

# 2

# XML

## sintaxe e estrutura

Helder da Rocha  
(helder@argonavis.com.br)

# Como criar um documento XML

- **XML não tem comandos, nem operadores, nem funções, nem tipos**
  - Não é exatamente uma "linguagem"
  - A especificação XML não estabelece nenhum vocabulário
  - Define apenas uma estrutura e sintaxe **geral** para a organização de informações estruturadas
- **Para criar o menor documento XML**
  - Abra um editor de textos qualquer
  - Salve o arquivo com extensão **.xml**
  - Escreva um elemento raiz vazio; por exemplo:  
**<hello/>**
  - Salve o arquivo
  - Abra em um browser como Firefox ou Internet Explorer



# Especificação XML

- *As regras para criação de documentos XML são definidas pelo World Wide Web Consortium (W3C) através de especificação*
  - <http://www.w3.org/TR/xml/>
- *A especificação não define*
  - *nomes de elementos e atributos (cada aplicação define os seus)*
  - *como escrever documentos **válidos**: a validade de um documento XML é definido pelo **autor da aplicação** em que ele é usado*
- *A especificação define*
  - *tokens, caracteres e formatos de texto que podem ser usados em documentos XML (basicamente texto Unicode)*
  - *elementos e atributos reservados (começam com o string **xml**)*
  - *regras mínimas que possibilitam a leitura por um processador XML: um documento que segue essas regras é **bem formado***
  - *como uma aplicação pode validar um documento XML usando um DTD (Document Type Definition) com sintaxe similar a SGML*



# Estrutura XML

- Um documento XML pode ser representado como uma **árvore**. A estrutura é formada por vários **nós** (galhos e folhas)

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

```
<!-- Isto é um comentário -->
```

informações usadas  
pelo processador XML

```
<cartao-simples>
```

```
  <logotipo href="/imagens/logo14bis.gif" />
```

```
  <nome>Alberto Santos Dumont</nome>
```

```
  <endereco>Rua do Encanto, 22 - 2o. andar -  
Centro - 25600-000 - Petrópolis - RJ</endereco>
```

```
  <email>dumont@14bis.com.br</email>
```

```
  <telefone tipo="residencial">
```

```
    <ddd>21</ddd>
```

```
    <numero>2313011</numero>
```

