



Projeto Apache/Jakarta



The **Apache Jakarta Project**

[http:// jakarta.apache.org/](http://jakarta.apache.org/)

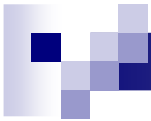
Rogério Napoleão Júnior (rogerionj@gmail.com)

Projeto Apache/Jakarta



■ Projeto Jakarta

- criou e mantém vários *softwares livres* para a plataforma Java. Ele é o "guarda-chuva" de vários outros projetos sob o controle da Apache Software Foundation. Todos os "produtos" Jakarta são licenciados através da licença Apache.



Subprojetos Jakarta



- **BCEL** - uma biblioteca para manipulação do bytecode da linguagem Java
- **BSF** - um *framework* para *scripting*
- **Cactus** - um *framework* para teste unitário de classes Java utilizadas em aplicações servidoras
- **ECS** – o Element Construction Set é uma API para geração de elementos para várias linguagens que suporta diretamente HTML 4.0 e XML
- **JMeter** - uma ferramenta para teste de carga e de desempenho de aplicações

Subprojetos Jakarta (cont)

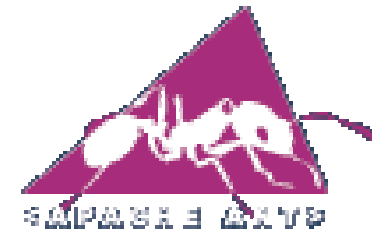


- **JCS** - é um sistema distribuído de caching em java para aplicações java no “lado-servidor”
- **ORO** - Conjunto de classes Java de processamento de texto que fornecem expressões regulares compatíveis com o Perl5, AWK-, expressões glob, e classes utilitário para realizar substituições, divisões, filtragem arquivo, etc.
- **Regexp** – 100% Pure Java Regular Expression package.
- **TagLibs** - Uma coleção de bibliotecas para JavaServer Pages (JSP) com tags customizadas útil na construção de aplicações web

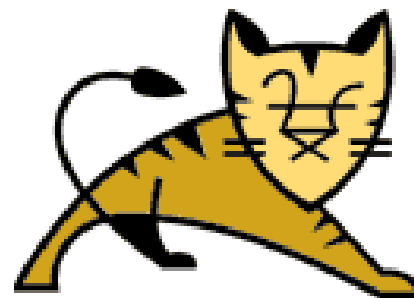
Alguns projetos Ex-Jakarta



ANT



APACHE TOMCAT

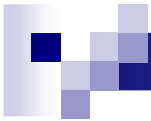




Agenda de estudos



- **XML** – Por que XML? Será usado no ANT, Tomcat ...
- **ANT**
- **Tomcat**



XML - Agenda



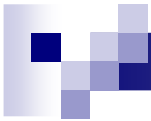
- Histórico
- O que é?
- HTML x XML
- Características
- Utilização do XML
- Sintaxe



XML - Histórico



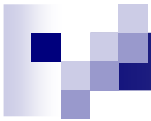
- Proposto em 1996 pelo W3C, juntamente com várias empresas.
- Lançada a primeira especificação em 1998.
- Simplificação do SGML.
- Crescimento acelerado.



XML – O que é?



- eXtensible Markup Language
- Padrão para marcação de dados na Web, com foco na descrição do 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



XML – Características



- Separar o conteúdo da apresentação
- Independência de armazenamento
 - Permite a visualização de dados provenientes de fontes heterogêneas
- Independência de apresentação
 - Permite que as aplicações apresentem/tratem os dados como lhes é conveniente
- Descrição do conteúdo



XML – Características

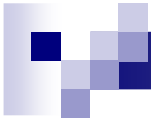


- **Simplicidade**

- ☐ Assim como o HTML, torna fácil a adesão

- **Não impõe um acordo**

- ☐ Cada um descreve seus dados como os vê
- ☐ Aplicações extraem os dados que interessam

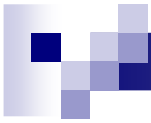


XML x HTML



- HTML: conjunto e semântica de tags pré-definidos
 - `<H1>`, `<TABLE>`, etc.

