Lenguajes de Marcas y Sistemas de Gestión de la Información

DEFINICION DE ESQUEMAS Y VOCABULARIOS EN XML

Objetivos:

- Describir la estructura de un documento XML con DTD.
- Conocer los elementos de los que se compone una DTD.
- Definicion de entidades de una DTD.
- Creación y asignaciómn de atributos a un elemento desde una DTD.
- Asignación de una DTD a un documento XML.
- Describir la estructura de un documento XML con un esquema.
- Conocer los elementos de los que se compone un esquema.
- Creación y asignación de atributos a un elemento desde un esquema.

Objetivos:

- Asignación de un esquema a un documento XML.
- Tipos básicos de los elementos de un esquema.
- Asignación de elementos hijos con modificación de ocurrencias.
- Distinguir entre documento bien formado y documento válido.
- Conocer qué herramientas de validación existen via web o aplicación local.

❑ Introducción:

- XML ha sido propuesto como un **estándar en el intercambio de información, independientemente de la plataforma** en la que se genere o emplee.
- Esto **funciona** siempre y cuando la **estructura de los documentos XML no cambien** entre intercambios.
- Veremos dos maneras de **especificar formatos** de comunicacion en XML: las **DTDs** y los **Esquemas.**

❑ DTD (Document Type Definition):

- Es la definición de cómo se construye un documento XML.
- Establece qué elementos son aceptados y en qué posiciones deben aparecer en un documento XML.
- Al definir una DTD establecemos dos tipos de relaciones:
  - qué lexico se espera.
  - qué reglas sintácticas debe cumplir nuestro lenguaje.
- Antes de crear un documento XML se debe analizar qué elementos y etiquetas se van a necesitar para la aplicación que estemos creando.

¿Por qué es importante la creación de DTDs?

- Porque pretendemos asegurar que la informacion que compartiremos mediante el documento XML sea comprendida por aquellas aplicaciones que la reciban.
- Por esto debemos asegurar que el documento XML esté bien formado y sea válido.

¿Donde declararemos la DTD?

- Podemos ubicar la **declaración** de la DTD :
  - en el propio **documento XML.**
  - En un **fichero externo**.
- **Recomendable**, en un **fichero externo**:
  - Por **economía de esfuerzos** de desarrollo: si el DTD está en el mismos documento y tenemos varios, cada modificación en la DTD supone modificar todos los documentos.
  - Por **economía de comunicaciones y procesado**: un archivo XML con la DTD incluida es mas pesado.

¿Donde declararemos la DTD? Ejemplos

- **Declaración** en el propio **documento XML**:

<!DOCTYPE **ElementoRaiz [**

declaraciones de elementos

...

]>

- **Declaración** en un **fichero externo.**

<!DOCTYPE **ElementoRaiz** SYSTEM **"URL_al_DTD.dtd"**>

Bloques para construir una DTD:

- **Elemento**: bloque principal con el que se contruye la DTD.
- **Atributo**: elemento de información añadida al elemento.
- **Entidad**: En XML existen algunos caracteres que tienen significado especial. Para poder emplearlos como datos del documetno, tendremos que sustituirlos por un codigo. Ejemplos:
  -   es espacio
  - > es >
  - ...

❑ Bloques para construir una DTD:

- también podemos **definir nuestras propias entidades**, de forma que el parseador realice la sustitucion:
  - **<!ENTITY mejorEquipo "Ese Betis">**, define que toda ocurrencia en el documento de **&mejorEquipo;** sea sustituida por la cadena **"Ese Betis"**.
- **PCData** (Parsed Character Data), indica que entre las etiquetas de apertura y cierre se almacenará texto y será analizado por el prseador.
- **CData** (Character Data), igual que PCData, pero el texto contenido no será analizado por el parseador.

Declaración de elementos:

- Los elementos se declaran de una de las dos siguientes maneras:
  - <!ELEMENT **nombreElemento categoria**>
    - Categoría puede tomar los valores:
      - **EMPTY**, indica elemento vacío.
      - **(#PCDATA)**
      - **(#CDATA)**
      - **(#ANY)**, indica cualquier elemento valido.
  - <!ELEMENT **nombreElemento (nodosHijos)**>
    - En este caso, se puede definir **nombreElemento** como el nodo del que cuelgan los nodos hijos descritos por **nodosHijos**.

