

[EJERCICIO 1] Validar un documento XML

Dado el documento ***“marcadores.xml”***:

```
<?xml version="1.0" encoding="UTF-8"?>
<marcadores xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="marcadores.xsd">
  <pagina>
    <nombre>OpenAI</nombre>
    <descripcion>Inteligencia artificial.</descripcion>
    <url>http://www.chatopenAI.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>
```

Y el archivo ***“marcadores.xsd”***:

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

Comprobar que ***“marcadores.xml”*** es válido.

[EJERCICIO 2] Definición de elementos simples

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.

[EJERCICIO 3] Puerta cerrada y ventana abierta

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**".

[EJERCICIO 4] Fichas de personas

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="1">
    <nombre>Ana Sanz Tin</nombre>
    <edad>22</edad>
  </ficha>
  <ficha numero="2">
    <nombre>Iker Rubio Mol</nombre>
    <edad>23</edad>
  </ficha>
</fichas>
```

Escribir el contenido del archivo "**fichas.xsd**" que permita validarlo.

[EJERCICIO 5] Edad entre 0 y 130 años

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="1">
    <nombre>Ana Sanz Tin</nombre>
    <edad>22</edad>
  </ficha>
  <ficha numero="2">
    <nombre>Iker Rubio Mol</nombre>
    <edad>23</edad>
  </ficha>
</fichas>
```

Escribir el contenido del archivo ***"fichas.xsd"*** que permita validarlo, teniendo en cuenta que se debe definir la "edad" con la restricción de que el valor que tome no pueda ser menor que 0 ni mayor que 130. Además, en vez de ***xs:minInclusive*** y ***xs:maxInclusive***, se debe utilizar:

- ***xs:minExclusive*** que sirve para especificar que el valor debe ser mayor que el indicado.
- ***xs:maxExclusive*** que sirve para especificar que el valor debe ser menor que el indicado.

[EJERCICIO 6] Precios de tres dígitos

Dado el siguiente documento XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<precios xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="precios.xsd">
  <precio>8</precio>
  <precio>2.6</precio>
  <precio>4.95</precio>
  <precio>187</precio>
</precios>
```

Escribir el contenido del archivo ***"precios.xsd"*** que permita validarlo, teniendo en cuenta que el elemento "precio" puede tomar por valor un número que contenga tres dígitos como máximo y, de ellos, solamente dos pueden ser decimales. Para ello, escribir una restricción que no podrá ser utilizada por otros elementos y, por otra parte, haga uso de:

- ***xs:totalDigits*** que sirve para especificar el número máximo de dígitos que puede tener un número, incluyendo a los decimales.
- ***xs:fractionDigits*** que sirve para especificar el número máximo de decimales que puede tener un número.

[EJERCICIO 7] Tipo de vehículo

Dada la siguiente definición:

```
<xs:element name="vehiculo">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="barco"/>
      <xs:enumeration value="bicicleta"/>
      <xs:enumeration value="coche"/>
      <xs:enumeration value="tren"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

¿De qué otro modo se puede definir el elemento “vehículo” y un **xs:simpleType** llamado **tipoDeVehiculo** que restringiese a "barco", "bicicleta", "coche" y "tren" como los únicos valores aceptables para el vehículo, de forma que dicho tipo pudiera ser también utilizado por otros elementos?

[EJERCICIO 8] Iniciales de personas famosas

Dado el archivo “*fichas.xsd*” cuyo contenido es:

```
<?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>
```

Corregir los errores cometidos en el siguiente documento XML, para que sea válido:

```
<?xml version="1.0" encoding="UTF-8"?>
<fichas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fichas.xsd">
  <ficha>
    <nombre>Antonio Machado Ruiz</nombre>
    <iniciales>AMR</iniciales>
    <edad>22</edad>
  </ficha>
  <ficha>
    <nombre>Mario Moreno</nombre>
    <iniciales>MM</iniciales>
    <edad>23</edad>
  </ficha>
  <ficha>
    <iniciales>ALO</iniciales>
    <nombre>Ada Lovelace</nombre>
    <edad>24</edad>
  </ficha>
  <ficha>
    <nombre>pablo ruiz picasso</nombre>
    <iniciales>prp</iniciales>
    <edad>24</edad>
  </ficha>
</fichas>
```

