



Linguagens de Marcação

Parte 2 – XML

Informática – CEFET

Prof. Dacy Câmara Lobosco

Prof. Rafael Escalfoni



XML

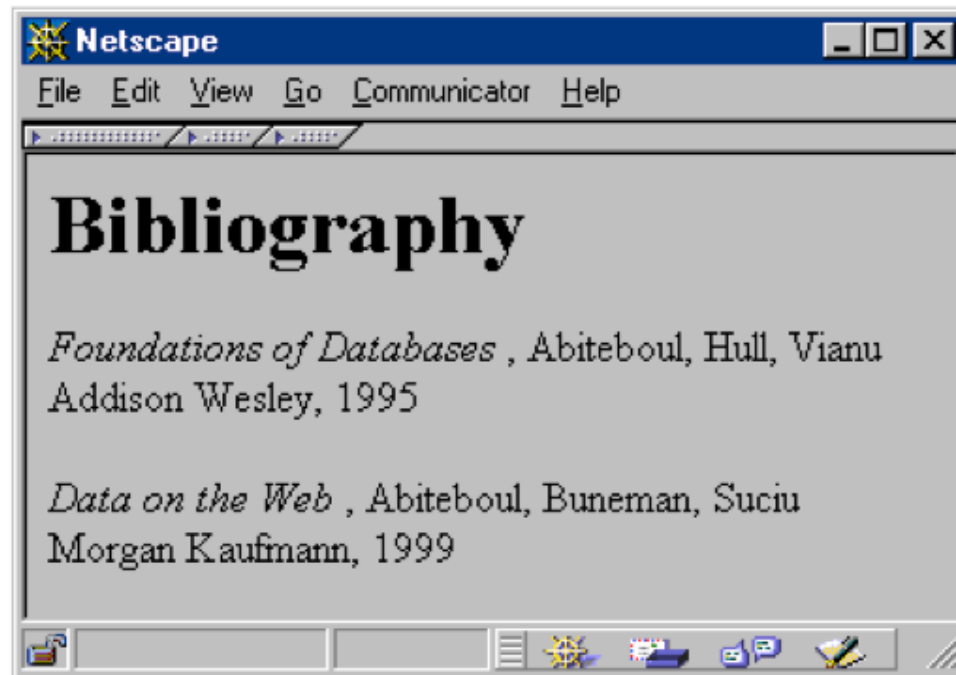
- XML é uma Recomendação W3C;
- XML apenas descreve os dados e o que eles significam
- XML foi desenvolvido para estruturar, armazenar e enviar os dados
- *"XML is a cross-platform, software and hardware independent tool for transmitting information" – W3Schools*
- Com XML a troca de dados entre sistemas incompatíveis é possível
- As *tags* do XML não são pré-definidas, você deve criar as suas próprias *tags*, obedecendo a um pequeno conjunto de regras de sintaxe.



XML vs HTML

- HTML – descreve a **formato** do documento
 - HTML tem um conjunto fixo de tags e não descreve conteúdo
- XML – descreve o **conteúdo** do documento
 - Usuário define suas próprias tags para criar uma estrutura
 - Um documento XML não tem nenhuma instrução para apresentação

De HTML para XML



HTML descreve a FORMA!



Fonte HTML

HTML: Conjunto pré-definido de elementos (tags) para especificação da estrutura e apresentação de um documento

```
<!DOCTYPE HTML>
```

```
<HTML>
```

```
<HEAD>
```

```
  <TITLE>A bibliography on Databases</TITLE>
```

```
  <META content="text/html; charset=utf-8" http-equiv=Content-Type>
```

```
</HEAD>
```

```
<BODY>
```

```
  <h1> Bibliography </h1>
```

```
  <p> <i> Foundations of Databases </i> Abiteboul, Hull, Vianu <br />  
  Addison Wesley, 1995
```

```
  </p>
```

```
  <p> <i> Data on the Web </i> Abiteoul, Buneman, Suciu <br />  
    Morgan Kaufmann, 1999
```

```
  </p>
```

```
</BODY>
```

```
</HTML>
```