Declaración de atributos:

- Los elementos se declaran de la siguiente manera:
  - <!ATTLIST **nombreElemento nombreAtributo tipoAtributo valorXDefecto modificador**>
  - **nombreElemento**, es el nombre del elemento al que le estamos asociando el atributo
  - **nombreAtributo**, es el nombre del atributo.
  - **tipoAtributo**, puede tomar los valores:
    - **CDATA**, texto que incluya cualquier caracter.
    - **ID**, identificador unico del elemento en el documento XML.
    - **IDREF**, identificador de otro elemento del propio documento XML.

❑ Declaración de atributos:

- **IDREFS**, lista de identificadores a otros elementos
- **tipo1 | tipo2 | tipo3 ...**, el valor es uno de los indicados en esta lista
- **NMTOKEN**, texto que solo contendrá letras digitos, "-", "_", "." o ":". Es lo que se llama **nombres válidos XML.**
- **NMTOKENS**, es una lista de nombres validos XML. Es como NMTOKEN, pero se incluyen los espacios en blanco, tabuladores y retornos de carro.
- **ENTITY**, el atributo es una entidad declarada anteriormente.

- **Declaración de atributos:**
  - **ENTITITES**, es una lista de entidades.
  - **modificador**, los atributos podrán ser declarados como:
    - **#REQUIRED**, obligatorio.
    - **#IMPLIED**, optativo.
    - **#FIXED**, de valor fijo.
  - **Ejemplo** de declaracion de atributos:
    - <! ATTLIST hora zona CDATA “GMT+1” #REQUIRED>

Secuencias de elementos: Estructuras con hijos

- En nuestro ejercicio del documento XML con datos de libros, cabía la posibilidad que un libro tuviese mas de un autor. Un ejemplo de documento XML basado en esa definición es:

```
<libro>
    <titulo>Lenguajes de Marcas</titulo>
    <autores>
        <autor>Javier S. Zurdo</autor>
        <autor>Pablo Toharia</autor>
    </autores>
    <editorial>Ra-Ma</editorial>
    <fechaPublicacion>2011</fechaPublicacion>
<libro>
```

Secuencias de elementos: Estructuras con hijos

- Su correspondiente DTD sería, según hemos visto hasta ahora:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE libro [
<!ELEMENT libro (titulo, autores, editorial, fechaPublicacion)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT autores (autor)>
<!ELEMENT autor (#PCDATA) >
<!ELEMENT editorial (#PCDATA)>
<!ELEMENT fechaPublicacion (#PCDATA)>
]>
```

Secuencias de elementos: Estructuras con hijos

- Si ponemos definicion y datos en el mismo documento y validamos:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE libro [
<!ELEMENT libro (titulo, autores, editorial, fechaPublicacion)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT autores (autor)>
<!ELEMENT autor (#PCDATA) >
<!ELEMENT editorial (#PCDATA)>
<!ELEMENT fechaPublicacion (#PCDATA)>
]>
<libro>
    <titulo>Lenguajes de Marcas</titulo>
    <autores>
        <autor>Javier S. Zurdo</autor>
        <autor>Pablo Toharia</autor>
    </autores>
    <editorial>Ra-Ma</editorial>
    <fechaPublicacion>2011</fechaPublicacion>
</libro>
```

Secuencias de elementos: Estructuras con hijos

- Nos encontraremos con el siguiente error de validación
- Este error es debido a que **no hemos indicado que el elemento autor puede aparecer varias** veces en el documento.
- Para indicar el numero de ocurrencias de un elemento, tenemos las siguientes opciones:
- "+" indica que puede haber una o mas ocurrencias.
- "*" indica que puede haber cero o mas ocurrencias.
- "?" indica que puede haber cero o una ocurrencia.
- El que aparezca el nombre del elemento indica que este debe aparecer una sola vez.

Secuencias de elementos: Estructuras con hijos

- Para solucionarlo, basta con modificar el DTD para que la definición del elemento autores quede de la siguiente forma:

```
<!ELEMENT autores (autor+)>
```

Esquemas XML

- Un **esquema XML** (XML Schema o **XSD** – XML Schema Definition) es un lenguaje que sirve como alternativa al uso de DTD en la declaración de gramáticas XML.
- XSD nació en 1998 y **desde el 2001 el W3C ha recomendado su uso**.
- Sus principales **características** son:
- Define qué **elementos y atributos** pueden aparecer en un documento XML
- Define qué **elementos son compuestos**, indicando qué elementos hijos deben aparecer y en qué orden.
- Define qué **tipos** se pueden emplear en cada elemento o atributo.
- Define la **obligatoriedad** o no de los elementos o atributos.