- XML: meta-linguagem para descrever linguagens de marcação.
 - XML provê recursos para definir tags e as relações estruturais entre elas.



Exemplo XML

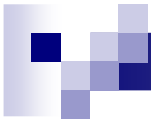


```
<?xml version="1.0" ?>
```

```
<HelloWorld>
```

Este é um teste de Hello Wold!

```
</HelloWorld>
```



Componentes de um XML



- **Prólogo**
 - ☐ Declaração XML
 - ☐ Comentário
 - ☐ Declaração de tipo de documento
 - ☐ Instrução de processamento
- **Comentários**
- **Instruções de Processamento**
- **Elementos**
- **Atributos**
- **Entidades**

Prólogo

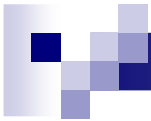


<?xml version="1.0" encoding="iso-8859-1" ?>

<!-- Isto é um comentário -->

<?comando tipo="simples" parametro ?>

<!DOCTYPE pessoa SYSTEM "pessoa.dtd">



Elementos

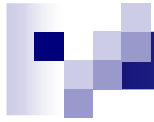


- Marcação mais comum
- Delimitados por < e >
- Exemplos:
 - `<elemento>` qualquer texto `</elemento>`
 - `<elemento/>`

Atributos



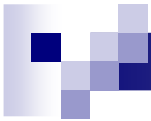
- Pares nome-valor que caracterizam o elemento
- Em XML, todos os atributos devem vir entre aspas.
- Exemplo:
 - `<peessoa nome="Rogério Napoleão Júnior">`



Caracteres Especiais e Entidades



<	<
>	>
“	"
’	'
&	&



Comentários



- Não são processados pelo interpretador XML
- Início: `<!--` Fim: `-->`
- Exemplo:
 - `<!-- Este é apenas um comentário -->`

Instruções de Processamento



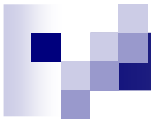
- Recurso para prover informações para uma aplicação
- Não são parte do documento XML, mas são passados às aplicações
- Instruções não reconhecidas são ignoradas
- Formato:
 - `<?name pidata?>`

