



**PROVAS DE TI**  
TUDO PARA VOCÊ PASSAR

# Questões XML

Prof. Rodrigo Macedo

# XML

- Separação do conteúdo da formatação.
- Simplicidade e legibilidade, tanto para humanos quanto para computadores.
- Possibilidade de criação de tags sem limitação.
- Criação de arquivos para validação de estrutura (chamados DTDs).
- Interligação de bancos de dados distintos



# DTD - Elementos

- O DTD define um conjunto de regras que definem a estrutura do documento indicando os elementos que podem ser usados e onde podem ser aplicados.
- Conector de sequência usa o conector "," (e)
- Conector de escolha usa o conector "|" (ou)
- Podemos misturar os dois conectores de escolha usando "()": Ex: **<!ELEMENT time (Santos, (Palmeiras|Sao Paulo))>**
- Elemento opcional sem repetição : "?" **<!ELEMENT autor (nome , sobrenome?)>**
- Elemento requerido e com repetição: "+" Ex: **<!ELEMENT autor (nome+)>**
- Quando um elemento é composto por texto: **<!ELEMENT titulo (#PCDATA)**

# DTD - Atributos

- Para um tipo de dados texto usamos a notação **CDATA**.
- Atributos obrigatórios (#Required): **<!ATTLIST autor codigo CDATA #REQUIRED>**
- Atributos ausentes (#Implied): **<!ATTLIST autor email CDATA #IMPLIED>**
- Se o atributo possuir valores padrão usamos:  
**<!ATTLIST autor sexo ( F| M) "F" >**  
**<!ATTLIST autor sexo CDATA "F">**
- **IDREF** - O valor é o ID de outro elemento

# XSD - Xml Schema

- Esta linguagem é uma alternativa ao DTD, cuja sintaxe não era baseada no formato XML.

Tipos suportados:

1. xsd:string – string de caracteres de comprimento ilimitado
2. xsd:boolean – valor booleano (true (1) ou false (0))
3. xsd:decimal – número decimal
4. xsd:float – ponto flutuante
5. xsd:date – Uma data no calendário gregoriano
6. xsd:dateTime – Um instante específico no calendário gregoriano
7. xsd:integer – Um número inteiro

```
<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

# Entidades Especiais

| Character                             | Entity   | Decimal | Hex     | Rendering in Your Browser |         |     |
|---------------------------------------|----------|---------|---------|---------------------------|---------|-----|
|                                       |          |         |         | Entity                    | Decimal | Hex |
| quotation mark = APL quote            | &quot;   | &#34;   | &#x22;  | "                         | "       | "   |
| ampersand                             | &amp;    | &#38;   | &#x26;  | &                         | &       | &   |
| less-than sign                        | &lt;     | &#60;   | &#x3C;  | <                         | <       | <   |
| greater-than sign                     | &gt;     | &#62;   | &#x3E;  | >                         | >       | >   |
| Latin capital ligature OE             | &OElig;  | &#338;  | &#x152; | Œ                         | Œ       | Œ   |
| Latin small ligature oe               | &oelig;  | &#339;  | &#x153; | œ                         | œ       | œ   |
| Latin capital letter S with caron     | &Scaron; | &#352;  | &#x160; | Š                         | Š       | Š   |
| Latin small letter s with caron       | &scaron; | &#353;  | &#x161; | š                         | š       | š   |
| Latin capital letter Y with diaeresis | &Yuml;   | &#376;  | &#x178; | ÿ                         | ÿ       | ÿ   |
| modifier letter circumflex accent     | &circ;   | &#710;  | &#x2C6; | ˆ                         | ˆ       | ˆ   |
| small tilde                           | &tilde;  | &#732;  | &#x2DC; | ˜                         | ˜       | ˜   |

## Q1) [CESPE TJ PA 2020]

```
<rua nome="Almirante Barroso" numero="3089" type="Avenida"/>
```

Considerando que a instrução precedente pertence a um código XML qualquer, assinale a opção correta.

- a) Em XML, o analisador sintático (parser) apresentará, no momento da compilação, um erro de sintaxe, pois os campos nome, numero e type não estão separados por vírgula.
- b) A instrução em questão é um comentário, pois está inserida entre os operadores de delimitação < e >.
- c) Para que o código não apresente erro de construção, deve-se inserir o caractere barra (/) imediatamente após o sinal de menor (<)
- d) Essa instrução é um exemplo típico de declaração XML, pois define o primeiro elemento do documento, o elemento raiz ou nó raiz; logo, deverá ser incluída na primeira linha do documento.
- e) Os atributos Almirante Barroso, 3089 e Avenida estão declarados corretamente, já que, em XML, eles devem estar entre aspas (").

## Q1) [CESPE TJ PA 2020]