Esquemas XML

- Se trata de la evolución natural de los DTDs y está basado en XML, por lo que presenta grandes ventajas sobre el primero:
  - Al estar basado en XML, es facilmente extensible a las futuras modificaciones o necesidades que se identifiquen.
  - Permite definir de manera muy clara los tipos de datos y los espacios de nombres.
  - Se pueden definir los tipos exactamente igual que si se tratase de una base de datos, facilitando la conversion de uno a otro y las transferencias de información.

Componentes de un esquema XML.

- Los documentos de esquema XML pueden contener los siguientes elementos:
  - Elemento raíz
  - Elementos simples.
  - Atributos
  - Restricciones
  - Elementos complejos
  - Secuencias de elementos

Componentes de un esquema XML. Elemento raíz

- Veamos un ejemplo de archivo de esquema XML:

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

<xs:schema xmlns:xs="http://www.fpada.com/XMLSchema"
targetNamespace="http://www.fpada.com"
xmlns="http://www.fpada.com"
elementFormDefault="qualified">

...

...

</xs:schema>
```

Componentes de un esquema XML. Elemento raíz

- El elemento **<schema>** es el elemento raiz de todo archivo de esquema XML.
- Puede contener algunos atributos interesantes:
  - xmlns:xs="http://www.fpada.com/XMLSchema". Este ya lo conocemos, se trata de la reserva de espacio de nombres para los elementos del esquema.
  - targetNamespace="http://www.fpada.com" indica que los elementos del esquema pertenecen al espacio de nombres http://www.fpada.com
  - xmlns="http://www.fpada.com" indica el espacio de nombres por defecto
  - elementFormDefault="qualified" indica que cualquier elemento definido en este esquema debe tener un espacio de nombres asociado.

Componentes de un esquema XML.. Elementos simples

- Un elemento simple es aquel que solo contiene información, pero no otros **elementos ni atributos**.
- La sintaxis de un elemento es:

```
<xs:element name="nombre_elemento" type="tipo_elemento"/>
```

- Los elementos pueden ser de los siguientes tipos:
  - **xs:string**, para representar texto.
  - **xs:date**, para representar fechas, en formato YYYY-MM-DD incluyendo o no la zona horaria. Ejemplos:

    ```
    <nacimiento>2013-01-05</nacimiento>
    ```

    ```
    <fundacion>2013-01-05+1:00</nacimiento>
    ```

Componentes de un esquema XML.. Elementos simples

- **xs:time**, para almacenar horas en el formato hh:mm:ss, tambien podemos indicar la zona horaria. Ejemplos:

  <horaInicio>07:00:00</horaInicio>

  <horaInicio>07:00:00+1:00</horaInicio>

- **xs:dateTime**, para almacenar una cadena de fecha y hora, segun el formato YYYY-MM-DDThh:mm:ss, donde T indica el comienzo de la parte de la hora. Ejemplos:

  <fecha>2012-05-02T08:00:00</fecha>

- **Componentes de un esquema XML.. Elementos simples**
  - **Tipos numericos, los mas empleados:**
    - **xs:decimal**, para indicar un valor numérico.

      Ejemplo:

      <precio>26.99</precio>
    - **xs:integer**
    - **xs:boolean**, para almacenar valores **true** o **false**
  - **Tipos binarios. hay dos, y se emplean para almacenar documentos o formatos binarios:**
    - **xs:hexBinary** (datos binarios codificados en base hexadecimal)
    - **xs:base64Binary** (datos binarios codificados en base 64)

Componentes de un esquema XML.. Atributos

- Los atributos son complementos de informacion que se pueden asignar a un elemento previamente declarado.
- Su sintaxis es:

<**xs:attribute name**="nombre_atributo" **type**="tipo_atributo"/>

donde **tipo_atributo** es de cualquiera de los tipos que hemos visto para los elementos.

- Ejemplo:

<xs:attribute name="Nombre" type="xs_string"/>

- **Componentes de un esquema XML.. Elementos complejos**
  - Los elementos complejos son todos aquellos que:
    - contienen otros elementos, y /o
    - contienen atributos.
  - La **definicion de un elemento complejo se realiza** como sigue:
    1. se define el elemento
    2. se indica que es un elemento complejo mediante la etiqueta <xs:complexType>
    3. Se añade un elemento <xs:sequence> y dentro de este, se definen el resto de elementos y atributos de los que se componga el elemento.

Componentes de un esquema XML.. Elementos complejos

- Ejemplo:

```
<profesor>
    <nombre>Bacterio</nombre>
    <asignatura>Tecnología</asignatura>
</profesor>
```

Componentes de un esquema XML.. Elementos complejos

- Su definición sería:

```
<xs:element name="profesor">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="nombre" type="xs:string"/>
            <xs:element name="asigantura" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Componentes de un esquema XML.. Secuencias

- Hemos visto la etiqueta <xs:sequence> para definir los elementos complejos.
- Existen **indicadores** que nos permiten indicar cuantas veces deben aparecer determinados elementos.
- Un **indicador** permite determinar los siguientes patrones:
  - El orden en el que se establecen los elementos hijos.
  - El numero de veces que aparecen
  - El grupo al que pertenecer los elementos hijos.

Componentes de un esquema XML.. Secuencias

- **Indicadores de orden**:
  - <xs:all>, indica que los elementos hijos pueden aparecer en cualquier orden, pero solo pueden aparecer una vez.
  - <xs:Choice>, especifica que de entre los elementos hijos solo puede aparecer uno u otro.
  - <xs:sequence>, permite indicar el orden especifico en el que deben aparecer los elementos hijos.
  - Todos ellos se pueden emplear como hijos de xs:complexType

Componentes de un esquema XML.. Secuencias

- **Indicadores de ocurrencia**:
  - <xs:maxOccurs>, indica el máximo número de veces que un elemento hijo puede aparecer.
  - <xs:minOccurs>, indica el mínimo número de veces que un elemento hijo puede aparecer.
  - En ambos indicadores el valor por defecto es 1. En caso que se quiera indicar que se puede repetir de forma ilimitada, indicaremos el valor **unbounded**.

Componentes de un esquema XML.. Secuencias

- **Indicadores de grupo**:
  - Los indicadores de grupo permiten establecer un conjunto de elementos asociados entre sí.
  - Podemos definir un grupo de elementos para posteriormente, emplearlos en otra parte del documento.
  - por ejemplo, si queremos representar las caracteristicas de un coche (marca, modelo, caballos), podríamos escribirlo de la siguiente manera:

Componentes de un esquema XML.. Secuencias

- **Indicadores de grupo**:

```
<xs:group name="grupoCoche">
    <xs:sequence>
      <xs:element name="marca" type="xs:string"/>
      <xs:element name="modelo" type="xs:string"/>
      <xs:element name="cvs" type="xs:string"/>
    <xs:sequence>
</xs:group>
...
```

Componentes de un esquema XML.. Secuencias

- **Indicadores de grupo**:

```
<xs:element name="coche">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="añoFab" type="xs:string"/>
            <xs:groupref="grupoCoche"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

- **Componentes de un esquema XML.. Restricciones:**
  - Sirven para restringir los rango de valores que pueden tomar los atributos de un elemento.
  - La forma de definirlos es sencilla: después de la definición del atributo, definiremos un nuevo elemento simpleType, que contendrá la restricción (elemento restriction).
  - Ejemplo:

```
<xs:atribute name="nota">
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="10"/>
        </xs:restriction>
    <xs:simpleType>
</xs:atrribute>
```

Componentes de un esquema XML.. Restricciones:

- Hay varios tipos de restricciones, segun el tipo del atributo.
- para el tipo xs:string
  - xs:length
  - xs:minLength
  - xs:maxLength
- para el tipo xs:decimal
  - xs:maxExclusive o xs:maxInclusive
  - xs:minExclusive o xs:minInclusive

Componentes de un esquema XML.. Restricciones:

- Hay otra restriccion interesante, que se nos posibilita establecer una lista de valores permitidos, empleando la palabra clave enumeration:

```
<xs:attribute name”sexo”>
    <xs:simpleType>
        <xs:restriction>
            <xs:enumeration value=”hombre”/>
            <xs:enumeration value=”mujer”/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
```

Componentes de un esquema XML.. Restricciones:

- Incluso podemos emplear otro tipo de restricciones, como son las expresiones regulares, como por ejemplo

  <xs:pattern value="[A-Z]"/>

- Mas sobre restricciones en:

  https://www.w3schools.com/xml/schema_facets.asp

Referencia:

XML Schema Tutorial:
https://www.w3schools.com/xml/schema_intro.asp