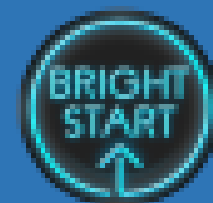


EXTENSIBLE MARKUP LANGUAGE

APOIO XML



FILIPESÁ FILIPES.A@ESTGV.IPV.PT



XML - INTRODUÇÃO

- O que é?
 - Linguagem para definir linguagens específicas de anotação (Ex. HTML, DocBook, EAD)
- É um standard ISO
- Ideal para armazenamento e intercâmbio
- Uma linguagem baseada em anotações

XML: MOTIVAÇÃO

- Integração de Dados:
 - Evitar INCOMPATIBILIDADES na troca de dados entre sistemas com software e hardware diferentes?
 - Solução fácil: dados codificados em ASCII

mas... só ASCII tem muitas limitações, é necessária alguma estrutura.

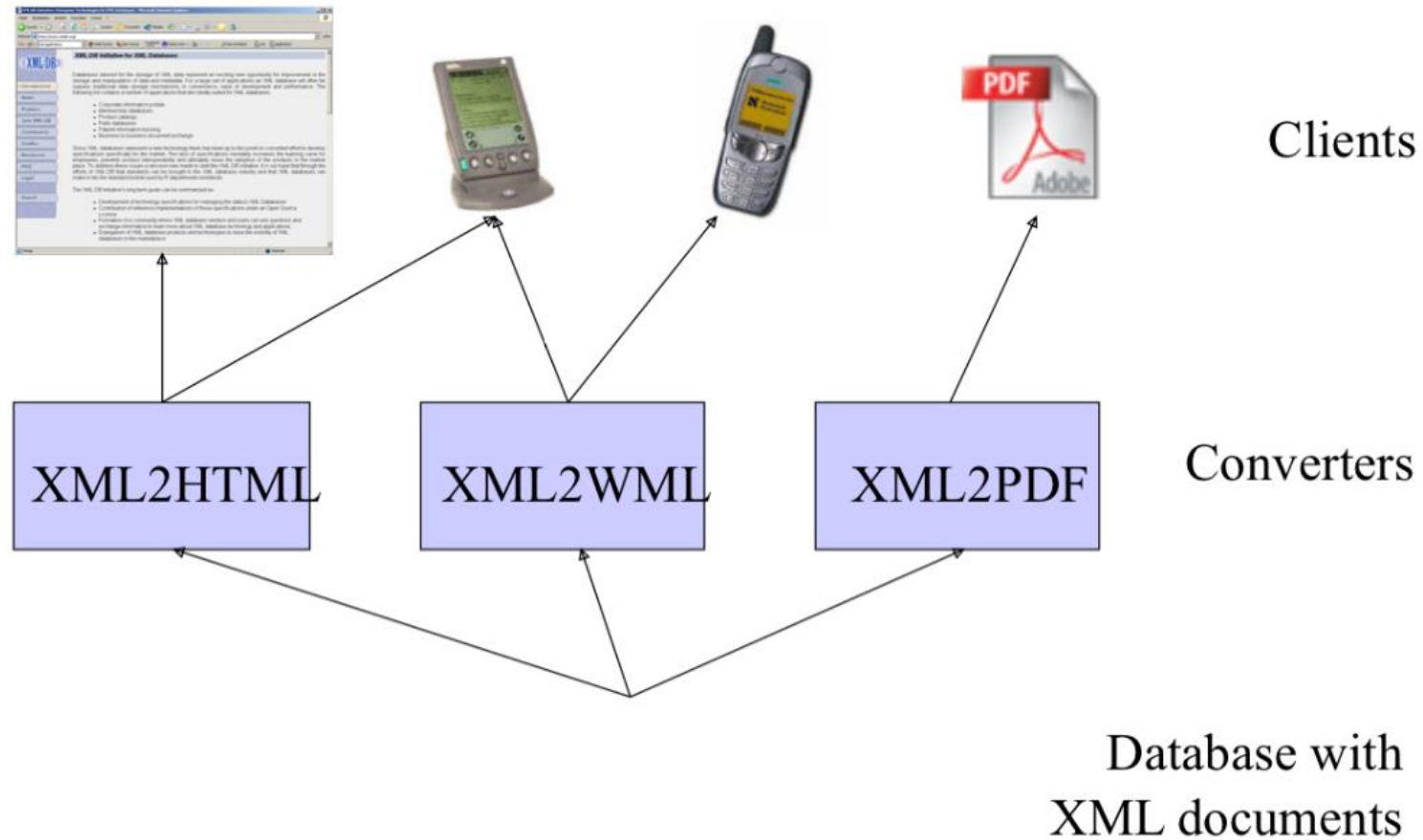
XML: MOTIVAÇÃO

- HTML é ASCII com alguma estrutura
 - é fácil de ler e visualizar
 - possui standards que o tornam independente dos browsers
 - várias formas de visualizar texto, imagens, formulários.

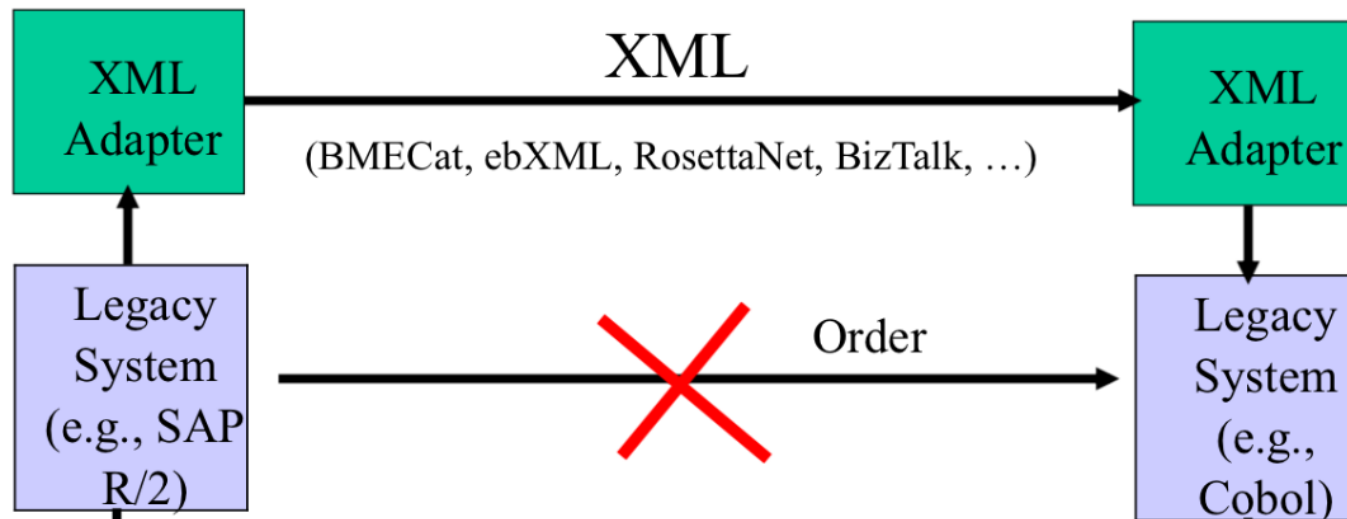
mas... tem limitações

- elementos e atributos fixos
- os dados e a formatação estão misturados
- difícil de automatizar a interpretação do significado dos dados (o que é title?, onde estão nomes, autores, ...)
- misturar informação de vários ficheiros HTML não é fácil - não tem uma linguagem de pesquisas

XML: MOTIVAÇÃO



XML: MOTIVAÇÃO



XML: ETAPAS

analisar a estrutura da informação e os atributos que a caracterizam

determinar quais as funções de processamento que produzirão o formato desejado para cada elemento

inserir as etiquetas/anotações no texto

XML: CARACTERÍSTICAS

XML = eXtensible Markup Language

- Criada em 1998
- Standard da W3C (WWW Consortium)

Independente dos mecanismos de armazenamento e transmissão

- Dados são codificados em ASCII/UNICODE (e.g. UTF-8)
- Pode ser enviada por email, ftp, armazenada em ficheiros, bases de dados, etc

XML: CARACTERÍSTICAS

Linguagem baseada em elementos (Tags) e atributos

- Ex: <peessoa tipo="func">Filipe Sá</peessoa>

Linguagem estruturada

- Os ficheiros são organizados em forma de árvore;
- DOM – Document Object Model

Extensível

- Os elementos e atributos não são pré-definidos e fixos;

Ferramentas/Tecnologias associadas:

- XML Schema, DTD, Xpath, XSLT, XQuery, SOAP, ...

