

**ESCOLA  
SUPERIOR  
DE TECNOLOGIA  
E GESTÃO**

**P.PORTO**

**Bruno Oliveira**  
2018/2019

## **Introdução a XML (eXtensible Markup Language)**

Processamento Estruturado de Informação

# Resumo

- Estrutura em árvore do XML
- Sintaxe do XML
- Elementos XML
- Atributos XML

# Introdução

- O XML (eXtensible Markup Language) é uma linguagem de marcação utilizada para armazenar e transportar dados;
- O XML foi desenvolvido para possibilitar a leitura não só para máquinas mas também para humanos;
- Atualmente, o XML tem um papel de revelo não só no suporte a vários sistemas de informação, mas também como meio de distribuição de dados através da internet;
- O XML é uma recomendação da W3C: <https://www.w3.org/XML/>

# XML

- Exemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<email>
  <para>joao@sapo.pt</para>
  <de>bruno@gmail.com</de>
  <cabecalho>Aviso</cabecalho>
  <corpo>Não te esqueças do que combinamos!</corpo>
</email>
```

# XML é extensível

- A maioria das aplicações continuarão a funcionar normalmente mesmo quando são realizadas **alterações** na estrutura;
- Exemplo:

```
<email>  
  <para>joao@sapo.pt</para>  
  <de>bruno@gmail.com</de>  
  <cabecalho>Aviso</cabecalho>  
  <corpo>Não te esqueças do que  
combinamos!</corpo>  
</email>
```

Adicionamos a <data> e  
removemos o <cabecalho>

```
<email>  
  <para>joao@sapo.pt</para>  
  <data>21-09-2017</data>  
  <de>bruno@gmail.com</de>  
  <corpo>Não te esqueças do que  
combinamos!</corpo>  
</email>
```

- A forma de organização do XML, permite que estas alterações sejam realizadas **sem comprometer** o funcionamento das aplicações.

## Como podemos utilizar XML?

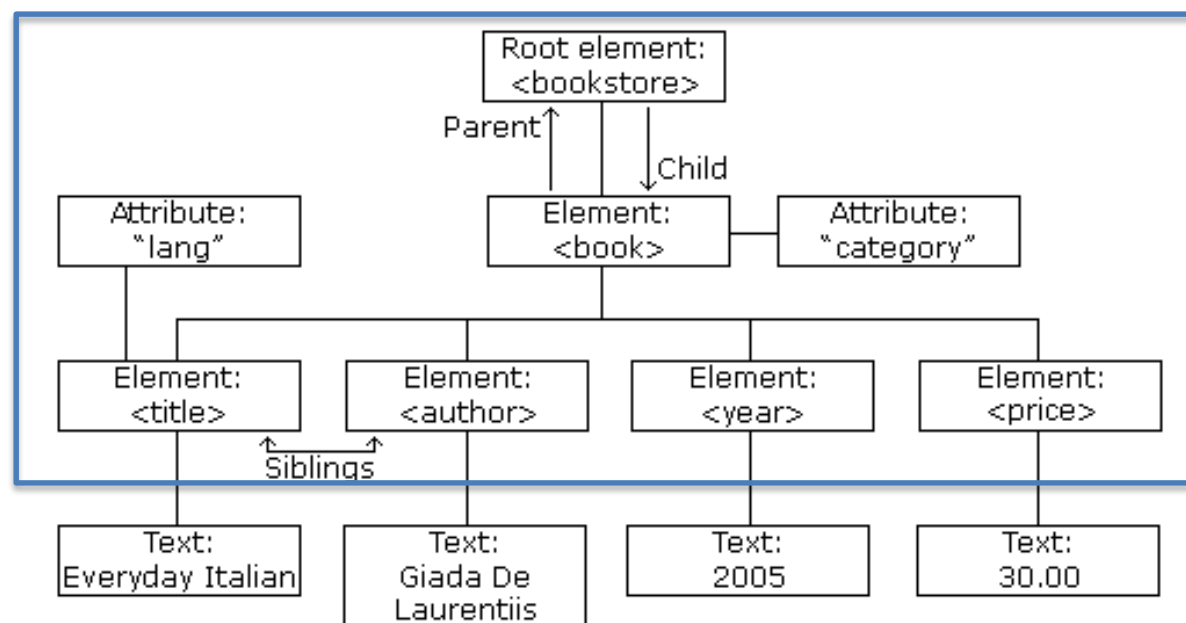
- O XML é frequentemente utilizado para **separar** a camada de **dados** da camada que trata da sua **apresentação**:
  - O XML **não** inclui informações sobre a forma como os dados são **apresentados**;
  - O mesmo XML pode ser utilizado em **várias** camadas de apresentação em vários cenários **diferentes**;
- Por exemplo, no desenvolvimento **Web**, o XML é frequentemente utilizado para **armazenar** e **transportar** dados e o formato HTML para providenciar a sua apresentação;

# Exemplo de documento

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>

  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
</bookstore>
```