Fonte XML

XML: Elementos (tags) definidos pelo usuário da linguagem e servindo para descrever o conteúdo e a estrutura.

```
<?xml version="1.0" encoding="ISO-8858-1"?>
<bibliography>
  <book>
    <title>Foundations of Databases</title>
    <author>Abiteboul</author>
    <author>Hull</author>
    <author>Vianu</author>
    <publisher>Addison Wesley</publisher>
    <year>1995</year>
  </book>
</bibliography>
```



Sintaxe XML

- Documento XML
 - Sequência de elementos que englobam texto ou outros elementos
 - **Elementos** podem conter atributos

Todos os documentos XML devem conter a “*declaração XML*”

- Define a versão do XML e a codificação de caracteres usada no documento
 - Parâmetros
 - **version** indica a versão da linguagem (1.0 ou 1.1) - obrigatório
 - **encoding** indica a codificação de caracteres utilizada no documento – opcional

```
<?xml version="1.0" encoding="ISO-8858-1"?>
```



XML

Todos os documentos XML devem conter um, e apenas um, elemento raiz

- `<raiz> ... </raiz>`
- Entre as *tags* do elemento raiz que estarão todas as outras *tags* do seu documento XML

Todos os elementos devem ter uma *tag* de fechamento

- Existem duas construções válidas:
 - `<teste>Teste 1</teste>` o elemento teste tem conteúdo
 - `<teste />` o elemento teste é vazio



XML

○ Elemento

- Delimitado por marcas (*tags*)
- Possuem uma marca inicial e uma marca final
- Tudo o que estiver delimitado por essas marcas faz parte do conteúdo do elemento
- Ex: `<empregado>João</empregado>`

○ Atributo

- Podem aparecer dentro da marca **inicial** de um elemento
- Ex: `<empregado cod="E01">João</empregado>`



XML

Todos os elementos devem ter uma *tag* de fechamento

○ Existem duas construções válidas:

- `<teste>Teste 1</teste>` o elemento teste tem conteúdo
- `<teste />` o elemento teste é vazio

XML é *case sensitive*

- `<teste>` é diferente de `<Teste>`

Os elementos XML devem estar corretamente aninhados

- Incorreto: `<i>texto da tag</i>`
- Correto: `<i>texto da tag</i>`

Tags podem ter Atributos, caso tenham devem estar entre aspas.

- Correto: `<aluno ID="2212"></aluno>`
- Incorreto: `<aluno=2212></aluno>`



XML

Os nomes dos elementos devem seguir as regras

- Nomes das tags podem possuir letras, números e outros caracteres.
- Nomes não podem começar com número ou caractere de pontuação.
- Nomes não podem começar com as letras XML e suas variações
- Nomes não podem conter espaços

Sintaxe dos comentários no documento:

<!--Comentário-->



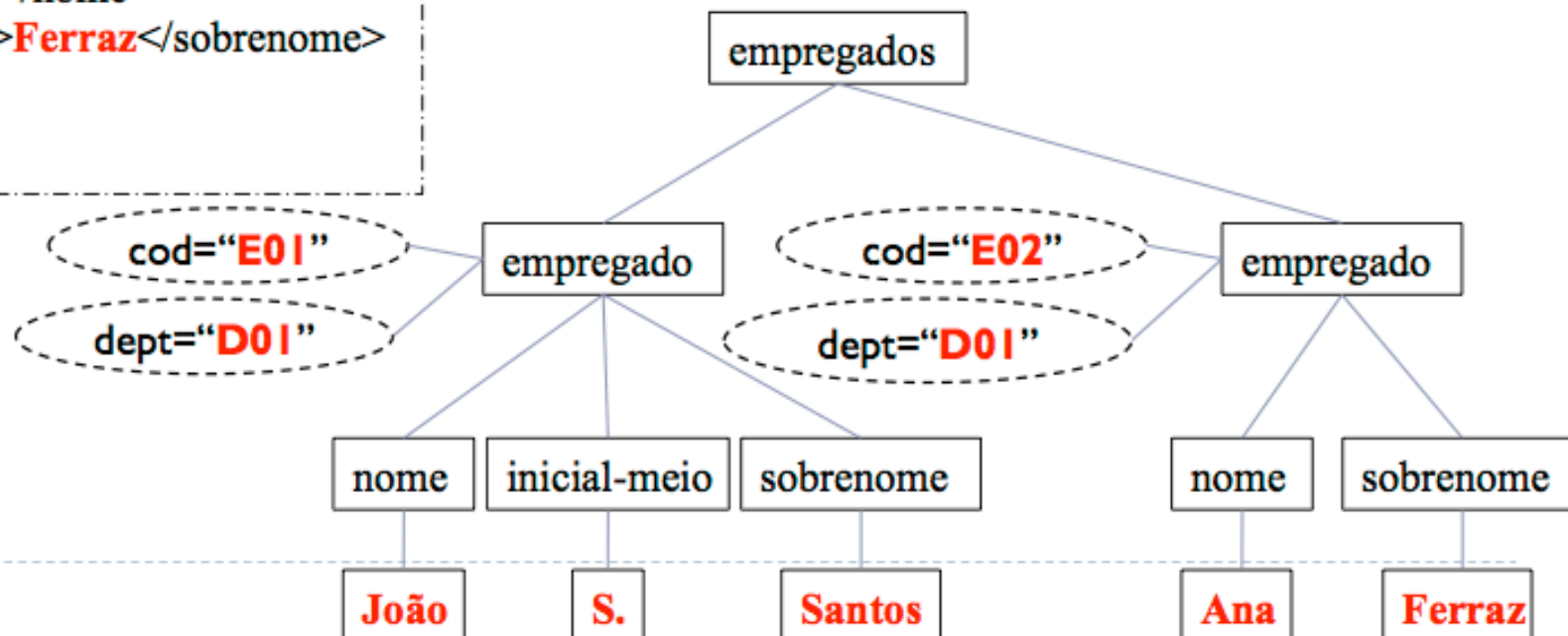
Exemplo de Documento XML

```
<? xml version="1.0" ?>
  <empregados>
    <empregado cod="E01" dept="D01">
      <nome>João</nome>
      <inicial-meio>S.</inicial-meio>
      <sobrenome>Santos</sobrenome>
    </empregado>
    <empregado cod="E02" dept="D01">
      <nome>Ana</nome>
      <sobrenome>Ferraz</sobrenome>
    </empregado>
  </empregados>
```

Documentos XML como árvores

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

- ▶ Elementos, atributos e **texto** → nodos
- ▶ Relações pai/filho → arestas





Documentos XML Bem-Formados

- Por representar uma estrutura de árvore, algumas restrições se aplicam a documentos XML
 - Raiz única (elemento documento ou elemento raiz)
 - Todas as marcas são fechadas
 - Elementos são bem aninhados (marcas fecham na ordem inversa à em que foram abertas)
 - Exemplo de elemento não bem-aninhado
<empregado><nome>João</empregado></nome>
 - Atributos não se repetem no mesmo elemento
 - Nomes de elementos são sensíveis a maiúsculas e minúsculas



Tipos de Elemento

- | | |
|---------------------------------|--|
| ○ Composto | <empregado> |
| • Contém outros (sub)-elementos | <nome>Ana</nome>
<sobrenome>Ferraz</sobrenome>
</empregado> |
| <hr/> | |
| ○ Textual | <nome>Ana</nome> |
| • Contém somente texto | |
| <hr/> | |
| ○ Misto | <endereco> Rua das Flores, 75 |
| • Contém texto e sub-elementos | <cidade>Rio de Janeiro</cidade>
</endereco> |
| <hr/> | |
| ○ Vazio | <engenheiro></engenheiro> |
| • Elemento sem conteúdo | <engenheiro/> |

Outras considerações importantes

- Elementos são ordenados
- Atributos **não são** ordenados

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
</empregados>
```

Documentos Diferentes!

Outras considerações importantes

- Elementos são ordenados
- Atributos **não são** ordenados