XML: CARACTERÍSTICAS

- XML fornece uma sintaxe, não fornece semântica
- As etiquetas (tags) não têm significado definido pela linguagem, mas podem fazer sentido para as aplicações
- Define apenas a estrutura/conteúdo de um documento, não o seu comportamento nem o seu tratamento

XML: ESTRUTURA

"Soneto Já Antigo"
(Álvaro de Campos)

Olha, Daisy: quando eu morrer tu hás-de
dizer aos meus amigos aí de Londres,
embora não o sintas, que tu escondes
a grande dor da minha morte.

Irás de Londres p'ra Iorque, onde nasceste
(dizes que eu nada que tu digas acredito),
contar àquele pobre rapazito
que me deu horas tão felizes,

em bora não o saibas, que morri...
Mesmo ele, a quem eu tanto julguei amar,
nada se importará... Depois vai dar

a notícia a essa estranha Cecily
que acreditava que eu seria grande...
Raios partam a vida e quem lá ande!

(1922)

Estrutura

Poema: título, autor, corpo, data

Corpo: quadra, quadra, terno,
terno.

Quadra: verso, verso, verso,
verso

Terno: verso, verso, verso

Verso: (texto | nome)+

Nome: texto

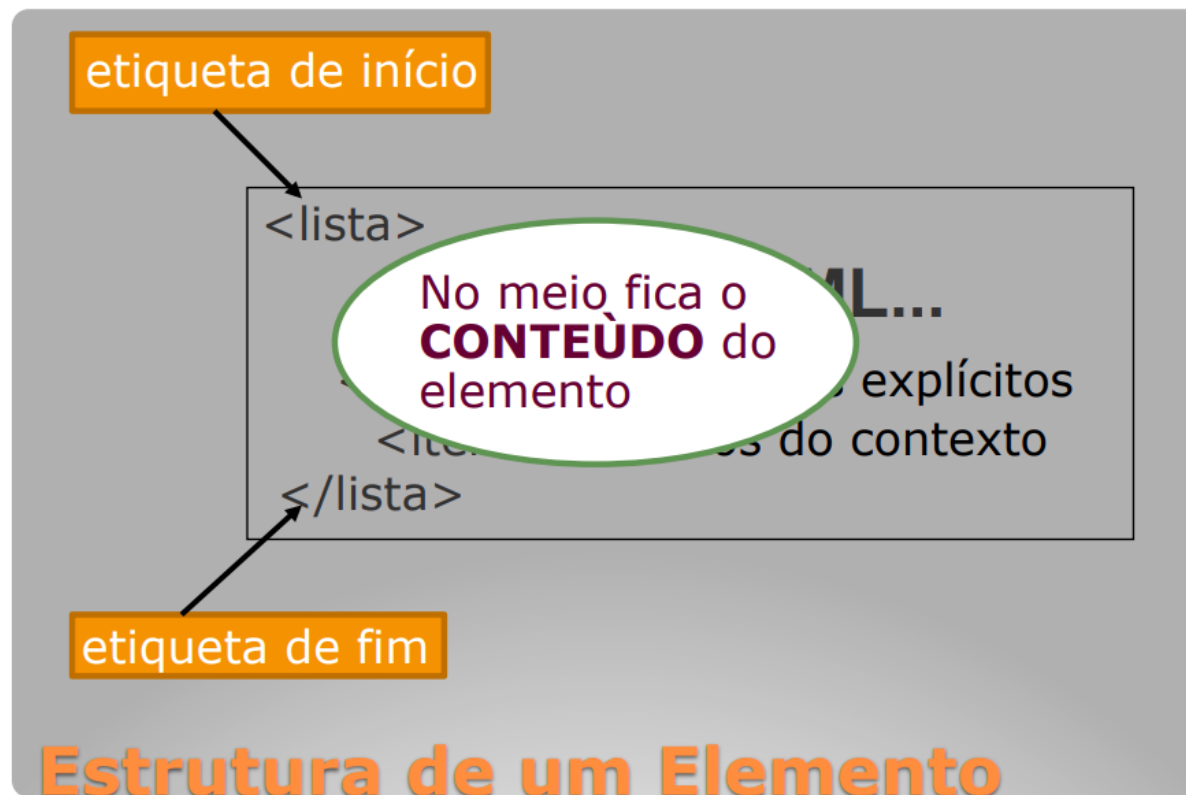
XML: ESTRUTURA

```
<lista>  
  <cabec>Os delimitadores podem ser: </cabec>  
  <item> explícitos  
  <item> inferidos do contexto  
</lista>
```

