

Relación de ejercicios: XML Schema

1. Definir un elemento llamado **puertaCerrada** de tipo lógico, que por defecto tenga el valor "**falso**", y otro elemento llamado **ventanaAbierta** también de tipo lógico, que tenga asignado el valor fijo "**verdadero**".
2. Escribir un XML Schema para el siguiente documento XML, e incluir los cambios necesarios en el mismo para referenciar al esquema creado. Se sabe que todos los elementos hijos del elemento "nota" son obligatorios y deben aparecer sólo una vez.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<nota>
  <para>Pedro</para>
  <de>Laura</de>
  <titulo>Recordatorio</titulo>
  <contenido>A las 7:00 pm en la puerta del teatro</contenido>
</nota>
```

3. Escribir un XML Schema para el siguiente documento XML, e incluir los cambios necesarios en el mismo para referenciar al esquema creado

```
<?xml version="1.0" encoding="UTF-8"?>

<listadeproductos>

  <producto>
    <nombre> Galletas integrales</nombre>
    <codigo>1001</codigo>
    <precio>6.45</precio>
    <tipo es_dietetico="si"/>
  </producto>

  <producto>
    <nombre>Cereales con chocolate</nombre>
    <codigo>1002</codigo>
    <precio>3.95</precio>
    <tipo es_dietetico="no"/>
  </producto>

</listadeproductos>
```

4. Validar, utilizando un XML Schema, un documento XML que contiene un registro de temperatura con la siguiente estructura:
- El elemento principal contiene un registro de temperaturas con el nombre de la provincia, la fecha del registro, la temperatura mínima y máxima y una relación de incidencias
 - La fecha de registro debe ser una fecha válida
 - La temperatura mínima no debe ser inferior a -50°C
 - La temperatura máxima no debe ser superior a 50°C
 - El número de incidencias no tiene límite
 - Las incidencias tienen un nombre que puede ser “frio”, “nieve”, “lluvia” o “calor”
 - Las incidencias tienen una severidad que puede ser “alta”, “media” o “baja”
5. Escribir un XML Schema para el siguiente documento XML, e incluir los cambios necesarios en el mismo para referenciar al esquema creado

```
<pedido>
  <cliente>
    <nombre>Juan</nombre>
    <apellidos>Delgado Martínez</apellidos>
    <direccion>calle Barco 4, tercer piso, letra A</direccion>
  </cliente>
  <articulo>bicicleta A2013</articulo>
  <fecha_entrega>19-5-2000</fecha_entrega>
</pedido>
```

6. Escribir un XML Schema para el siguiente documento XML, e incluir los cambios necesarios en el mismo para referenciar al esquema creado

```
<?xml version="1.0" encoding="utf-8"?>
<blog>
  <articulo>
    <referencia>38</referencia>
    <titulo>Bizcocho de chocolate</titulo>
    <fecha>2018-03-01</fecha>
    <publicado>true</publicado>
  </articulo>
  <articulo>
    <referencia>21</referencia>
    <titulo>Tarta de manzana</titulo>
    <fecha>2018-10-01</fecha>
    <publicado>false</publicado>
  </articulo>
</blog>
```

7. Escribir un XML Schema para el siguiente documento XML, e incluir los cambios necesarios en el mismo para referenciar al esquema creado. Se debe cumplir también lo siguiente:
- Todos los elementos que aparecen en el documento son obligatorios y deben aparecer siempre en el mismo orden.
 - Se deben definir dos tipos de datos:
 - “tipo persona” : los elementos “persona” serán de este tipo
 - “info”: los elementos “datos” serán de este tipo
 - El atributo “nacimiento” es de tipo “date”.
 - El elemento “comentario” no es obligatorio; si aparece lo hace sólo 1 vez.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<persona nacimiento="1999-10-20">
  <datos>
    <nombre>Pepe</nombre>
    <apellidos>Garcia</apellidos>
    <dni>25390952</dni>
  </datos>
  <comentario>buena gente...</comentario>
</persona>
```

