

XML

Modelo de dados e sintaxe

Carlos A. Heuser
UFRGS

02/1

Documento XML

□ Representação textual:

```
<?xml version="1.0"?>
<nota>
  <de>Carlos</de>
  <para>Eliane</para>
  Nao esquecer <ingles>birthday</ingles> amanha
</nota>
```

□ Qual é o modelo de dados por trás?

02/2

Alguns conceitos

```
<?xml version="1.0"?>
<nota>
  <de>Carlos</de>
  <para>Eliane</para>
  Nao esquecer <ingles>birthday</ingles>
  amanha
</nota>
```

02/3

Alguns conceitos - elemento

```
<?xml version="1.0"?>
<nota>
  <de>Carlos</de>
  <para>Eliane</para>
  Nao esquecer <ingles>birthday</ingles>
  amanha
</nota>
```

um elemento

02/4

Alguns conceitos - elemento

```
<?xml version="1.0"?>
<nota>
  <de>Carlos</de>
  <para>Eliane</para>
  Nao esquecer <ingles>birthday</ingles>
  amanha
</nota>
```

um elemento

02/5

Alguns conceitos - rótulo

```
<?xml version="1.0"?>
<nota>
  <de>Carlos</de>
  <para>Eliane</para>
  Nao esquecer <ingles>birthday</ingles>
  amanha
</nota>
```

um elemento

Rótulos (*tags*) de início (*start-tag*) e de fim (*end-tag*)

02/6

Alguns conceitos - Conteúdo de elemento

```
<?xml version="1.0"?>
<nota>
  <de>Carlos</de>
  <para>Eliane</para>
  Nao esquecer <ingles>birthday</ingles>
  amanha
</nota>
```

Conteúdo do elemento

02/7

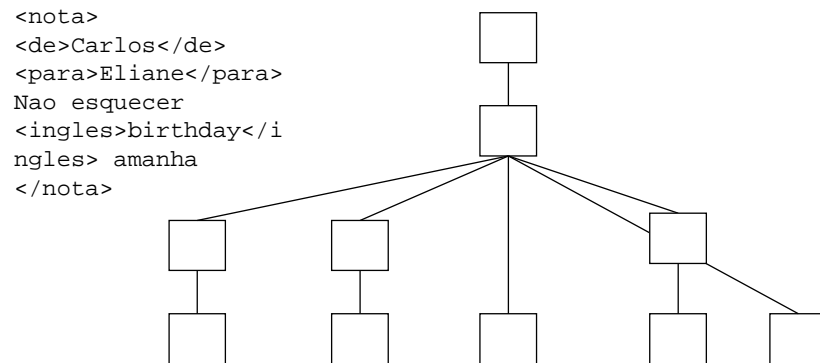
Modelo de dados

- Por trás da sintaxe XML há um **modelo de dados**
- Na realidade, há vários modelos (XML básico, XML Schema, Xquery,...)
- Modelo usado na programação:
 - DOM (*document object model*)

02/8

Modelo de dados - DOM

- Um documento XML é uma **árvore** (*tree*) formada por **nodos** (*node*) , ligados por arcos.
- Nodos filhos estão **ordenados**



02/9

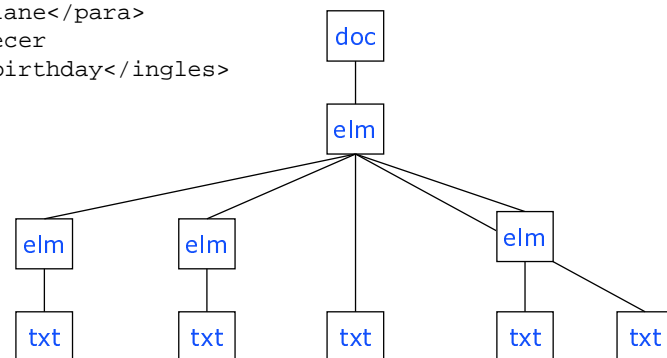
Tipos de nodos

- Cada nodo tem um *tipo* (*nodeType*)
- Alguns tipos:
 - Document* (doc)
 - Element* (elm)
 - Text* (txt)

02/10

Tipos de nodos

```
<nota>
<de>Carlos</de>
<para>Eliane</para>
Nao esquecer
<ingles>birthday</ingles>
amanha
</nota>
```

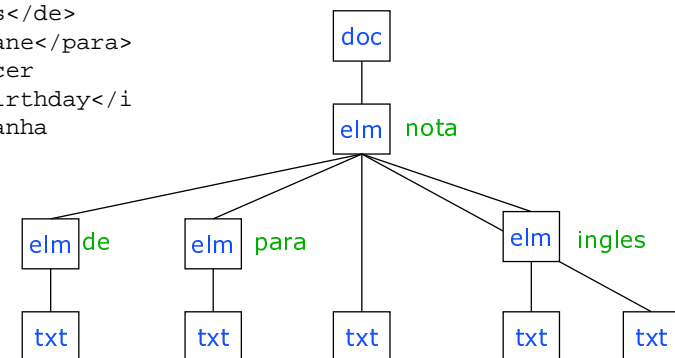


02/11

Nomes de nodos (*nodeName*)

