #REQUIRED>

Uso: Os capítulos <referencia cap="A2 A3"/> apresentam …

Usando o ID e o IDREF

O valor atribuído ao atributo **cap** deve ser válido, ou seja, deve haver um atributo do tipo ID com aquele valor no doc. XML

- Na DTD:

```
<!ELEMENT capitulo(…)>
<!ATTLIST capitulo nr ID #REQUIRED>
<!ELEMENT referencia EMPTY>
<!ATTLIST referencia cap IDREF #REQUIRED>
```

- Na instância (documento) XML

```
…
<capitulo nr="A1">
   …
</capitulo>
<conclusao> Enfim, deve-se …. Como apresentado inicialmente no
   capitulo <referencia cap="A1"/> deve-se tomar cuidado ao definir …
</conclusao>
…
```

Exercício 1

- Crie uma DTD para a receita médica definida no exercício da aula de Introdução (pegar arquivo no site da disciplina)
  - Modifique o documento para incluir a declaração da DTD
  - Teste declaração interna, e depois declaração externa

Exercício 2

- Altere a DTD do exercício anterior para adicionar as seguintes características
  - um elemento vazio para o logo da clínica
  - um elemento observações que tenha como conteúdo QUALQUER elemento já definido nesta DTD

- Altere o documento XML de receita médica para que fique válido com a nova versão da DTD

Exercício 3

- Acrescente à DTD do exercício anterior as seguintes informações na forma de atributos:
  - Data de criação da receita, obrigatória no documento
  - A descrição do logo da clínica, obrigatória
  - O tipo da figura que representa o logo, com valor default GIF e grupo de escolha: GIF, JPG e BMP
- Altere a definição do elemento observações. Agora ele deve ser declarado como um elemento com conteúdo misto e subelementos **importante**.

Exercício 4

- Crie um documento XML que obedeça a seguinte DTD (artigo.dtd)

```
<!ELEMENT artigo (autor+, titulo, resumo, secao, bibliografia)>
<!ATTLIST artigo data CDATA #REQUIRED
    ultima_revisao CDATA #IMPLIED ultimo_revisor CDATA #IMPLIED
    versao CDATA #IMPLIED status CDATA #IMPLIED>
<!ELEMENT autor (nome)>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT resumo (#PCDATA)>
<!ELEMENT secao (#PCDATA|figura)*>
<!ELEMENT figura EMPTY>
<!ATTLIST figura nome CDATA #IMPLIED>
<!ELEMENT bibliografia (referencia)+>
<!ELEMENT referencia (obra, autor+, ano?, local?)>
<!ATTLIST referencia id ID #REQUIRED>
<!ELEMENT obra (#PCDATA)>
<!ELEMENT ano (#PCDATA)>
<!ELEMENT local (#PCDATA)>
```

Outras declarações na DTD

- Seções Condicionais

- Entidades de parâmetro
- Declaração de Notação

Relacionadas à estrutura física dos Documentos XML.