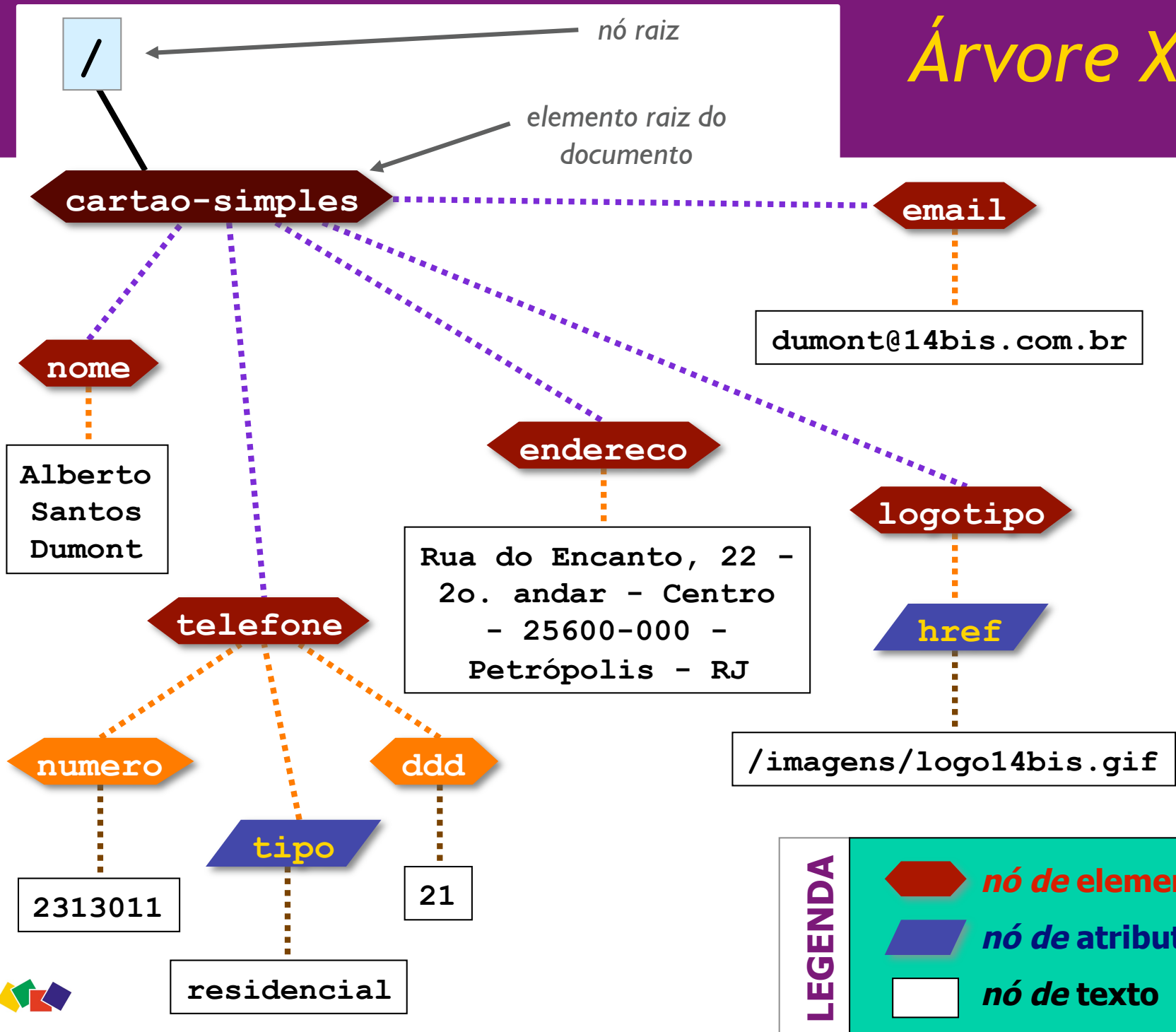
```
  </telefone>
```

```
</cartao-simples>
```

um "nó" pode ser ...

- um elemento,
- um atributo,
- um bloco de texto,
- um comentário,
- uma instrução,
- uma declaração,
- uma entidade, ...

# Árvore XML



# Componentes de um documento

- *Um documento pode conter*
  - *Prólogo*
  - *Comentários*
  - *Instruções de processamento*
  - *Atributos nos elementos*
  - *Nós de texto dentro dos elementos*
  - *Elementos aninhados (sendo apenas um na raiz)*
  - *Conteúdo misto (elemento e texto) dentro de elementos*
  - *Entidades gerais*
  - *Entidades de caractere*
  - *Blocos CDATA*



Declaração XML

Comentário (pode aparecer em qualquer lugar)

Instrução de processamento

Declaração de tipo de documento

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!-- Isto é um comentário -->
<?comando tipo="simples" parametro ?>
<!DOCTYPE cartao-simples SYSTEM "cartoes.dtd">
<cartao-simples>
  <logotipo href="/imagens/logo14bis.gif" />
  <nome>Alberto Santos Dumont</nome>
  <endereco>Rua do Encanto, 22 - 2o. andar -
  Centro - 25600-000 - Petrópolis - RJ</endereco>
  <email>dumont@14bis.com.br</email>
  <telefone tipo="residencial" >
    <ddd>21</ddd>
    <numero>2313011</numero>
  </telefone>
</cartao-simples>
```



# Nó raiz e elementos

elemento raiz do documento

nó raiz (/)

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

```
<cartao-simples>
```

```
<logotipo href="/imagens/logo14bis.gif" />
```

```
<nome>Alberto Santos Dumont</nome>
```

```
<endereco>Rua do Encanto, 22 - 2o. andar -  
Centro - 25600-000 - Petrópolis - RJ</endereco>
```

```
<email>dumont@14bis.com.br</email>
```

```
<telefone tipo="residencial" >
```

```
<ddd>21</ddd>
```

```
<numero>2313011</numero>
```

```
</telefone>
```

```
</cartao-simples>
```

elementos

elementos



- Só podem conter um descendente: nó de texto

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