[EJERCICIO 9] Iniciales al revés

Dado el archivo *"fichas.xsd"*:

```
<?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>
```

Realizar los cambios necesarios en **“fichas.xsd”**, para que el siguiente documento XML sea válido:

```
<?xml version="1.0" encoding="UTF-8"?>
<fichas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fichas.xsd">
  <ficha>
    <nombre>Ana Sanz Tin</nombre>
    <iniciales>AST</iniciales>
    <edad>22</edad>
    <iniciales-al-reves>TSA</iniciales-al-reves>
  </ficha>
  <ficha>
    <nombre>Iker Rubio Mol</nombre>
    <iniciales>IRM</iniciales>
    <edad>23</edad>
    <iniciales-al-reves>MRI</iniciales-al-reves>
  </ficha>
</fichas>
```

Nota: hay que tener en cuenta que el elemento **iniciales-al-reves** debe ser del mismo tipo que **iniciales**, y dicho tipo solamente deberá definirse una vez.

[Ampliación del ejercicio]

Si se quisiera permitir que las tres letras de **iniciales** e **iniciales-al-reves** sean minúsculas o mayúsculas indistintamente; en vez de **[A-Z] [A-Z] [A-Z]** ¿qué se podría escribir?

[EJERCICIO 10] 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?

[Ampliación del ejercicio]

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

[EJERCICIO 11] 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>
```

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.

[EJERCICIO 12] Escribir expresiones regulares

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

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

Teniendo en cuenta, solamente, los símbolos mostrados en la tabla anterior, escribir las posibles expresiones regulares que permitan representar los siguientes valores:

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 ó 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"...

[Ampliación del ejercicio]

Dado el siguiente documento XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<expresiones-regulares xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="expresiones-regulares.xsd">
  <expresiones>
    <expresion1>Capítulo 0</expresion1>
    <expresion2>Capítulo 0</expresion2>
    <expresion3>Capítulo 1</expresion3>
    <expresion4>Capítulo 0</expresion4>
    <expresion5>aZ</expresion5>
    <expresion6>ABBC</expresion6>
    <expresion7>RSS</expresion7>
    <expresion8>COD645pera</expresion8>
  </expresiones>
  <expresiones>
    <expresion1>Capítulo 9</expresion1>
    <expresion2>Capítulo 99</expresion2>
    <expresion3>Capítulo 99</expresion3>
    <expresion4>Capítulo 99999</expresion4>
    <expresion5>?Z</expresion5>
    <expresion6>ABBBBBBC</expresion6>
    <expresion7>RSSSSSSSSSSS7</expresion7>
    <expresion8>COD646manzana</expresion8>
  </expresiones>
</expresiones-regulares>
```

Escribir el contenido del archivo **"expresiones-regulares.xsd"** que permita validarlo utilizando las expresiones regulares escritas en este ejercicio.

[EJERCICIO 13] 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]***

Realizar 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.

[EJERCICIO 14] Longitud fija de una clave

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.

- Solución 1: utilizando las facetas **xs:pattern** y **xs:length**.
- Solución 2: utilizando solamente **xs:pattern**.

[EJERCICIO 15] Longitud mínima y máxima de una clave

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>
    <nombre>Ana Sanz Tin</nombre>
    <clave>G8w2</clave>
  </ficha>
  <ficha>
    <nombre>Iker Rubio Mol</nombre>
    <clave>ag32Ue7AFF</clave>
  </ficha>
</fichas>
```

Escribir el contenido del archivo ***"fichas.xsd"*** que permita validarlo, teniendo en cuenta que el elemento "clave" debe poder tomar por valor un mínimo de cuatro caracteres y un máximo de diez. Dichos caracteres pueden ser indistintamente letras mayúsculas o minúsculas de la "a" a la "z", o dígitos del "0" al "9". La restricción solamente podrá aplicarse al elemento "clave".

Para ello, se debe utilizar **xs:pattern** y también:

- **xs:minLength** que permite especificar la longitud mínima.
- **xs:maxLength** que permite especificar la longitud máxima.