Seções Condicionais

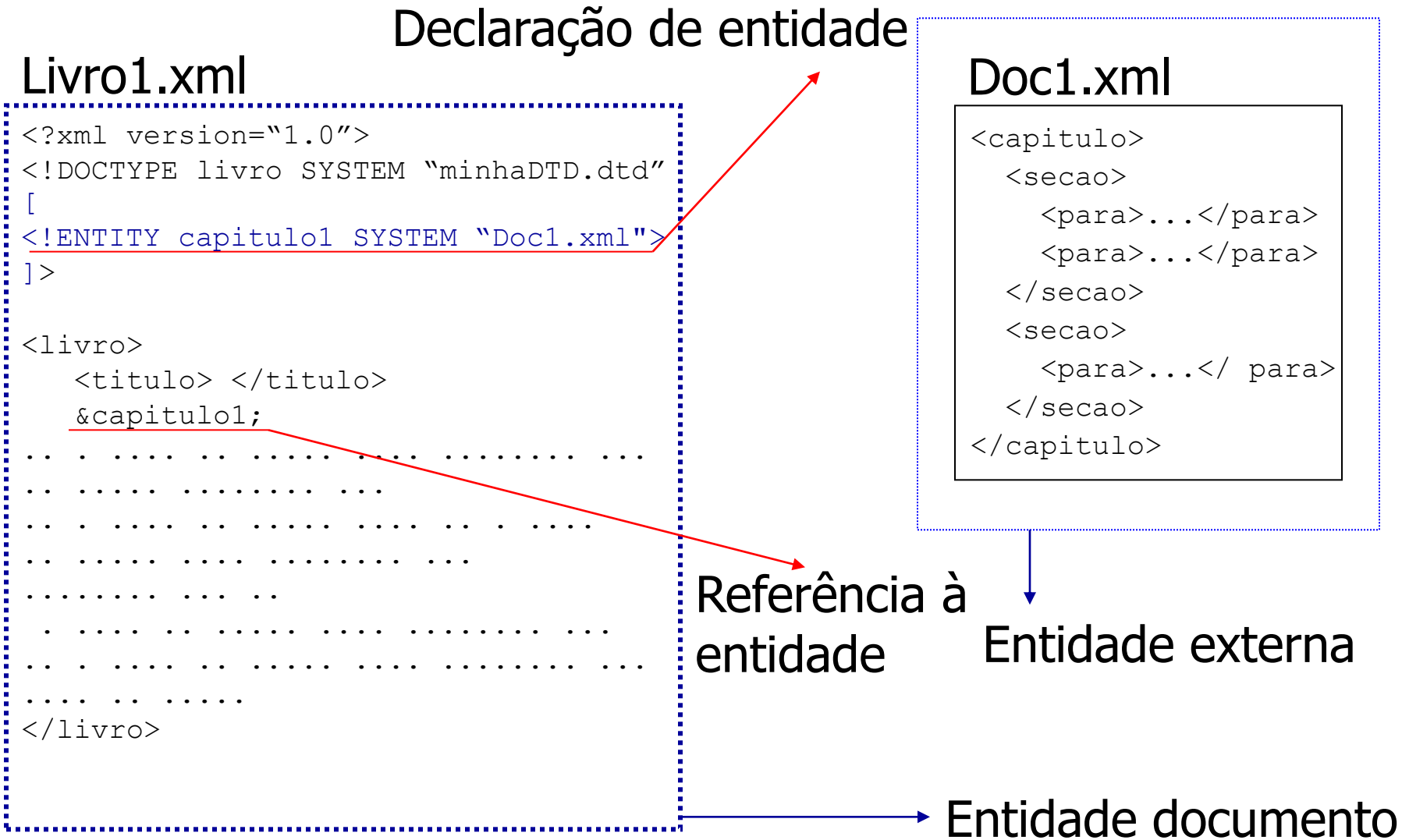
- Porções da DTD podem ser consideradas opcionais. Esta característica é fornecida pelo uso de seções condicionais
  - As seções que serão consideradas pelo processador XML devem ser marcadas como uma seção incluída .Por exemplo:
  - <![INCLUDE [.. ....]]>
  - Para que ela não seja processada pelo processador XML, a palavra INCLUDE deve ser trocada por IGNORE, marcando uma seção ignorada :
  - <![IGNORE [.. ....]]>

Estrutura física: Entidades

Entidades

- USO:
  - Separar um documento em vários
  - Permitir reuso
    - Ex.: Um livro onde cada capítulo é armazenado separadamente
  - Permitir inclusão de dados não XML
    - Ex.: As figuras utilizadas em um livro
- Podem ser dados XML ou arquivos binários

Conceitos


Declaração de entidade
Livro1.xml
<?xml version="1.0">
<!DOCTYPE livro SYSTEM "minhaDTD.dtd"
[
<!ENTITY capitulo1 SYSTEM "Doc1.xml">
]>
<livro>
<titulo> </titulo>
&capitulo1;
.. . .
.. ..
.. . .
.. ..
....
. ...
.. . .
....
</livro>
Doc1.xml
<capitulo>
<secao>
<para>...</para>
<para>...</para>
</secao>
<secao>
<para>...</ para>
</secao>
</capitulo>
Referência à entidade
Entidade externa
Entidade documento

Parsed × Unparsed

- Entidade *Parsed*
  - O conteúdo é considerado como parte integrante do documento e deve ser bem formado.
  - Conteúdo **é analisado** pelo parser
  - Documentos XML
- Entidade *Unparsed*
  - O conteúdo pode ser qualquer dado não-XML, arquivos binários, pdf, ps, etc, até mesmo conteúdo XML. Não existe restrição ao conteúdo deste tipo de entidade
  - Conteúdo **não é analisado** pelo parser
  - Normalmente utilizado para **dados binários**

Definição de uma Entidade

- Através de uma declaração
  - <!ENTITY ...>
- Deve ser definida **antes** de ser referenciada
- Nome é sensível a maiúsculas e minúsculas
- É possível definir a mesma entidade mais de uma vez, mas só a primeira será considerada pelo parser
  - <!ENTITY MinhaEntidade " ">
  - <!ENTITY MinhaEntidade " ">
- Uma entidade pode ser referenciada **n** vezes

Armazenamento de uma Entidade

- Dentro do documento XML
  - ENTIDADE INTERNA
    - Entidade de texto

- Em um arquivo separado
  - ENTIDADE EXTERNA
    - Entidade de texto
    - Entidade binária

Entidade de Texto Interna

- O conteúdo está contido entre “ “ . Exemplo
  - <!ENTITY xml "Extensible Markup Language">
- Ao ser referenciada o caracter “**&**” inicia a marcação e “**;**” finaliza. Por exemplo:
  - "... Em **&xml;** o conceito de entidades significa ..."
- O processador resolve:
  - "... Em Extensible Markup Language o conceito de entidades significa ..."

