XML Schema

Joan Sales
IES Provençana
DAWBIO1-M04-UF1

Anteriormente con DTD

```
<!ELEMENT juegos (juego+)>
<!ELEMENT juego (Nombre,EdadMinima)>
<!ELEMENT Nombre (#PCDATA)>
<!ELEMENT EdadMinima (#PCDATA)>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE juegos SYSTEM "juegos.dtd">
<juegos>
  <juego>
    <Nombre>FIFA</Nombre>
    <EdadMinima>Portugal</EdadMinima>
  </juego>
</juegos>
```

¿Qué ocurre aquí?

Anteriormente con DTD

```
<!ELEMENT juegos (juego+)>
<!ELEMENT juego (Nombre,EdadMinima)>
<!ELEMENT Nombre (#PCDATA)>
<!ELEMENT EdadMinima (#PCDATA)>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE juegos SYSTEM "juegos.dtd">
<juegos>
  <juego>
    <Nombre>FIFA</Nombre>
    <EdadMinima>Portugal</EdadMinima>
  </juego>
</juegos>
```

DTD No permite especificar con tanto detalle el contenido de los elementos…

Declaración XSD

Otra forma de definir la estructura de elementos XML es con XML Schema.

XML

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

<root element  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="esquema.xsd">

</root element >
```

Declaración XSD

XSD

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >

</xs:schema>
```

Declaración XSD

XML

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

XSD

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

Elementos simples

Tipos simples y tipos complejos

Tipos simples

-No contienen atributos
-No contienen otras etiquetas
-No están vacíos
-Solo contienen datos
-Tipos: (¡Hay más!)

- xs:string
- xs:decimal
- xs:integer
- xs:boolean
- xs:date
- xs:time

Tipos complejos

-Pueden contener atributos o

-Pueden contener otras etiquetas o

-Pueden estar vacíos o

-Puede contener datos y etiquetas o

-Combinaciones de lo anterior

Elementos simples

Tipos simples. Sintaxis

```
<xs:element name="ElementoSimple">
  <xs:simpleType>
    <xs:restriction base="xs:Tipo">
       Restricciones...
    </xs:restriction>
  </xs:simpleType>
</xs:element>

<xs:element name="ElementoSimple type="xs:Tipo"/>
```

Elementos simples

Un elemento simple puede tener valores default o fixed asignados

<xs:element name=”TipoBotella” type=”xs:string” default=”Plástico”>

Cuidado, los elementos vacíos también validan aunque hayamos indicado un fixed:

<TipoBotella></TipoBotella>

Elementos complejos

Tipos complejos. Sintaxis

```
<xs:element name="ElementoComplejo">
  <xs:complexType>
    Atributos...
    <xs:sequence> O bien all o bien choice
      Sequencia de elementos...(O ninguno si está vacío)
    </xs:sequence>
  </xs:complexType>

</xs:element>
```

Elementos complejos

Tipos complejos

Indicadores:

Sequence
Varios elementos en ese orden.

```
<xs:element name=”Name”>
 <xs:complexType>
  <xs:sequence>

  </xs:sequence>
 </xs:complexType>
</xs:element>
```

Choice
Opción entre varios elementos.

```
<xs:element name=”Name”>
 <xs:complexType>
  <xs:choice>

  </xs:choice>
 </xs:complexType>
</xs:element>
```

All
Varios elementos en cualquier orden, sólo una vez cada elemento.

```
<xs:element name=”Name”>
 <xs:complexType>
  <xs:all>

  </xs:all>
 </xs:complexType>
</xs:element>
```

Elementos complejos

Tipos complejos

Ocurrencias

```
<xs:element name=”Name”>
 <xs:complexType>
  <xs:sequence>
   <element ref=”Referencia” minOccurs=”1” maxOccurs=”20”/>
  </xs:sequence>
 </xs:complexType>
</xs:element>
```

Por defecto tanto minOccurs como maxOccurs es 1.
Si queremos tener un número ilimitado podemos indicar maxOccurs=”unbounded”

XML Schema

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

<!--Definiciones de elementos simples-->
<xs:element name="Tipo"/>
<xs:element name="x" type="xs:float"/>
<xs:element name="y" type="xs:float"/>
<xs:element name="z" type="xs:float" fixed="2"/>


<!--Definiciones de elementos complejos-->
<xs:element name="Mueble">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Tipo"/>
      <xs:element ref="x"/>
      <xs:element ref="y"/>
      <xs:element ref="z"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="Muebles">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Mueble"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>


</xs:schema>
```

Separamos un poco las definiciones...

Podemos referenciar otros elementos complejos.

Restricciones

Un elemento simple puede tener restricciones asignadas:

```
<xs:element name=”Edad”>
 <xs:simpleType>
  <xs:restriction base=”xs:integer>
   <xs:minInclusive value=”18”/>
   <xs:maxInclusive value=”200”/>
  </xs:restriction>
 </xs:simpleType>
</xs:element>
```

Hay que definirlo con simpleType

Restricciones

Posibles restricciones:

Rango:

```
<xs:element name=”Edad”>
 <xs:simpleType>
  <xs:restriction base=”xs:integer>
   <xs:minInclusive value=”18”/>
   <xs:maxInclusive value=”200”/>
  </xs:restriction>
 </xs:simpleType>
</xs:element>
```

Ciertos valores:

```
<xs:element name=”TipoBotella”>
 <xs:simpleType>
  <xs:restriction base=”xs:string>
   <xs:enumeration value=”Plástico”/>
   <xs:enumeration value=”Metal”/>
   <xs:enumeration value=”Cerámica”/>
  </xs:restriction>
 </xs:simpleType>
</xs:element>
```

Muchas más...

length
fractiondigits
whiteSpace
...

Restricciones (expresiones regulares)

Posibles restricciones:

Expresiones regulares:

```
<xs:element name=”dni”>
 <xs:simpleType>
  <xs:restriction base=”xs:string>
   <xs:pattern value=”[0-9]{8}-[A-Z]”/>
  </xs:restriction>
 </xs:simpleType>
</xs:element>
```

[]= Entre corchetes: valores válidos
    [0-9]: Del 0 al 9
    [a-zA-Z]:De la ‘a’ a la ‘z’ y de la ‘A’ a la ‘Z’
        Recordad tabla unicode…
*= Ninguna aparición o muchas
+= Una o más apariciones
    [0-9]+: Del 0 al 9 varias veces, como 888721
|= or. Un valor u otro.
    [soltero|casado]= O bien soltero, o bien casado
{}=Exactamente ese número de veces
    {8} Lo anterior se debe repetir exactamente 8 veces.

Atributos

Atributos

<xs:attribute name=”Color” type=”xs:string”/>

Los atributos pueden llevar default o fixed(igual que los elementos simples)

Por defecto los atributos son opcionales, si queremos indicar que son obligatorios podemos utilizar use=”required”

Recordemos: Solo se pueden especificar atributos si son tipos complejos.
La especificación del atributo(es decir, cómo indicamos qué etiqueta tiene ese atributo) se hace dentro del tipo complejo.

También podemos aplicar restricciones a los atributos…

```
<!--Definiciones de atributos-->
<xs:attribute name="Color">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="Rojo|Azul"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```