

# Tarea para LMSGI04

Alumno: Derimán Tejera Fumero

Fecha: 26/11/2022

## ÍNDICE

Ejercicio 1 .....	2
Ejercicio 2 .....	5

## Ejercicio 1

1.- Una empresa utiliza ficheros XML para realizar el inventario, entre otras cosas, de impresoras. Para cada impresora se quiere guardar su número de serie, marca, modelo, peso, tamaños de papel con los que puede trabajar, el cartucho que usa, el tipo, el año de compra y si está en red o no.

Un ejemplo de estos ficheros sería:

```
1  <impresoras>
2    <impresora numSerie="i245" tipo="láser" compra="2010">
3      <marca>Epson</marca>
4      <modelo>EPL300</modelo>
5      <peso>4.52</peso>
6      <tamaño>A4</tamaño>
7      <tamaño>A5</tamaño>
8      <cartucho>C-123BV</cartucho>
9      <enred/>
10   </impresora>
11   <impresora numSerie="i246" tipo="matricial">
12     <marca>HP</marca>
13     <modelo>LaserJet 2410</modelo>
14     <peso>3.2</peso>
15     <tamaño>A4</tamaño>
16     <cartucho>C-456P</cartucho>
17   </impresora>
18 </impresoras>
```

Escribe un XML Schema para estos ficheros, eligiendo los tipos de datos más apropiados y teniendo en cuenta que:

- El peso es un número positivo y no puede tener más de dos decimales.
- Puede haber más de un tamaño.
- El atributo tipo sólo puede tomar los valores 'matricial', 'láser' y 'tinta'. Es obligatorio.
- El atributo numSerie es obligatorio y hace de identificador.
- El cartucho está formado por una C mayúscula, un guión, tres números y una o dos letras mayúsculas.
- Todos los elementos son obligatorios, menos enred.
- El elemento enred es opcional. De estar presente, es un elemento vacío.
- El atributo compra, opcional, guarda el año de compra. Es un número entero positivo.
- En cada fichero hay datos para una o más impresoras.

El primer archivo:

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <impresoras
4      xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
5      xs:noNamespaceSchemaLocation="tejera_fumero_deriman_impresoras.xsd">
6
7      <impresora numSerie="i245" tipo="láser" compra="2010">
8          <marca>Epson</marca>
9          <modelo>EPL300</modelo>
10         <peso>4.52</peso>
11         <tamaño>A4</tamaño>
12         <tamaño>A5</tamaño>
13         <cartucho>C-123BV</cartucho>
14         <enred/>
15     </impresora>
16     <impresora numSerie="i246" tipo="matricial">
17         <marca>HP</marca>
18         <modelo>LaserJet 2410</modelo>
19         <peso>3.2</peso>
20         <tamaño>A4</tamaño>
21         <cartucho>C-456P</cartucho>
22         <enred></enred>
23     </impresora>
24 </impresoras>
25
26

```

El segundo archivo:

