

Ejercicios XSD 1

Dados los siguientes xsd, comentar si el XML que lo acompaña permite su validación o no y en caso de no validar, modificar el XML para que lo valide.

1) Colores A

Fichero XSD:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="color">
    <xs:complexType>
      <xs:sequence minOccurs="0" maxOccurs="1">
        <xs:element ref="colores"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="colores" type="xs:string"/>
</xs:schema>
```

Fichero xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<color>
<colores>blanco</colores>
<colores>rojo</colores>
</color>
```

SOLUCION:

La cardinalidad mínima es 0 y la máxima 1, lo que equivale a "?" en un DTD, por tanto un XML válido sería:

```
<color>
  <colores>blanco</colores>
</color>
```

2) Colores B

Fichero XSD:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="color">
    <xs:complexType>
      <xs:sequence minOccurs="1" maxOccurs="2">
        <xs:element ref="colores"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="colores" type="xs:string"/>
</xs:schema>
```

Fichero XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<color>
<colores>blanco</colores>
<colores>rojo</colores>
<colores>azul</colores>
</color>
```

SOLUCION:

Aquí se muestra la potencia de XSD frente a DTD pues podemos controlar exactamente la cardinalidad, en este caso la mínima es 1 y la máxima 2, lo que se puede realizar en DTD de forma mas engorrosa, especialmente con cardinalidades elevadas:

```
<?xml version="1.0" encoding="UTF-8"?>
<color>
<colores>rojo</colores>
<colores>azul</colores>
</color>
```

3) Colores C**Fichero XSD:**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="color">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="colores"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="colores" type="xs:string"/>
</xs:schema>
```

Fichero XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<color>
<colores>blanco</colores>
<colores>rojo</colores>
<colores>azul</colores>
</color>
```

SOLUCION: En este caso al omitirse las cardinalidades mínima y máxima se entiende que son 1 y 1 es decir que el elemento debe aparecer exactamente una vez.

4) Colores D

Fichero XSD:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="color">
    <xs:complexType>
      <xs:sequence minOccurs="1" maxOccurs="3">
        <xs:element ref="colores"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="colores" type="xs:integer"/>
</xs:schema>
```

Fichero XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<color>
  <colores>blanco</colores>
  <colores>rojo</colores>
  <colores>azul</colores>
</color>
```

SOLUCION:

Aunque la cardinalidad es correcta, el tipo permitido ahora no es string sino integer, y por tanto habría que cambiar el contenido a números enteros.

5) Colores D**Fichero XSD:**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="color">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="colores"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="colores" type="xs:string"/>
</xs:schema>
```

Fichero XML:

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

SOLUCION:

El xml es inválido por que no se permiten elementos vacíos en la secuencia al ser minOccurs="1" implícitamente.

6) Animales

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="animales">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="perro"/>
        <xs:element ref="gato"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="perro" type="xs:string"/>
  <xs:element name="gato" type="xs:string"/>
</xs:schema>
```

Fichero XML:

```
<animales>
  <animal>Perro</animal>
  <animal>Gato</animal>
</animales>
```

SOLUCION:

Animal se compone de dos elementos, uno "perro" y otro "gato", en este caso los contenidos de los elementos animal si tienen los valores "perro" y "gato" pero deben estar en la etiqueta y no en los valores.

```
<animales>
  <perro>Perro</perro>
  <gato>Gato</gato>
</animales>
```

7) Escritores A**Fichero XSD:**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="escritores">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element ref="escritor"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="escritor">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="nombre"/>
        <xs:element ref="nacimiento"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="nombre" type="xs:string"/>
  <xs:element name="nacimiento" type="xs:string"/>
</xs:schema>
```

Fichero XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<escritores>
  <escritor>
    <nombre>Mario Vargas Llosa</nombre>
    <nacimiento>28 de marzo de 1936</nacimiento>
  </escritor>
  <escritor>
    <nacimiento>1 de abril de 1929</nacimiento>
    <nombre>Milan Kundera</nombre>
  </escritor>
</escritores>
```

SOLUCION:

Escritores se compone de 1 o varios elementos escritor, el cual a su vez es una secuencia de nombre y nacimiento de tipo String cada uno de ellos. Al estar el orden cambiado en el segundo bloque el XSD no valida. La solución es poner nombre y nacimiento en el orden exacto que se especifica en el elemento sequence "escritor".

8) Escritores B

Fichero XSD:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="escritores">
    <xs:complexType>
      <xs:sequence maxOccurs="2">
        <xs:element ref="escritor"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="escritor">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="nombre"/>
        <xs:element ref="nacimiento"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="nombre" type="xs:string"/>
  <xs:element name="nacimiento" type="xs:string"/>
</xs:schema>
```

Fichero XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<escritores>
  <escritor>
    <nombre>Mario Vargas Llosa</nombre>
    <nacimiento>28 de marzo de 1936</nacimiento>
  </escritor>
  <escritor>
    <nacimiento>1 de abril de 1929</nacimiento>
    <nombre>Milan Kundera</nombre>
  </escritor>
  <escritor>
    <nacimiento>19 de Enero de 1809</nacimiento>
    <nombre>E.A. Poe</nombre>
  </escritor>
</escritores>
```

SOLUCION:

Hay dos errores uno de ellos es que solo pueden aparecer dos grupos escritor y el otro que escritor se define como una secuencia de nombre y nacimiento y en el xml aparecen dos grupos con <nacimiento> <nombre>, habría que cambiar el orden.

