

## UT5: Ejercicios XML-XSD

### Ejercicio 1 - Pedido

CREACIÓN DE UN XSD: A partir del siguiente XML que representa el envío de un pedido (shiporder) de libros para un cliente, crea un XSD capaz de validar los siguientes aspectos:

- Cada pedido se identifica por un id único llamado orderId y tiene 3 elementos: el nombre del cliente, la dirección de envío y los artículos que viajan en el envío.
- El nombre es texto simple.
- La dirección estará compuesta por el nombre de la persona a la que se entregará el pedido, la dirección, la localidad y el país.
- Se podrán enviar varios libros en cada pedido indicando la cantidad y el precio incluido en el envío. Opcionalmente se podrá incluir una nota en cada uno de los libros enviados.

Ejemplo XML válido:

```
<?xml version="1.0" encoding="UTF-8"?>
<shiporder orderId="ID889923" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="shiporder.xsd">
  <orderperson>John Smith</orderperson>
  <shipto>
    <name>Ola Nordmann</name>
    <address>Langgt 23</address>
    <city>4000 Stavanger</city>
    <country>Norway</country>
  </shipto>
  <item>
    <title>Empire Burlesque</title>
    <note>Special Edition</note>
    <quantity>1</quantity>
    <price>10.90</price>
  </item>
  <item>
    <title>Hide your heart</title>
    <quantity>1</quantity>
    <price>9.90</price>
  </item>
</shiporder>
```

## Ejercicio 2 - Planes de vuelo

Si para representar la siguiente información ficticia:

PANEL DE VUELOS DE SALIDA DEL AEROPUERTO JFK – 20/12/2013						
Código	Diario	Origen	Destino	Hora salida	Hora llegada	Estado
V22	SI	New York	Chicago	9:30	11:30	R
V23	NO	New York	Miami	10:15	11:15	C