Entidade Externa (de texto e binária)

- Uso de um verificador de sistema

- Indicado pela palavra-chave SYSTEM

```
<!ENTITY minhaEnt SYSTEM "/ent/minhaEnt.xml">
```

Usando uma entidade

livro.xml

```
<?xml version="1.0">
<!DOCTYPE livro SYSTEM "livro.dtd"[
<!ENTITY capitulo1 SYSTEM "capitulo1.xml">
<!ENTITY capitulo2 SYSTEM "capitulo2.xml">
<!ENTITY capitulo3 SYSTEM "capitulo3.xml">
]>
<livro>
   <titulo> </titulo>
   &capitulo1;
   &capitulo2;
   &capitulo3;
</livro>
```

capitulo1.xml

```
<capitulo>
  <secao>
    <para>...</para>
    <para>...</para>
  </secao>
  <secao>
    <para>...</ para>
  </secao>
</capitulo>
```

capitulo3.xml

```
<capitulo>
  <secao>
    <para>...</para>
    <para>...</para>
  </secao>
  <secao>
    <para>...</ para>
  </secao>
</capitulo>
```

capitulo2.xml

```
<capitulo>
  <secao>
    <para>...</para>
    <para>...</para>
  </secao>
  <secao>
    <para>...</ para>
  </secao>
</capitulo>
```

livro.dtd

```
<!ELEMENT livro(titulo, capitulo+)>
<!ELEMENT titulo(#PCDATA)>
<!ELEMENT capitulo(secao+)>
<!ELEMENT secao(para+)>
...
```

Analisando o exemplo...

- Percebemos que uma entidade é conteúdo!
  - São definidas no documento XML

```
<?xml version="1.0">
<!DOCTYPE livro SYSTEM "livro.dtd"[
<!ENTITY cap1 SYSTEM "capitulo1.xml">
<!ENTITY cap2 SYSTEM "capitulo2.xml">
<!ENTITY cap3 SYSTEM "capitulo3.xml">
]>
...
```

  - Na DTD define-se APENAS a estrutura, jamais as entidades
    - Caso contrário existirá mistura de estrutura com conteúdo

Entidades *built in*

- O processador XML possui algumas entidades pré-definidas
  - < produz <
  - > produz >
  - & produz &
  - ' produz ’
  - " produz “

Entidades de parâmetro

- Elementos com conteúdo de escolha que são comuns em um conjunto de documento, podem ser definidos como entidades. Ex:

  <!ENTITY %comum "(parag | lista | tabela)">

- Dentro da declaração de elemento:

  <!ELEMENT capitulo ((%comum;)*, secao*) >

  <!ELEMENT secao ((%comum;)*) >

- Procurar utilizar "()" para separar o quantificador
- Utilizar exclusivamente em DTDs (estas entidades definem estrutura, e não conteúdo)

Reuso de entidades de texto interna

- Reuso de definição de entidade de texto interna
- Usa-se no topo do XML, juntamente com a definição das demais entidades:

```
<DOCTYPE livro SYSTEM “livro.dtd”[
<!ENTITY % minhasEntidades SYSTEM “mEnt.ent">
%minhasEntidades;
]>
```

- O arquivo **mEnt.ent** pode conter, por exemplo:

```
<!ENTITY xml “eXtensible Markup Language">
<!ENTITY html “HyperText Markup Language">
<!ENTITY sgml “Standard Generalized Markup Language”>
```

Reuso de entidades de parâmetro

- Reuso de definição de entidade de parâmetro
- Usa- se na DTD

```
<!ENTITY % minhasEntPar SYSTEM “mEntPar.ent">
%minhasEntPar;
]>
```

Vejam que isso permite que a DTD seja dividida em vários arquivos

- O arquivo mEntPar.ent pode conter:

```
<!ENTITY % comum “(lista, tabela)”>
<!ENTITY % parRef “(paragrafo, referencia)”>
<!ENTITY % endereco “(rua, numero, bairro)”>
```

Entidades Binárias

- Uso de SYSTEM – porque é externa!!