Seções CDATA

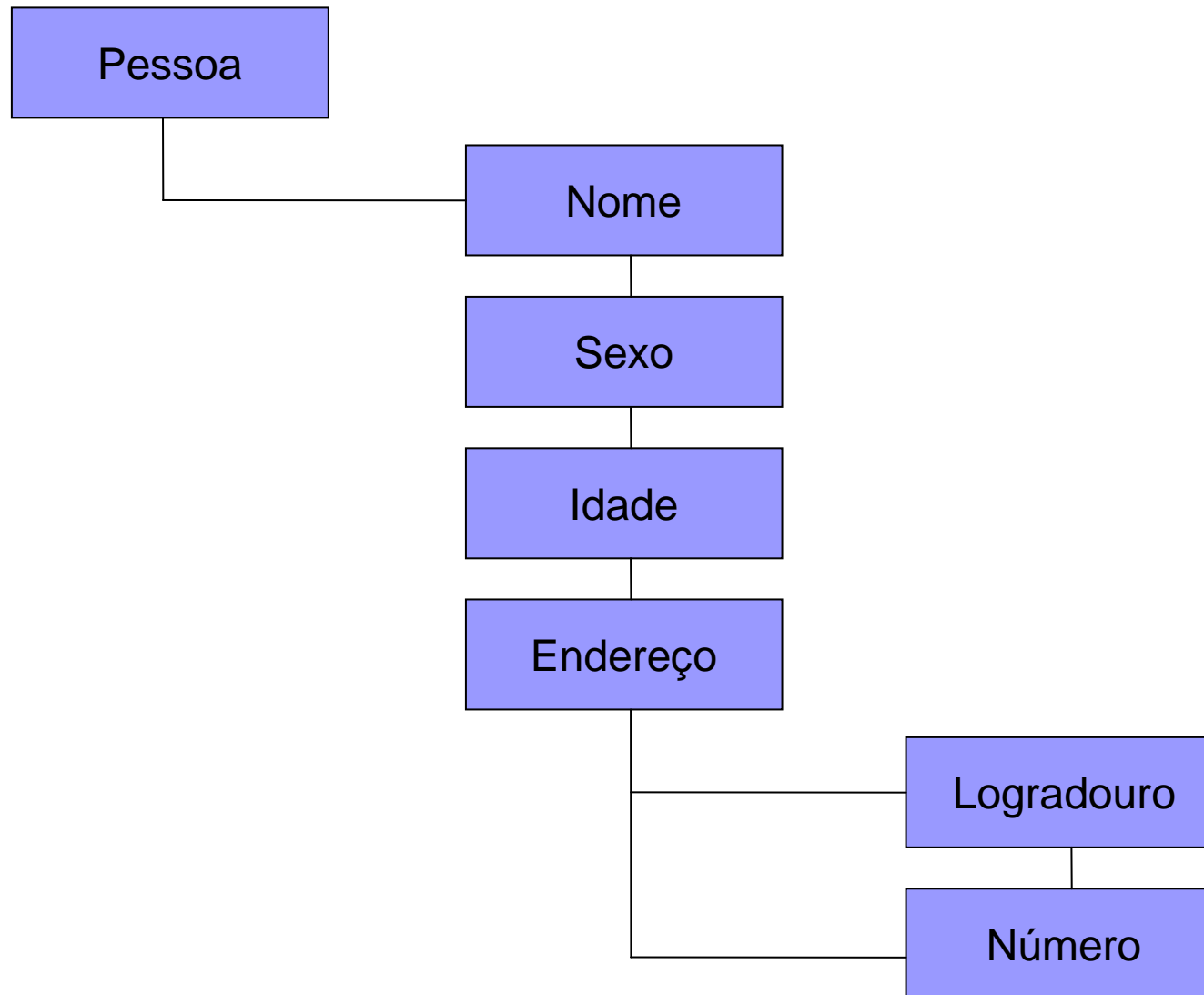


- Desabilita as marcações
- Exemplo:

```
<![CDATA[  
    *p= &q;  
    b = (I <=3);  
]]>
```



Exemplo

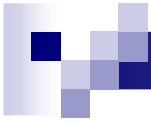




Declaração de Tipos de Documentos (DTD)



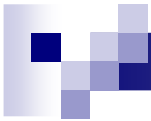
- Permite a criação de Tags
- Permite definição de semântica diferenciada
- Tipos:
 - Declarações de elementos
 - Declarações de listas de atributos
 - Declarações de entidades
 - Declarações de notação



DTD - Exemplo



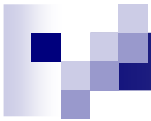
```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Pessoa (Nome,Sexo,Idade,Endereco)>
<!ELEMENT Endereco (Logradouro, Numero)>
<!ELEMENT Nome (#PCDATA)>
<!ELEMENT Sexo (#PCDATA)>
<!ELEMENT Idade (#PCDATA)>
<!ELEMENT Logradouro (#PCDATA)>
<!ELEMENT Numero (#PCDATA)>
```