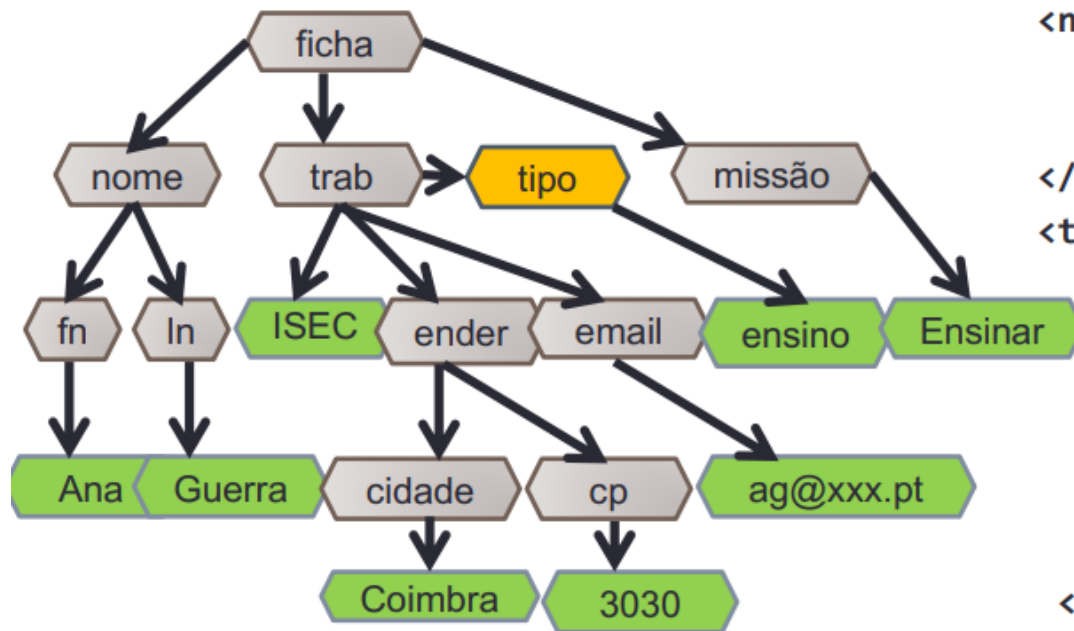
- O elemento do tipo LISTA é formado por um elemento CABEC, seguido por dois elementos do tipo ITEM.

Estrutura de um Documento

XML: ESTRUTURA



XML: EXEMPLO



Elementos

Atributos

Dados



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

```
<ficha>
```

```
<nome>
```

```
<fn>Ana</fn>
```

```
<ln>Guerra</ln>
```

```
</nome>
```

```
<trab tipo="ensino">
```

```
ISEC
```

```
<ender>
```

```
<cidade>Coimbra</cidade>
```

```
<cp>3030</cp>
```

```
</ender>
```

```
<email>ag@xxxx.pt</email>
```

```
</trab>
```

```
<missão>Ensinar</missão>
```

```
</ficha>
```

XML: EXEMPLO DE UM FICHEIRO

```
<?xml version="1.0"?>
<catalogo>
  <livro tipo="bd" id="001">
    <titulo>Lucky Luke, Sob o Céu do Oeste</titulo>
    <autor>Morris</autor>
    <ano>2005</ano>
  </livro>
  <livro tipo="ficção" id="002">
    <titulo>O Senhor dos Anéis I</titulo>
    <autor>Tolkien</autor>
    <ano>1997</ano>
  </livro>
</catalogo>
```