Árvore de representação XML:



Estrutura

# Exemplo

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">(...)</book>
</bookstore>
```

Conversão  
através de um  
parser

HTML

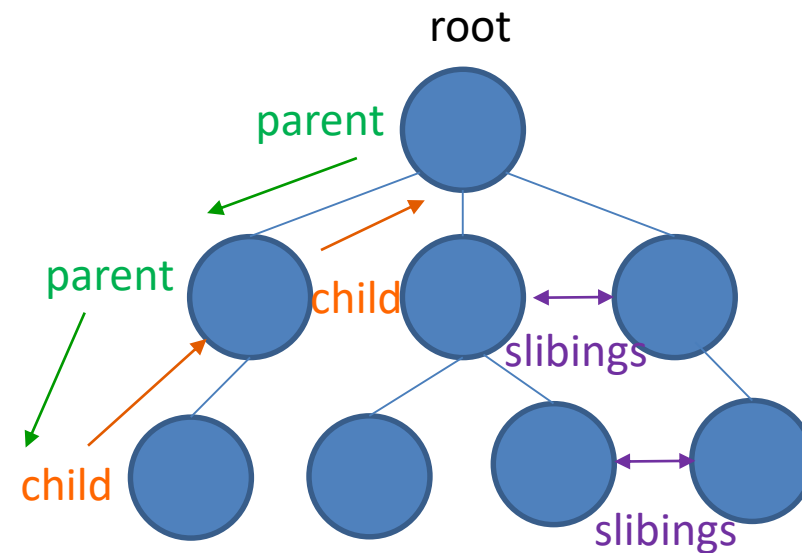
```
<table id="demo" class="w3-table-all">
  <tbody>
    <tr>
      <th>Title</th>
      <th>Author</th>
    </tr>
    <tr>
      <td>Everyday Italian</td>
      <td>Giada De Laurentiis</td>
    </tr>
    <tr>
      <td>Harry Potter</td>
      <td>J K. Rowling</td>
    </tr>
    <tr>
      <td>XQuery Kick Start</td>
      <td>James McGovern</td>
    </tr>
    <tr>
      <td>Learning XML</td>
      <td>Erik T. Ray</td>
    </tr>
  </tbody>
</table>
```

Title	Author
Everyday Italian	Giada De Laurentiis
Harry Potter	J K. Rowling
XQuery Kick Start	James McGovern
Learning XML	Erik T. Ray



## Estrutura em árvore do XML

- Os documentos XML formam uma estrutura em **árvore** que se inicia na **raíz** (*root*) e através de **ramos** permite o acesso às “**folhas**” (elementos);
- O XML **DOM** (*Document Object Model*) define a abordagem padrão para aceder e manipular documentos XML:




## Estrutura em árvore do XML

- Todos os elementos podem conter **subelementos** (filhos)
- Os termos: *parent* (pai), *child* (filho) e *sibling* (irmão) são utilizados para descrever *relacionamentos* entre elementos;
- Todos os elementos podem conter **texto** como conteúdo (“Senhor dos Anéis”) e também **atributos** (categoria=“fantasia”).
- Exemplos:

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