```
<rua nome="Almirante Barroso" numero="3089" type="Avenida"/>
```

Considerando que a instrução precedente pertence a um código XML qualquer, assinale a opção correta.

- a) Em XML, o analisador sintático (parser) apresentará, no momento da compilação, um erro de sintaxe, pois os campos nome, numero e type não estão separados por vírgula.
- b) A instrução em questão é um comentário, pois está inserida entre os operadores de delimitação < e >.
- c) Para que o código não apresente erro de construção, deve-se inserir o caractere barra (/) imediatamente após o sinal de menor (<)
- d) Essa instrução é um exemplo típico de declaração XML, pois define o primeiro elemento do documento, o elemento raiz ou nó raiz; logo, deverá ser incluída na primeira linha do documento.
- e) Os atributos Almirante Barroso, 3089 e Avenida estão declarados corretamente, já que, em XML, eles devem estar entre aspas (").



- Q7) [FCC TRF3 2019]** A instrução `< !ELEMENT funcionário (nome, (RG | CPF) + )>` , presente em um DTD para um documento XML, indica que
- a) no interior do elemento funcionário deve existir ou o elemento RG ou CPF , sendo aceitável qualquer um dos dois.
  - b) RG e CPF são elementos opcionais, ou seja, não precisa existir nem RG nem CPF no interior do elemento funcionário.
  - c) há obrigatoriedade dos elementos RG e CPF no interior do elemento nome.
  - d) o elemento nome é opcional, mas o elemento RG é obrigatório no interior do elemento funcionário.
  - e) há obrigatoriedade dos elementos RG e CPF no interior do elemento funcionário.

**Q7) [FCC TRF3 2019]** A instrução < !ELEMENT funcionário (nome, (RG | CPF) + )> , presente em um DTD para um documento XML, indica que

- a) no interior do elemento funcionário deve existir ou o elemento RG ou CPF , sendo aceitável qualquer um dos dois.
- b) RG e CPF são elementos opcionais, ou seja, não precisa existir nem RG nem CPF no interior do elemento funcionário.
- c) há obrigatoriedade dos elementos RG e CPF no interior do elemento nome.
- d) o elemento nome é opcional, mas o elemento RG é obrigatório no interior do elemento funcionário.
- e) há obrigatoriedade dos elementos RG e CPF no interior do elemento funcionário.

**Q9) [FCC TRF3 2019]** Considere o documento XML abaixo.

Para definir que o elemento name deve possuir um atributo id e que este atributo deve obrigatoriamente conter um valor, a lacuna I deve ser corretamente preenchida por

a) <!ATTLIST name id CDATA #IMPLIED>

b) <!ELEMENT name ATTRIBUTE id PCDATA REQUIRED>

c) <!ATTLIST name id CDATA #FIXED id >

d) <!ELEMENT name ATTRIBUTE id CDATA #REQUIRED>

e) <!ATTLIST name id CDATA #REQUIRED>

```
<?xml version = "1.0"?>
<!DOCTYPE address [
  <!ELEMENT address ( name )>
  <!ELEMENT name ( #PCDATA )>
    I
    .....
  ]>
<address>
  <name id="987">
    João Paulo da Silva
  </name>
</address>
```

**Q9) [FCC TRF3 2019]** Considere o documento XML abaixo.

Para definir que o elemento name deve possuir um atributo id e que este atributo deve obrigatoriamente conter um valor, a lacuna I deve ser corretamente preenchida por

a) <!ATTLIST name id CDATA #IMPLIED>

b) <!ELEMENT name ATTRIBUTE id PCDATA REQUIRED>

c) <!ATTLIST name id CDATA #FIXED id >

d) <!ELEMENT name ATTRIBUTE id CDATA #REQUIRED>

e) <!ATTLIST name id CDATA #REQUIRED>

```
<?xml version = "1.0"?>
<!DOCTYPE address [
  <!ELEMENT address ( name )>
  <!ELEMENT name ( #PCDATA )>
    I
    .....
  ]>
<address>
  <name id="987">
    João Paulo da Silva
  </name>
</address>
```

**Q11) [FCC TRF3 2019]** Para definir o elemento `cliente` como tipo `string` em um XML Schema utiliza-se a instrução

a) `<xml : element name="cliente type="string">Paulo</xml:element>`

b) `<xsl : element id = "cliente" type = "xsl : string"/>`

c) `<xs : element name = "cliente" type = "xs : string" / >`

d) `<xs : element id = "cliente" type = "xs : varchar "/> Paulo</xml : element>`

e) `<xml: element name = "cliente" type = "string "value="Paulo"/ >`

**Q11) [FCC TRF3 2019]** Para definir o elemento `cliente` como tipo `string` em um XML Schema utiliza-se a instrução

a) `<xml : element name="cliente type="string">Paulo</xml:element>`

b) `<xsl : element id = "cliente" type = "xsl : string"/>`

c) `<xs : element name = "cliente" type = "xs : string" / >`

d) `<xs : element id = "cliente" type = "xs : varchar "/> Paulo</xml : element>`

e) `<xml: element name = "cliente" type = "string "value="Paulo"/ >`

**Q14) [FGV MPE-RJ 2019]** A troca de dados entre sistemas computacionais é normalmente realizada por meio de arquivos que seguem padrões de formato e organização. Desse modo, diferentes agentes com diferentes equipamentos podem enviar e receber dados estruturados muito facilmente. Nesse contexto, analise um trecho do conteúdo de um dado arquivo a seguir.

```
<nota> <para>Rita</para> <de>Bernardo</de> <título>Lembrete</título>  
<texto>O pacote &lt;chegou&gt; ...</texto> </nota>
```

Com base nesse trecho, é correto deduzir que a organização desse arquivo segue o padrão conhecido como:

- a) CSS
- b) CSV
- c) ODF
- d) PDF
- e) XML

**Q14) [FGV MPE-RJ 2019]** A troca de dados entre sistemas computacionais é normalmente realizada por meio de arquivos que seguem padrões de formato e organização. Desse modo, diferentes agentes com diferentes equipamentos podem enviar e receber dados estruturados muito facilmente. Nesse contexto, analise um trecho do conteúdo de um dado arquivo a seguir.

