

BANCO DE DADOS II

XML e JSON

TRANSMISSÃO DE DADOS

Muitas aplicações trocam e acessam informações de diversas fontes de dados e várias trocam informações entre si.

Qual padrão utilizado para essa comunicação? Principalmente entre sistemas independentes e por vezes de fornecedores independentes.

Exemplo:

"A nota fiscal eletrônica tem validade em todos os estados da Federação e já é uma realidade na legislação brasileira desde outubro de 2005, com a aprovação do Ajuste SINIEF 07/05 que instituiu nacionalmente a Nota Fiscal Eletrônica e o Documento Auxiliar da Nota Fiscal Eletrônica - DANFE."

Protocolo ICMS 10/07 e alterações

De 18/05/2007, dispõe sobre a obrigatoriedade de emissão de NF-e.

Protocolo ICMS 42/09 e alterações

De 03/07/2009, estabelece a obrigatoriedade da utilização da Nota Fiscal Eletrônica (NF-e) em substituição à Nota Fiscal, modelo 1 ou 1A, pelo critério de CNAE e operações com os destinatários que especifica.

<http://www.nfe.fazenda.gov.br/portal/perguntasFrequentes.aspx?tipoConteudo=E4+tmY+ODf4=>

DADOS ESTRUTURADOS, SEMIESTRUTURADOS E NÃO ESTRUTURADOS

Dados Estruturados:

- Bancos de dados relacionais armazenam dados estruturados, representando-os de modo estrito.
- Cada tupla de uma tabela segue o mesmo formato das demais.
- Dados estruturados seguem cuidadosamente um schema.

Algumas vezes dados são coletados e armazenados de uma maneira não tão cuidadosa. As vezes coletados e armazenados para depois serem analisados.

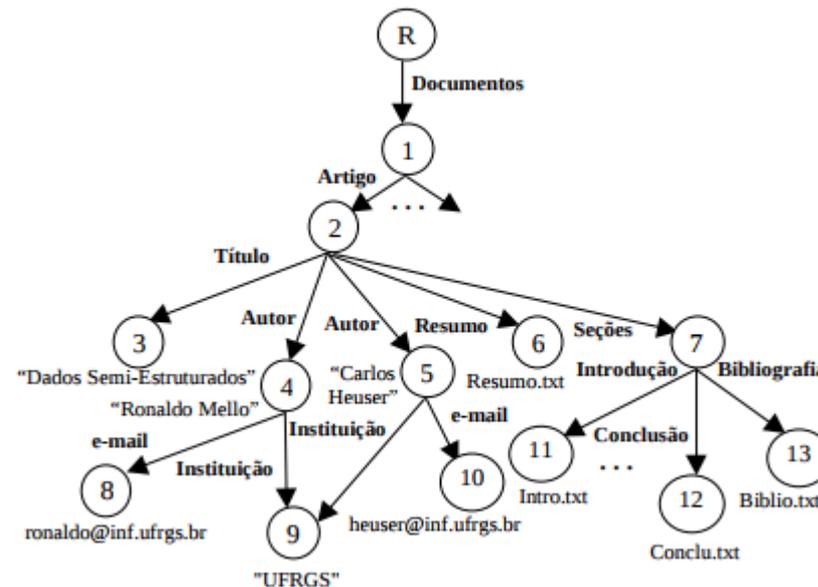
Dados Semiestruturados:

- Podem ter estrutura, mas nem toda a informação seguirá uma estrutura idêntica.
- Alguns atributos podem ser compartilhados entre todas as entidades.
- Alguns outros atributos podem existir apenas em alguns casos.
- Atributos novos podem ser adicionados nos dados que serão armazenados.

DADOS ESTRUTURADOS, SEMIESTRUTURADOS E NÃO ESTRUTURADOS

Dados Semiestruturados:

- A informação do esquema e os valores dos dados estão misturadas (**dados autodescritivos**).
- Comumente são representados em grafos ou árvores
 - Arestas direcionadas rotuladas com os nomes de esquema.