```
<? xml version="1.0" ?>
<empregados>
  <empregado dept="D01" cod="E01" >
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

```
<? xml version="1.0" ?>
<empregados>
  <empregado cod="E01" dept="D01">
    <nome>João</nome>
    <inicial-meio>S.</inicial-meio>
    <sobrenome>Santos</sobrenome>
  </empregado>
  <empregado cod="E02" dept="D01">
    <nome>Ana</nome>
    <sobrenome>Ferraz</sobrenome>
  </empregado>
</empregados>
```

Documentos Iguais

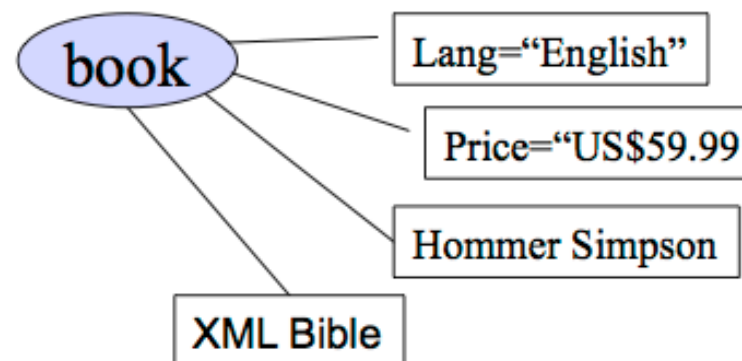


Elementos x Atributos

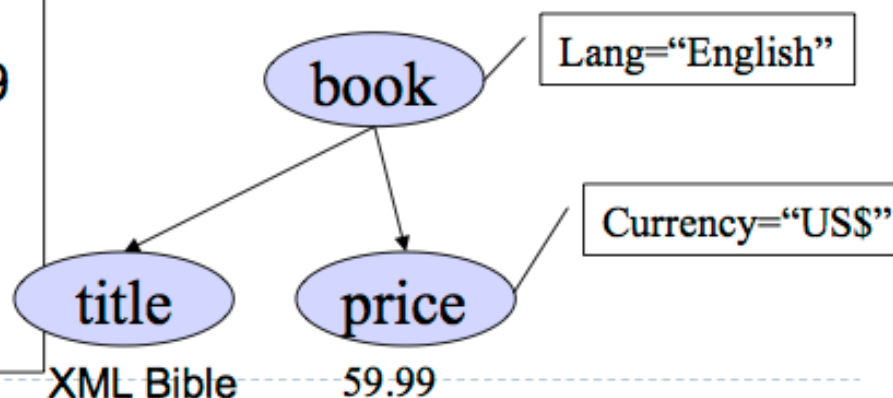
- Não há regras
- Atributos apresentam algumas restrições
 - Não são extensíveis
 - Não permitem múltiplos valores
 - Não descrevem estruturas
- Recomendação: em geral, **preferir elementos**, e **usar atributos para informações secundárias**
- Metadados (dados sobre os dados) devem ser representados como atributos
Ex: <price currency="US">59.99</price>

Elementos e Atributos

```
<book lang="English" price="US  
$59.99"  
title="XML Bible"  
author="Hommer Simpson">  
...  
</book>
```



```
<book lang="English">  
  <price currency="US$"> 59.99  
  </price>  
  <title>XML Bible </title>  
  ...  
</book>
```

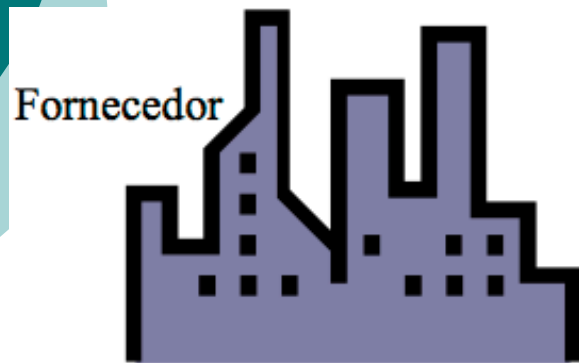




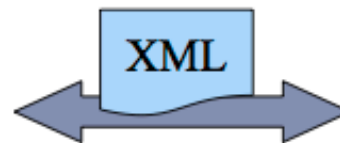
Interoperabilidade

- Chave do sucesso de XML:
 - Usuários podem definir suas próprias marcas
- E a interoperabilidade, como fica
 - Vamos considerar um exemplo prático

