

Unidad 8:

Schema



JJ Taboada León
IES San Sebastián, Departamento de Informática
LENGUAJE DE MARCASY SGI
Curso 2011 / 2012

Guión del tema

Contenidos

- ¿Qué son los esquemas?
- Finalidad de un esquemas XML
- Estructura de un esquema XML
- Elementos simples y complejos

Actividades

- La expuestas en el cuaderno de actividades

¿Qué son los esquemas XML?

- Un esquema XML describe la estructura de un documento XML

nota.xsd

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

<xs:element name="nota">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="para" type="xs:string"/>
      <xs:element name="de" type="xs:string"/>
      <xs:element name="asunto" type="xs:string"/>
      <xs:element name="cuerpo" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

nota.xml

(sin referencia al esquema)

```
<?xml version="1.0"?>
<nota>
  <para>Antonio</para>
  <de>Juan</de>
  <asunto>Recuerdo</asunto>
  <cuerpo>¡No me olvides!</cuerpo>
</nota>
```

¿Qué son los esquemas XML?(II)

- XML Schema es una alternativa basada en XML para la DTD.
- Un esquema XML describe la estructura de un documento XML.
- El lenguaje de esquema XML también se conoce como definición de esquema XML (XSD).

Finalidad de un esquemas XML

- Definir los elementos válidos de un documento XML, al igual que un DTD.

Un esquema XML:

- Define los elementos que pueden aparecer en un documento
- Define los atributos que pueden aparecer en un documento
- Define los elementos que son elementos secundarios
- Define el orden de los elementos secundarios
- Define el número de elementos secundarios
- Define si un elemento está vacío o puede incluir texto
- Define los tipos de datos de elementos y atributos
- Define los valores por defecto y fijos para elementos y atributos

Referencia a un esquema (*nota.xsd*) en un documento XML

```
<?xml version="1.0" encoding="ISO-8859-1?>
<nota xmlns="http://www.w3schools.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:schemaLocation="
http://roble.pntic.mec.es/jtal0007/xmlfp/XSD/
nota.xsd" >
  <para>Antonio</para>
  <de>Juan</de>
  <asunto>Recuerdo</asunto>
  <cuerpo>No me olvides este weekend!</cuerpo>
</nota>
```

nota.xml
(con referencia al esquema)

Caso de un esquema local

xmlns="http://www.w3schools.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="nota.xsd"

Estructura de un Esquema XML(I)

Elemento raíz

```
<?xml version="1.0"?>
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">
```

Indica que los elementos y tipos de datos utilizados en el esquema provienen del espacio de nombres: "http://www.w3.org/2001/XMLSchema"

```
...
...
</xs:schema>
```

Todos los elementos y atributos declarados globalmente en el esquema deben aparecer calificados con un espacio de nombres

Indica que los elementos definidos por este esquema (nota, para, de, cuerpo.) provienen del espacio de nombres: "http://www.w3schools.com"

Espacio de nombres por defecto: "http://www.w3schools.com".

Estructura de un Esquema XML(II)

- Un esquema contiene elementos que pueden ser
 - **Simples:** No pueden tener ni elementos ni atributos
 - **Complejos:** Pueden contener otros elementos y atributos

```
<xs:element name="nota">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="para" type="xs:string"/>
      <xs:element name="de" type="xs:string"/>
      <xs:element name="asunto" type="xs:string"/>
      <xs:element name="cuerpo" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Nota es un elemento
Complejo pues
contiene hijos en el
documento XML

Elementos
Simples

XSD Elementos simples

- Los esquemas XML definen los elementos de los documentos XML.
- Un elemento simple es un elemento XML que sólo contiene texto. *(No puede contener otros elementos o atributos)* .

```
<xs:element nombre ="xxx" type="yyy" />
```

- Los tipos Pueden ser

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

XML	XSD
<pre><nombre>Juan</nombre> <edad>18</edad> <fecha>15-03-2012</fecha></pre>	<pre><xs:element name="nombre" type="xs:string"/> <xs:element name="edad" type="xs:integer"/> <xs:element name="fecha" type="xs:date"/></pre>

Valores por defecto y valores fijos para elementos simples

- `<xs:element name="color" type="xs:string" default="rojo"/>`
- `<xs:element name="color" type="xs:string" fixed="rojo"/>`

- Actividades: Realizar simple1 y simple2 del cuaderno de actividades de Schema

XSD Restricciones o facetas (I)

- Podemos restringir valores para los elementos o atributos XML.
- El siguiente ejemplo define un elemento llamado “temperatura ”, con una restricción. El valor de la temperatura no puede ser inferior a -20 o superior a 40:

Restricciones sobre los valores