```
<desportos>
  <desporto categoria="futebol">
    <jogador>Ronaldo</jogador>
  </desporto>
</desportos>
```



Todos os elementos têm de  
possuir uma tag de fecho

## Estrutura em árvore do XML

- O XML utiliza uma sintaxe que **autodescreve** o seu conteúdo;
- A primeira linha de um documento XML descreve a **versão** e o **tipo** de codificação utilizado: `<?xml version="1.0" encoding="UTF-8"?>`
- A segunda linha identifica o elemento **root** do documento:

`<lojaLivros>`

- As restantes linhas descrevem o **conteúdo** do documento:


`<livro categoria="fantasia">  
(...)`

## Estrutura em árvore do XML

- O elemento <livro> tem 4 elementos filho: <título>, <autor>, <ano>, <preco>.

```
<titulo idioma="pt">O Senhor dos Anéis</titulo>  
<autor>J.R.R. Tolkien</autor>  
<ano>2010</ano>  
<preco>30.00</preco>
```

Todos os elementos  
têm de possuir uma tag  
de fecho



# O Prolog XML

- A primeira linha XML é chamada de **prolog**:  
`<?xml version="1.0" encoding="UTF-8"?>`
- O atributo **version** permite identificar a versão do XML utilizada, de forma a que os interpretadores de XML possam interpretar a estrutura do documento;
- A versão 1.1 alterou a forma como os nomes de elementos e atributos poderiam ser definidos para abranger os caracteres definidos pela codificação **Unicode 2.0** (O UTF-8 é uma forma de codificar o formato Unicode);
- Mais informação: <https://www.w3.org/TR/xml11/#sec-xml11>

## O Prolog XML

- Os documentos XML podem conter caracteres **especiais** como øæå ou êèé.
- Para evitar erros, o tipo de **encoding** deve ser especificado;
- **UTF-8** é o encoding de caracteres por defeito para documentos XML.
- Mais informação:
  - <https://www.w3schools.com/charsets/default.asp>

# Regras para a criação de documentos XML válidos

- Todos os elementos XML têm de possuir uma **tag** de **fecho**;
- As tags XML são **sensíveis** a maiúsculas e minúsculas;
- Os elementos devem estar **agrupados** corretamente;
- Os valores dos atributos XML devem ser colocados entre

aspas:

```
<nota data="12/11/2007">  
  <de>Tove</de>  
  <para>Jani</para>  
</nota>
```

# Regras para a criação de documentos XML válidos

- Caracteres especiais (Entity references):

```
<mensagem>salary &lt; 1000</mensagem>
```

- Exemplos:

&lt;	<	less than
&gt;	>	greater than
&amp;	&	ampersand
&apos;	'	apostrophe
&quot;	"	quotation mark



# Regras para a criação de documentos XML válidos

- Sintaxe base de um comentário XML:  
`<!-- This is a comment -->`
- O espaço em XML é preservado;
- Aplicações windows armazenam a sinalização de nova linha como: *carriage return* e *line feed* (CR+LF) - &#10.
- Unix utiliza LF - &#xA;;
- Versões antigas do Mac até 9.0 utilizam &#xD;
- O XML armazena a nova linha como LF - &#xA;

# Elementos XML

- Exemplo de elemento: `<preco>29.99</preco>`
- Um elemento pode conter:
  - Texto
  - Atributos
  - Outros elementos
  - Atributos e elementos/texto

## Elementos XML

- `<title>`, `<author>`, `<year>`, e `<price>` possuem **conteúdo** de **texto** (como 29.99).
- `<bookstore>` e `<book>` possuem **conteúdo** que são também **elementos**;
- `<book>` tem um **atributo** (`category="children"`).