```
<cartao-simples>
```

```
  <logotipo href="/imagens/logo14bis.gif" />
```

```
  <nome>Alberto Santos Dumont</nome>
```

```
  <endereco>Rua do Encanto, 22 - 2o. andar -  
Centro - 25600-000 - Petrópolis - RJ</endereco>
```

```
  <email>dumont@14bis.com.br</email>
```

```
  <telefone tipo="residencial" >
```

```
    <ddd>21</ddd>
```

```
    <numero>2313011</numero>
```

```
  </telefone>
```

```
</cartao-simples>
```

*atributos*



# Nós de texto

- Não podem ter descendentes (são as folhas da árvore)

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

```
<cartao-simples>
```

```
  <logotipo href="/imagens/logo14bis.gif" />
```

```
  <nome>Alberto Santos Dumont</nome>
```

```
  <endereco>Rua do Encanto, 22 - 2o. andar -  
Centro - 25600-000 - Petrópolis - RJ</endereco>
```

```
  <email>dumont@14bis.com.br</email>
```

```
  <telefone tipo="residencial" >
```

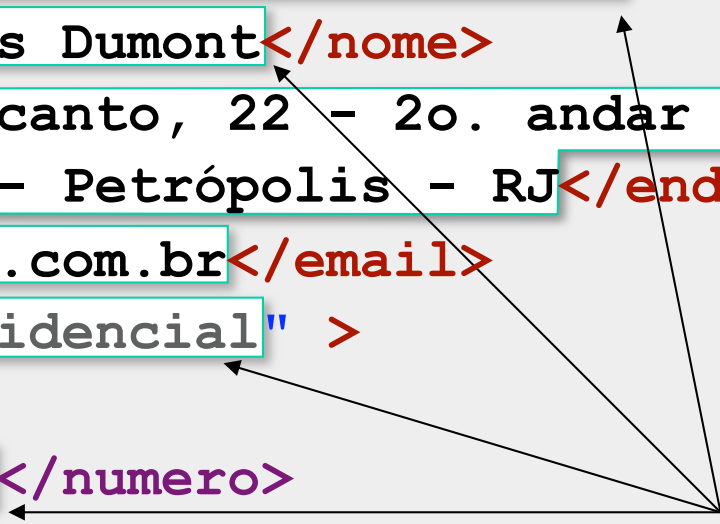
```
    <ddd>21</ddd>
```

```
    <numero>2313011</numero>
```

```
  </telefone>
```

```
</cartao-simples>
```

nós de  
texto



# Entidades gerais

- *São constantes associadas a um valor de texto*
  - *Podem aparecer em qualquer lugar do documento*
  - *São substituídas **durante** o processamento do documento*
  - *Podem ser definidas pelo usuário (via DTD)*
- *Sintaxe: **&entidade;***
- *Exemplo:*
  - *&data\_de\_hoje;*
- *Entidades pré-definidas:*
  - **&lt;**      *que corresponde a*      **<**
  - **&gt;**      *que corresponde a*      **>**
  - **&amp;**      *que corresponde a*      **&**
  - **&quot;**    *que corresponde a*      **"**
  - **&apos;**    *que corresponde a*      **'**



# Entidades de caracteres

- *Também são substituídas durante o processamento do documento*
- *Sintaxe:*
  - **&#CÓDIGO\_16b-decimal;**
  - **&#xCÓDIGO\_16b-hexadecimal;**
- *Exemplos:*
  - **&#065;** **%#x0042;**
  - **&#x0020;** *representa um espaço em Unicode*
  - *Veja mais em [www.unicode.org/charts/](http://www.unicode.org/charts/)*
- *XML não define por default as entidades do HTML*
  - *Não existe **&nbsp;** ou **&atilde;** a menos que sejam definidas em um DTD (como no DTD do XHTML)*



# Elementos e atributos: regras básicas

- Etiqueta **inicial** e **final** têm que ter o mesmo **nome** (considerando diferença de maiúscula e minúscula)
  - Não pode haver espaço depois do `<` nas etiquetas iniciais nem depois do `</` nas finais
- Atributos têm sempre a forma **nome**, seguido de `'='`, seguido do **valor** entre aspas ou apóstrofes
  - `nome="valor"` ou `nome = 'valor'` são válidos
  - aspas podem ser usadas entre apóstrofes
  - apóstrofes podem ser usados entre aspas
  - aspas e apóstrofes não podem ser neutralizados, mas podem ser representados pelas entidades `&quot;` e `&apos;`
  - Não pode haver atributos na etiqueta final
  - Atributos não podem se repetir no mesmo elemento



# Um documento é bem formado quando

- *Tem um único elemento raiz*
- *Todas as etiquetas iniciais e finais dos seus elementos combinam (levando em conta maiúsculos e minúsculos)*
- *Seus elementos estão bem aninhados*
  - *Não acontece nada do tipo `<a><b></a></b>`*
- *Valores dos atributos estão entre aspas ou apóstrofes*
- *Os atributos não se repetem*
- *Elementos e atributos têm identificadores válidos*
- *Comentários não aparecem dentro de etiquetas*
- *Sinais `<` ou `&` nunca ocorrem dentro de atributos ou nós de texto do documento*



# Elementos e atributos

- *Elementos mal formados*