- Cada nodo tem um **nome** (para elementos é o rótulo)

```
<nota>
<de>Carlos</de>
<para>Eliane</para>
Nao esquecer
<ingles>birthday</i
ngles> amanha
</nota>
```

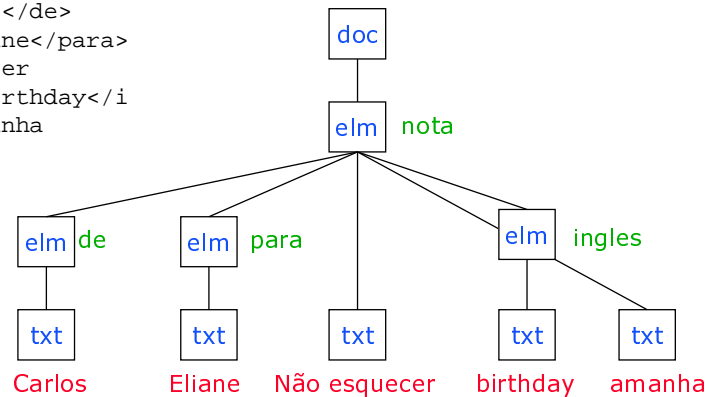


02/12

Valores de nodos (*nodeValue*)

- Cada nodo tem um **valor**

```
<nota>
<de>Carlos</de>
<para>Eliane</para>
Não esquecer
<ingles>birthday</i
ngles> amanhã
</nota>
```



02/13

Sintaxe – regras de formação

- Um documento na sintaxe correta é dito **bem formado** (*well-formed*)

02/14

Sintaxe - Regras de formação

- Todos os elementos devem ter o rótulo de fim:

`<p>Parágrafo em HTML` ✗

`<parag>Um possível parágrafo em XML</parag>` ✓

- Os rótulos XML são *case sensitive* (maiúsculas e minúsculas são diferenciadas)

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

`<MENSAGEM>Isto é correto</MENSAGEM>` ✓

`<mensagem>Isto é correto</mensagem>` ✓

02/15

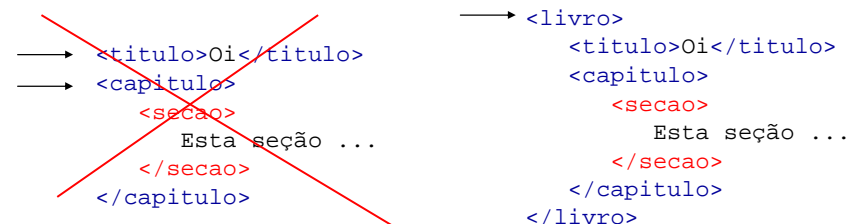
Sintaxe - Regras de formação

- Os rótulos devem estar **bem aninhados**

`<i>Isto é permitido em HTML, mas não em XML</i>`

`<i>Isto é permitido em XML</i>`

- TODO documento XML deve possuir um único **elemento raiz**



02/16

Sintaxe - Nomes de elementos

- Podem conter letras e números
- Não podem iniciar com números e caracteres de pontuação
- Não podem iniciar com a palavra xml,(XML, Xml, etc.)
 - Atualmente os *parsers* não validam isso, assim é permitido, mas o padrão NÃO aconselha o uso
- Não podem conter espaços em branco

02/17

Elementos - Estrutura recursiva

```
<list>
| <item>Primeiro item</item>
| <item>Segundo item</item>
| <item>
| | <list>
| | | <item>Primeiro sub-item</item>
| | | <item>Segundo sub-item</item>
| | </list>
| </item>
</list>
```

02/18

Sintaxe – elemento vazio

```
<?xml version="1.0"?>
<nota>
  <data></data>
  <de>Carlos</de>
  <para>Eliane</para>
  Nao esquecer <ingles>birthday</ingles>
  amanha
</nota>

<?xml version="1.0"?>
<nota>
  <data/>
  <de>Carlos</de>
  <para>Eliane</para>
  Nao esquecer <ingles>birthday</ingles>
  amanha
</nota>
```