Declaração XML

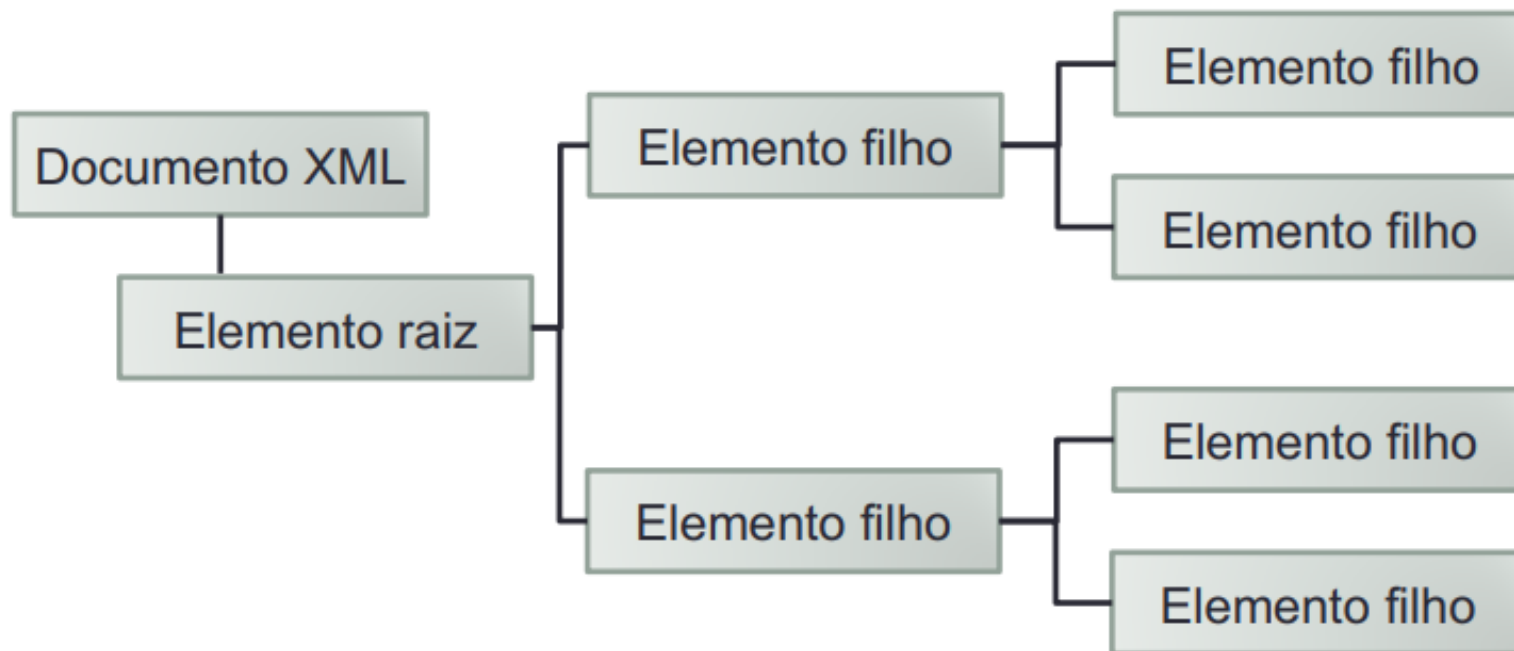
Elemento raiz

Atributos

Dados (informação)

XML: COMPONENTES DE UM FICHEIRO

- **Declaração XML** ou prólogo
- **Elementos**, tags ou etiquetas: têm estrutura hierárquica



- **Atributos**: caracterizam elementos

XML: COMPONENTES DE UM FICHEIRO

- A declaração XML: 1ª linha do ficheiro XML
 - Diz que se trata de XML:
`<?xml version="1.0" standalone = "yes" encoding = "UTF-8"?>`
 - Pode conter três atributos:
 - **Version:**
 - indica a versão de XML que está a ser utilizada
 - é obrigatório
 - **Standalone:**
 - indica se o documento é auto contido (toma o valor "yes") ou não ("no")
 - é opcional
 - **Encoding**
 - indica qual a codificação usada para os caracteres
 - o valor por omissão é "UTF-8"
 - para caracteres portugueses, usa-se "iso-8859-1"
 - é opcional