```
<xs:element name="temperatura">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="-20"/>
      <xs:maxInclusive value="40"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

XSD Restricciones (II)

- Podemos limitar el contenido de un elemento XML a un conjunto de valores aceptables, usando la restricción de enumeración.
- El siguiente ejemplo define un elemento llamado “semana” con una restricción. Los únicos valores aceptables son: Lunes, Miércoles y Viernes:

Restricciones a un conjunto de valores

```
<xs:element name="semana">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Lunes"/>
      <xs:enumeration value="Miércoles"/>
      <xs:enumeration value="Viernes"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

XSD Restricciones (III)

- El ejemplo anterior también se puede escribir así

Restricciones a un conjunto de valores

```
<xs:element name="semana" type="diasSemana" />
.....
<xs:simpleType name="diasSemana">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Lunes" />
    <xs:enumeration value="Miércoles" />
    <xs:enumeration value="Viernes" />
  </xs:restriction>
</xs:simpleType>
```

- Se trata de un tipo personalizado “diasSemana”. La ventaja es que el tipo diasSemana podemos utilizarlo en otras partes del esquema

Restricciones usando patrones

- El siguiente ejemplo define un elemento denominado "carta" con una restricción. El único valor aceptable es UNA de las letras minúsculas de la A a la Z:

Restricciones usando patrones

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

- Otros patrones
 - <xs:pattern value="[A-Z][A-Z][A-Z]" />
 - <xs:pattern value="[0-9][0-9][0-9][0-9][0-9]" />

Restringir la longitud de lo contenido en un elemento

Restricciones de longitud

```
<xs:element name="dni">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:length value="10" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

- **xs:length:** establece una longitud fija
- **xs:minLength:** establece un mínimo en la longitud
- **xs:maxLength:** establece un máximo en la longitud

Resumen de restricciones/facetas

Restricción	Descripción
enumeración	Define una lista de valores aceptables
fractionDigits	Especifica el número máximo de decimales permitidos. Debe ser igual o mayor que cero
length	Especifica el número exacto de caracteres o elementos de lista permitidos. Debe ser igual o mayor que cero
maxExclusive	Especifica los límites superiores para valores numéricos (el valor debe ser inferior a este valor)
maxInclusive	Especifica los límites superiores para valores numéricos (el valor debe ser menor o igual a este valor)
maxLength	Especifica el número máximo de caracteres o elementos de lista permitidos. Debe ser igual o mayor que cero
minExclusive	Especifica los límites inferiores de los valores numéricos (el valor debe ser mayor que este valor)
minInclusive	Especifica los límites inferiores de los valores numéricos (el valor debe ser mayor o igual a este valor)
minLength	Especifica el número mínimo de caracteres o elementos de lista permitidos. Debe ser igual o mayor que cero
patrón	Define la secuencia exacta de caracteres que son aceptables
totalDigits	Especifica el número máximo de dígitos permitidos. Debe ser mayor que cero
whiteSpace	Especifica el espacio en blanco (los saltos de línea, tabuladores, espacios y retornos de carro) se maneja

- **Actividades:** Realizar las actividades restriccion1 a 4 del cuaderno de actividades schema.

Elementos complejos

Son elementos XML que contienen otros elementos y / o atributos.

Hay cuatro tipos de elementos complejos:

- Elementos con contenido vacío y atributos
- Elementos con contenido y atributos
- Elementos que contienen hijos
- Elementos que contienen elementos y texto (mixto)

```
<alumno>  
  <nombre>Juan</nombre>  
  <apellido>León</apellido>  
</alumno>
```

```
<xs:element name="alumno">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element name="alumno" type="xs:string"/>  
      <xs:element name="apellido" type="xs:string"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

Declaración

(Indica que los siguientes elementos son hijos y en ese orden)

XSD Atributos

- Los elementos con atributos son elementos complejos
- Los atributos se declaran como tipos simples.