- Formatos não XML.
  - Exemplo de definição:

<!ENTITY foto SYSTEM "/ent/foto.gif“ ...>

- Ao referenciá- la no texto:
  - Foto de Fulano de Tal:<IMAGEM nome="foto"/>
  - Referenciar somente como valor de atributo, sem **&** e **;**
  - O atributo deve ser do tipo ENTITY ou ENTITIES

Entidades Binárias

- Uma entidade binária não é um documento XML
- Como indicar ao processador que a entidade binária é *unparsed,* e qual aplicação é capaz de processá-la??
  - Necessário o uso de notação => Estas declarações devem ser feitas DENTRO do documento XML

<!NOTATION GIF SYSTEM “c:/gimp/gimp.exe”>

<!ENTITY foto SYSTEM "/ent/foto.gif“ NDATA GIF>

Notações

- Uso da aplicação que indica o software que processa a notação é opcional
  - Forma mínima:

<!NOTATION GIF SYSTEM “”>

- Uso do tipo de entidade, quando binária, é obrigatório

<!ENTITY foto SYSTEM "/ent/foto.gif“ NDATA GIF>

Entidades Binárias – exemplo

- Na DTD:

```
<!ELEMENT figura EMPTY>
<!ATTLIST figura nome ENTITY #REQUIRED>
```

- No documento XML:

```
...
<!NOTATION GIF SYSTEM “c:/gimp/gimp.exe">
<!ENTITY foto SYSTEM "/ent/foto.gif" NDATA GIF>
...
<pessoa>
  <figura nome="foto“/>
<pessoa>
....
```

Onde referenciar uma entidade

- Locais que uma entidade pode ser referenciada:
  - Dentro do conteúdo de texto de um elemento
  - Em um atributo
  - Em um valor de uma entidade
  - Dentro das declarações de construtores da DTD
  - No entanto, existem restrições dependendo de onde ela é referenciada

Onde referenciar uma entidade

- Entidades de texto não podem aparecer nas DTD.
  - Exemplo abaixo está ERRADO

  - <!ELEMENT paragr(#PCDATA | Emph | **&Entidade;**)>

Onde referenciar uma entidade

- Atributos não podem referenciar uma entidade de texto EXTERNA, mas podem referenciar uma entidade de texto INTERNA.

```
<!ENTITY Empresa "Minha Empresa">
<!ENTITY Endereco SYSTEM "endereco.xml">
...
<livro proprietario="&Empresa; em &Endereco;">
```

Onde referenciar uma entidade

- O uso de entidades binárias só é permitido como conteúdo de atributos
- NA DTD:

```
<!ELEMENT fig EMPTY>
<!ATTLIST fig nome ENTITY #REQUIRED>
...
```

- No Documento XML:

```
<!DOCTYPE titulo SYSTEM “titulo.dtd”[
<!NOTATION TIFF SYSTEM “c:/gimp/gimp.exe”>
<!ENTITY MinhaFig SYSTEM "../figuras/fig.tif" NDATA TIFF>
]>
<titulo>
   Relato <fig nome=“MinhaFig”/>
</titulo>
```

Gerenciamento de Entidades

- Uso de identificadores
  - Um gerenciador de entidade usa um identificador para localizar entidades externas

- Mecanismo adicional
  - Usar identificador público indicado pela palavra-chave PUBLIC

```
<!ENTITY minhaEnt
PUBLIC "-//MinhaEmpresa/Entidades Superscript
   Charts//EN...">
```

  - Usado para manter a compatibilidade com o SGML

Gerenciamento de Entidades Externas

- Uso de identificadores públicos formais
  - Estrutura rígida composta por:
    - Tipo do identificador (PUBLIC – entidade de domínio público)
    - **Identificador proprietário** (**+** proprietário registrado(padrão), **-** proprietário não registrado)
    - **Classe de texto público** (em XML pode ser DTD, Entidades ou Notações)
    - **Descrição de texto público** (fornece informação adicional)
    - **Linguagem de texto público** (linguagem usada)
- Exemplo:

<!ENTITY ... PUBLIC "**+//MyCorp//TEXT...**">

<!ENTITY ... PUBLIC "**-//MyCorp//ENTITY Superscript Chars**//**EN**" >

<!ENTITY ... PUBLIC "**ISO 8879:1986**//...">

Identificador proprietário da ISO (A ISO não necessita o uso do "+" ou "–")

Exercício 5

- Altere a receita médica para incluir a imagem do logo da clínica e da assinatura do médico.
- É necessário alterar o documento XML e a DTD? Ou só o documento XML?

Exercício 6

- Separe o conteúdo da receita médica em vários arquivos XML, cada um contendo um medicamento