Mello, Ronaldo dos Santos, et al. Dados Semi-Estruturados.
<<https://www.ime.usp.br/~jef/semi-estruturado.pdf>>

DADOS ESTRUTURADOS, SEMIESTRUTURADOS E NÃO ESTRUTURADOS

Dados não estruturados:

- Dados dos quais não existe indicação sobre o tipo dos dados, ou alguma muito limitada.
- Exemplo:
 - Textos
 - Algumas páginas HTML

Qual as diferenças, vantagens e desvantagens desses tipos de dados?

XML - EXTENSIBLE MARKUP LANGUAGE

A linguagem XML (Extensible Markup Language) surgiu como um padrão para a estruturação e troca de dados entre sistemas.

XML pode fornecer informações sobre a estrutura e o significado dos dados, tendo como propósito ser lida e compreendida tanto por humanos quanto por máquinas.

EXEMPLO:

- Um estabelecimento comercial deve enviar para a SEFAZ um XML com informações da venda para a autorização e emissão da NF-e.
- A SEFAZ responde com um XML contendo autorização.
- A partir dos dados desses XML a Nota Fiscal é gerada e enviada aos clientes.

XML - EXTENSIBLE MARKUP LANGUAGE

O objeto básico em XML é o **documento XML** que contém uma estrutura hierárquica baseado em dois conceitos:

- **Elementos**
- **Atributos**
 - Fornecem informações adicionais a um elemento.

```
note.xml
```

```
<note>  
  <to>Tove</to>  
  <from>Jani</from>  
  <heading>Reminder</heading>  
  <body>Don't forget me this weekend!</body>  
</note>
```

DIFERENÇAS ENTRE XML E HTML

- XML foi projetada para transportar dados;
- HTML foi projetada para mostrar dados;
- As tags XML não são pré-definidas
 - As tags do exemplo anterior não são um padrão XML, elas foram definidas pelo autor do documento.
 - HTML funciona com tags pré-definidas, como <p>, <h1>,...
 - Em XML o autor deve definir as tags e a estrutura do documento.
 - A maioria das aplicações XML devem continuar funcionando mesmo que um dado seja adicionado ou removido dos novos documentos.
- XML simplifica tarefas
 - Simplifica o compartilhamento de dados;
 - Simplifica o transporte de dados;
 - Simplifica mudanças na plataforma;
 - Simplifica a disponibilidade dos dados;

INTERPRETE ESSES 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>
```

```
    <book category="web">
      <title lang="en">XQuery Kick Start</title>
      <author>James McGovern</author>
      <author>Per Bothner</author>
      <author>Kurt Cagle</author>
      <author>James Linn</author>
      <author>Vaidyanathan Nagarajan</author>
      <year>2003</year>
      <price>49.99</price>
    </book>

    <book category="web" cover="paperback">
      <title lang="en">Learning XML</title>
      <author>Erik T. Ray</author>
      <year>2003</year>
      <price>39.95</price>
    </book>