```
<xs:attribute nombre="xxx" type="yyy"/>
```

- Los tipos Pueden ser

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

XML	XSD
<code><curso letra="A">1</curso></code>	<code><xs:element name="curso" type="xs:integer"/> <xs:attribute name="letra" type="xs:string"/></code>

Valores por defecto, fijos u opcionales

```
<xs:attribute name="letra" type="xs:string" default="A"/>  
<xs:attribute name="letra" type="xs:string" fixed="A"/>  
<xs:attribute name="letra" type="xs:string" use="required"/>
```

Elementos complejos vacíos y con atributos

XML

```
<producto id="1345" />
```

XSD

```
<xs:element name="producto">  
  <xs:complexType>  
    <xs:attribute name="id" type="xs:positiveInteger"/>  
  </xs:complexType>  
</xs:element>
```

Elementos complejos con contenido y atributos

XML

```
<nombre repetidor="No" >juan</nombre>
```

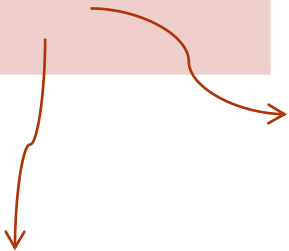
XSD

```
<xs:element name="nombre" >  
  <xs:complexType>  
    <xs:simpleContent>  
      <xs:extension base="xs:string">  
        <xs:attribute name="repetidor" type="xs:string" />  
      </xs:extension>  
    </xs:simpleContent>  
  </xs:complexType>  
</xs:element>
```

Tipo complejo con sólo hijos

XML

```
<alumno>
  <nombre>Juan</nombre>
  <apellido>León</apellido>
</alumno>
```



XSD

```
<xs:element name="alumno">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="alumno"
type="xs:string"/>
      <xs:element name="apellido"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

XSD usando un tipo personalizado "tipopersona"

```
<xs:element name="alumno" type="tipoperso">
<xs:element name="tipoPerso">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="alumno" type="xs:string"/>
      <xs:element name="apellido" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Tipos complejos con contenido mixto

XML

```
<carta>  
  Estimado Sr.<nombre>Juan León</nombre>.  
  Su pedido <pedido>1032</pedido>  
  será enviado el <fechaenvio>25-03-2012</fechaenvio>.  
</carta>
```

XSD

```
<xs:element name="carta">  
  <xs:complexType mixed="true">  
    <xs:sequence>  
      <xs:element name="nombre" type="xs:string"/>  
      <xs:element name="pedido" type="xs:positiveInteger"/>  
      <xs:element name="fechaenvio" type="xs:date"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

Otras forma de declarar elementos complejos (tipos personalizados)

XSD

```
<xsd:element name="empleado" type="tipoPersona"/>
<xsd:element name="estudiante" type="tipoPersona"/>

<xsd:complexType name="tipoPersona">
  <xsd:sequence>
    <xsd:element name="nombre" type="xsd:string"/>
    <xsd:element name="apellidos" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
```

Otras forma de declarar elementos complejos (referencias ref)

- **<!-- Definición de elementos-->**

```
<xs:element name="nombre" type="xsd:string" >  
<xs:element name="apellidos" type="xsd:string" >
```

<!-- Definición de atributos -->

```
<xs:attribute name="repetidor" type="xs:string" />
```

<!-- Definición de elementos complejos -->

```
<xsd:element name="alumno"  
  <xsd:complexType >  
    <xsd:sequence>  
      <xsd:element ref="nombre"/>  
      <xsd:element ref="apellidos"/>  
      <xsd:attribute ref="repetidor" />  
    </xsd:sequence>  
  </xsd:complexType>  
</xsd:element>
```


Actividades

- **Actividades:** Realizar las actividades complejo 1 a 3 del cuaderno de actividades schema.

Indicadores

- Podemos controlar cómo los elementos se van a utilizar en los documentos con indicadores.

Hay siete indicadores:

Indicadores de orden:

Todo

Elección

Secuencia

Indicadores de ocurrencia:

maxOccurs

minOccurs

Indicadores de grupo:

Nombre del grupo

attributeGroup nombre

Indicadores de orden: all

- Se utiliza para definir el orden de los elementos

ALL	XSD
Los elementos que contiene pueden aparecer en cualquier orden, pero como máximo sólo una vez	<pre><xs:element name="persona"> <xs:complexType> <xs:all> <xs:element name="nombre" type="xs:string"/> <xs:element name="apellidos" type="xs:string"/> </xs:all> </xs:complexType> </xs:element></pre>

Indicador de orden: elección

ALL	XSD
<p>Puede aparecer sólo uno de los elementos que contiene</p>	<pre><xs:element name="persona"> <xs:complexType> <xs:choice> <xs:element name="empleado" type="empleado"/> <xs:element name="miembro" type="miembro"/> </xs:choice> </xs:complexType> </xs:element></pre>

