Caso Práctico de XSD

Objetivo

Crear un esquema de validación en XSD para el siguiente XML.

```
<libros>
    <libro disponible="true">
        <titulo>The Elements</titulo>
        <fechaPubli>2000-11-24</fechaPubli>
        <ref>23-TE</ref>
        <precio>12.45</precio>
    </libro>
    <libro disponible="true">
        <titulo>Alquimia</titulo>
        <fechaPubli>1988-01-02</fechaPubli>
        <ref>13-Al</ref>
        <precio divisa="$">25.24</precio>
    </libro>
    <libro disponible="false">
        <titulo>El Médico</titulo>
        <fechaPubli>2007-07-30</fechaPubli>
        <ref>09-EM</ref>
        <precio divisa="€">30.12</precio>
    </libro>
</libros>
```

Paso 1: Definir el elemento raíz y sus hijos

Un elemento se define con la etiqueta:

```
<xs:element name="nomnbre_elemento">
```

Si es un elemento final, su tipo se especifica como un argumento y se puede escoger unos de los múltiples tipos básicos existente en XSD. Por ejemplo:

```
<xs:element name="nombre" type="xs:string"/>
<xs:element name="edad" type="xs:positiveInteger"/>
<xs:element name="fechaNacimiento" type="xs:date"/>
```

Pero si el elemento va a contener otros elementos, Los elementos hijos se declaran dentro de un elemento **complejo**. Además, también tendremos que especificar si los elementos hijos son

- una secuencia “sequence”.
- se escoge uno solo “choice”.
- pueden aparecer todos en cualquier orden “all”.

```
<xs:element name="raíz">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="hijo1"/>
      <xs:element name="hijo2"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

En nuestro ejemplo, vamos a declarar el elemento raíz **“libros”** y el elemento hijo **“libro”**.

```
<xs:element name="libros">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="libro"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

El siguiente paso es indicar la **repetición** del elemento <libro>. En XSD hay dos atributos para definirla:

- minOccurs: número mínimo de ocurrencias.
- maxOccurs: número máximo de ocurrencias, y *unbounded* para indicar *indefinidas* veces.

Ampliando el ejemplo anterior

```
<xs:element name="libros">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="libro" minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Con este XSD, podemos validar, a modo ilustrativo, el siguiente XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<libros xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="libros00.xsd">
    <libro/>
    <libro></libro>
</libros>
```

Paso 2: Definir los hijos de <libro>

En este paso definiremos los elementos hijos del xml. En este caso, estos nodos ya no serán de *tipo complejo,* sino de *tipo simple,* así que haremos uso de los tipos que trae XSD por defecto.

Primero, como el contenido de “libro” son varios hijos, seguiremos con un elemento *complejo* y una *secuencia*.

En cada elemento se especifica el tipo de contenido mediante el atributo “type”.

```
<xs:element name="libro" minOccurs="1" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="titulo" type="xs:string"/>
            <xs:element name="fechaPubli" type="xs:date"/>
            <xs:element name="ref" type="xs:string"/>
            <xs:element name="precio" type="xs:decimal"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Paso 3: Añadir el atributo <libro disponible="">

Para definir un atributo usaremos, por ejemplo:

```
<xs:attribute name="ID" type="xs:string" use="optional">
```

Donde vemos que los atributos también usan los tipos definidos en XSD, y su uso puede ser *"optional"* o *"required"*.

Los atributos **solo** son parte de elementos **complejos**, y se definen dentro de un elemento <xs:complexType>; así que, en el ejemplo, el atributo disponible de <libro> se define **después** de *sequence*.

```
<xs:element name="libro" maxOccurs="unbounded" minOccurs="1">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="titulo" type="xs:string"/>
            <xs:element name="fechaPubli" type="xs:date"/>
            <xs:element name="ref" type="xs:string"/>
            <xs:element name="precio" type="xs:decimal"/>
        </xs:sequence>
        <xs:attribute name="disponible" type="xs:boolean" default="true"
use="required"/>
    </xs:complexType>
</xs:element>
```

Paso 4: Añadir atributo a un elemento final <precio divisa="">

Añadir un atributo a un elemento final, no es tan obvio como parece. Uno podría pensar que solo habría que escribir, para añadir el atributo divisa, algo como esto:

```
<xs:element name="precio" type="xs:decimal">
    <xs:complexType>
        <xs:attribute name="divisa" type="xs:string" default="€" use="optional"/>
    </xs:complexType>
</xs:element>
```

Pero esto no es así.

En realidad al añadir el atributo a través de un <xs:complexType>, el atributo type="xs:decimal" queda invalidado. La manera de resolver esta situación és:

```
<xs:element name="precio">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:decimal">
                <xs:attribute name="divisa" type="xs:string" default="€"
                 use="optional"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
```

Y con esto ya tenemos la primera versión del XSD, que nos permite validar el ejemplo inicial.

El código completo es:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="libros">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="libro" maxOccurs="unbounded" minOccurs="1">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="titulo" type="xs:string"/>
                            <xs:element name="fechaPubli" type="xs:date"/>
                            <xs:element name="ref" type="xs:string"/>
                            <xs:element name="precio">
                                <xs:complexType>
                                    <xs:simpleContent>
                                        <xs:extension base="xs:decimal">
                                            <xs:attribute name="divisa"
                                            type="xs:string" default="€"
                                            use="optional"/>
                                        </xs:extension>
                                    </xs:simpleContent>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                        <xs:attribute name="disponible" type="xs:boolean"
                        default="true" use="required"/>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```