8. Realiza la validación Schema del siguiente archivo XML:

```
<marcadores>
  <pagina>
    <nombre>Abrirllave</nombre>
    <descripcion>Tutoriales de informática.</descripcion>
    <url>http://www.abrirllave.com/</url>
  </pagina>
  <pagina>
    <nombre>Wikipedia</nombre>
    <descripcion>La enciclopedia libre.</descripcion>
    <url>http://www.wikipedia.org/</url>
  </pagina>
  <pagina>
    <nombre>W3C</nombre>
    <descripcion>World Wide Web Consortium.</descripcion>
    <url>http://www.w3.org/</url>
  </pagina>
</marcadores>
```

9. Para los siguientes elementos:

```
<ciudad>Roma</ciudad>
<fecha-de-nacimiento>1996-12-18</fecha-de-nacimiento>
<hora>18:29:45</hora>
<nota>7.5</nota>
<apto>true</apto>
```

Escribir sus definiciones de elementos simples correspondientes. En una segunda versión solo permitimos notas de [0 a 10] aumentando en 0.5.

10. Modifica el ejercicio 2 y realiza una adecuada validación teniendo en cuenta que:

- a. El código ha de ser un valor entero superior a 1000 e inferior a 5000.
- b. El precio y precio oferta ha de tener 2 dígitos decimales.
- c. El atributo es_dietetico solo admite dos valores "si" y "no". Además es obligatorio

```
<?xml version="1.0" encoding="UTF-8"?>
<listadeproductos>
  <producto>
    <nombre> Galletas integrales</nombre>
    <codigo>1001</codigo>
    <precio>6.45</precio>
    <tipo es_dietetico="si"/>
  </producto>
  <producto>
    <nombre>Cereales con chocolate</nombre>
    <codigo>1002</codigo>
    <precio>3.95</precio>
    <tipo es_dietetico="no"/>
  </producto>
</listadeproductos>
```

11. Dado el siguiente archivo .xml realiza el correspondiente .xsd para validarlo.

```
<?xml version="1.0" encoding="UTF-8"?>
<ubicaciones xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ubicaciones.xsd">
  <ubicacion metros="32">norte</ubicacion>
  <ubicacion metros="25">este</ubicacion>
  <ubicacion metros="64">este</ubicacion>
</ubicaciones>
```

- Crear un **simpleType** llamado "**direccion**" que pueda contener los valores: norte, sur, este y oeste.
- Crear un nuevo elemento **complexType** llamado "infoUbicacion" que amplíe la definición de "direccion" añadiendo el atributo "**metros**" (tendría pues texto y atributo).

12. Escribir un XML Schema para el siguiente documento XML, e incluir los cambios necesarios en el mismo para referenciar al esquema creado. Se debe cumplir también lo siguiente:

- Los elementos “vehículo”, “nombre” y “modificacion” deben aparecer mínimo una vez, y el máximo no está limitado. El resto de los elementos deben aparecer 1 vez.
- Todos los elementos que aparecen son obligatorios y deben aparecer siempre en el mismo orden.
- Los elementos que contienen información de fecha son todos de tipo cadena.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<vehiculos>
  <vehiculo>
    <nombre>Count Zero</nombre>
    <modelo>Series I, 80"</modelo>
    <fabricacion>
      <inicio>
        <dia>21</dia>
        <mes>July</mes>
        <anyo>1949</anyo>
      </inicio>
      <fin>
        <dia>9</dia>
        <mes>August</mes>
        <anyo>1949</anyo>
      </fin>
    </fabricacion>
    <modificaciones>
      <modificacion>Change Engine</modificacion>
      <modificacion>Change pedals</modificacion>
      <modificacion>Change gearbox</modificacion>
      <modificacion>Fit Rollcage</modificacion>
    </modificaciones>
  </vehiculo>
</vehiculos>
```

13. Diseña el esquema XSD para el siguiente archivo XML con las siguientes consideraciones:

- El elemento número será un entero entre 0 y 500.
- El elemento teléfono será un entero y dicho elemento podrá aparecer entre 0 y 5 veces.
- El atributo dni será una plantilla de 8 números y una letra mayúscula.