Indicador de orden: secuencia

ALL	XSD
Los elementos secundarios deben aparecer en un orden específico	<pre><xs:element name="persona"> <xs:complexType> <xs:sequence> <xs:element name="nombre" type="xs:string"/> <xs:element name="apellidos" type="xs:string"/> </xs:sequence> </xs:complexType> </xs:element></pre>

Indicadores de ocurrencia: maxOccurs y minOccurs

- Especifica el número máximo y mínimo de veces que puede aparecer un elemento hijo de un elemento complejo. El atributo maxOccurs puede tomar el valor “unbounded”, que indica que no existe ningún límite

XSD

```
<xsd:element name="persona">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="nombre" type="xsd:string"/>
      <xsd:element name="nombre_hijo" type="xsd:string" maxOccurs="10"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Ejemplo: familia.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<personas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="familia.xsd">

  <persona>
    <nombre>Hege Refsnes</nombre>
    <nombre_hijo>Cecilie</nombre_hijo>
  </persona>

  <persona>
    <nombre>Tove Refsnes</nombre>
    <nombre_hijo>Hege</nombre_hijo>
    <nombre_hijo>Stale</nombre_hijo>
    <nombre_hijo>Jim</nombre_hijo>
    <nombre_hijo>Borge</nombre_hijo>
  </persona>

  <persona>
    <nombre>Stale Refsnes</nombre>
  </persona>
</personas>
```

Ejemplo: familia.xsd

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
            elementFormDefault="qualified">

  <xs:element name="personas">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="persona" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="nombre" type="xs:string"/>
              <xs:element name="nombre_hijo" type="xs:string"
                minOccurs="0" maxOccurs="5"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

- **Actividades:** Realizar las actividades Indicadores 1 a 4 del cuaderno de actividades schema.

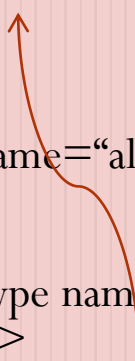
Indicadores de Grupo

- Se utiliza para definir conjuntos relacionados de elementos.

```
<xs:group name="nombre_grupo">  
...  
</xs:group>
```

Se debe usar en ALL, CHOICE o
SEQUENCE

```
<xs:group name="grupoAlumno">  
  <xs:sequence>  
    <xs:element name="nombre" type="xs:string"/>  
    <xs:element name="apellido" type="xs:string"/>  
    <xs:element name="cumpleaños" type="xs:date"/>  
  </xs:sequence>  
</xs:group>  
  
<xs:element name="alumno" type="infoAlumno"/>  
  
<xs:complexType name="infoAlumno">  
  <xs:sequence>  
    <xs:group ref="grupoAlumno"/>  
    <xs:element name="ciudad" type="xs:string"/>  
  </xs:sequence>  
</xs:complexType>
```

A curved arrow originates from the `<xs:group ref="grupoAlumno"/>` line in the `infoAlumno` complex type definition and points to the `</xs:group>` line of the `grupoAlumno` group definition above it.

Indicadores de Grupo de Atributos

- Se utiliza para definir conjuntos relacionados de atributos.

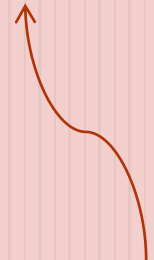
```
<xs:attributeGroup ame="nombregrupo">
```

```
...
```

```
</xs:attributeGroup>
```

```
<xs:attributeGroup name="grupoAtributosAlumno">  
  <xs:attribute name="nombre" type="xs:string" />  
  <xs:attribute name="apellido" type="xs:string" />  
  <xs:attribute name="cumpleaños" type="xs:date" />  
</xs:attributeGroup>
```

```
<xs:element name="alumno">  
  <xs:complexType>  
    <xs:attributeGroup ref="grupoAtributosAlumno" />  
  </xs:complexType>  
</xs:element>
```



Modelo de contenido: ANY

- Nos permite incluir elementos no declarados inicialmente en el documento XML

```
<xsd:element name="person">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="firstname" type="xsd:string"/>
      <xsd:element name="lastname" type="xsd:string"/>
      <xsd:any minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Podemos añadir un nuevo elemento en el documento XML después de lastname

```
<person>
  <firstname>Hege</firstname>
  <lastname>Refsnes</lastname>
  <children>
    <childname>Cecilie</childname>
  </children>
</person>
```

```
<person>
  <firstname>Stale</firstname>
  <lastname>Refsnes</lastname>
</person>
```

El modelo de contenido: anyAttribute

- Nos permite incluir atributos no declarados inicialmente en el documento XML

ACTIVIDADES

- **Actividades:** Realizar las actividad Schema1 del cuaderno de actividades schema.