9) Musicos

Fichero XSD:

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

  <xs:element name="musicos">
    <xs:complexType>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="musico"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="musico">
    <xs:complexType>
      <xs:sequence>
        <xs:choice>
          <xs:element name="nombre" type="xs:string"/>
          <xs:element name="apodo" type="xs:string"/>
        </xs:choice>
        <xs:element name="fechaNacimiento" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Fichero XML:

```
<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>
```

SOLUCION:

<Musicos> es una secuencia formada por 0 o varias repeticiones de <musico>, el cual a su vez es una secuencia formada por uno de dos elementos (no a la vez): <nombre> o <apodo> seguidos de <fechaNacimiento> todos de tipos string. Sin embargo en el XML <nombre> y <apodo> aparecen a la vez lo que es erróneo.

10) Sistema solar A

Fichero XSD:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="sistemaSolar">
    <xs:complexType>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="cuerpo"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="cuerpo">
    <xs:complexType>
      <xs:choice>
        <xs:element name="planeta" type="xs:string"/>
        <xs:element name="satelite" type="xs:string"/>
        <xs:element name="asteroide" type="xs:string"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Fichero XML:

```
<sistemaSolar>
  <cuerpo>
    <planeta>Tierra</planeta>
    <satelite>Luna</satelite>
  </cuerpo>
  <asteroide>Ceres</asteroide>
</sistemaSolar>
```

SOLUCION:

En este caso <sistemaSolar> es una secuencia de 0 o varios elementos <cuerpo>, el cual es un "choice" de <planeta> <satélite> <asteroide>, "choice" permite una aparición de los elementos de la secuencia nada mas, y dado que no tienen cardinalidad expresada se asume mínima y máxima uno, con lo cual han de aparecer una vez, pero solo uno de ellos; sin embargo en el XML en cuerpo están <planeta> y <satélite> y fuera de <cuerpo> aparece <asteroide> lo que es incorrecto. La solución pasa por borrar <asteroide> y <planeta> o <satélite>

11) Sistema Solar B

Fichero XSD:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="sistemaSolar">
    <xs:complexType>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="cuerpo"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="cuerpo">
    <xs:complexType>
      <xs:all>
        <xs:element name="planeta" type="xs:string"/>
        <xs:element name="satelite" type="xs:string"/>
        <xs:element name="asteroide" type="xs:string"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Fichero XML:

```
<sistemaSolar>
  <cuerpo>
    <planeta>Tierra</planeta>
    <satelite>Luna</satelite>
  </cuerpo>
  <asteroide>Ceres</asteroide>
</sistemaSolar>
```

SOLUCION:

En este caso <sistemaSolar> es una secuencia de 0 o varios elementos <cuerpo>, el cual es un "all" de <planeta> <satélite> <asteroide>, "ALL" permite la aparición de todos los elementos en un orden cualquiera, y dado que no tienen cardinalidad expresada se asume mínima y máxima uno, con lo cual han de aparecer una vez cada uno de ellos; sin embargo en el XML en cuerpo están <planeta> y <satélite> y fuera de <cuerpo> aparece <asteroide> lo que es incorrecto. La solución pasa por poner <asteroide> dentro de <cuerpo>.

12) Escribir un XSD que valide el XML, e incluir los cambios necesarios en el mismo para referenciar al esquema creado. Todos los elementos hijos del elemento "email" son obligatorios y deben aparecer sólo una vez.

Números

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

<email>

    <para>Jose</para>
    <de>Maria</de>
    <asunto>Reunion</asunto>
    <cuerpo>Reunión en la planta quinta</cuerpo>

</email>
```

SOLUCION:

Se trata de una secuencia formada por los cuatro elementos de <email>, en este caso se entiende que tienen que aparecer en ese orden exactamente lo que es satisfecho por el elemento <xs:sequence>

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="email">
<xs:complexType>
<xs:sequence>
    <xs:element name="para" type="xs:string"/>
    <xs:element name="de" type="xs:string"/>
    <xs:element name="asunto" type="xs:string"/>
    <xs:element name="cuerpo" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

13) Realizar el XSD que valide 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 "coche", "modelo" y "mejora" 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 en el documento 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"?>
<coches>
  <coche>
    <modelo>Count Zero</modelo>
    <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>
    <mejoras>
      <mejora>Change Engine</mejora>
      <mejora>Change pedals</mejora>
      <mejora>Change gearbox</mejora>
      <mejora>Fit Rollcage</mejora>
    </mejoras>
  </coche>
</coches>
```

SOLUCION:

```
<?xml version = "1.0" encoding="ISO-8859-1"?>
<xsd:schema
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
  <xsd:element name = "coches">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref = "coche" maxOccurs = "unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name = "coche">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name = "nombre" type = "xsd:string" maxOccurs = "unbounded"/>
        <xsd:element name = "modelo" type = "xsd:string"/>
        <xsd:element name = "fabricacion">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element ref = "inicio"/>
              <xsd:element ref = "fin"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
```

```

        <xsd:element name = "mejoras">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name = "mejora" type = "xsd:string"
maxOccurs = "unbounded"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name = "inicio">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref = "dia"/>
            <xsd:element ref = "mes"/>
            <xsd:element ref = "anyo"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name = "fin">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref = "dia"/>
            <xsd:element ref = "mes"/>
            <xsd:element ref = "anyo"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name = "dia" type = "xsd:string"/>
<xsd:element name = "mes" type = "xsd:string"/>
<xsd:element name = "anyo" type = "xsd:string"/>
</xsd:schema>

```

14) Escribir un XML que valide el XSD adjunto:

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="root">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:pattern value="\d{2}-[A-Z]{2}"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
</xs:schema>

```

SOLUCION:

En este caso la expresión regular requiere la presencia de un dígito de dos cifras seguido de "-" y una cadena de mayúsculas de 2 caracteres:

```
<root>12-AB</root>
```