```

1  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
2      <!-- Comienzo creando el elemento "impresoras" que a su vez contendrá 1 o mas elementos "impresora" -->
3      <xs:element name="impresoras">
4          <xs:complexType>
5              <xs:sequence>
6                  <xs:element ref="impresora" minOccurs="1" maxOccurs="unbounded"/> <!-- Utilizo ref en lugar de name ya que en realidad se hace referencia al elemento "impresora" definido mas abajo -->
7              </xs:sequence>
8          </xs:complexType>
9      </xs:element>
10
11      <!-- Aquí se define el elemento "impresora" y dentro cada uno de los elementos que lo componen: "marca", "modelo", "peso", "tamaño", "cartucho", "enred"... -->
12      <xs:element name="impresora">
13          <xs:complexType>
14              <xs:sequence>
15                  <xs:element name="marca" type="xs:string" minOccurs="1" maxOccurs="unbounded"/> <!-- Al especificar minOccurs="1" estamos obligando a que al menos aparezca 1 vez, con lo cual estamos haciendo que sea obligatorio su aparición -->
16                  <xs:element name="modelo" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>
17                  <xs:element name="peso" minOccurs="1" maxOccurs="unbounded"/> <!-- Peso debe aparecer el menos una vez -->
18                  <xs:simpleType>
19                      <xs:restriction base="xs:decimal"/> <!-- Debe decimal -->
20                      <xs:fractionDigits value="2"/> <!-- Debe tener 2 cifras -->
21                      <xs:minInclusive value="1"/> <!-- Debe ser un número positivo -->
22                  </xs:restriction>
23                  <xs:simpleType>
24                      <xs:restriction base="xs:string"/>
25                      <xs:pattern value="[C]([A-Z]{1,2})"/> <!-- Expresión regular que define: El cartucho está formado por una C mayúscula, un guión, tres números y una o dos letras mayúsculas -->
26                  </xs:restriction>
27                  <xs:simpleType>
28                      <xs:restriction base="xs:string" minOccurs="1" maxOccurs="unbounded"/>
29                  </xs:restriction>
30                  <xs:simpleType>
31                      <xs:restriction base="xs:string"/>
32                      <xs:pattern value="[C]([A-Z]{1,2})"/> <!-- Expresión regular que define: El cartucho está formado por una C mayúscula, un guión, tres números y una o dos letras mayúsculas -->
33                  </xs:restriction>
34                  <xs:simpleType>
35                      <xs:restriction base="xs:string" default="" minOccurs="0" maxOccurs="unbounded"/> <!-- El elemento enred es opcional. De estar presente, es un elemento vacío, por eso default="" -->
36                  </xs:restriction>
37              </xs:sequence>
38          </xs:complexType>
39      </xs:element>
40
41      <xs:attribute name="numSerie" use="required"/> <!-- Aplico: use="required" porque es obligatorio que aparezca -->
42      <xs:simpleType>
43          <xs:restriction base="xs:ID"/> <!-- El atributo numSerie es obligatorio y hace de identificador, por eso utilizo xs:ID, ya que no puede haber otro igual en el XML -->
44      </xs:restriction>
45      <xs:simpleType>
46          <xs:restriction base="xs:string" default="" minOccurs="0" maxOccurs="unbounded"/>
47      </xs:restriction>
48      <xs:attribute>

```

```

49 |
50 | <xs:attribute name="tipo" use="required">!--El atributo "tipo", sólo puede tomar uno de los valores indicados: "matricial", "láser", "tinta".-->
51 | <xs:simpleType>
52 | <xs:restriction base="xs:string">
53 | <xs:enumeration value="matricial"/>
54 | <xs:enumeration value="láser"/>
55 | <xs:enumeration value="tinta"/>
56 | </xs:restriction>
57 | </xs:simpleType>
58 | </xs:attribute>
59 |
60 | <xs:attribute name="compra" type="xs:positiveInteger" use="optional"/>!--El atributo "compra" debe ser un número positivo, además, es opcional su aparición.-->
61 |
62 | </xs:complexType>
63 | </xs:element>
64 |
65 | </xs:schema>
66 |

```

#### Información



impresoras.xml es bien-formado

#### Información



impresoras.xml is valid

#### Explicación:

Por motivos de legibilidad, la explicación de lo que ocurre en el .xsd y las decisiones tomadas se han hecho con comentarios en el propio archivo.

## Ejercicio 2

2.- Un asociación de jugadores de mus utiliza ficheros XML para almacenar la información de los torneos que organiza. Para cada torneo se quiere guardar el año en que se realizó, el anterior ganador (su id) y la información de los participantes. De los participantes se quiere almacenar su identificador, el de su pareja, su nombre completo, edad, país y también si es cabeza de serie o no.

Un ejemplo de estos ficheros sería:

```
1  <torneo edicion="1998" anteriorGanador="j01">
2    <participante idP="j01" pareja="j02">
3      <nombre>Manuel Pérez</nombre>
4      <edad>23</edad>
5      <pais> España</pais>
6      <cabezaDeSerie/>
7    </participante>
8    <participante idP="j02" pareja="j01">
9      <nombre>Manuel Gómez</nombre>
10     <edad>25</edad>
11     <pais>España</pais>
12   </participante>
13   <participante idP="j03" pareja="j04">
14     <nombre>Ana Puertas</nombre>
15     <edad>22</edad>
16     <pais> E5paña</pais>
17     <cabezaDeSerie/>
18   </participante>
19   <participante idP="j04" pareja="j03">
20     <nombre>Paco Fraile</nombre>
21     <edad>45</edad>
22     <pais>España</pais>
23   </participante>
24 </torneo>
```

Escribe un DTD adecuado para estos ficheros, teniendo en cuenta que:

- Todos los atributos son obligatorios.
- El atributo idP hace de identificador.
- El atributo pareja es el identificador de otro participante que forzosamente debe estar presente en el fichero.
- El elemento cabezaDeSerie es opcional, los demás son obligatorios.
- En cada fichero hay datos para un solo torneo.
- El torneo tiene que tener participantes.

Primer archivo:

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE torneo SYSTEM "tejera_fumero_deriman_torneo.dtd">
3
4  <torneo edicion="1998" anteriorGanador="j01">
5    <participante idP="j01" pareja="j02">
6      <nombre>Manuel Pérez</nombre>
7      <edad>23</edad>
8      <pais> España</pais>
9      <cabezaDeSerie/>
10   </participante>
11   <participante idP="j02" pareja="j01">
12     <nombre>Manuel Gómez</nombre>
13     <edad>25</edad>
14     <pais>España</pais>
15   </participante>
16   <participante idP="j03" pareja="j04">
17     <nombre>Ana Puertas</nombre>
18     <edad>22</edad>
19     <pais> E5paña</pais>
20     <cabezaDeSerie/>
21   </participante>
22   <participante idP="j04" pareja="j03">
23     <nombre>Paco Fraile</nombre>
24     <edad>45</edad>
25     <pais>España</pais>
26   </participante>
27 </torneo>

```

Segundo archivo:

```

1  <ELEMENT torneo (participante+)><!--Creo el elemento raíz "torneo", que contendrá el elemento "participante" que aparecerá 1 o mas veces (+)-->
2  <IA TLIST torneo edicion CDATA #REQUIRED><!--Existe un atributo "edicion" en el elemento "torneo" con poseerá datos de tipo cadena de caracteres -->
3  <IA TLIST torneo anteriorGanador CDATA #REQUIRED><!--Existe un atributo "anteriorGanador" en el elemento "torneo" con poseerá datos de tipo cadena de caracteres -->
4  <ELEMENT participante (nombre+, edad+, pais+, cabezaDeSerie?)><!--El elemento "participante" está compuesto a su vez por los elementos: "nombre"(que aparece una o mas veces), "edad" etc. -->
5  <IA TLIST participante idP ID #REQUIRED><!--Existe un atributo "idP" en el elemento "participante" que es de tipo ID (es único) y además es obligatorio -->
6  <IA TLIST participante pareja IDREF #REQUIRED><!--Existe un atributo "pareja" en el elemento "participante" que es de tipo IDREF (ya que hace referencia a un ID ya existente) y además es obligatorio -->
7  <ELEMENT nombre (#PCDATA)><!--Existe un elemento "nombre" que poseerá datos de tipo cadena de caracteres -->
8  <ELEMENT edad (#PCDATA)><!--Existe un elemento "edad" que poseerá datos de tipo cadena de caracteres -->
9  <ELEMENT pais (#PCDATA)><!--Existe un elemento "pais" que poseerá datos de tipo cadena de caracteres -->
10 <ELEMENT cabezaDeSerie (#PCDATA)><!--Existe un elemento "cabezaDeSerie" que poseerá datos de tipo cadena de caracteres -->

```

#### Información



torneo.xml es bien-formado

#### Información



torneo.xml is valid

### Explicación:

Por motivos de legibilidad, la mayor parte de la explicación de lo que ocurre en el .dtd y las decisiones tomadas se han hecho con comentarios en el propio archivo.

En el archivo .xml específico dónde puede encontrar el archivo asociado .dtd, al estar en la misma carpeta, con añadir simplemente el nombre ya es suficiente.

En el archivo .dtd, defino sus elementos y sus hijos por orden, luego el tipo de dato a almacenar, en la mayoría de casos #PCDATA.

En los atributos además, defino aquellos datos que son requeridos.