Métodos de diseño XSD XMLSchema

Hay varias maneras de abordar el diseño de un XMLSchema. Usaremos una u otra, o una combinación de varias, dependiendo de factores tales como la complejidad, extensión y el tipo de documentos que estamos definiendo (por ejemplo, sin son documentos donde predomina una colección de datos estructurados, o son documentos con mucho texto libre).

A continuación, se describen tres formas de diseñar XMLSchemas:

- Diseño Anidado o de muñecas rusas. Se llama así porque se anidan declaraciones de elementos unas dentro de otras. Se describe cada elemento y atributo en el mismo lugar donde se declaran. Consiste en, partiendo de un documento XML, seguir la estructura del mismo e ir definiendo los elementos que aparecen en el mismo de forma secuencial, incluyendo la definición completa de cada elemento en el mismo orden en el que aparecen en el documento instancia. Este método de diseño es muy sencillo, pero puede dar lugar a esquemas XML difíciles de leer y mantener cuando los documentos son complejos. Además, produce duplicidades en la descripción de elementos con tipos iguales y puede haber elementos con igual nombre y distintas descripciones. Es el método de diseño menos recomendable.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http;//www.w3.org/2001/XMLSchema">
   <xs:element name="libro">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="titulo" type="xs:string"/>
            <xs:element name="autor" type="xs:string"/>
            <xs:element name= "personaje" minOccurs= "O"
                  maxOccurs =" unbounded" >
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="nombre" type="xs:string"/>
                           <xs:element name="amigoDe" type= "xs : string"
                              minOccurs="O" maxOccurs="unbounded"/>
                     <xs:element name="desde" type="xs:date"/>
                     <xs:element name="calificación" type="xs:string"/>
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
         </xs:sequence>
         <xs:attribute name="isbn" type="xs:string"/>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

- Diseño Plano o de uso de referencias a elementos y atributos. En este método primero se definen los diferentes elementos y atributos, para después referenciarlos utilizando el atributo “ref”. La declaración y su definición están en diferentes sitios. Esta técnica es la que más se parece al diseño con DTDs. En el ejemplo que mostramos a continuación se comienza a definir los elementos más simples para terminar con los más complejos, pero puede

hacerse al revés (como en los DTDs) comenzando por definir los elementos de mayor nivel y, descendiendo, ir definiendo los elementos de menor nivel hasta llegar a los elementos simples.

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

   <!-- definición de elementos de tipo simple-->
   <xs:element name="titulo" type="xs:string"/>
   <xs:element name="autor" type="xs:string"/>
   <xs:element name="nombre" type="xs:string"/>
   <xs:element name="amigoDe" type="xs:string"/>
   <xs:element name="desde" type= "xs:date"/>
   <xs:element name="calificacion" type="xs:string"/>

   <!-- definición de atributos -->
   <xs:attribute name="isbn" type="xs:string"/>

   <!-- definición de elementos de tipo complejo -->
   <xs:element name="personaje">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="nombre"/>
            <xs:element ref="amigoDe" minOccurs="O"
maxOccurs="unbounded"/>
            <xs:element ref="desde"/>
            <xs:element ref="calificación"/>
            <!-- los elementos de tipo simple se referencian ref -->
            <!-- la definición de cardinalidad en elemento referenciado-->
         </xs:sequence>
      </xs:complexType>
   </xs:element>

   <xs:element name="libro">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="titulo"/>
            <xs:element ref="autor"/>
            <xs:element ref="personaje" minOccurs="O"
maxOccurs="unbounded"/>
         </xs:sequence>
         <xs:attribute ref="isbn"/>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

- Diseño con tipos con nombre. Con este método se definen clases o tipos utilizando los elementos de XMLSchema “simpleType” y “complexType” con un nombre. Estos tipos definidos se pueden utilizar mediante el tributo “type” de los elementos. Este mecanismo permite reutilizar de definiciones de tipos en diferentes puntos del documento. Es el método más aconsejado ya que permite la reutilización. En el ejemplo, al igual que en el caso anterior, se ha

seguido un diseño ascendente (definiendo primero el de menor nivel) pero se puede hacer un diseño descendente.

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

   <!-- definición de tipos simples-->
   <xs:simpleType name="TipoNombre" >
      <xs:restriction base="xs:string">
      <xs:raaxLength value="32"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="TipoDesde">
      <xs:restriction base="xs:date"/>
   </xs:simpleType>
   <xs:simpleType name= "TipoDescripcion" >
      <xs:restriction base="xs:string"/>
   </xs:simpleType >
   <xs:simpleType name="TipoISBN">
      <xs:restriction base="xs:string">
      <xs:pattern value="[0-9]{13}"/>
      </xs:restriction>
   </xs:simpleType>

   <!-- definición of tipos complejos -->
   <xs:complexType name="TipoPersonaje">
      <xs:sequence>
         <xs:element name="nombre" type="TipoNombre"/>
         <xs:element name="amigoDe" type="TipoNombre" minOccurs = "0"
                maxOccurs="unbounded"/>
         <xs:element name="desde" type="TipoDesde"/>
         <xs:element name="calificacion" type="TipoDescripcion"/>
      </xs:sequence>

   </xs:complexType>
   <xs:complexType name="TipoLibro">
      <xs:sequence>
         <xs:element name="titulo" type="TipoNombre"/>
         <xs:element name="autor" type="TipoNombre"/>
         <xs:element name="personaje" type="TipoPersonaje"
minOccurs="O"/>
      </xs:sequence >
      <xs:attribute name="isbn" type="TipoISBN" use="required"/>
   </xs:complexType>

   <!-- definición del elemento raíz libro usando el tipo complejo TipoLibro-->
   <xs:element name="libro" type="TipoLibro"/>
</xs:schema>
```