```
<nota> <para>Rita</para> <de>Bernardo</de> <título>Lembrete</título>  
<texto>O pacote &lt;chegou&gt; ...</texto> </nota>
```

Com base nesse trecho, é correto deduzir que a organização desse arquivo segue o padrão conhecido como:

- a) CSS
- b) CSV
- c) ODF
- d) PDF
- e) XML



**Q22) [FCC TJMA 2019]** Considere as instruções abaixo:

Trata-se de um DTD que define

- a) a estrutura e os elementos de um arquivo XML.
- b) quais os elementos são válidos em uma página web.
- c) os metadados de uma página criada com ASP.NET.
- d) os atributos de uma tabela de banco de dados relacional.
- e) os formatos de elementos para versionamento de documentos com GIT.

```
<!DOCTYPE cliente  
[  
  <!ELEMENT cliente (nome,cidade)>  
  <!ELEMENT nome (#PCDATA)>  
  <!ELEMENT cidade (#PCDATA)>  

```

**Q22) [FCC TJMA 2019]** Considere as instruções abaixo:

Trata-se de um DTD que define

a) a estrutura e os elementos de um arquivo XML.

b) quais os elementos são válidos em uma página web.

c) os metadados de uma página criada com ASP.NET.

d) os atributos de uma tabela de banco de dados relacional.

e) os formatos de elementos para versionamento de documentos com GIT.

```
<!DOCTYPE cliente
[
  <!ELEMENT cliente (nome,cidade)>
  <!ELEMENT nome (#PCDATA)>
  <!ELEMENT cidade (#PCDATA)>
]>
```

**Q24) [FCC SANASA 2019]** Considere o código XML abaixo.

O código não está bem formatado e NÃO é válido porque

a) o atributo lider do funcionário João não contém um número de funcionário válido, pois este deve aparecer em um dos atributos num\_funcionario nas linhas anteriores.

b) o elemento nome\_funcionario não pode ser do tipo #PCDATA, mas sim do tipo #STRING ou #TEXT.

c) o atributo lider referenciado como IDREF não pode ser #IMPLIED, mas sim #REQUIRED.

d) o elemento nome\_funcionario não pode ter mais de um atributo, ou seja, ou se utiliza o atributo num\_funcionario ou o atributo lider.

e) para que o elemento equipe suporte vários elementos nome\_funcionario o asterisco (\*) da linha `<!ELEMENT equipe (nome_funcionario)*>` deve ser substituído pelo sinal de adição (+).

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE equipe [
  <!ELEMENT equipe (nome_funcionario)*>
  <!ELEMENT nome_funcionario (#PCDATA)>
  <!ATTLIST nome_funcionario num_funcionario ID #REQUIRED>
  <!ATTLIST nome_funcionario lider IDREF #IMPLIED>
]>
<equipe>
  <nome_funcionario num_funcionario="a8904885">Marcelo</nome_funcionario>
  <nome_funcionario num_funcionario="a9011133">Andrea</nome_funcionario>
  <nome_funcionario num_funcionario="a9216735"
    lider="a9211135">João</nome_funcionario>
</equipe>
```

**Q24) [FCC SANASA 2019]** Considere o código XML abaixo.

O código não está bem formatado e NÃO é válido porque

a) o atributo lider do funcionário João não contém um número de funcionário válido, pois este deve aparecer em um dos atributos num\_funcionario nas linhas anteriores.

b) o elemento nome\_funcionario não pode ser do tipo #PCDATA, mas sim do tipo #STRING ou #TEXT.

c) o atributo lider referenciado como IDREF não pode ser #IMPLIED, mas sim #REQUIRED.

d) o elemento nome\_funcionario não pode ter mais de um atributo, ou seja, ou se utiliza o atributo num\_funcionario ou o atributo lider.

e) para que o elemento equipe suporte vários elementos nome\_funcionario o asterisco (\*) da linha `< ! ELEMENT equipe (nome_funcionario) *>` deve ser substituído pelo sinal de adição (+).

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE equipe [
  <!ELEMENT equipe (nome_funcionario)*>
  <!ELEMENT nome_funcionario (#PCDATA)>
  <!ATTLIST nome_funcionario num_funcionario ID #REQUIRED>
  <!ATTLIST nome_funcionario lider IDREF #IMPLIED>
]>
<equipe>
  <nome_funcionario num_funcionario="a8904885">Marcelo</nome_funcionario>
  <nome_funcionario num_funcionario="a9011133">Andrea</nome_funcionario>
  <nome_funcionario num_funcionario="a9216735"
    lider="a9211135">João</nome_funcionario>
</equipe>
```

**Q28) [FCC Prefeitura de Manaus 2019]** Para que o documento XML abaixo fosse considerado bem formatado e válido, um programador criou um DTD chamado Contribuinte.dtd com regras para esse documento.

