

RELACIÓN DE EJERCICIOS 3

XML SCHEMA

1. Descárgate el siguiente fichero XML usando la opción "Guardar como..." del navegador:

https://www.w3schools.com/xml/cd_catalog.xml

Crea el XML Schema que describa la gramática de dicho fichero XML.

Verifica que el fichero XML es válido para la gramática que hemos creado.

2. Crea la gramática de un lenguaje basado en XML que permita almacenar datos de los **alumnos** de un **aula**. Un aula tiene como mínimo 2 alumnos y como máximo 30. Usa XML Schema.

De cada alumno del aula se desea almacenar la información que se indica a continuación. Cada alumno tiene:

- Un atributo "dni". Un dni está formado por 8 dígitos y una letra en mayúscula. Es obligatorio.
- Nombre.
- Apellidos.
- Repetidor o No repetidor. Serán dos etiquetas vacías. La segunda etiqueta tendrá el atributo "n_veces" que indica las veces que ha repetido.
- Sexo. Solo puede tomar los valores "hombre" y "mujer".
- Fecha de nacimiento, donde una fecha tiene un día, un mes y año. Los valores de dichos elementos deben ser numéricos.
- Dirección, donde una dirección tiene una calle, número, localidad, provincia. Además, tiene un atributo código postal.
- Teléfono, donde un teléfono tiene un atributo tipo que toma los valores de tipo cadena (fijo, móvil, trabajo, casa, etc.).
- Nota media del ciclo.

Crea a modo de ejemplo un documento XML válido que almacene los datos de dos alumnos. Verifica que el fichero XML creado es válido para la gramática definida.

3. Crea la gramática de un lenguaje basado en XML que permita almacenar datos de las **películas** que hay ahora mismo en una **cartelera**. La cartelera puede tener o no tener películas. Usa XML Schema.

De cada película de la cartelera se desea almacenar la información que se indica a continuación. Cada película tiene:

- Un atributo "estreno" que almacena el año de estreno. Este atributo es opcional.
- Título de película.
- Nacionalidad
- Duración, donde la duración está formada por las etiquetas horas y minutos. La hora tomará valores entre 0 y 5 (no habrá películas que duren más de 6 horas) y los minutos tomarán valores entre 0 y 59.
- Director.

- Actores, donde un actor es:
 - Actor con atributo "tipo" y valor "principal" o "reparto" y atributo "sexo" con valor "hombre" o "mujer":
 - Nombre.
 - Apellidos.
 - Cada película almacena los datos de cómo mínimo un actor.
- Descripción, donde la descripción será un elemento mixto donde se pueden marcar ciertas palabras con la etiqueta "<clave>" para indicar que esa palabra es muy importante en la descripción de la película.

Crea a modo de ejemplo un fichero XML que cumpla dicha gramática.

4. Crea la gramática de un lenguaje basado en XML que permita almacenar datos de los **equipos** de curling que participan en una **competición**. En una competición participan como mínimo 2 equipos y como máximo 10. Usa XML Schema.

De cada equipo se desea almacenar la información que se indica a continuación. Cada equipo tiene:

- Nombre del equipo.
- Nacionalidad.
- Presupuesto. Este elemento es opcional.
- Tipo de equipo. Esto se hará con dos etiquetas sin contenido para indicar si el equipo es un equipo normal de 4 jugadores o doble de 2 jugadores:
"<normal />" o "<doble />"

Además, cada tipo tendrá un atributo "mixto" que tomará los valores "si" o "no" para indicar si dicho equipo es mixto o no.

- Sweepers (o jugadores), donde un sweeper es:
 - Sweeper:
 - Tiene un atributo "dni" que es obligatorio. Un dni está formado por 8 dígitos y una letra.
 - Nombre y apellidos. En una sola etiqueta.
 - Edad. La edad tomará valores entre 18 y 80.
 - Sexo. El sexo tomará el valor "hombre" o "mujer".

Cada equipo está formado por un sweeper como mínimo y por ocho como máximo.

- Pista. Con un atributo asientos que indica la capacidad de la pista.
 - Nombre.
 - Dirección:
 - Calle.
 - Número.
 - Población. Donde población está formada por:
 - CP: este elemento es opcional, y tiene un total de 5 dígitos.
 - Provincia.
- Fecha de fundación. Será una cadena con el siguiente formato DD-MM-YYYY.

Crea a modo de ejemplo un fichero XML que cumpla dicha gramática.

5. Crea la gramática de un lenguaje basado en XML que permita almacenar datos de los **cursos** que se imparten en una determinada **escuela** . Usa XML Schema. La escuela imparte como mínimo un curso y como máximo no tiene definido ningún límite.