Multiplicidade DTD

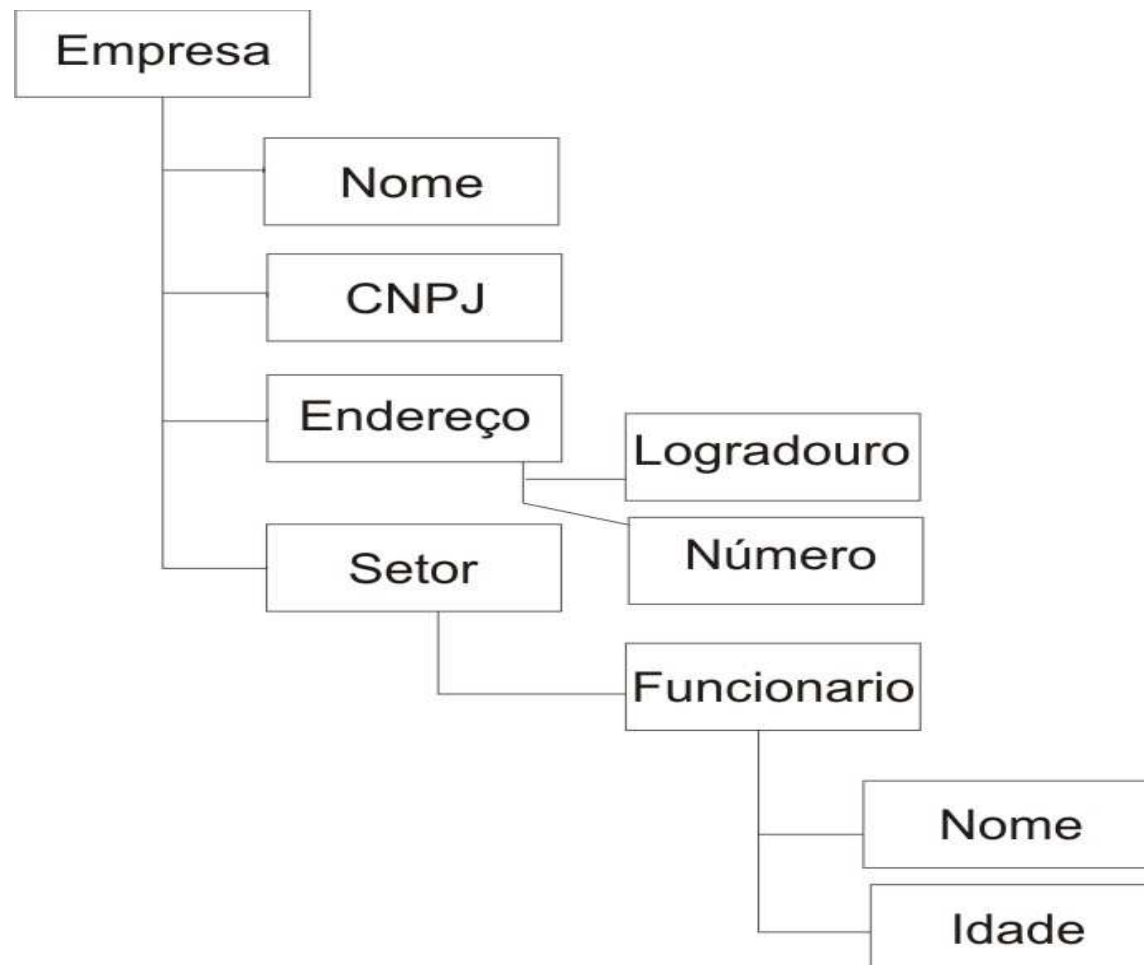
- * - 0 ou muitos elementos
- + - 1 ou muitos elementos
- ? - 1 ou 0

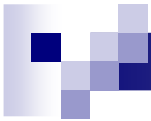
- Exemplo:
 - `<!ELEMENT Pessoa (nome, esposa?, filho*, casa+)>`



Exercício

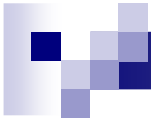
- Crie um arquivo xml e dtd conforme estrutura a seguir (Use as regras de multiplicidade)





XSD e DTD

- Um XML pode ser validado por outro documento:
 - XSD (XML Schema Definition): formato XML
 - DTD (Data Type Definition): formato não-XML
 - Ambos definem a estrutura de elementos e atributos;
- XSD é o sucessor do DTD:
 - Suporta tipos de dados
 - É mais extensível
 - Sua sintaxe segue XML