```
<bookstore>
  <book category="children">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

## Regras de nomes

- Elementos sem conteúdo são chamados de **vazios**:

`<element></element>` Ou `<element />`

- Os nomes dos elementos devem começar com uma **letra** ou **underscore**;
- Os nomes dos elementos **não** podem começar com as letras XML, ou *XML*, etc;
- Os nomes dos elementos **podem** conter letras, dígitos, hifens, underscores, e pontos;
- Os nomes dos elementos **não** podem conter **espaços**;

## Boas práticas para nomes

- Usar nomes **descriptivos**, como `<essoa>`, `<primeironome>`, `<ultimonome>`.
- Usar nomes **pequenos**, como: `<titulo_livro>` e não: `<o_titulo_do_livro>`.
- **Evitar** “-”, uma vez que pode ser confundido com o operador de subtração;
- **Evitar** “.”
  - Nomes como: “nome.primeiro”, pode levar que se interprete que primeiro é propriedade de nome;
- **Evitar** “:”. Os dois pontos estão reservados para namespaces;
- Ter cuidado com caracteres com **acentuação**, uma vez que podem trazer problemas para a componente aplicacional;

# Atributos XML

- **Atributos** descrevem **dados relacionados** com os seus elementos;
- Todos os valores de atributos devem ser colocados entre **aspas** ou **plicas**:

```
<essoa genero="feminino">
```

```
<essoa genero='feminino'>
```

```
<essoa nome='George "Shotgun" Ziegler'>
```

# Elementos vs. Atributos

- Exemplos:

```
<peessoa genero="feminino">  
  <primeiroNome>Anna</primeiroNome>  
  <ultimoNome>Smith</ultimoNome>  
</peessoa>
```

VS.

```
<peessoa>  
  <genero>feminino</genero>  
  <primeiroNome>Anna</primeiroNome>  
  <ultimoNome>Smith</ultimoNome>  
</peessoa>
```

## Evitar o uso de atributos?

- Devemos considerar:
  - Os atributos **não podem conter múltiplos valores** (os elementos podem)
  - Os atributos **não podem ter estrutura em árvore** (os elementos podem)
  - Os atributos **não podem ser facilmente estendidos** (mudanças futuras)
- Os **meta dados** devem ser armazenadas como atributos e os dados devem ser armazenados em elementos:

```
<peessoa id="501">
```



## Exemplos de aplicação de XML

- Utilizado no desenvolvimento web para **separar** os dados da sua componente de apresentação;
- **Transportar** informação entre diversos dialetos para suportar negócios específicos: ações, medicina, matemática e serviços noticiosos;
- Criação de **vocabulários** entre organizações (por exemplo: SAFT<sup>1</sup>);
- Para suportar a **comunicação** com serviços web: WSDL, SOAP, RDF, RSS;
- Serviço Web dos CTT: [http://www.ctt.pt/pdcp/xml\\_pdcp](http://www.ctt.pt/pdcp/xml_pdcp)
  - Exemplo:  
[http://www.ctt.pt/pdcp/xml\\_pdcp?indistrito=viana+castelo&inlocal=lisboa](http://www.ctt.pt/pdcp/xml_pdcp?indistrito=viana+castelo&inlocal=lisboa)

<sup>1</sup>[http://info.portaldasfinancas.gov.pt/pt/apoio\\_contribuinte/news\\_saf-t\\_pt\\_vers\\_inglesa.htm](http://info.portaldasfinancas.gov.pt/pt/apoio_contribuinte/news_saf-t_pt_vers_inglesa.htm)

## Editores/ferramentas para explorar XML

- SublimeText: <https://www.sublimetext.com> + Exalt: <https://www.sublimetext.com>
- Liquid Studio: <https://www.liquid-technologies.com/liquid-community-edition>
- BaseX - <http://basex.org/products/gui/>

# Exercício – Utilizando um editor de texto

- Considere o seguinte documento XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<email>
  <to>Tove@gmail.com</to>
  <from>Jani@gmail.com</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</email>
```

- Crie uma representação em árvore do documento XML apresentado;
- Altere a estrutura (utilizando inicialmente a representação em árvore) de forma a suportar o armazenamento de vários emails;
- Altere a estrutura (utilizando inicialmente a representação em árvore) de forma a incluir a data do email e ainda o tipo de email: “Pessoal”, “Trabalho”;

**ESCOLA  
SUPERIOR  
DE TECNOLOGIA  
E GESTÃO**

**P.PORTO**

**Bruno Oliveira**  
2018/2019

## **Introdução a XML (eXtensible Markup Language)**

Processamento Estruturado de Informação