```
<?xml version="1.0" encoding="UTF-8"?>
<alumno dni="12345678A">
  <nombre>Juan Garcia</nombre>
  <direccion>
    <calle>Avenida de la Fuente</calle>
    <numero>6</numero>
    <ciudad>Zafra</ciudad>
    <provincia>Badajoz</provincia>
  </direccion>
  <telefono>924555555</telefono>
  <telefono>658741236</telefono>
</alumno>
```

14. Escribir un XML Schema para el siguiente documento XML, e incluir los cambios necesarios en el mismo para referenciar al esquema creado.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE matricula SYSTEM "matricula.dtd">
<matricula>
  <personal>
    <dni>99223366M</dni>
    <nombre>Juan Pardo Martín</nombre>
    <titulacion>Ingeniería Informática</titulacion>
    <curso_academico>1997/1998</curso_academico>
    <domicilios>
      <domicilio tipo="familiar">
        <nombre>C/ Principal nº1</nombre>
      </domicilio>
      <domicilio tipo="habitual">
        <nombre>C/ Secundaria nº2</nombre>
      </domicilio>
    </domicilios>
  </personal>
  <pago>
    <tipo_matricula>Matrícula Ordinaria</tipo_matricula>
  </pago>
</matricula>
```

15. Diseña el esquema XSD para el siguiente archivo XML con las siguientes consideraciones:

- Como primer elemento de helado podemos elegir, o bien chocolate, o bien fresa (pero no los dos al mismo tiempo). Estos elementos almacenarán la cantidad en gramos (como un número entero entre 0 y 1000).
- Como segundo elemento de helado se podrá elegir uno de los siguientes vainilla, turrón o nata. Estos elementos almacenarán la cantidad en gramos (como un número entero entre 0 y 1000).
- El atributo fabricación indicará la fecha en la que se fabricó el mismo mediante el tipo de dato fecha (**obligatorio**)

```
<?xml version="1.0" encoding="UTF-8"?>
<heladeria>
  <helado fabricación="2015-01-30">
    <chocolate>250</chocolate>
    <turrón>300</turrón>
  </helado>
  <helado fabricación="2015-01-30">
    <fresa>500</fresa>
    <vainilla>200</vainilla>
  </helado>
</heladeria>
```

16. Dado el fichero fichas.xsd escribe un archivo .xml válido.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="fichas">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ficha" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="nombre" type="xs:string"/>
              <xs:element name="iniciales">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:pattern value="[A-Z][A-Z][A-Z]"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="edad" type="xs:integer"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

17. Dado el siguiente documento XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<fichas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fichas.xsd">
  <ficha numero="01" letra="Z">
    <codigo>11</codigo>
    <nombre>Ana Sanz Tin</nombre>
  </ficha>
  <ficha numero="02">
    <codigo>12</codigo>
    <nombre>Iker Rubio Mol</nombre>
  </ficha>
</fichas>

```

Escribir el contenido del archivo **"fichas.xsd"** que permita validarlo, teniendo en cuenta que:

- Tanto el atributo **numero** como el elemento **"código"** utilizan la misma restricción que solamente les permite tomar un valor entero expresado con dos dígitos comprendidos entre "00" y "19".
- El atributo **letra** puede tomar por valor una de las siguientes letras: "X", "Y" o "Z". La restricción debe definirse de forma que solamente pueda ser utilizada por dicho atributo. □ Para cada ficha se tiene que indicar un número, obligatoriamente. Sin embargo, la letra es opcional.

18. Definir un elemento "clave" que pueda tomar por valor exactamente diez caracteres, los cuales podrán ser letras mayúsculas o minúsculas de la "a" a la "z",

o dígitos del "0" al "9". Por ejemplo, serán válidos los valores siguientes:
"abcde12345", "Clave55ABC", "1A2b3c4D5f", etc.

19. Dado el siguiente archivo .xml realiza el correspondiente .xsd para validarlo.

```
<?xml version="1.0" encoding="UTF-8"?>
<articulos xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="articulos.xsd">
  <articulo>
    <nombre>Mesa</nombre>
    <precio moneda="Euro">50</precio>
  </articulo>
  <articulo>
    <nombre>Silla</nombre>
    <precio moneda="Dólar">78.99</precio>
  </articulo>
</articulos>
```

- El nombre de los artículos puede ser: Mesa, Silla y Sofá.
- La moneda puede ser: Euro, Dólar y Libra.
- El elemento nombre ha de declararse global. Debemos usar **ref**

20. Dado el siguiente archivo .xml realiza el correspondiente .xsd para validarlo.

```
<?xml version="1.0" encoding="UTF-8"?>
<bingo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="bingo.xsd">
  <bola numero="17"/>
  <bola numero="23"/>
  <bola numero="65"/>
</bingo>
```