# Tecnologías Multimedia, Curso 10/11 - Práctica 6 Lenguajes para la Web

## **Objetivos:**

Se trata de ofrecer una visión a nivel práctico de los lenguajes de marcado y de presentación utilizados para la publicación de documentos en la web. Dado el poco tiempo disponible y la vasta cantidad de tecnologías a explorar se ha pretendido que el alumno adquiera una "cultura general" acerca de las distintas posibilidades que estos lenguajes proporcionan, de modo que más adelante pueda entrar en mayor profundidad en las tecnologías que más le interesen.

## Desarrollo:

Antes de comenzar el desarrollo de la práctica, descarga e instala el programa  $\underline{XML-Spy\ Editor}^1$ . Se trata de una versión de evaluación gratuita que debes activar mediante una clave que te envían por email, y que dura 30 días.

## **Documentos XML bien formados**

Vamos a utilizar el lenguaje XML para crear una base de datos de películas, definiendo nosotros mismos los campos que nos interesa guardar de cada una. En este caso, elegimos, como en el ejemplo de teoría, "título", "audiencia" y "comentario". Un ejemplo de esta base de datos podría ser el siguiente.

**NOTA:** A la hora de diseñar un formato XML para nuestra base de datos, hemos elegido que los campos de la película sean elementos anidados. También podría haberse hecho que fueran atributos del elemento película, por ejemplo: <película titulo="Rambo">. Ambas alternativas son correctas en XML, pero no intercambiables. En nuestro caso debemos ceñirnos al modelo anidado, que es el elegido.

- Abre el editor <u>Notepad++</u><sup>2</sup>, elige Archivo->Nuevo y asegúrate de que en el menú "Formato" la opción elegida es "Codificar en UTF-8 sin BOM"
- Copia el código de ejemplo dado más arriba. Si te apetece puedes cambiar los datos de las películas por otras. Guárdalo como pelis1.xml
- Visualiza el ejemplo anterior con Internet Explorer (por ejemplo, arrastra el fichero y suéltalo en el navegador). Observarás que en lugar de la "ñ" aparece "C1". Tampoco salen correctamente los caracteres acentuados. Esto se debe a que en la primera línea del fichero hemos puesto encoding="ascii", lo que no es cierto porque Notepad++ lo ha guardado en utf8 (además, en ascii no sería posible introducir la eñe ni las vocales con acentos)

<sup>&</sup>lt;sup>1</sup> <u>http://www.altova.com/xml-editor</u>

<sup>&</sup>lt;sup>2</sup> <u>http://notepad-plus-plus.org/</u>

- Modifica el encoding y pon "utf-8". Guárdalo como pelis2.xml. Visualiza de nuevo y comprueba que se ve bien. También puedes usar Firefox para visualizarlo (el cual además te permite "colapsar" o "expandir" cada elemento XML en el documento).
- Si un documento está bien o mal formado se puede comprobar con el editor XMLSpy. Para ello abre tu archivo .xml con dicho editor y pulsa la tecla F7. También puedes comprobarlo con el navegador Internet Explorer, o con Firefox. Cualquiera de los dos te muestra un error si detecta que el xml está mal formado.
- Introduce un error en algún tag. Por ejemplo, pon </pelucula> en lugar de la primera</película>. Guárdalo como pelis3.xml. Verifica si está bien formado con las herramientas mencionadas.
- Haz ahora que el correspondiente tag de apertura sea también <pelucula>. Guárdalo como pelis4.xml. Comprueba si está bien formado. En esta ocasión, al estar las etiquetas correctamente balanceadas, ¡no hay error! Es XML bien formado, y el navegador no puede saber si <pelucula> o <pelicula> son elementos válidos o no, ya que el documento no referencia ningún DTD (Document Type Definition).

## Documentos XML válidos

- Vamos a definir qué formato sería el válido para nuestra base de datos. La especificación sería la siguiente:
  - 1. El elemento raíz del documento es lista\_de\_peliculas
  - 2. El elemento raíz podrá contener cero o más elementos película.
  - 3. Cada elemento película contendrá obligatoriamente los elementos titulo y audiencia y opcionalmente un elemento comentario
- Esta especificación se escribiría en un fichero (llamémosle peliculas.dtd) con el siguiente contenido.



