

Introdução a XML (eXtensible Markup Language)

Agenda

- Introdução a XML:
 - DOM (*Document Object Model*);
 - Sintaxe;
 - Elementos;
 - Atributos;
- Introdução a XML Schema.

Introdução

Informação Estruturada:

Número	Nome	País	Profissão
1	Rui	Portugal	Gestor
2	Eva	Espanha	Médica
3	Camille	França	Bióloga

Informação Semi-Estruturada:

```
<workshop>
  <orador numero="1">
    <nome>Rui</nome>
    <pais>Portugal</pais>
    <profissao>Gestor</profissao>
  </orador>
  <orador numero="2">
    <nome>Eva</nome>
    <pais>Espanha</pais>
    <profissao>Médica</profissao>
  </orador>
  <orador numero="3">
    <nome>Camille</nome>
    <pais>França</pais>
    <profissao>Bióloga</profissao>
  </orador>
</workshop>
```

Informação Não Estruturada:

O *workshop* de línguas conta com a participação de vários oradores de diferentes partes do mundo. Os oradores vão falar acerca dos costumes do seu país.

O orador número 1 é o Rui. O Rui é português e gestor. A Eva, que vem de Espanha, é a oradora número 2 e é médica. A terceira oradora, a Camille, é bióloga e vive em França.

Introdução

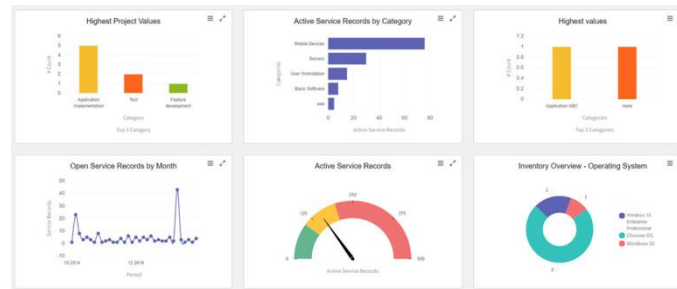
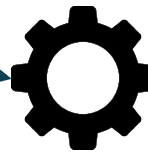
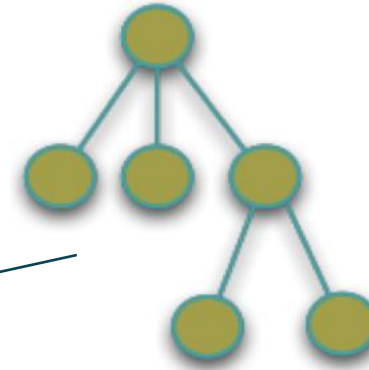
- O XML (eXtensible Markup Language) é uma linguagem de **marcação** utilizada para **armazenar** e **transportar** dados utilizando **documentos**;
- O XML tornou-se um dos **standards** mais relevantes para suportar a **representação**, **transferência** e **integração** de dados em sistemas **heterogêneos**;
- 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>.

Problema?

Informação não estruturada

- 1.Bata as claras
- 2.Misture as gemas, a margarina e o açúcar atéAcrescente o leite e a farinha de trigo aos poucos, sem parar de bater.
- 3.Por último, adicione as claras em neve e o ferment
- 4.(...)

Estrutura



Exemplo de documento XML

```
<?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?

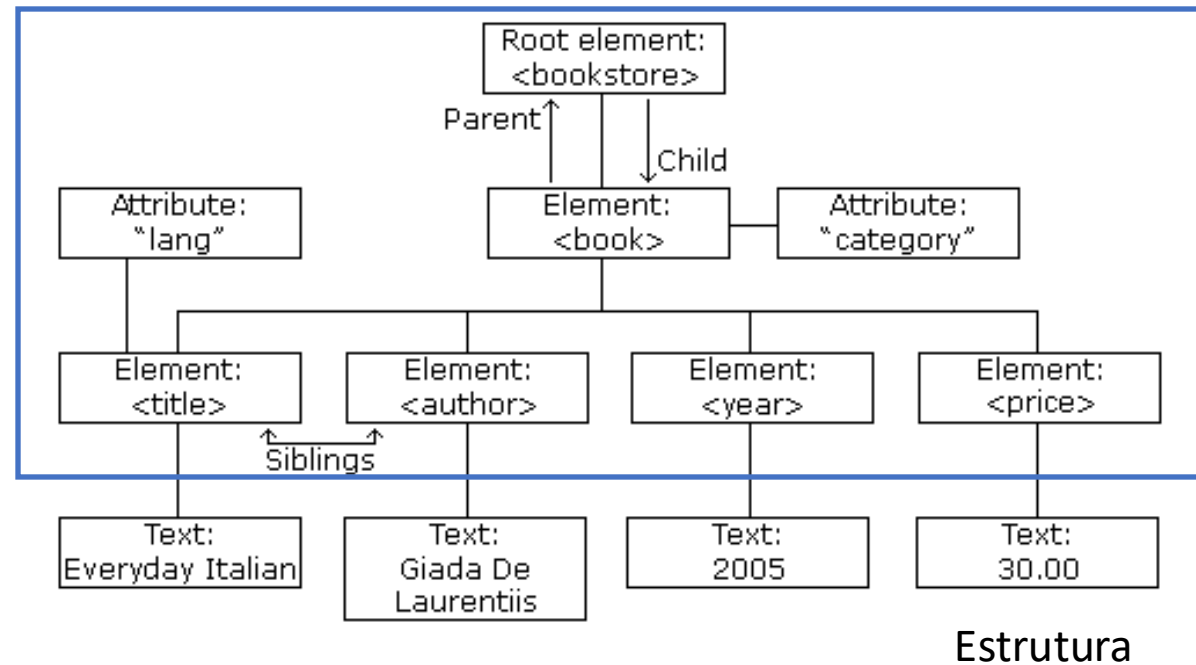
- O XML é frequentemente utilizado para **separar** a camada de **dados** da camada que trata da sua **apresentação**:
- O XML **não** inclui informações sobre a forma como os dados são apresentados;
- O mesmo XML pode ser utilizado em várias **camadas** de apresentação em vários cenários diferentes;

Representação hierárquica de dados