De cada curso de la escuela se desea almacenar la información que se indica a continuación. Cada curso tiene:

- Un atributo "identificador" que almacena el código del curso.
- Nombre del curso.
- Duración en horas del curso. Un curso tendrá como mínimo 25 horas y como máximo 2000 horas.
- Descripción.
- Opción a beca. Esto se hará con dos etiquetas sin contenido para indicar si el curso permite solicitar una beca o no:
"<becas />" o "<no_becas />"
Si se puede solicitar beca si indicará con un atributo "institución" que institución ofrece la beca.
- Aula con atributo "informática" y valor "si" o "no". El aula estará formada por 3 números. El primer número indicará la planta y el otros dos el número de aula.
- Precio.
- Semanas, donde una semana es:
 - Semana con atributo "horario" y valor "m" o "t" para indicar mañana o tarde:
 - Inicio: Será una cadena con el formato DD-MM-YYYY.
 - Fin: Será una cadena con el formato DD-MM-YYYY.

Cada curso se imparte como mínimo una vez y como máximo tres.

- Profesores, donde un profesor es:
 - Profesor con atributo "tipo" y valor "teoría" o "prácticas":
 - Nombre.
 - Apellidos.

Cada curso almacena los datos de cómo mínimo dos profesores.

- Temas, donde un tema es:
 - Tema con atributo "numero" para almacenar el número del tema y el atributo "cod_curso" que tendrá el valor del código del curso. Ambos atributos son obligatorios.
 - Título del tema.
 - Palabras clave.
 - Resumen donde el resumen será un elemento mixto donde se pueden marcar ciertas palabras con la etiqueta "<clave>" para indicar que esa palabra es muy importante en el resumen del tema.

Cada curso tiene como mínimo un tema.

Crea a modo de ejemplo un fichero XML que cumpla dicha gramática.

6. Dado el siguiente fichero XML:

```
1      <?xml version="1.0" encoding="UTF-8"?>
2      <shiporder orderid="889923"
3          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4          xsi:noNamespaceSchemaLocation="shiporder.xsd">
5          <orderperson>John Smith</orderperson>
6          <shipto>
7              <name>Ola Nordmann</name>
8              <address>Langgt 23</address>
9              <city>4000 Stavanger</city>
10             <country>Norway</country>
11         </shipto>
12         <item>
13             <title>Empire Burlesque</title>
14             <note>Special Edition</note>
15             <quantity>1</quantity>
16             <price>10.90</price>
17         </item>
18         <item>
19             <title>Hide your heart</title>
20             <quantity>1</quantity>
21             <price>9.90</price>
22         </item>
23     </shiporder>
```

Crea la gramática XSD que describa dicho lenguaje. Indica en el fichero XSD usando comentarios todas las decisiones que tomes. Decisiones del tipo qué atributo es obligatorio y cual es opcional, qué etiquetas deben aparecer obligatoriamente, etc.

Verifica que el fichero XML es válido para la gramática que hemos creado.

