

Ejercicios XML y DTD del libro

Contenido

1.	Cuenta el número de elementos del siguiente código XML:.....	2
2.	Contesta a las siguientes preguntas:	2
3.	Dado el siguiente documento XML, donde se almacenan los datos relacionados de una librería con libros de informática, crea el documento DTD asociado.	4
	DTD	4
	¿Qué modificación harías en el documento DTD para que puedan añadirse varios precios, teniendo en cuenta que puede haber libros en papel y online?	4
4.	Pon un ejemplo de los siguientes elementos:	5
	a. Elementos sin atributos con datos.	5
	b. Elementos con atributos y datos.	5
	c. Elementos con atributos sin datos.	5
5.	Crea un XML a partir de este DTD	5

1. Cuenta el número de elementos del siguiente código XML:

```
<item>
  <title>
  <link>http://www.librolm.com/html</link>
  <description>Apuntes agregadores</description>
  <guid>B4BA124D-D535-4830-B28C-E0F0BD49D80E</guid>
  <pubDate>Tue, 26 Dec 2017 17:02:39 GMT</pubDate>
</item>
```

Este código contiene seis elementos: item, title, link, description, guid y pubdate.

2. Contesta a las siguientes preguntas:

- ¿Está bien formado el siguiente XML teniendo en cuenta su esquema?
- ¿Es válido el siguiente XML teniendo en cuenta su esquema?

XML

```
<?xml version="1.0" encoding="iso-8859-1"?>
<bdproductos xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="zax.xsd">
  <producto>
    <nombre>teclado</nombre>
    <código>C001</código>
    <iva>21</iva>
    <categoria>A</categoria>
    <precio>15</precio>
    <descuento>10</descuento>
  </producto>
  <producto>
    <nombre>monitor</nombre>
    <código>C002</código>
    <categoria>B</categoria>
    <iva>21</iva>
    <precio>147.2</precio>
    <descuento>9</descuento>
  </producto>
</bdproductos>
```

XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="bdproductos">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="producto"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="producto">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="nombre"/>
        <xs:element ref="codigo"/>
        <xs:element ref="categoria"/>
        <xs:element ref="iva"/>
        <xs:element ref="precio"/>
        <xs:element ref="descuento"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

<xs:element name="nombre" type="xs:NCName"/>
<xs:element name="codigo" type="Tlongitudentre"/>
<xs:element name="categoria" type="Tcat1"/>
<xs:element name="iva" type="xs:integer"/>
<xs:element name="precio" type="Tdecimal"/>
<xs:element name="descuento" type="Tdescuento"/>
<xs:simpleType name="Tdescuento">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="10"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Tdecimal">
  <xs:restriction base="xs:decimal">
    <xs:totalDigits value="4"/>
    <xs:fractionDigits value="2"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Tcat">
  <xs:restriction base="xs:string">
    <xs:enumeration value="A"/>
    <xs:enumeration value="B"/>
    <xs:enumeration value="C"/>
    <xs:enumeration value="D"/>
    <xs:enumeration value="E"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Tcat1">
  <xs:restriction base="xs:string">
    <xs:pattern value="[A-E]"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Tlongitud">
  <xs:restriction base="xs:string">
    <xs:length value="4"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Tlongitudentre">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="4"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

- a. Sigue la estructura jerárquica según el XSD y todas las etiquetas están correctamente formadas. Por lo que sí, el XML está bien formado.
- b. Para validarlo, deberemos asegurarnos de que se cumplen los tipos de datos declarados:
 - nombre → NCName: Cumple el tipo, no tiene espacios ni caracteres especiales.
 - código → Tlongitudentre: Cumple el formato, su longitud está entre 1 y 4.
 - iva → integer: Los valores son numéricos.
 - categoría → Tcat1: Están delimitados correctamente entre A y B.
 - precio → Tdecimal: Los valores están delimitados en el formato.
 - descuento → Tdescuento: Se encuentran entre 1 y 10.

Además, los elementos aparecen en el orden en el que se han especificado, así que sí, es correcto.

3. Dado el siguiente documento XML, donde se almacenan los datos relacionados de una librería con libros de informática, crea el documento DTD asociado.

XML

```
<?xml version="1.0" encoding="iso-8859-1"?>
<librería>
  <libro>
    <título>Lenguaje de marcas</título>
    <autor>Autor2</autor>
    <autor>Autor1</autor>
    <ISBN></ISBN>
    <páginas>158</páginas>
    <precio>15</precio>
  </libro>
  <libro>
    <título>XML</título>
    <autor>Autor1</autor>
    <ISBN></ISBN>
    <páginas>278</páginas>
    <precio>30</precio>
  </libro>
</librería>
```

DTD

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!ELEMENT librería (libro+)>
<!ELEMENT libro (título, autor+, ISBN, páginas, precio)>
<!ELEMENT título (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT ISBN (#PCDATA)>
<!ELEMENT páginas (#PCDATA)>
<!ELEMENT precio (#PCDATA)>
```

¿Qué modificación harías en el documento DTD para que puedan añadirse varios precios, teniendo en cuenta que puede haber libros en papel y online?

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!ELEMENT librería (libro+)>
<!ELEMENT libro (título, autor+, ISBN, páginas, precio, precioOnline?)>
<!ELEMENT título (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT ISBN (#PCDATA)>
<!ELEMENT páginas (#PCDATA)>
<!ELEMENT precio (#PCDATA)>
<!ELEMENT precioOnline (#PCDATA)>
```

4. Pon un ejemplo de los siguientes elementos:

a. Elementos sin atributos con datos.

```
| <nombre>Pepito</nombre>
```

b. Elementos con atributos y datos.

```
| <libro categoría="Suspense">Cómo hacer la declaración de la renta</libro>
```

c. Elementos con atributos sin datos.

```
| <imagen enlace="imgs/imagen.png"/ >
```

5. Crea un XML a partir de este DTD

```
| <!-- ELEMENT arbol (especie)+>
| <!-- ELEMENT especie (nombre, variedad, (origen)+,color_fruto, (maduración)+,
| (riego)?, (precio)?>
| <!-- ELEMENT nombre (#PCDATA)>
| <!-- ELEMENT variedad (#PCDATA)>
| <!-- ELEMENT origen (#PCDATA)>
| <!-- ELEMENT color_fruto (#PCDATA)>
| <!-- ELEMENT maduración (#PCDATA)>
| <!-- ELEMENT riego (#PCDATA)>
| <!-- ELEMENT precio (#PCDATA)>
```

XML

```
| <arbol>
|   <especie>
|     <nombre>Limonero</nombre>
|     <variedad>Una</variedad>
|     <origen>Italia</origen>
|     <color_fruto>Amarillo</color_fruto>
|     <maduración>16</maduración>
|     <riego>Abundante</riego>
|     <precio>40</precio>
|   </especie>
| </arbol>
```

Aparece una especie, cumpliendo el (especie)+, también aparecen riego y precio aunque son opcionales (riego)?, (precio)?.