Exemplo prático: interoperabilidade



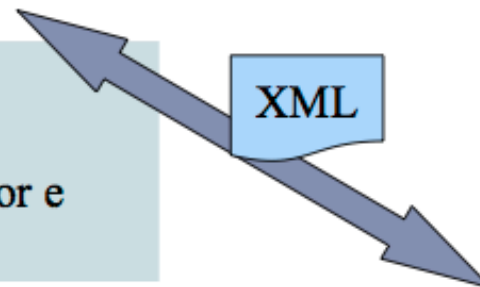
Fornecedor



Cliente X

Aceita pedidos em XML

Pedidos são recebidos pelo servidor e processados automaticamente



Cliente Y

Exemplo prático: interoperabilidade

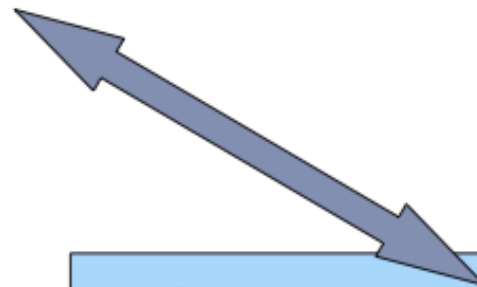
Fornecedor



```
<produto>  
  <nome>caneta</nome>  
  <quant>100</quant>  
</produto>
```



Cliente X

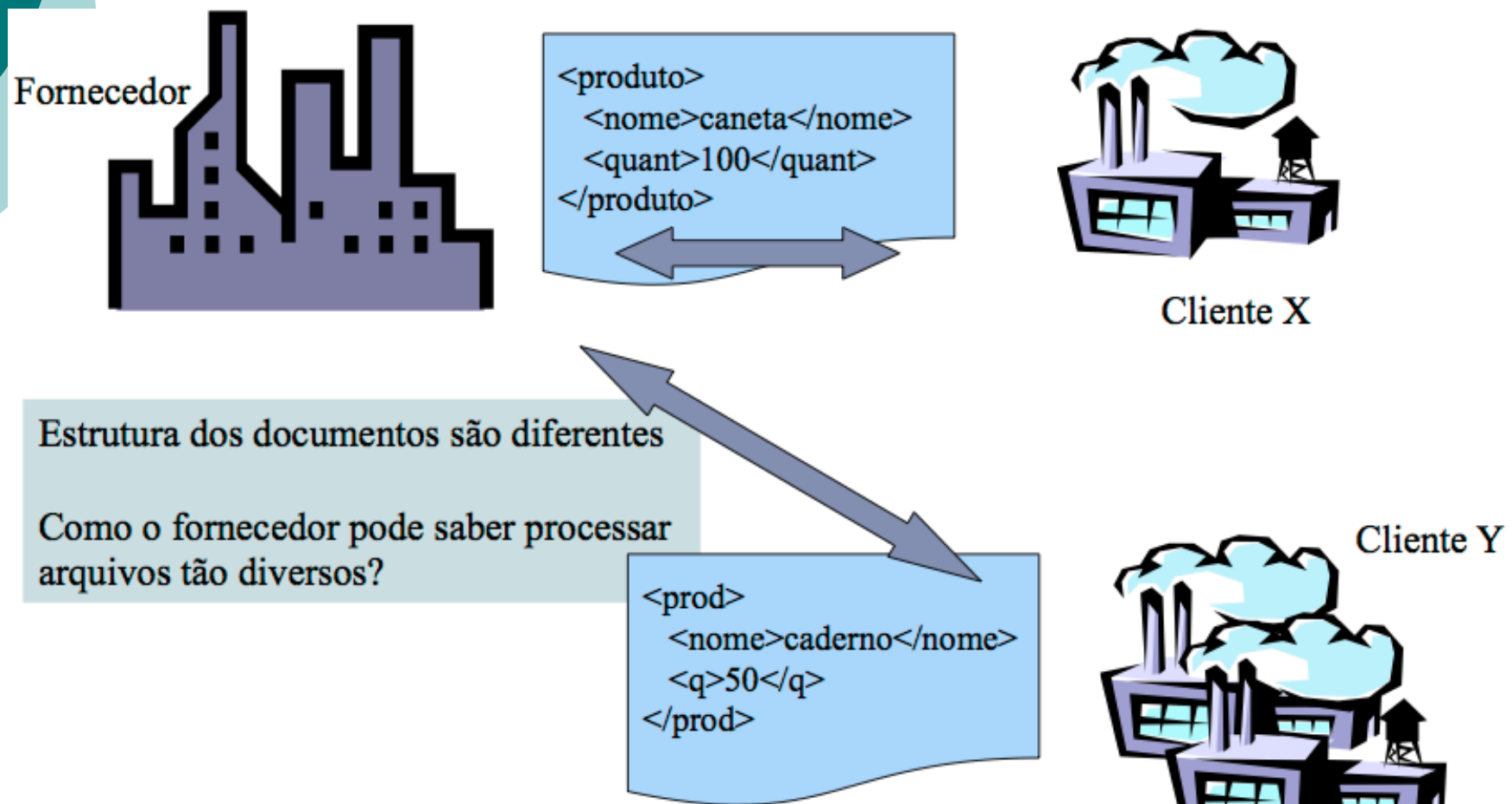


```
<prod>  
  <nome>caderno</nome>  
  <q>50</q>  
</prod>
```