```
<?xml version="1.0" encoding="UTF-8"?>
<contribuinte>
  <nome>Antônio Andrade Silva</nome>
  <cpf>154.246.045-14</cpf>
</contribuinte>
```

Para fazer referência ao DTD criado, na linha seguinte à tag <?xml version="1.0" encoding="UTF-8"? o programador deverá colocar o comando:

- a) <link type="text/dtd" href="Contribuinte.dtd">
- b) <!DOCTYPE contribuinte SYSTEM "Contribuinte.dtd">
- c) <!INCLUDE contribuinte SYSTEM "Contribuinte.dtd">
- d) <!DOCTYPE contribuinte FILE "Contribuinte.dtd" FILETYPE "text/dtd">
- e) <link type="text/dtd" src="Contribuinte.dtd">

**Q28) [FCC Prefeitura de Manaus 2019]** Para que o documento XML abaixo fosse considerado bem formatado e válido, um programador criou um DTD chamado Contribuinte.dtd com regras para esse documento.

```
<?xml version="1.0" encoding="UTF-8"?>
<contribuinte>
  <nome>Antônio Andrade Silva</nome>
  <cpf>154.246.045-14</cpf>
</contribuinte>
```

Para fazer referência ao DTD criado, na linha seguinte à tag <?xml version="1.0" encoding="UTF-8"? o programador deverá colocar o comando:

- a) <link type="text/dtd" href="Contribuinte.dtd">
- b) <!DOCTYPE contribuinte SYSTEM "Contribuinte.dtd">
- c) <!INCLUDE contribuinte SYSTEM "Contribuinte.dtd">
- d) <!DOCTYPE contribuinte FILE "Contribuinte.dtd" FILETYPE "text/dtd">
- e) <link type="text/dtd" src="Contribuinte.dtd">

**Q29) [IADES BRB 2019]** O objetivo de um XML Schema é definir quais são os blocos de construção permitidos em um documento XML. O XML Schema tem potencial para substituir a tecnologia

a) Namespaces.

b) Postgree.

c) DTD.

d) SWT

e) EJBQL.

**Q29) [IADES BRB 2019]** O objetivo de um XML Schema é definir quais são os blocos de construção permitidos em um documento XML. O XML Schema tem potencial para substituir a tecnologia

a) Namespaces.

b) Postgree.

c) DTD.

d) SWT

e) EJBQL.



**Q32) [FCC TRF4 2019]** Considere o bloco XML Schema abaixo.

:

Para que o valor de idade não possa ser menor que 0 (zero), a lacuna I deve ser preenchida por

- a) <xs:upToInt value="0"/>
- b) <xs:biggerThan value="0"/>
- c) <xs:minAcceptValue value="0"/>
- d) <xs:minLength value="0"/>
- e) <xs:minInclusive value="0"/>

```
<xs:element name="idade">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
          I    
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**Q32) [FCC TRF4 2019]** Considere o bloco XML Schema abaixo.

:

Para que o valor de idade não possa ser menor que 0 (zero), a lacuna I deve ser preenchida por

- a) `<xs:upToInt value="0"/>`
- b) `<xs:biggerThan value="0"/>`
- c) `<xs:minAcceptValue value="0"/>`
- d) `<xs:minLength value="0"/>`
- e) `<xs:minInclusive value="0"/>`

```
<xs:element name="idade">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
          I    
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

**Q36) [CCV-UFC UFC 2019]** Sobre as características da linguagem XML (eXtensible Markup Language), é correto afirmar:

- a) o usuário da linguagem XML pode definir novas tags para melhor estruturar a informação contida no arquivo.
- b) a XML é uma linguagem de programação que necessita de um compilador específico para gerar o arquivo binário a ser executado em algum sistema.
- c) a desvantagem da XML é a dependência da plataforma sobre a qual está executando, sendo necessárias adequações para cada tipo de sistema.
- d) a característica de extensibilidade da linguagem está relacionada ao fato de ser possível criar funções a partir de um conjunto fixo de tags fornecido pela linguagem.
- e) a linguagem XML fornece o recurso de tipagem dos dados, de forma que, por exemplo, é possível definir números inteiros e realizar operações sobre eles no programa XML.

**Q36) [CCV-UFC UFC 2019]** Sobre as características da linguagem XML (eXtensible Markup Language), é correto afirmar:

a) o usuário da linguagem XML pode definir novas tags para melhor estruturar a informação contida no arquivo.

b) a XML é uma linguagem de programação que necessita de um compilador específico para gerar o arquivo binário a ser executado em algum sistema.

c) a desvantagem da XML é a dependência da plataforma sobre a qual está executando, sendo necessárias adequações para cada tipo de sistema.

d) a característica de extensibilidade da linguagem está relacionada ao fato de ser possível criar funções a partir de um conjunto fixo de tags fornecido pela linguagem.

e) a linguagem XML fornece o recurso de tipagem dos dados, de forma que, por exemplo, é possível definir números inteiros e realizar operações sobre eles no programa XML.

**Q37) [FGV DPE-RJ 2019]** Considere os trechos XML exibidos a seguir.

I. <p>Um primeiro exemplo (/p>  
<br/>

II. <message>Texto breve</message>

III. <b><i>Texto com destaque.</b></i>

IV. <p>Note que, para  $x > 1$ , a resposta é sim.</p>

O número de trechos válidos é:

a) 0 (zero)

b) 1

c) 2

d) 3

e) 4

**Q37) [FGV DPE-RJ 2019]** Considere os trechos XML exibidos a seguir.

I. <p>Um primeiro exemplo (/p>

<br/>

II. <message>Texto breve</message>

III. <b><i>Texto com destaque.</b></i>

IV. <p>Note que, para  $x > 1$ , a resposta é sim.</p>

O número de trechos válidos é:

a) 0 (zero)

b) 1

c) 2

d) 3

e) 4

**Q41) [FCC SEMEF 2019]** Para que o documento XML abaixo fosse considerado bem formatado e válido, um programador criou um DTD chamado Contribuinte.dtd com regras para esse documento.

Para fazer referência ao DTD criado, na linha seguinte à tag `<?xml version="1.0" encoding="UTF-8"?>` o programador deverá colocar o comando:

a) `<link type="text/dtd" href="Contribuinte.dtd">`

b) `< ! DOCTYPE contribuinte SYSTEM "Contribuinte.dtd">`

c) `< ! INCLUDE contribuinte SYSTEM "Contribuinte.dtd">`

d) `< ! DOCTYPE contribuinte FILE "Contribuinte.dtd" FILETYPE "text/dtd">`

e) `<link type="text/dtd" src="Contribuinte.dtd">`