se ha escrito el siguiente documento XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<aeropuerto xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="aeropuerto.xsd">
  <nombre>JFK</nombre>
  <vuelos>
    <vuelo código="V22" estado="R">
      <diario />
      <origen>New York</origen>
      <destino>Chicago</destino>
      <hora-salida>09:30:00</hora-salida>
      <hora-llegada>11:30:00</hora-llegada>
    </vuelo>
    <vuelo código="V23" estado="C">
      <origen>New York</origen>
      <destino>Miami</destino>
      <hora-salida>10:15:00</hora-salida>
      <hora-llegada>11:15:00</hora-llegada>
    </vuelo>
  </vuelos>
  <fecha>2013-12-20</fecha>
</aeropuerto>
```

Escribir el código del archivo "aeropuerto.xsd" que permita validarlo, teniendo en cuenta que:

- No debe utilizarse ni group ni attributeGroup.
- El nombre del aeropuerto, los vuelos y la fecha pueden aparecer en distinto orden.
- Se tiene que indicar que el código ha de ser único (esto se puede hacer definiéndolo de tipo xs:ID) y obligatorio para cada vuelo.

- Haciendo uso pattern indicar que los posibles estados de un vuelo son C (Cancelado), E (En hora), R (Retrasado). Dicha restricción sólo debe poder ser utilizada por el atributo estado. El valor por defecto debe ser E.
- Debe permitirse aparecer desde cero hasta ilimitados elementos vuelo y, para cada uno de ellos, se tiene que guardar la información en el mismo orden en el que aparece en el panel.
- Para indicar si un vuelo es diario, se debe utilizar un elemento vacío que, respecto a cada vuelo, podrá aparecer (en el caso de sí ser diario) o no aparecer (en el caso contrario).
- Respecto a los elementos nombre, origen, destino, hora-llegada, hora-salida y fecha, cada uno de ellos debe definirse del tipo que se considere más apropiado, de entre los proporcionados por XML Schema.

### Ejercicio 3 - Factura

Escribir un documento XML con un XSD que represente la información contenida en la siguiente factura ficticia:

FACTURA NÚMERO 19 – FECHA: 23/02/2021					
DATOS EMISOR:			DATOS CLIENTE:		
Librería Coslada			IES Luis Braille		
CIF: 12345678B			CIF: 87654321A		
Teléfono: 666000000			Teléfono: 606066666		
DETALLE FACTURA:					
CÓDIGO-ARTÍCULO	TIPO	DESCRIPCIÓN	CANTIDAD	OFERTA	PVP
AW7	Libro	Aplicaciones Web	2	SI	25€
LM5	PDF	Lenguaje de Marcas	1	NO	30€
IMPORTE:					85€

Escribir el código del archivo "factura.xsd" que permita validarlo, teniendo en cuenta que:

- Exceptuando los elementos datos-emisor, datos-cliente y detalle-factura, que no tienen por qué aparecer en este orden, el resto de elementos representados en el documento XML, sí deben escribirse obligatoriamente en el orden en el que aparecen.

- Excepto para los hijos directos del elemento factura, siempre que sea posible agrupar al menos dos elementos o dos atributos, se debe usar group o attributeGroup, respectivamente.
- Respecto al número de la factura (que debe ser un valor entero mayor que 0) y su fecha de emisión (de tipo xs:date), hay que indicar que son atributos obligatorios.
- El atributo moneda debe indicarse que es un valor fijo.
- Los nombres del emisor y cliente, así como, la descripción de cada artículo, deben ser del mismo tipo, al que llamaremos tipoTexto, y donde debe indicarse que los valores admitidos para dichos elementos pueden ser cadenas de un mínimo de 5 caracteres y un máximo de 20.
- Haciendo uso pattern hay que indicar que el valor del cif debe estar formado por una cadena de ocho dígitos del 0 al 9, seguidos de un guión "-" y una letra mayúscula de la "A" a la "Z". Dicha restricción sólo debe poder ser utilizada por el elemento cif.
- Haciendo uso pattern hay que indicar que el valor del teléfono debe estar formado por una cadena de nueve dígitos del 0 al 9. Dicha restricción sólo debe poder ser utilizada por el teléfono.
- Al menos tiene que aparecer una línea de detalle y como máximo 15.
- El importe debe indicarse que es obligatorio.
- El importe y el pvp deben ser del mismo tipo, al que llamaremos tipoPrecio, y donde debe indicarse, sin hacer uso de pattern, que los valores admitidos por este tipo pueden ser números decimales mayores que 0, pero no mayores que 999. Además, dichos valores podrán contener cinco dígitos como máximo y, de ellos, sólo dos podrán ser decimales.
- El código del artículo ha de ser único y obligatorio para cada artículo.
- Sin hacer uso pattern indicar que los posibles tipos de un artículo son Libro, DVD o Varios, no permitiéndose otro valor. Para ello, se debe definir un tipo de dato llamado tipoArtículo, que debe poder ser utilizado por otros atributos o elementos. Ahora bien, hay que tener en cuenta que este atributo es opcional.
- La cantidad de artículos indicada en cada línea, debe ser un valor entero mayor que 0.
- Para indicar si un artículo está de oferta, se debe utilizar un elemento vacío que, respecto a cada artículo, podrá aparecer (en el caso de sí estar de oferta) o no aparecer (en el caso contrario).
- No hay que definir más tipos de datos que los especificados en el ejercicio: tipoTexto, tipoArtículo y tipoPrecio.

## Ejercicio 4 - Conexiones

Escribe el XML y el XSD.

REGISTRO DE CONEXIONES DE USUARIOS Y EMPLEADOS DE UNA EMPRESA	
<b>USUARIOS:</b>	
IDENTIFICADOR NOMBRE APELLIDOS EMAIL	CONEXIONES (FECHA HORA TIEMPO)
U123&Ana&Sanz Tapia&asanz@jmail.com	2014-02-23&19:15:40&122 2014-02-23&20:30:22&617 2014-02-24&11:18:31&25
U96&Pedro&Ruiz Hierro&pruiz@jotmail.com	2014-02-25&20:33:55&390
<b>EMPLEADOS:</b>	
IDENTIFICADOR NOMBRE APELLIDOS EMAIL DEPARTAMENTO	CONEXIONES (FECHA HORA TIEMPO)
E4&Marta&Vera Gil&mvera@yajoo.es&Marketing	(Ninguna)

Escribir el código del archivo "registro.xsd" que permita validarlo, teniendo en cuenta que:

- Todos los elementos y atributos son obligatorios, a menos que se indique lo contrario.
- Siempre que sea posible agrupar al menos dos elementos o dos atributos, se debe utilizar group o attributeGroup, respectivamente.
- Pueden aparecer de cero a ilimitados usuarios y, a continuación, de cero a ilimitados empleados.
- Usuario debe ser de un tipo definido por nosotros al que llamaremos tipoPersona, donde hay que definir los elementos apellidos-y-nombre, email y conexiones. Por otro lado, empleado ha de ser de otro tipo llamado tipoPersonaAmpliado, definido como una extensión de tipoPersona añadiéndosele el elemento departamento. El orden en que tienen que aparecer los elementos hijo de usuario y empleado, debe ser obligatoriamente el escrito en el documento XML.
- El valor del identificador debe ser una cadena formada por una letra "U" o "E" mayúscula, seguida de uno a cinco dígitos del 0 al 9.
- El valor del elemento apellidos-y-nombre debe ser una cadena de entre 1 a 30 caracteres (de la "a" a la "z", mayúsculas o minúsculas, o el carácter espacio en blanco) para los apellidos, seguida del carácter coma ",", y de entre 1 a otras 20 letras (de la "a" a la "z", también mayúsculas o minúsculas, o el carácter espacio en blanco) para el nombre.

- El valor del email puede ser una cadena formada por 1 a 15 caracteres de la "a" a la "z", seguida del carácter "@", seguido de entre 1 a otras 25 letras de la "a" a la "z", seguidas del carácter punto "." y de entre otras 2 a 4 letras de la "a" a la "z".
- De cada usuario y empleado se reflejan sus conexiones, indicando para cada conexión la cantidad de segundos que duró, que debe ser un número entero mayor que cero. Hay que tener en cuenta que, como se puede ver en el documento XML, pueden aparecer desde cero hasta ilimitados elementos conexión.
- Respecto a los atributos fecha y hora, cada uno de ellos debe definirse del tipo que se considere más apropiado, de entre los proporcionados por XML Schema.
- Los posibles departamentos de la empresa a los que puede pertenecer un empleado son administración, informática o marketing. De tal forma que, para cada empleado, sólo uno de ellos debe escribirse en el documento XML mediante un elemento vacío, como en este caso se ha escrito <marketing />.
- No hay que definir más tipos de datos que los indicados en el ejercicio: tipoPersona y tipoPersonaAmpliado.

## Ejercicio 5 - Colegio

A través del XML proporcionado deberás crear un XSD, donde deberás indicar las características del mismo