7. Dado el siguiente esquema XSD crea un documento XML válido que almacene los datos de dos personas:

```
1      <?xml version="1.0" encoding="UTF-8"?>
2      <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
3          <!-- ===== -->
4          <!-- Definiciones de tipos simples -->
5          <!-- ===== -->
6          <!-- ===== Tipo "tipoDia" ===== -->
7          <xsd:simpleType name="tipoDia">
8              <xsd:restriction base="xsd:positiveInteger">
9                  <xsd:minInclusive value="1"/>
10                 <xsd:maxInclusive value="31"/>
11             </xsd:restriction>
12         </xsd:simpleType>
```

```
13      <!-- ===== Tipo "tipoMes" ===== -->
14      <xsd:simpleType name="tipoMes">
15          <xsd:restriction base="xsd:string">
16              <xsd:enumeration value="Enero"/>|
17              <xsd:enumeration value="Febrero"/>
18              <xsd:enumeration value="Marzo"/>
19              <xsd:enumeration value="Abril"/>
20              <xsd:enumeration value="Mayo"/>
21              <xsd:enumeration value="Junio"/>
22              <xsd:enumeration value="Julio"/>
23              <xsd:enumeration value="Agosto"/>
24              <xsd:enumeration value="Septiembre"/>
25              <xsd:enumeration value="Octubre"/>
26              <xsd:enumeration value="Noviembre"/>
27              <xsd:enumeration value="Diciembre"/>
28          </xsd:restriction>
29      </xsd:simpleType>
30      <!-- ===== Tipo "tipoAnyo" ===== -->
31      <xsd:simpleType name="tipoAnyo">
32          <xsd:restriction base="xsd:positiveInteger">
33              <xsd:minInclusive value="1900"/>
34              <xsd:maxInclusive value="2011"/>
35          </xsd:restriction>
36      </xsd:simpleType>
37      <!-- ===== Tipo "tipoNombre" ===== -->
38      <xsd:simpleType name="tipoNombre">
39          <xsd:restriction base="xsd:string">
40              <xsd:maxLength value="50"/>
41          </xsd:restriction>
42      </xsd:simpleType>
43      <!-- ===== Tipo "tipoCodPostal" ===== -->
44      <xsd:simpleType name="tipoCodPostal">
45          <xsd:restriction base="xsd:positiveInteger">
46              <xsd:totalDigits value="5"/>
47          </xsd:restriction>
48      </xsd:simpleType>
49      <!-- ===== Definiciones de elementos de tipos simples ===== -->
50      <!-- Definiciones de elementos de tipos simples -->
51      <!-- ===== Elementos de tipo string ===== -->
52      <!-- ===== Elementos de tipo string ===== -->
53      <xsd:element name="nombre" type="tipoNombre"/>
54      <xsd:element name="apellidos" type="tipoNombre"/>
55      <xsd:element name="calle" type="tipoNombre"/>
56      <xsd:element name="poblacion" type="tipoNombre"/>
57      <xsd:element name="provincia" type="tipoNombre"/>
58      <!-- ===== Otros elementos de tipo simple ===== -->
59      <xsd:element name="varon"/>
60      <xsd:element name="hembra"/>
61      <xsd:element name="cpostal" type="tipoCodPostal"/>
```

```
62      <!-- ===== Elemento "nacimient" ===== -->
63      <!-- Definiciones de elementos de tipo complejo -->
64      <!-- ===== Elemento "nacimient" ===== -->
65      <!-- ===== Elemento "nacimient" ===== -->
66      <xsd:element name="nacimient">
67          <xsd:complexType>
68              <xsd:simpleContent>
69                  <xsd:extension base="xsd:string">
70                      <xsd:attribute name="dia" type="tipoDia"/>
71                      <xsd:attribute name="mes" type="tipoMes"/>
72                      <xsd:attribute name="anyo" type="tipoAnyo"/>
73                  </xsd:extension>
74              </xsd:simpleContent>
75          </xsd:complexType>
76      </xsd:element>
77      <!-- ===== Elemento "direccion" ===== -->
78      <xsd:element name="direccion">
79          <xsd:complexType>
80              <xsd:sequence>
81                  <xsd:element ref="calle" maxOccurs="unbounded"/>
82                  <xsd:element ref="poblacion" maxOccurs="unbounded"/>
83                  <xsd:element ref="provincia" maxOccurs="unbounded"/>
84                  <xsd:element ref="cpostal" maxOccurs="unbounded"/>
85              </xsd:sequence>
86          </xsd:complexType>
87      </xsd:element>
88      <!-- ===== Elemento "personas" ===== -->
89      <xsd:element name="personas">
90          <xsd:complexType>
91              <xsd:sequence>
92                  <xsd:element ref="persona" maxOccurs="unbounded"/>
93              </xsd:sequence>
94          </xsd:complexType>
95      </xsd:element>
96      <!-- ===== Elemento "persona" ===== -->
97      <xsd:element name="persona">
98          <xsd:complexType>
99              <xsd:sequence>
100                  <xsd:element ref="nombre"/>
101                  <xsd:element ref="nacimient" minOccurs="0" maxOccurs="1"/>
102                  <xsd:element ref="direccion" maxOccurs="unbounded"/>
103                  <xsd:choice>
104                      <xsd:element ref="varon"/>
105                      <xsd:element ref="hembra"/>
106                  </xsd:choice>
107              </xsd:sequence>
108          </xsd:complexType>
109      </xsd:element>
110  </xsd:schema>
```

Verifica que el fichero XML creado es válido para la gramática definida.

Criterios de evaluación

La presente relación de ejercicios sirve para evaluar todos los criterios de evaluación del RA4.

Los ejercicios del 1 al 5 tienen una puntuación de 1 punto cada uno, el 6 y el 7 tienen una puntuación de 2,5 puntos cada uno.

Método de entrega

Nombra todos los ejercicios de tal forma que se identifique de manera clara (p.e.: ejercicio2.dtd) y comprímelos en un solo archivo con el siguiente formato de nombre: Apellidos_Nombre_LMSGI_UD4R3.zip, y súbelo a la tarea correspondiente de la plataforma Moodle.