

# Programação Orientada a Serviços

**Prof. Diego Cirilo**

**Aula 06: DTD**

# XML DTD

- *Document Type Definition*
- Define a estrutura e os elementos e atributos permitidos no documento.
- Padronização.
- Validação de documentos externos.

# Validação de XML

- Um XML que segue os padrões W3C é considerado *well-formed*, ou bem formado.
- Exemplo:
  - Possui um, e apenas um, elemento raiz.
  - Todos os elementos possuem uma tag de fim `</fim>`.
  - Respeitam a diferença de maiúsculas/minúsculas.
  - Aninhados corretamente `<b><i>texto</i></b>`.
  - Atributos são definidos entre aspas.
- Para ser considerado **válido**, deve atender ao DTD definido.

# Exemplo

- O DTD pode ser interno (no mesmo arquivo) ou externo.
- Interno:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE nota [
<!ELEMENT nota (para, de, cabecalho, corpo)>
<!ELEMENT para (#PCDATA)>
<!ELEMENT de (#PCDATA)>
<!ELEMENT cabecalho (#PCDATA)>
<!ELEMENT corpo (#PCDATA)>
]>
<nota>
  <para>Chucky</para>
  <de>Annabelle</de>
  <cabecalho>Bilhetinho</cabecalho>
  <corpo>Oi sumido, rs</corpo>
</nota>
```

# Exemplo

- Externo:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE nota SYSTEM "definicoes.dtd">
<nota>
  <para>Chucky</para>
  <de>Annabelle</de>
  <cabecalho>Bilhetinho</cabecalho>
  <corpo>Oi sumido, rs</corpo>
</nota>
```

- definicoes.dtd

```
<?xml version="1.0" encoding="utf-8"?>
<!ELEMENT nota (para,de,cabecalho,corpo)>
<!ELEMENT para (#PCDATA)>
<!ELEMENT de (#PCDATA)>
<!ELEMENT cabecalho (#PCDATA)>
<!ELEMENT corpo (#PCDATA)>
```

# Sintaxe

- `<!DOCTYPE nomeraiz ...>` - Define o DTD e qual o elemento raiz.
- `<!ELEMENT ...>` - Informações do elemento.
- `<!ATTLIST ...>` - Informações dos atributos.
- `<!ENTITY ...>` - Informações da entidade.
- Tipos de dados:
  - PCDATA - *Parsed Character Data*
  - CDATA - *Character Data*
- `SYSTEM "nomedoarquivo.dtd"` - Arquivos externos (locais ou URL)

# Elementos

- `<!ELEMENT nome (tipos-de-dados)>`
- Tipos de dados:
  - `#PCDATA`
  - `EMPTY`
  - Sub-elementos na forma de lista:
    - `(sub-elem1, sub-elem2)`
    - `(sub-elem1|sub-elem2)`
    - `((sub-elem1|sub-elem2), sub-elem3)`
    - `(#PCDATA|sub-elem1|sub-elem2)`
- `|` - operador "ou"
- A ordem dos sub-elementos importa!

# Elementos

- Cardinalidade (quantidade de elementos):
  - **+** um ou mais
  - **\*** zero ou mais
  - **?** zero ou um
  - Se não houver símbolo de cardinalidade ao lado do elemento, ele é obrigatório e único.
  - Pode ser aplicado em parênteses.



# Exemplos

```
<!ELEMENT usuario (nome, cpf, endereco)>
```

```
<!ELEMENT usuario (nome, cpf?, endereco*)>
```

```
<!ELEMENT usuario (nome, (email|cpf)?, endereco*)>
```

```
<!ELEMENT livro (autor+, titulo, num_paginas?)>
```

```
<!ELEMENT livro (autor+, titulo, subtitulo?, editora)>
```

# Atributos

- `<!ATTLIST nomeelemento nomeatributo tipo valor>`
- Tipo
  - CDATA, ID (não pode iniciar com número), etc.
  - Lista. Ex. (valor1|valor2|valor3)
- Valor
  - Padrão "valor-padrao"
  - `#REQUIRED` , `#IMPLIED` , `#FIXED valor`

# Exemplos

```
<!ATTLIST livro isbn CDATA #REQUIRED>
```

```
<!ATTLIST ingrediente unidade (g|l|kg) #REQUIRED>
```

```
<!ATTLIST pagamento tipo (credito|debito|pix) "pix">
```

# Entidades

- Atalhos para caracteres especiais ou textos.
- Entidades padrão ([lista](#))
- `<!ENTITY nome "valor">`

```
<!ENTITY escritor "Pato Donald">  
<!ENTITY copyright "Copyright Disney">
```

- No XML chamamos com `&nome;`, ex:
  - `<autor>&escritor;&copyright;</autor>`

# Tarefa

- Crie um DTD `definicoes.dtd` para o XML da aula passada, os requisitos são os seguintes:
  - Elemento raiz é `cardapio` e deve conter um ou mais elementos `prato`.
  - Elemento `prato` deve conter obrigatoriamente um atributo `id` do tipo ID;
  - Elemento `prato` deve conter os sub-elementos `nome`, `descricao`, `ingredientes`, `preco`, `calorias` e `tempoPreparo`;
  - Elemento `ingredientes` deve conter 1 ou mais sub-elementos `ingrediente`;
  - Os demais elementos são do tipo PCDATA;
  - O elemento `preco` deve conter um atributo `moeda`, com as opções `BRL` e `USD`, o padrão deve ser `BRL`.
  - Defina uma entidade chamada `reais` com o valor `R$`.
- Atualize o seu XML para o padrão desse DTD e coloque a tag `<!DOCTYPE...`, indicando o elemento raiz `cardapio` e apontando para o arquivo `definicoes.dtd`

# Tarefa

- Crie um XML `imobiliaria.xml` com DTD **interno** para os dados de uma imobiliária que cumpra os seguintes requisitos:
  - O elemento raiz é `imobiliária` e deve conter um ou mais elementos `imovel`.
  - O elemento `imovel` deve conter os sub-elementos `descricao`, `proprietario`, `endereco`, `caracteristicas` e `valor`.
  - O elemento `proprietario` deve conter os sub-elementos `nome` e pelo menos um `email` ou `telefone` (pode ter mais de um e pode ter os dois).
  - O elemento `endereco` deve conter os sub-elementos `rua`, `bairro`, `cidade` e `número`, que deve ser opcional.
  - O elemento `caracteristicas` deve conter os sub-elementos `tamanho`, `numQuartos` e `numBanheiros`.
  - O tipo de dado padrão para os demais elementos é PCDATA.
- Crie pelo menos 5 imóveis válidos.
  - Inclua pelo menos um proprietário com 2 telefones e um email e outro apenas com um telefone.
  - Teste também um imóvel sem número.

# Tarefa

- Crie um XML válido com pelo menos 5 questões de 4 alternativas para o [quiz.dtd](#)
- Use o VSCode para validar o XML.

# Dúvidas?