é o mesmo que

02/19

Atributo – outro tipo de nodo

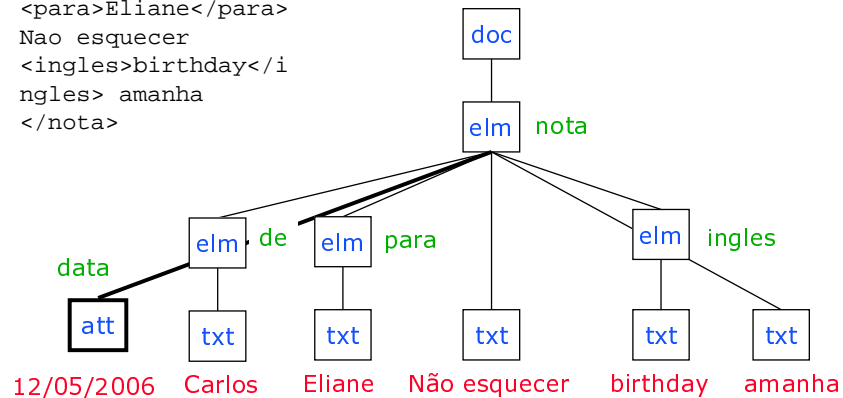
```
<?xml version="1.0"?>
<nota data="12/05/2006">
  <de>Carlos</de>
  <para>Eliane</para>
  Nao esquecer <ingles>birthday</ingles>
  amanha
</nota>
```

- Atributo aparece junto ao **rótulo de início** do elemento
- Atributo sempre contém texto, **não pode conter outros elementos**

02/20

Atributo na árvore DOM

```
<nota  
  data="12/05/2006">  
  <de>Carlos</de>  
  <para>Eliane</para>  
  Nao esquecer  
  <ingles>birthday</i  
  ngles> amanha  
</nota>
```



02/21

Atributo vs elemento

```
<?xml version="1.0"?>  
<nota data="12/05/2006">  
  <de>Carlos</de>  
  <para>Eliane</para>  
  Nao esquecer <ingles>birthday</ingles>  
  amanha  
</nota>
```

OU

```
<?xml version="1.0"?>  
<nota data="12/05/2006" de="Carlos" para="Eliane">  
  Nao esquecer <ingles>birthday</ingles>  
  amanha  
</nota>
```

02/22

Outros tipos de nodos

- Instruções de processamento
 - Instrução XML
- Declarações
 - Comentário
 - Seção CDATA
 - ... (outras serão vistas depois)

02/23

Instrução de processamento

- Trecho delimitado por <? e ?>
- Contém uma instrução a ser processada por uma aplicação que lê o documento
- Instrução especial XML com a qual começa um arquivo XML:
 <?xml version="1.0"?>

02/24

Instrução especial XML - sintaxe

- É uma instrução de processamento especial

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

□ Parâmetros

- `version` indica a versão do documento (atualmente, existe somente uma versão) - **obrigatório**

- `encoding` indica o código de caracteres utilizado no documento - **opcional**
cada codificação pode representar um número "x" de caracteres

Por *default*, a codificação "UTF-8" é utilizada. Outros valores possíveis para o atributo *encoding* são:

ISO-8859-1, UTF-16, ISO-10646, UCS-2, UCS-4, US_ASCII, entre outros.

02/25

Instrução especial XML - sintaxe

- É uma instrução de processamento especial

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

□ Parâmetros

- `standalone` - **opcional**

"yes" indica que não declarações externas que afetam a interpretação (*default*)

"no" indica que um conjunto de declarações definido externamente contém informação que afeta a interpretação do conteúdo do documento. O valor "no" deve ser usado se qualquer elemento, atributo ou entidade externa for definida em uma DTD externa

02/26

Instrução de processamento XML – *encoding* para Português

- Exemplo de conteúdo com acentos

- Exige encoding correspondente

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<nota data="12/05/2006">
  <de>Carlos</de>
  <para>Eliane</para>
  Não esquecer <ingles>birthday</ingles>
  amanhã
</nota>
```

02/27

Declarações

- Declarações são delimitadas por `<!` e `>`

- Observar a ausência do ponto de exclamação ao fechar

- Algumas declarações

- Declaração CDATA

- Permite incluir qualquer texto, inclusive marcações no documento

- Comentário

02/28

Declaração CDATA

- ❑ Em um segmento de texto CDATA (*character data*) os rótulos de marcação não são processados

- Possibilita o uso de *rótulos* no texto
- Delimitadas por "<![CDATA[" e "]">"
- Exemplo:

```
<![CDATA[Um exemplo do elemento: <titulo>]]>
```

- ❑ Resultado depois do documento ser processado:

Um exemplo do elemento: <titulo>

- ❑ Aparece como um nodo separado na árvore DOM

02/29

Declaração comentário

- ❑ Comentário bem formado

```
<!-- Este é um comentário -->
```

- ❑ Não pode conter --
- ❑ Comentário com erro

```
<!-- Este é um comentário -- ILEGAL- - -->
```

02/30

Outras declarações (usadas na DTD - vistas depois)

- ❑ <!ENTITY >
 - Definição de entidades
- ❑ <!NOTATION >
 - Definição de notações
- ❑ <!ELEMENT >
 - Definição de elementos
- ❑ <!ATTLIST >
 - Definição de lista de atributos
- ❑ <![IGNORE[.....]]>
 - Definição de seção a ser ignorada no processamento
- ❑ <![INCLUDE[.....]]>
 - Definição de seção a ser incluída no processamento

02/31