```
<?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>
```

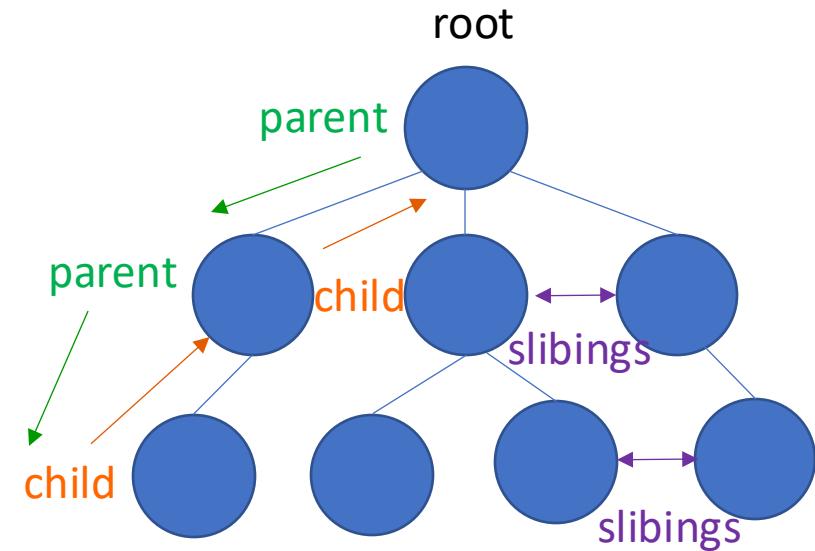
Árvore de representação XML:



Fonte: https://www.w3schools.com/xml/xml_tree.asp

Estrutura em árvore do XML

- Os documentos XML formam uma estrutura em **árvore** que se inicia na raiz (**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”).

- O XML utiliza uma sintaxe que autodescreve o seu conteúdo;
- A primeira linha de um documento XML (também conhecido como prolog) 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">  
(...)
```

- 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>
```

Sintaxe base de um comentário XML: `<!-- This is a comment -->`

- Caracteres especiais (Entity references):

`<mensagem>salary < 1000</mensagem>`

- Exemplos:

<	<	less than
>	>	greater than
&	&	ampersand
'	'	apostrophe
"	"	quotation mark

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").

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

- 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**:

Boas práticas para nomes

- Usar nomes **descriptivos**, como <peessoa>, <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** descrevem dados relacionados com os seus elementos;
- Todos os valores de atributos devem ser colocados entre **aspas** ou **plicas**:

```
<peessoa genero="feminino">
```

```
<peessoa genero='feminino'>
```

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

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">
```

Documentos bem formados

- “Well-formed XML” document” é um documento xml que respeita as regras definidas anteriormente;
- Isto significa que o documento XML tem de estar conforme com um conjunto de regras sintáticas;
- Por exemplo:
 - Os elementos devem estar devidamente aninhados;
 - Os elementos não vazios têm de possuir uma marcação de início e de fim;
 - Atributos devem estar incluídos em aspas;
 - (...)

Quando bem formado não é suficiente

- Um documento XML bem formado é um documento que cumpre as **regras de sintaxe** do XML (deve iniciar com a declaração XML, todas as tags devem ser corretamente encerradas, etc).
- Mesmo estando bem **formados**, os documentos podem conter **erros** considerando a âmbito em que estão inseridos;

Introdução a XML (eXtensible Markup Language)