  </bookstore>
```

DADOS DE TRANSAÇÃO

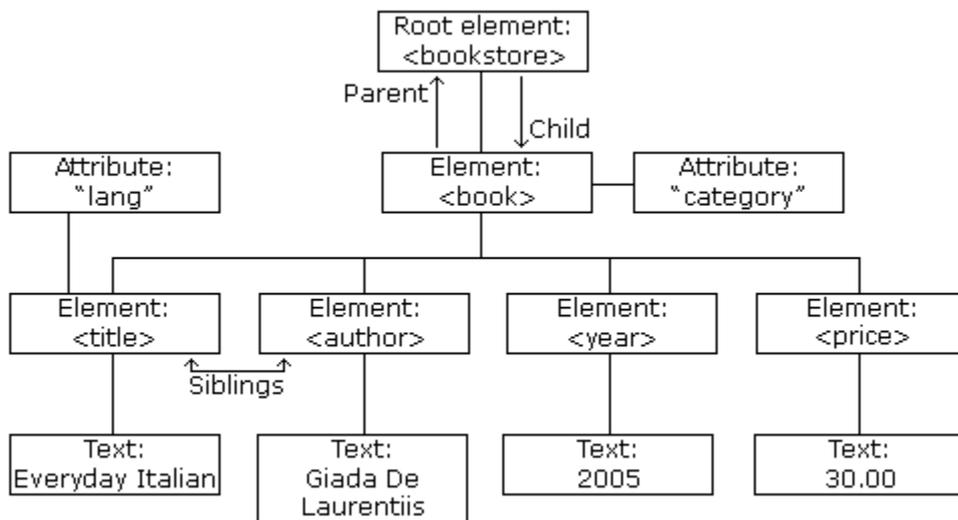
Existem milhares de formatos, em diferentes indústrias, para descrever transações corriqueiras:

- Transações financeiras
- Dados médicos
- Medições Científicas
- Notícias
- Serviços climáticos

```
<?xml version="1.0" encoding="UTF-8"?>
<nitf>
  <head>
    <title>Colombia Earthquake</title>
  </head>
  <body>
    <headline>
      <h1>143 Dead in Colombia Earthquake</h1>
    </headline>
    <byline>
      <bytag>By Jared Kotler, Associated Press Writer</bytag>
    </byline>
    <dateline>
      <location>Bogota, Colombia</location>
      <date>Monday January 25 1999 7:28 ET</date>
    </dateline>
  </body>
</nitf>
```

ÁRVORE XML

- Documentos XML possuem uma estrutura em árvore que inicia em um **elemento raiz** e ramifica para as folhas.
- Todo elemento pode ter elementos filhos (sub elementos).



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

XML - SINTÁXE

O primeiro elemento de um documento, introduzindo a árvore, deve ser a declaração do XML e codificação utilizada.

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

Na sequencia, o elemento raiz é declarado.

```
<bookstore>
```

Um elemento pode ser adicionado dentro da tag raiz.

```
<book category="cooking">
```

XML - SINTÁXE

1. Todo documento XML deve conter um elemento raiz que será pai de todos os outros elementos do documento.
2. O Prólogo XML `<?xml version="1.0" encoding="UTF-8"?>` é opcional, mas se ele existir deve ser a primeira declaração do documento.
 1. Use UTF-8 para evitar erros, reconhecendo todos os caracteres.
3. Todos os elementos devem ter uma tag de fechamento.
4. Tags XML são case sensitive.
5. Elementos XML devem ser aninhados corretamente
 1. Qual o correto???
 1. `<i>This text is bold and italic</i>`
 2. `<i>This text is bold and italic</i>`
6. Espaços em branco são preservados.

XML - ELEMENTOS

Elementos Vazios

- Um elemento sem conteúdo é dito vazio.
- Outros elementos
- Texto e atributos e elementos.

Regras para nomes de elementos:

- São case-sensitive
- Devem iniciar com letra ou _
- Não pode iniciar com a sequência xml
- Podem conter letras, -, _,
- Não pode conter espaços

Boas práticas:

- Usar nomes descritivos, como < pessoa >, <primeironome>, <ultimonome>
- Use nomes curtos e simples, como <livro_titulo>, em vez de <titulo_do_livro>.
- Evite "-" e ".", eles podem carregar algum significado não desejado.

XML - ATRIBUTOS

Elementos XML podem ter atributos (nome = "valor"), como em HTML.
Em XML o valor do atributo deve sempre estar entre aspas.

```
<note date="12/11/2007">  
  <to>Tove</to>  
  <from>Jani</from>  
</note>
```

- Atributos não podem conter múltiplos valores;
- Atributos não podem conter estrutura em árvore;
- Atributos não são expansíveis;

QUAL A DIFERENÇA?