XML: COMPONENTES DE UM FICHEIRO

Tipos de Elementos (tags):

- **agregadores**, contêm outros elementos
- **folha**, contêm os dados
 - Ambos podem ter **atributos**
- **vazios**
- **Exemplo:**

```
<listareceitas edicao="3">
  <titulo>O Meu Livro de Receitas</titulo>
  <receita origem="Portugal">
    <titulo> Bolo </titulo>
    <ingrediente> 500g de farinha </ingrediente>
    <ingrediente> 200g de açúcar </ingrediente>
    <ingrediente> 300g de manteiga </ingrediente>
    <imagem src="http://...."/>
  </receita>
</listareceitas>
```

XML: COMPONENTES DE UM FICHEIRO

Atributos

- Um elemento pode ter **um ou mais** atributos
- Podem ser **opcionais** ou **obrigatórios**
- Visam qualificar o elemento a que estão associados
 - Isto é uma casa --- `<casa>`
 - Isto é uma casa verde --- `<casa cor="verde">`
- Definem-se usando um par constituído por um **nome** e um **valor**:
 - O **nome** e o valor devem estar separados pelo sinal =
 - O **valor** deverá estar colocado dentro de aspas simples ou duplas.
 - Exemplo: `<receita origem = "Portugal">`

↑
Nome

↑
Valor

XML: COMPONENTES DE UM FICHEIRO

- Elementos + Atributos

permitem várias formas de representar a mesma informação:

```
<data> 10 setembro 2018 </data>
```

```
<data><a>2018</a><m>09</m><d>10</d></data>
```

```
<data formato = 'ISO-8601'>2018-09-10</data>
```

XML: COMPONENTES DE UM FICHEIRO

- Comentários: igual ao HTML
 - Começam com `<!--` e terminam com `-->`

```
<?xml version="1.0" encoding="iso-8859-1"?>  
<!--Isto é um comentário no início-->  
<texto>Olá Mundo!!!</texto>
```

Restrições:

- Não podem aparecer antes da declaração (prólogo).
- Não podem aparecer dentro de um elemento.
- Não se pode utilizar a sequência de caracteres `--` dentro de um comentário.

XML: DOCUMENTO BEM FORMATADO

- Deve começar com a declaração XML
- Deve conter um ou mais elementos
 - Elemento raiz (é obrigatório e engloba todos os outros)
 - Elementos filhos
- Cada elemento deve ter elemento de abertura e de fecho com mesmo nome (excepto para elementos sem conteúdo)

`<nome>Anabela Simões</nome>`

- Os elementos devem estar aninhados de forma correcta

Correcto: `<nome><primeiro>Anabela</primeiro></nome>`

Incorrecto: `<nome><primeiro>Anabela</nome></primeiro>`

XML: DOCUMENTO BEM FORMATADO

- Elementos

- Elementos sem informação

`<emstock></emstock>` ✓

`<emstock/>` ✓

- Distinção entre maiúsculas e minúsculas:

Nome ≠ NOME ≠ nome

`<nome>Anabela</nome>` ✓

`<nome>Anabela</Nome>` ✗

XML: DOCUMENTO BEM FORMATADO

- Restrições nos nomes dos elementos
 - O primeiro caracter:
 - uma letra ou
 - um underscore ou
 - um sinal de dois pontos;
- Os caracteres seguintes:
 - letras
 - dígitos
 - underscores, hífens, pontos e dois pontos;
- O espaço em branco não pode aparecer no nome de um elemento.

XML: DOCUMENTO BEM FORMATADO

- Restrições nos nomes das elementos (exemplos)

- **nomes válidos:**

- `<Doc:princ>` `</Doc:princ>`
- `<documento>` `</documento>`
- `<_secreto>` `</_secreto>`
- `<aluno4>` `</aluno4>`

- **nomes inválidos:**

- `<1documento>` `</1documento>` -- começa por dígito
- `<aluno(4)>` `</aluno(4)>` -- tem parêntesis
- `<DB tab5>` `</DB tab5>` -- tem espaço em branco

XML: DOCUMENTO BEM FORMADO

- Restrições : caracteres especiais (exemplos)
- não se devem usar os caracteres '<' e '>'
- em lugar deles devem-se usar, respetivamente, as entidades do tipo caracter '<' e '>' ;
- Em alternativa, usar CDATA