```
<Profissão>Arquiteto</profissão>  
<TR><TD>item um</td></tr>  
<td>item um</ td>  
<equacao>x + y < z + k</equacao>  
<ДЕНГИЙ>139.00</денгий>
```

- *Atributos mal formados*

```
<profissao tipo=1>Arquiteto</profissão>  
<chave x="X$%9_"PZ99" />  
<a x="1" y="2" z="3" x="0" />
```

- *Elementos e atributos bem formados*

```
<profissao tipo='2'>Físico</profissão>  
<chave x = "X$%9_&quot;PZ99" />
```



# Quando usar elementos ou atributos?

- *Há várias maneiras de representar a mesma informação em XML*

```
<data>23/02/1998</data>
```

```
<data dia="23" mes="02" ano="1998" />
```

```
<data>  
  <dia>23</dia>  
  <mes>02</mes>  
  <ano>1998</ano>  
</data>
```



# Quando usar elementos ou atributos?

- *Uma questão de design*
  - *Elementos geralmente referem-se a **coisas** que têm atributos*
  - *Atributos geralmente são **características** dessas coisas que podem ser descritas com poucas palavras*
- *Uma questão de suporte tecnológico*
  - *Atributos não podem conter subelementos*
  - *Atributos são mais fáceis de serem validados num DTD*
- *Sempre que possível, priorize os argumentos de design aos de suporte tecnológico*



# Identificadores de elementos e atributos

- Nomes de atributos e elementos
- Podem conter
  - qualquer caractere alfanumérico ou ideograma
  - . (ponto)
  - - (hífen)
  - \_ (sublinhado)
- Não podem **começar** com
  - ponto
  - hífen
  - número
- Não podem começar com a seqüência 'xml'
  - É reservada para atributos e elementos com significado especial, definido em especificação (ex: xmlns, xml:lang)



# Identificadores de elementos e atributos

- *Elementos bem formados*

```
<αριστοτελεσ>περι ποιητικησ</αριστοτελεσ>
```

```
<книга xml:lang='ru' >
```

```
  <название>Евгений Онегин</название>
```

```
  <автор рождение="1799"
```

```
    смерть="1837">Александр Сергеевич Пушкин</автор>
```

```
</книга>
```

```
<_1_/>
```

```
<cdd:gen.inf cdd:cod="005">Introdução a XML</cdd:gen.inf>
```

- *Elementos mal formados*

```
<3-intro>Fundamentos</3-intro>
```

```
<cartão de crédito>1234567887654321</cartão de crédito>
```

```
<xmligue>Coisas</xmligue>
```



- *Texto misturado com elementos XML*

<trecho>

<secao>2</secao>

<paragrafo>A unidade de informação dentro de um documento XML é o

<definicao>elemento</definicao>. Um elemento é formado por duas

<definicao>etiquetas</definicao> que atribuem algum significado ao conteúdo. </paragrafo>

</trecho>



- *Ignora efeitos especiais dos caracteres*