```
<?xml version="1.0" encoding="UTF-8"?>
<contribuinte>
  <nome>Antônio Andrade Silva</nome>
  <cpf>154.246.045-14</cpf>
</contribuinte>
```

**Q41) [FCC SEMEF 2019]** Para que o documento XML abaixo fosse considerado bem formatado e válido, um programador criou um DTD chamado Contribuinte.dtd com regras para esse documento.

Para fazer referência ao DTD criado, na linha seguinte à tag `<?xml version="1.0" encoding="UTF-8"?>` o programador deverá colocar o comando:

a) `<link type="text/dtd" href="Contribuinte.dtd">`

b) `< ! DOCTYPE contribuinte SYSTEM "Contribuinte.dtd">`

c) `< ! INCLUDE contribuinte SYSTEM "Contribuinte.dtd">`

d) `< ! DOCTYPE contribuinte FILE "Contribuinte.dtd" FILETYPE "text/dtd">`

e) `<link type="text/dtd" src="Contribuinte.dtd">`

```
<?xml version="1.0" encoding="UTF-8"?>
<contribuinte>
  <nome>Antônio Andrade Silva</nome>
  <cpf>154.246.045-14</cpf>
</contribuinte>
```



**Q47) [CESGRANRIO TRANSPETRO 2018]** Documentos XML são estruturados segundo uma hierarquia de unidades informacionais chamadas de nós. Qual tecnologia XML fornece ao desenvolvedor uma API para adicionar, editar e remover esses nós?

- a) XMI
- b) XSDL
- c) XSLT
- d) XML DOM
- e) XML Schema

**Q47) [CESGRANRIO TRANSPETRO 2018]** Documentos XML são estruturados segundo uma hierarquia de unidades informacionais chamadas de nós. Qual tecnologia XML fornece ao desenvolvedor uma API para adicionar, editar e remover esses nós?

a) XMI

b) XSDL

c) XSLT

d) XML DOM

e) XML Schema

**Q53) [FAURGS UFRGS 2018]** Um documento \_\_\_\_\_ bem formado pode ser analisado por processadores genéricos, denominados \_\_\_\_\_ que percorrem o documento e criam uma representação no formato de \_\_\_\_\_. A \_\_\_\_\_ permite que os programas manipulem a sua representação correspondente; entretanto, o documento inteiro precisa ser analisado de antemão para conversão de sua estrutura interna. A \_\_\_\_\_ permite o processamento do documento por meio de eventos, sempre que uma tag de início ou de fim é encontrada. Isso facilita o processamento de documentos grandes.

Assinale a alternativa que preenche, correta e respectivamente, as lacunas do texto acima.

- a) XML – APIs (Application Programming Interface) – árvore – API DOM (Document Object Model) – API SAX (Simple API for XML)
- b) HTML – parsers HTML – fila – API SAX (Simple API for XML) – API (Application Programming Interface)
- c) XML – APIs DOM (Document Object Model) – lista – API (Application Programming Interface) – API SAX (Simple API for XML)
- d) HTML – APIs (Application Programming Interface) – árvore – API SAX (Simple API for XML) – API DOM (Document Object Model)
- e) XML – parsers HTML – pilha – API DOM (Document Object Model) – API (Application Programming Interface)

**Q53) [FAURGS UFRGS 2018]** Um documento \_\_\_\_\_ bem formado pode ser analisado por processadores genéricos, denominados \_\_\_\_\_ que percorrem o documento e criam uma representação no formato de \_\_\_\_\_. A \_\_\_\_\_ permite que os programas manipulem a sua representação correspondente; entretanto, o documento inteiro precisa ser analisado de antemão para conversão de sua estrutura interna. A \_\_\_\_\_ permite o processamento do documento por meio de eventos, sempre que uma tag de início ou de fim é encontrada. Isso facilita o processamento de documentos grandes.

Assinale a alternativa que preenche, correta e respectivamente, as lacunas do texto acima.

- a) XML – APIs (Application Programming Interface) – árvore – API DOM (Document Object Model) – API SAX (Simple API for XML)
- b) HTML – parsers HTML – fila – API SAX (Simple API for XML) – API (Application Programming Interface)
- c) XML – APIs DOM (Document Object Model) – lista – API (Application Programming Interface) – API SAX (Simple API for XML)
- d) HTML – APIs (Application Programming Interface) – árvore – API SAX (Simple API for XML) – API DOM (Document Object Model)
- e) XML – parsers HTML – pilha – API DOM (Document Object Model) – API (Application Programming Interface)

**Q55) [FGV Prefeitura de Niterói 2018]** Considere a declaração do tipo de documento (DTD) a seguir.