Cliente Y



Exemplo prático: interoperabilidade





Solução

- Pode-se definir um vocabulário usando uma linguagem de esquemas para XML (DTD ou XML Schema)
 - Fornecedor define o vocabulário (estrutura, nomes das marcas, tipos de dados)
 - Cliente usa o vocabulário para enviar os pedidos

Interoperabilidade

Fornecedor

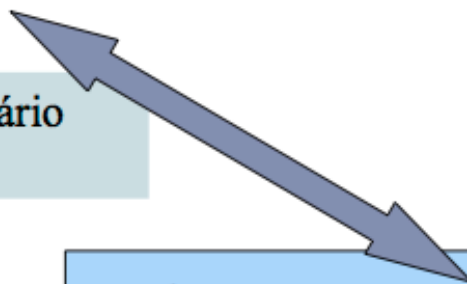


```
<produto>  
  <nome>caneta</nome>  
  <quant>100</quant>  
</produto>
```



Cliente X

Clientes usando o mesmo vocabulário
para enviar os pedidos



```
<produto>  
  <nome>caderno</nome>  
  <quant>50</quant>  
</produto>
```



Cliente Y



Exercícios

- Escrever um documento XML para representar uma receita médico
 - Lembre-se: é importante pensar em como estes documentos serão **estruturados**, e não em como serão **apresentados**

Exemplo de Receituário médico



Ana Maria Marina , 7 anos

Uso interno

Xarope SemTosse

1 colher 3x ao dia

Uso Externo

Gyellow

aplicar no braço 1x ao dia ao deitar



Dr. Juca

20/10/2001



Exercícios

- Faça o seu currículo em XML
- Informações obrigatórias:
 - Dados pessoais
 - Formação
 - Idiomas
 - Cursos adicionais



Leituras complementares

Moro, M., Braganholo, V., “Desmistificando XML: da Pesquisa à Prática Industrial”, disponível em:
<http://www2.ic.uff.br/~vanessa/papers/moro2009-jai.pdf>

“Extensible Markup Language (XML)”, disponível em:
<http://www.w3.org/XML/>

“XML Tutorial” disponível em:
<http://www.w3schools.com/xml/>