```
<titulo>Curso de XML</titulo>
```

```
<exemplo>Considere o seguinte trecho de XML:
```

```
<![CDATA[
```

```
    <empresa>
```

```
        <nome>João & Maria S/A</nome>
```

```
    </empresa>
```

```
]]>
```

```
</exemplo>
```



# Instruções de processamento

- *Instruções dependentes do processador*
- *Funcionam como comentários para os processadores que não a conhecem*

```
<?nome-do-alvo área de dados ?>
```

```
<?query-sql  
    select nome, email  
        from agenda  
        where id=25  
?>
```



- *Iguais aos comentários HTML*

```
<!-- Isto é um comentário -->
```

- *Comentários não podem conter a seqüência --*

```
<!-- isto é um erro -- sério! -->
```



# Declaração XML

- *É uma instrução de processamento para o processador XML*
- *É opcional*
  - *Exceto quando o encoding não for UTF-8 (default)*

```
<?xml version="1.0"  
      encoding="iso-8859-1"  
      standalone="yes" ?>
```



# XML Namespaces

- *Permite que elementos de mesmo nome de diferentes aplicações sejam misturados sem que haja conflitos*
- *Um **namespace** (universo de nomes) é declarado usando atributos reservados*
  - ***xmlns="identificador"** (namespace default): associa o identificador com todos os elementos contidos no elemento que declara o atributo que não estão **qualificados** com prefixo. Ex: <nome>*
  - ***xmlns:prefixo="identificador"**: associa o identificador com os elementos e atributos contidos no elemento que declara o atributo cujo nome local é precedido do prefixo. Ex <prefixo:nome>*
- *O prefixo é arbitrário e só existe dentro do documento*
- *O identificador (geralmente uma URI) deve ser reconhecido pela aplicação para validar o documento*
- **XML Namespaces** é uma especificação a parte
  - <http://www.w3.org/TR/xml-names/>



# Exemplo

Vale para todo o elemento <cartao>

Esta URI está associada a este prefixo

```
<ct:cartao
  xmlns:ct="urn:B1-01.234.567/cartoes">
  <ct:nome>Alberto Santos Dumont</ct:nome>
  <ct:endereco>Rua do Encanto, 22 - Centro
  25600-000 - Petrópolis - RJ</ct:endereco>
  <ct:email>dumont@14bis.com.br</ct:email>
  <ct:telefone tipo="residencial">
    <ct:ddd>21</ct:ddd>
    <ct:numero>2313011</ct:numero>
  </ct:telefone>
</ct:cartao>
```



# Exemplo com 3 namespaces

```
<departamento
  xmlns:ct="urn:B1-01.234.567/cartoes"
  xmlns="emp:E2-3349.9.0001-89/empresa"
  xmlns:html="http://www.w3.org/WD/REC-HTML/Strict">

  <ct:nome>Fulano de Tal</ct:nome>
  <nome>Contabilidade</nome>
  <endereco>Rua Projetada, 33</endereco>
  <html:a href="web.html">
    <html:strong>link negrito HTML</html:strong>
  </html:a>
  <urgencia><ct:numero>2313011</ct:numero></urgencia>
</departamento>
```

Namespace default

URI padrão XHTML



# Observações importantes sobre namespaces

- O escopo da declaração **xmlns** (sem prefixo) inclui
  - O elemento onde ela acontece
  - Os elementos-filho
- O escopo da declaração **xmlns:prefixo** inclui
  - O próprio elemento se qualificado com mesmo prefixo
  - Os elementos-filho qualificados com o prefixo
  - Os atributos do elemento onde ocorre a declaração e elementos filho qualificados com o prefixo
- O identificador **não** representa endereço na internet
  - Geralmente é escrito como URL, porque URLs são unívocas
  - O identificador é **string** e **não endereço**: omitir ou incluir uma / final faz diferença
- Declarar e usar um namespace **pode** ser opcional
  - Depende da aplicação que irá processar o documento