```
<!ELEMENT blog (nome, autor+, artigo*, permalink?) >  
<!ELEMENT nome (#PCDATA)>  
<!ELEMENT autor (#PCDATA)>  
<!ELEMENT artigo (#PCDATA)>  
<!ELEMENT permalink (#PCDATA)>
```

Em um documento XML, que obedece a esse conjunto de regras,

- a) deve haver precisamente um elemento do tipo artigo.
- b) podem ocorrer vários elementos do tipo permalink.
- c) pode conter um elemento do tipo nome e outro elemento do tipo autor, em qualquer ordem.
- d) deve existir um elemento do tipo autor, sendo permitidas múltiplas ocorrências deste tipo de elemento.
- e) não é necessário haver um elemento do tipo nome.

**Q55) [FGV Prefeitura de Niterói 2018]** Considere a declaração do tipo de documento (DTD) a seguir.

```
<!ELEMENT blog (nome, autor+, artigo*, permalink?) >  
<!ELEMENT nome (#PCDATA)>  
<!ELEMENT autor (#PCDATA)>  
<!ELEMENT artigo (#PCDATA)>  
<!ELEMENT permalink (#PCDATA)>
```

Em um documento XML, que obedece a esse conjunto de regras,

- a) deve haver precisamente um elemento do tipo artigo.
- b) podem ocorrer vários elementos do tipo permalink.
- c) pode conter um elemento do tipo nome e outro elemento do tipo autor, em qualquer ordem.
- d) deve existir um elemento do tipo autor, sendo permitidas múltiplas ocorrências deste tipo de elemento.
- e) não é necessário haver um elemento do tipo nome.

**Q60) [CCV-UFC UFC 2018]** Qual dos seguintes fragmentos representa um fragmento XML bem formado?

- a) `<myElement myAttribute="someValue"/>`
- b) `<myElement myAttribute="someValue' />`
- c) `<myElement myAttribute='someValue'>`
- d) `<myElement myAttribute=someValue/>`
- e) `<myElement myAttribute=someValue>`

**Q60) [CCV-UFC UFC 2018]** Qual dos seguintes fragmentos representa um fragmento XML bem formado?

a) `<myElement myAttribute="someValue"/>`

b) `<myElement myAttribute="someValue' />`

c) `<myElement myAttribute='someValue'>`

d) `<myElement myAttribute=someValue/>`

e) `<myElement myAttribute=someValue>`



**Q61) [CESPE TCE MG 2018]** A recomendação do World Wide Web Consortium (W3C) que especifica como descrever formalmente os elementos em um documento XML para verificar se cada item de conteúdo no documento adere à descrição do elemento no qual o conteúdo deve ser colocado é designada

- a) extensible stylesheet language for transformation.
- b) XSD.
- c) definição de tipo de documento (DTD).
- d) XLSX.
- e) XML path language (XPath).

**Q61) [CESPE TCE MG 2018]** A recomendação do World Wide Web Consortium (W3C) que especifica como descrever formalmente os elementos em um documento XML para verificar se cada item de conteúdo no documento adere à descrição do elemento no qual o conteúdo deve ser colocado é designada

a) extensible stylesheet language for transformation.

b) XSD.

c) definição de tipo de documento (DTD).

d) XLSX.

e) XML path language (XPath).

**Q65) [UFPR COREN-PR 2018]** O principal documento de referência de interoperabilidade no Brasil é o Padrão de Interoperabilidade do Governo Eletrônico (ePING). Em suas especificações técnicas de organização e intercâmbio de informação, esse documento define que as duas linguagens adotadas para tratamento e transferência de dados são

a) HTML e JSON

b) HTML e XML

c) JSON e XML

d) XML e XLST

e) XSL e XML

**Q65) [UFPR COREN-PR 2018]** O principal documento de referência de interoperabilidade no Brasil é o Padrão de Interoperabilidade do Governo Eletrônico (ePING). Em suas especificações técnicas de organização e intercâmbio de informação, esse documento define que as duas linguagens adotadas para tratamento e transferência de dados são

a) HTML e JSON