```
<note>  
  <date>2008-01-10</date>  
  <to>Tove</to>  
  <from>Jani</from>  
</note>
```

```
<note date="2008-01-10">  
  <to>Tove</to>  
  <from>Jani</from>  
</note>
```

```
<note>  
  <date>  
    <year>2008</year>  
    <month>01</month>  
    <day>10</day>  
  </date>  
  <to>Tove</to>  
  <from>Jani</from>  
</note>
```

XML NAMESPACES

Quando pode haver conflito de nomes entre elementos de documentos usados em um mesmo contexto, pode-se utilizar um prefixo para diferenciá-los.

```
<table>
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```

```
<table>
  <name>African Coffee Table</name>
  <width>80</width>
  <length>120</length>
</table>
```

XML NAMESPACES

XML Namespaces - XMLNS

- Um namespace deve ser declarado para especificar o "vocabulário" que se está utilizando.
- O atributo xmlns deve ser declarado na tag inicial de um elemento.

```
<root>

<h:table xmlns:h="http://www.w3.org/TR/html4/">
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>

<f:table xmlns:f="https://www.w3schools.com/furniture">
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>

</root>
```

```
<root xmlns:h="http://www.w3.org/TR/html4/"
xmlns:f="https://www.w3schools.com/furniture">

  <h:table>
    ...
```

ATIVIDADE

Elabore um XML que represente os cursos e suas disciplinas. A disciplina pode conter pré-requisitos.

Elabore um XML para os cursos do DBAcademico, considerando apenas 3 disciplinas por curso.

XPATH

- XPath pode ser utilizado para navegar através dos elementos e atributos em um documento XML.
 - **XML Path Language**
 - Utiliza o padrão de caminho(path) para identificar e navegar pelos nós em um documento XML.
 - Utiliza expressões para selecionar nós ou conjuntos de nós.
 - Muitas linguagens permitem a utilização de XPATH para processar XML.

Expressão	Descrição
nodename	Seleciona todos os nós com o nome especificado.
/	Seleciona a partir do nó raiz.
//	Seleciona nós a partir do nó atual.
.	Seleciona o nó atual.
..	Seleciona o nó pai do atual.
@	Seleciona atributos

XPATH

Relacionamento entre nós:

- **Parent**
- **Children**
- **Siblings**
- **Ancestor**
- **Descendants**

```
<bookstore>

<book>
  <title>Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>

</bookstore>
```

XPATH

Leia o link abaixo sobre sintaxe XPATH e então utilize os exemplos.

- **Sintaxe XPATH:**
 - https://www.w3schools.com/xml/xpath_syntax.asp
- **Exemplos XPATH:**
 - https://www.w3schools.com/xml/xpath_examples.asp
- **Documentação XPATH:**
 - [https://msdn.microsoft.com/en-us/library/ms256115\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ms256115(v=vs.110).aspx)

ATIVIDADE

1. Especifique um XPATH que selecione os nomes de todas as disciplinas.
2. Especifique um XPATH que selecione todos os nomes dos cursos.
3. Especifique um XPATH que selecione o nome de todos os cursos com carga horária maior que 200.
4. Especifique um XPATH que selecione todos os elementos das disciplinas.
5. Utilize a função XMLIMPORT em uma planilha Gdocs para extrair todas as manchetes de um site de notícias de sua preferência.

DÚVIDAS?



REFERÊNCIAS BIBLIOGRÁFICAS

Mello. Ronaldo dos Santos, et.al. Dados Semi-Estruturados.
<<https://www.ime.usp.br/~jef/semi-estruturado.pdf>>

Introduction to XML. Disponível em:
<https://www.w3schools.com/xml/xml_what_is.asp>. Acesso em 16 Nov. 2017.

Introduction to XPATH. Disponível em:
<https://www.w3schools.com/xml/xpath_nodes.asp>. Acesso em 16 Nov. 2017.