XML: DOCUMENTO BEM FORMADO

- **O que está errado neste exemplo:**



XML: BOAS PRÁTICAS

Menor ramificação

```
<curso>
  <disciplina ano="1">Programação</disciplina>
  <docente teoricas="sim">Francisco Pereira</docente>
  <docente teoricas="nao">Teresa Raquel</docente>
  <disciplina ano="2">Integração de Dados</disciplina>
  <docente teoricas="sim">Anabela Simões</docente>
</curso>
```

Dificuldades:

- **quais os docentes de uma disciplina?**
- **dificuldades na validação da ordem dos elementos**

XML: BOAS PRÁTICAS

Elementos versus atributos

```
<agenda>
  <entrada id="001" tipo="pessoa">
    <nome>Anabela Simões</nome>
    <email>abs@isec.pt</email>
    <telefone>239 790000</telefone>
  </entrada>
  ...
</agenda>
```

Informação nos elementos

OU

```
<agenda>
  <entrada id="001" tipo="pessoa" nome="Anabela Simões" email="abs@isec.pt"
    telefone="239 790000"/>
  ...
</agenda>
```

Informação nos atributos

XML: BOAS PRÁTICAS

Elementos versus atributos: regras

- Elementos podem ter **hierarquia**, os atributos não:

```
<nomes>
  <primeiro>Anabela</primeiro>
  <apelido>Simões</apelido>
</nomes>
```

- Elementos podem armazenar **valores múltiplos**, os atributos não:

```
<nomes>
  <primeiro>Anabela</primeiro>
  <apelido>Simões</apelido>
  <primeiro>João</primeiro>
  <apelido>Silva</apelido>
</nomes>
```

- Identificadores são normalmente atributos:

```
<nomes id="001"> ... </nomes>
```

XML: BOAS PRÁTICAS

- Um ficheiro XML bem formado permite:
 - Armazenar informação/dados
 - Efetuar pesquisas sobre os dados
 - Transformar, ordenar, filtrar os dados criando novos ficheiros de output:
 - Texto
 - HTML
 - XML

XML: EXEMPLO 1

- Considere o seguinte resultado de uma página no seu browser:

LIVROS			
Titulo	Autor	Gênero	Preço
JAVA 6.5	Maria Gomes	Técnico	32.30
XML e DTD	Carlos Cunha	Técnico	25.00
MoGel	Filipe Sá	Técnico	18.00
O Futuro da Terra	Gigi de Lourdes	Ficção	12.00

- Especifique o documento XML que o representa

XML: EXEMPLO 1- SOLUÇÃO

- `<?xml version="1.0" encoding="UTF-8" standalone = "yes" ?>`
- `<livros>`
 - `<livro>`
 - `<titulo>Java 6.5</titulo>`
 - `<autor>Maria Gomes</autor>`
 - `<genero>Técnico</genero>`
 - `<preco>32.30</preco>`
 - `</livro>`
 - ...
 - `<livro>`
 - `<titulo>O futuro da terra</titulo>`
 - `<autor>Gigi de Lourdes</autor>`
 - `<genero>Ficção</genero>`
 - `<preco>12.00</preco>`
 - `</livro>`
- `</livros>`

XML: EXERCÍCIO 2

- Considere o seguinte resultado de uma página no seu browser:

Filmes		
Título	Quantidade	Tipo
O Judas anda à solta	2	VHS
O regresso dos que não foram	1	DVD
E tudo a chuva molhou	1	DVD
Não é de gatos que o cão foge	3	VHS

- Especifique o documento XML que o representa. Faça uso de atributos

XML: EXERCÍCIO 2 - SOLUÇÃO

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

<filmes>
  <filme titulo=" O Judas anda à solta" quantidade="2"
    tipo="VHS"/>
  <filme titulo=" O regresso dos que não foram"
    quantidade="2" tipo="VHS"/>
  <filme titulo=" E tudo a chuva molhou"
    quantidade="2" tipo="VHS"/>
  <filme titulo=" Não é de gatos que o cão foge"
    quantidade="2" tipo="VHS"/>
</filmes>
```