XSD - Exemplo



```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="pessoa" type="Pessoa" />
  <xs:complexType name="Pessoa">
    <xs:sequence>
      <xs:element name="nome" type="xs:string" minOccurs="0" maxOccurs="1" />
      <xs:element name="sexo" type="xs:string" />
      <xs:element name="idade" type="xs:string" />
      <xs:element name="endereco" type="Endereco" />
    </xs:sequence>
  </xs:complexType>

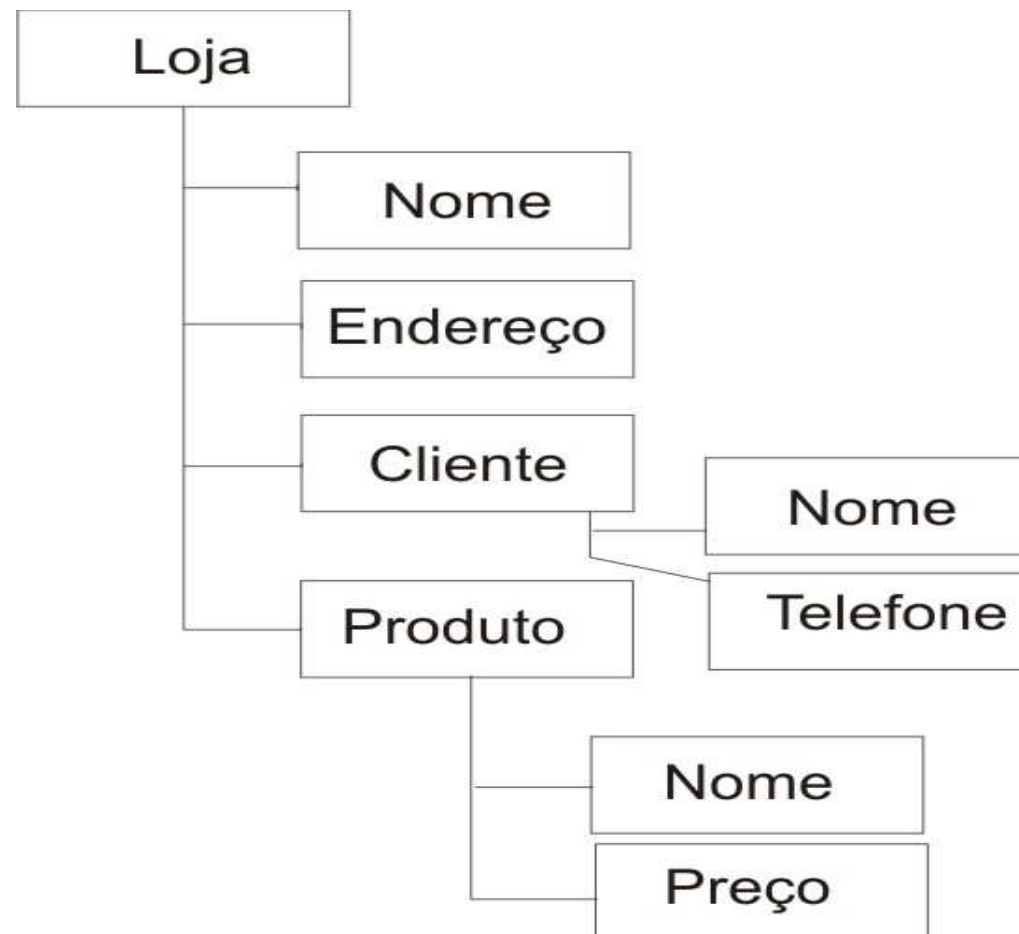
  <xs:complexType name="Endereco">
    <xs:sequence>
      <xs:element name="Logradouro" type="xs:string" />
      <xs:element name="Numero" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="tipo" type="xs:string" use="required" />
  </xs:complexType>
</xs:schema>
```

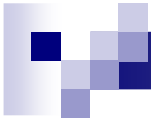


Exercício



- Crie um arquivo xml e xsd conforme estrutura a seguir (Use as regras de multiplicidade)





Links XSD



- <http://www.dei.unicap.br/~almir/seminarios/2003.1/ns06/xmlschemas/schema.html>
- <http://www.w3schools.com/schema/default.asp>