

EXAMEN XML, DTD y XSD

1. Diseña un XML bien formado que permita estructurar la información de una tienda de ropa. Se deberá mostrar la siguiente información:
 - Descripción
 - Tipo: cada artículo puede tener un solo tipo.
 - Color: un artículo tiene un solo color.
 - Precio: es el mismo para todas las tallas.
 - Origen (*made in): un origen para cada artículo.
 - Talla: el mismo artículo puede tener varias tallas diferentes.
 - Stock: para cada talla puede haber un número diferente de unidades.

Por ejemplo, para unos vaqueros azules puedes tener:

Talla S: 5 unidades

Talla M: 8 unidades

Talla L: 3 unidades

- a. Crea una estructura XML lo más compacta posible en un documento evitando dar información redundante e incluyendo datos para 1 artículo de ejemplo. Incluye al menos 2 atributos donde veas más conveniente y escribe al menos un comentario donde veas necesidad de aclarar la composición del documento. **(1,5 puntos)**
 - b. Dibuja una estructura de árbol para el ejemplo anterior. **(0,5 puntos)**
2. Corrige los errores que encuentres en los siguientes documentos para convertirlos en documentos XML bien formados:
 - a. **(0,75 puntos)**

```
<?xml version=1.0 encoding="ISO-8859-1"?>
<catalog>
  <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985
  <cd>
    <title/>Greatest Hits</title>
    <artist>Dolly Parton</artist>
    <country>USA</country>
    <company>RCA
    <price>9.90</price>
    <year>1982</year>
  </company>
</cd>
</catalog>
```

b. (0,75 puntos)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<comedor>
  <mesa tipo="redonda" madera="arce">
    <fabricante>Muebles Albacete</fabricante>
    <precio>40000</precio>
    <silla madera=arce>
      <cantidad>2</cantidad>
      <calidad>excelente</calidad>
      <cojín incluido="sí">
        <color>azul</color>
      </cojín>
    <silla madera="roble">
      <cantidad>3</cantidad>
      <calidad>normal</calidad>
    </silla>
  </comedor>
```

3. Dada la estructura XML en el documento bills.txt del directorio del examen, ten en cuenta las siguientes indicaciones:

- El documento debe albergar de 0 a ilimitadas “factura”
- Cada “factura” tiene un identificador (id) obligatorio y único, de tipo entero mayor a 0
- Dentro de cada “factura” tendremos un “recibo” y una o varias “entrega”
- Dentro del “recibo” puede haber un número ilimitado (al menos uno) “producto” y un código del receptor al final (cod_receptor)
- El producto puede o no tener considerarse “fragil” que indicará si el producto se debe manejar con más cuidados, dicha etiqueta aparecerá vacía en caso de que se considere en esta categoría.
- La cantidad indicará en qué unidad se mide el producto, pudiendo ser por unidades (unidad) o por peso (kg). Se indicará en la etiqueta “total” pudiendo ser un número decimal con máximo 2 cifras decimales.
- Cada entrega constará de la “distancia” desde el almacén y el “coste” del pedido, siendo el coste un número decimal con máximo 2 cifras decimales. Respecto a la distancia, por temas de logística no podrá sobrepasar los 999 km.

- a. Utiliza el documento bills_a.xml para con la estructura de bills.txt diseñar un documento XML con DTD interna para validar su estructura. **(1,75 puntos)**
- b. Utiliza el documento bills_b.xml para contener la estructura de bills.txt, luego diseñar un XSD llamado bills_b.xsd y asociarlo a dicho XML para validar su estructura. **(1,75 puntos)**

4. El siguiente documento no es válido porque contiene algún error (los errores están en el XML, no en la DTD interna). Propón una solución al XML. **(1 punto)**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE musicos [
  <!ELEMENT musicos (musico*)>
  <!ELEMENT musico ((nombre | apodo), fechaNacimiento)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT apodo (#PCDATA)>
  <!ELEMENT fechaNacimiento (#PCDATA)>
]>

<musicos>
  <musico>
    <nombre>Antonio Vivaldi</nombre>
    <apodo>El cura pelirrojillo</apodo>
    <fechaNacimiento>4 de marzo de 1678</fechaNacimiento>
  </musico>
  <musico>
    <nombre>Johann Sebastian Bach</nombre>
    <apodo>El viejo peluca</apodo>
    <fechaNacimiento>21 de marzo de 1685</fechaNacimiento>
  </musico>
</musicos>
```

5. El siguiente documento no es válido porque contiene algún error en la DTD interna. Damos por correcto los datos y estructura del XML. Propón una solución al DTD. **(1 punto)**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paises [
  <!ELEMENT pais (nombre, unionEuropea, otan)>
  <!ELEMENT nombre EMPTY>
  <!ELEMENT unionEuropea EMPTY>
  <!ELEMENT otan EMPTY>
]>

<paises>
  <pais>
    <nombre>España</nombre>
    <unionEuropea />
    <otan />
  </pais>
  <pais>
    <nombre>Noruega</nombre>
    <otan />
  </pais>
  <pais>
    <nombre>Austria</nombre>
    <unionEuropea />
  </pais>
</paises>
```

6. Completa el código que define el elemento **alfa_code** que cumpla lo siguiente:

Tanto el atributo **serial_num** como el elemento **codigo** utilizan la misma restricción que solamente les permite tomar un código con las siguientes características. Constará de un número de 3 cifras con valor menor a 300, es decir, de - 000 a 299, expresado siempre con 3 cifras, seguido de un guión y dos letras mayúsculas que marcarán el país de referencia de dicho elemento solo si es fuera de la UE. Ejemplos válidos:

000, 190, 200-UK, 111-CN, 003-AU...

(1 punto)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="proveedores">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="proveedor" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="codigo" type="alfa_code" />
              <xs:element name="nombre" type="xs:string" />
            </xs:sequence>
            <xs:attribute name="serial_num" type="alfa_code" />
            <xs:attribute name="letra" type="xs:string"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <!--DEFINE AQUÍ EL ELEMENTO alfa_code-->

  <xs:complexType _____
    <xs:restriction _____
      <xs:pattern value= _____
    </_____
  </xs:_____

  <!--DEFINE AQUÍ EL ELEMENTO alfa_code -->

</xs:schema>
```