

**06.02. Validación XML - XSD II****Ejercicio 1 - Respuestas admitidas**

En el siguiente ejemplo se define un elemento llamado "respuesta" con la restricción de que el único valor aceptable es una de las siguientes letras: "A", "B", "C", "D" o "E":

```
<xs:element name="respuesta">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[ABCDE]" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

En vez de escribiendo "[ABCDE]", ¿de qué otras formas se podría especificar la misma restricción?

Sin hacer uso de xs:pattern, ¿de qué otro modo podríamos especificar lo mismo que lo expresado con <xs:pattern value="[ABCDE]"/>?

**Ejercicio 2 - Números y letras**

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>
```

Escribe 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.

### Ejercicio 3 - Escribir expresiones regulares

En las expresiones regulares se pueden utilizar –entre otros– los siguientes símbolos:

<i>Símbolos</i>	<i>Significado</i>
<b>.</b>	Cualquier carácter.
<b>\d</b>	Cualquier dígito del <b>0</b> al <b>9</b> .
<b>\D</b>	Cualquier carácter que no sea un dígito del <b>0</b> al <b>9</b> .
<b>x*</b>	<b>x</b> puede aparecer cero o más veces.
<b>x+</b>	<b>x</b> debe aparecer al menos una vez.
<b>x?</b>	<b>x</b> puede aparecer una vez o no aparecer.
<b>[abc]</b> o <b>[a b c]</b>	Cualquier carácter indicado entre los corchetes: <b>a</b> , <b>b</b> o <b>c</b> .
<b>[a-z]</b>	Cualquier carácter de la <b>a</b> a la <b>z</b> .
<b>x{n}</b>	<b>x</b> debe aparecer <b>n</b> veces.
<b>x{n,m}</b>	<b>x</b> debe aparecer entre <b>n</b> y <b>m</b> veces.
<b>x{n,}</b>	<b>x</b> debe aparecer al menos <b>n</b> veces.

Teniendo en cuenta, solamente, los símbolos mostrados en la tabla anterior, escribir las posibles expresiones regulares que permitan representar los siguientes valores (siempre hay al menos dos opciones distintas):

1. "Capítulo 0", "Capítulo 1", "Capítulo 2"... "Capítulo 9". (Solo se permite un dígito).
2. "Capítulo 0", "Capítulo 1", "Capítulo 2"... "Capítulo 99". (Uno o dos dígitos).
3. "Capítulo 1", "Capítulo 2", "Capítulo 3"... "Capítulo 99". (No se permite "Capítulo 0").
4. "Capítulo 0", "Capítulo 1", "Capítulo 2"... "Capítulo 99"... "Capítulo 100"... (Uno o más dígitos).
5. Cualquier valor de dos caracteres, cuyo primer carácter sea distinto de un dígito (0-9) y cuyo segundo carácter sea "Z": "aZ" ... "zZ", "AZ" ... "ZZ", "?Z", "=Z", "\*Z"...
6. "ABBC", "ABBBC", "ABBBBC", "ABBBBBBC".
7. El primer carácter debe ser "R". A continuación, deben aparecer obligatoriamente dos o más "S". Finalmente, puede aparecer o no, un dígito del 3 al 8: "RSS", "RSSS"... "RSS3"... "RSS8", "RSSS3"... "RSSS8"... "RSSSSSSSSSSS7"...
8. Cualquier valor que contenga en primer lugar "COD", después tres dígitos (0-9) y, finalmente, uno o más caracteres cualesquiera: "COD645pera", "COD646manzana"...

#### Ejercicio 4 - Letras admitidas

En el siguiente ejemplo se define un elemento llamado "letras" con la restricción de que puede tomar por valor cero o más (\*) letras minúsculas de la "a" a la "z":

```
<xs:element name="letras">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="([a-z])*"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Nota: los paréntesis de la expresión regular se pueden omitir, escribiendo simplemente: [a-z]\*

Realiza los cambios necesarios en el código del ejemplo anterior para que "letras" pueda tomar por valor uno o más pares (+) de letras, y cada par de letras deberá estar formado por una letra mayúscula seguida de otra minúscula. Por ejemplo, "HoLa" sería admitido, pero no lo sería "Hola", "HOLa", "hola", etc.