```
<?xml version="1.0" encoding="UTF-8"?>
<colegio xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Ejercicio28.xsd">
  <alumno codigo="A100" nombre="Miguel" apellido1="Alcantara" apellido2="Alejo">
    <padre>P100</padre>
    <madre>M100</madre>
    <hermano>A125</hermano>
    <hermano>A232</hermano>
    <curso>C01</curso>
    <nacimiento>2010-04-10</nacimiento>
  </alumno>
  <alumno codigo="A125" nombre="Eduardo" apellido1="Alcantara" apellido2="Alejo">
    <padre>P100</padre>
    <madre>M100</madre>
    <hermano>A100</hermano>
    <hermano>A232</hermano>
    <curso>C03</curso>
    <nacimiento>2008-05-11</nacimiento>
  </alumno>
  <alumno codigo="A232" nombre="Sonia" apellido1="Alcantara" apellido2="Alejo">
    <padre>P100</padre>
    <madre>M100</madre>
    <hermano>A100</hermano>
    <hermano>A125</hermano>
    <curso>C05</curso>
    <nacimiento>2006-08-11</nacimiento>
  </alumno>
  <alumno codigo="A101" nombre="Alicia" apellido1="Pérez" apellido2="López">
    <madre>M101</madre>
    <curso>C01</curso>
    <nacimiento>2010-10-12</nacimiento>
  </alumno>
  <padre>
    <codigo>P100</codigo>
    <nombre>"Miguel"</nombre>
    <apellido1>"Alcantara"</apellido1>
    <apellido2>"García"</apellido2>
    <telefono>654786675</telefono>
  </padre>
  <madre>
    <codigo>M100</codigo>
    <nombre>"Sara"</nombre>
    <apellido1>"Alejo"</apellido1>
    <apellido2>"Hernández"</apellido2>
    <telefono>655211123</telefono>
  </madre>
  <madre>
    <codigo>M101</codigo>
    <nombre>"María Teresa"</nombre>
    <apellido1>"Pérez"</apellido1>
    <apellido2>"Padrones"</apellido2>
    <telefono>623141455</telefono>
  </madre>
  <curso>
    <codigo>C01</codigo>
    <descripcion>Primero de Infantil</descripcion>
  </curso>
  <curso>
    <codigo>C03</codigo>
    <descripcion>Primero</descripcion>
  </curso>
```

```
</curso>  
<curso>  
  <codigo>C05</codigo>  
  <descripcion>Tercero</descripcion>  
</curso>  
</colegio>
```