b) HTML e XML

c) JSON e XML

d) XML e XLST

e) XSL e XML

**Q66) [CESGRANRIO TRANSPETRO 2018]** Considerando a linguagem XML, qual é o exemplo correto de uso de um atributo chamado “src” que recebe o valor “computador.gif” em um elemento de nome “img”?

- a) ``
- b) ``
- c) `<img> "src=computador.gif " </img>`
- d) `<img> <src> computador.gif </src> </img>`
- e) `<img> src="computador.gif " </img>`

**Q66) [CESGRANRIO TRANSPETRO 2018]** Considerando a linguagem XML, qual é o exemplo correto de uso de um atributo chamado “src” que recebe o valor “computador.gif” em um elemento de nome “img”?

- a) ``
- b) ``
- c) `<img> "src=computador.gif " </img>`
- d) `<img> <src> computador.gif </src> </img>`
- e) `<img> src="computador.gif " </img>`

**Q67) [CESPE PF 2018]** Julgue o item a seguir, a respeito da teoria da informação e de metadados de arquivos.

Em arquivos no formato XML, as tags não são consideradas metadados.

**Q67) [CESPE PF 2018]** Julgue o item a seguir, a respeito da teoria da informação e de metadados de arquivos.

Em arquivos no formato XML, as tags não são consideradas metadados. ERRADO.



**Q69) [FAUGRS TJ-RS 2018]** Quanto à especificação de elementos na notação DTD (Document Type Definition), assinale a alternativa correta.

- a) O sinal “/”, após o nome do elemento, significa que esse nome pode ser repetido uma ou mais vezes no documento.
- b) Um símbolo de barra (e1 | e2) especifica que e1 ou e2 podem aparecer no documento.
- c) O sinal “\*”, após o nome do elemento, significa que esse nome pode ser repetido zero ou uma vez no documento.
- d) Um elemento de único valor (não repetitivo) obrigatório pode aparecer zero ou mais vezes no documento.
- e) O sinal “&”, após o nome do elemento, significa que esse nome pode ser repetido zero ou mais vezes no documento.

**Q69) [FAUGRS TJ-RS 2018]** Quanto à especificação de elementos na notação DTD (Document Type Definition), assinale a alternativa correta.

- a) O sinal “/”, após o nome do elemento, significa que esse nome pode ser repetido uma ou mais vezes no documento.
- b) Um símbolo de barra (e1 | e2) especifica que e1 ou e2 podem aparecer no documento.
- c) O sinal “\*”, após o nome do elemento, significa que esse nome pode ser repetido zero ou uma vez no documento.
- d) Um elemento de único valor (não repetitivo) obrigatório pode aparecer zero ou mais vezes no documento.
- e) O sinal “&”, após o nome do elemento, significa que esse nome pode ser repetido zero ou mais vezes no documento.

**Q73) [FGV MPE AL 2018]** No XML, a sequência de símbolos &lt; representa

a) <

b) >

c) "

d) &

e) '

**Q73) [FGV MPE AL 2018]** No XML, a sequência de símbolos &lt; representa

a) <

b) >

c) "

d) &

e) '

**Q74) [SUGEP-UFRPE UFRPE 2018]** A linguagem PHP permite a instalação de extensões que podem aumentar sua gama de funcionalidades. Uma das funcionalidades extras que podem ser adicionadas se refere à manipulação de arquivos XML. A extensão que possui várias classes que podem ser instanciadas para a leitura e gravação de arquivos XML é chamada:

- a) DOM.
- b) XML-RPC.
- c) Ctype.
- d) SCA.
- e) YAZ.

**Q74) [SUGEP-UFRPE UFRPE 2018]** A linguagem PHP permite a instalação de extensões que podem aumentar sua gama de funcionalidades. Uma das funcionalidades extras que podem ser adicionadas se refere à manipulação de arquivos XML. A extensão que possui várias classes que podem ser instanciadas para a leitura e gravação de arquivos XML é chamada:

- a) DOM.
- b) XML-RPC.
- c) Ctype.
- d) SCA.
- e) YAZ.

**Q76) [CESGRANRIO BASA 2018]** Considere o esquema XML a seguir:

```
<xs:element name="rectangle" type="area"/>
<xs:complexType name="area">
  <xs:attribute name="x1" type="xs:decimal" />
  <xs:attribute name="y1" type="xs:decimal" />
  <xs:attribute name="x2" type="xs:decimal" />
  <xs:attribute name="y2" type="xs:decimal" />
</xs:complexType>
```

Um elemento XML válido, segundo esse esquema, é:

- a) <area><x1>1</x1><y1>1</y1><x2>2</x2><y2>3</y2></area>
- b) <area x1="1" y1="1" x2="4" y2="5" />
- c) <rectangle x1="5" y1="4" x2="1" y2="1"/>
- d) <area><x1>4</x1><y1>4</y1><x2>2</x2><y2>3</y2></area>
- e) <rectangle><x1>1</x1><y1>1</y1><x2>2</x2><y2>3</y2></rectangle>

**Q76) [CESGRANRIO BASA 2018]** Considere o esquema XML a seguir:

```
<xs:element name="rectangle" type="area"/>
<xs:complexType name="area">
  <xs:attribute name="x1" type="xs:decimal" />
  <xs:attribute name="y1" type="xs:decimal" />
  <xs:attribute name="x2" type="xs:decimal" />
  <xs:attribute name="y2" type="xs:decimal" />
</xs:complexType>
```

Um elemento XML válido, segundo esse esquema, é:

a) <area><x1>1</x1><y1>1</y1><x2>2</x2><y2>3</y2></area>

b) <area x1="1" y1="1" x2="4" y2="5" />

c) <rectangle x1="5" y1="4" x2="1" y2="1"/>

d) <area><x1>4</x1><y1>4</y1><x2>2</x2><y2>3</y2></area>

e) <rectangle><x1>1</x1><y1>1</y1><x2>2</x2><y2>3</y2></rectangle>



**Q78) [CS-UFG UFG 2018]** XML Schema Definition (XSD) é usada para a definição de regras de validação em documentos construídos no formato XML. Ela fornece recursos como namespaces e datatypes para os elementos e atributos. XSD é uma alternativa ao

a) JavaScript.

b) CSS.

c) DOM.

d) DTD.

**Q78) [CS-UFG UFG 2018]** XML Schema Definition (XSD) é usada para a definição de regras de validação em documentos construídos no formato XML. Ela fornece recursos como namespaces e datatypes para os elementos e atributos. XSD é uma alternativa ao

a) JavaScript.

b) CSS.

c) DOM.

d) DTD.

**Q80) [CESGRANRIO PETROBRAS 2018]** Qual linguagem de marcação, fundamental para o estabelecimento de serviços Web, que compõe uma Arquitetura Orientada a Serviços, é usada para que dados sejam apresentados, comunicados e armazenados?

- a) HTML
- b) XML
- c) Java
- d) Javascript
- e) C#

**Q80) [CESGRANRIO PETROBRAS 2018]** Qual linguagem de marcação, fundamental para o estabelecimento de serviços Web, que compõe uma Arquitetura Orientada a Serviços, é usada para que dados sejam apresentados, comunicados e armazenados?

a) HTML

b) XML

c) Java

d) Javascript

e) C#

**Q84) [FGV Câmara de Salvador 2018]** Analise o conteúdo XML de um arquivo de seis linhas, exibido a seguir.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <database catalogo="BD" user="U1">
3  <SQL>
4      select * FROM T where a < 10
5  </SQL>
6  </database>
```

A validação desse arquivo apontaria um erro na linha de número:

- a) 1;
- b) 2;
- c) 3;
- d) 4;
- e) 6.

**Q84) [FGV Câmara de Salvador 2018]** Analise o conteúdo XML de um arquivo de seis linhas, exibido a seguir.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <database catalogo="BD" user="U1">
3  <SQL>
4      select * FROM T where a < 10
5  </SQL>
6  </database>
```

A validação desse arquivo apontaria um erro na linha de número:

- a) 1;
- b) 2;
- c) 3;
- d) 4;
- e) 6.

**Q85) [CESPE STM 2018]** Acerca de XML e SOAP, julgue o próximo item.

O script XML a seguir, que faz referência ao esquema verifica.xsd, está sintaticamente incorreto porque UTF-8 não é suportado no XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<addresses xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:noNamespaceSchemaLocation='verifica.xsd'>
  <endereco>
    <nome>JoaoTestador</nome>
    <rua>CPD Informatica 00</rua>
  </endereco>
```

**Q85) [CESPE STM 2018]** Acerca de XML e SOAP, julgue o próximo item.

O script XML a seguir, que faz referência ao esquema verifica.xsd, está sintaticamente incorreto porque UTF-8 não é suportado no XML. ERRADO.

```
<?xml version="1.0" encoding="UTF-8"?>
<addresses xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation='verifica.xsd'>
  <endereco>
    <nome>JoaoTestador</nome>
    <rua>CPD Informatica 00</rua>
  </endereco>
```



**Q88) [CONSULPLAN Câmara BH 2018]** “É uma alternativa à XML para representar dados; comumente utilizado em aplicativos Ajax. Seus objetos são representados como uma lista de nomes e valores de propriedade entre colchetes. Trata-se de um formato simples que facilita leitura, criação e análise de objetos e, ainda, permite que programas transmitam dados eficientemente pela internet.” Trata-se de:

- a) POJO – Plain Old Java Object.
- b) JSON – JavaScript Object Notation.
- c) SOAP – Simple Object Access Protocol.
- d) REST – Representational State Transfer.

**Q88) [CONSULPLAN Câmara BH 2018]** “É uma alternativa à XML para representar dados; comumente utilizado em aplicativos Ajax. Seus objetos são representados como uma lista de nomes e valores de propriedade entre colchetes. Trata-se de um formato simples que facilita leitura, criação e análise de objetos e, ainda, permite que programas transmitam dados eficientemente pela internet.” Trata-se de:

a) POJO – Plain Old Java Object.

b) JSON – JavaScript Object Notation.

c) SOAP – Simple Object Access Protocol.

d) REST – Representational State Transfer.

**Q91) [FCC DPE AM 2018]** Em uma DTD de um documento XML, os atributos são declarados pela instrução ATTLIST, que possui a sintaxe abaixo.

<!ATTLIST element-name attribute-name attribute-type attribute-value>

Nessa sintaxe, attribute-value pode ser

- a) #REQUIRED, #IMPLIED ou #FIXED.
- b) CDATA, PCDATA ou ANYDATA.
- c) REQUIRED, CHECK ou CONSTRAINT.
- d) #CDATA, #PCDATA ou #DATA.
- e) #ID, #IDREF ou #NOTATION.

**Q91) [FCC DPE AM 2018]** Em uma DTD de um documento XML, os atributos são declarados pela instrução ATTLIST, que possui a sintaxe abaixo.

<!ATTLIST element-name attribute-name attribute-type attribute-value>

Nessa sintaxe, attribute-value pode ser

- a) **#REQUIRED, #IMPLIED ou #FIXED.**
- b) CDATA, PCDATA ou ANYDATA.
- c) REQUIRED, CHECK ou CONSTRAINT.
- d) #CDATA, #PCDATA ou #DATA.
- e) #ID, #IDREF ou #NOTATION.

# Referências

- [http://www.macoratti.net/vb\\_xml2.htm](http://www.macoratti.net/vb_xml2.htm)
- [https://www.w3schools.com/xml/xml\\_schema.asp](https://www.w3schools.com/xml/xml_schema.asp)