• Copia el fichero anterior a la misma carpeta que el fichero .xml inicial. Modifica pelis2.xml para añadirle la siguiente línea (debajo de la primera), y guárdalo como pelis5.xml:

<!DOCTYPE lista\_de\_peliculas SYSTEM "peliculas.dtd">

- Carga pelis5.xml en XMLSpy y comprueba si es XML válido (pulsando F8).
- Modifica pelis4.xml para asociarlo con peliculas.dtd y prueba a validarlo con XML.
- Juega con el documento probando a violar alguna de las restricciones. Por ejemplo ¿puede una película tener dos títulos? ¿Y dos comentarios? ¿Y cero comentarios? ¿Pueden ponerse en diferente orden el titulo y la audiencia? ¿Qué pasa si omites el elemento raíz lista\_de\_películas? Etc.
- Usa XMLSpy para editar pelis5.xml y añadir una película más a la base de datos. Gracias a que el documento tiene un DTD asociado, XMLSpy sabe qué elementos puedes poner y cuáles no, por lo que te ayuda durante la edición del documento. Compruébalo.
- En un DTD, el carácter "\*" significa que el elemento en cuestión puede aparecer cualquier número de veces (incluso cero), como el caso del elemento película. El carácter "?" significa que es opcional, es decir, puede aparecer cero veces o una vez, como el caso del elemento comentario. El carácter "+" indicaría que debe aparecer al menos una vez, pero puede repetirse más veces (no hay ejemplo de este caso en el DTD anterior). Puedes probar a usar estos caracteres. Por ejemplo, haz que se admitan varios comentarios por película. Genera un xml válido de acuerdo con el nuevo DTD.

#### Uso de esquemas: XML Schema

 Otra forma de especificar la sintaxis válida de un xml, aparte de usar un DTD, es hacerlo mediante un fichero Schema. Este fichero es a su vez xml<sup>3</sup>. Copia el siguiente código y guárdalo en un archivo llamado películas.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="lista_de_peliculas">
      <xs:complexType>
         <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:element name="pelicula">
               <xs:complexType>
                  <xs:sequence>
         <xs:element name="titulo" type="xs:string"/>
         <xs:element name="audiencia" type="xs:string"/>
         <xs:element name="comentario" type="xs:string" minOccurs="0"/>
         </r></r></r>
               </r></r></r>
  </r></r></r>
         </xs:sequence>
      </xs:complexType>
   </r></r></r>
</xs:schema>
```

- Estudia un momento el fichero anterior e intenta entender "por encima" su significado (no es muy complicado, si piensas que está especificando el mismo formato del que ya hablamos en el DTD)
- Cárgalo en XMLSpy y verifícalo (así nos aseguramos de que no se nos ha colado un error de sintaxis al copiarlo, y que es un Schema válido). Al cargarlo, verás que no se muestra el código del mismo (se muestra de otra forma, más cómoda para escribir Schemas, que veremos más adelante). Pero en este momento nos interesa ver el código, de modo que cambia la vista mediante el menú View->Text View. Pulsa después F8 para validarlo.
- Modifica pelis5.xml y guardálo como pelis6.xml para que en lugar de asociarse con el DTD del apartado previo lo haga con el .xsd que acabas de escribir. Para ello elimina el <!DOCTYPE> y modifica el elemento raíz <lista\_de\_peliculas> para darle los atributos siguientes:

```
de_peliculas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="peliculas.xsd" >
```

o Abre el documento con XMLSpy y valídalo (F8).

#### Creación de un Schema

Como has podido comprobar, escribir un Schema "a mano" es tedioso y proclive a errores, ya que es una gran cantidad de texto el que hay que escribir, incluso para un formato tan simple como el de nuestra base de datos de películas. Las herramientas de edición de XML como XMLSpy son de gran ayuda en este caso. De hecho, XMLSpy posee incluso un modo en el cual los ficheros Schema se pueden crear "gráficamente" en lugar de tener que escribirlos.

Los siguientes pasos te muestran cómo crear un Schema como el anterior, pero con el editor gráfico.

- Abre XMLSpy y crea un nuevo documento. Te aparece una lista para que elijas el tipo de documento a crear. Selecciona "xsd W3C XML Schema".
- o Aparece una pantalla como la siguiente

<sup>&</sup>lt;sup>3</sup> Lo que implica que existe un DTD que especifica Schema. (<u>http://www.w3.org/TR/xmlschema-1/#nonnormative-schemaDTD</u>) ¡Incluso existe un fichero Schema que especifica Schema! (http://www.w3.org/TR/xmlschema-1/#normative-schemaSchema)

31 문

elen

- Pulsa en el pequeño símbolo con forma de estructura-árbol, a la izquierda de la palabra "element" (señalado con una flecha en la figura). Esto cambia la visualización al modo gráfico.
- Donde aparece ENTER\_NAME\_OF\_ROOT\_ELEMENT\_HERE debe escribirse el nombre del elemento raíz: lista\_de\_peliculas
- Seleccionar el elemento raíz con el botón derecho del ratón, elegir la opción Add Child y dentro de ella Sequence.
- En el nuevo nodo que aparece a la derecha, activa de nuevo la opción Add Child y dentro de ella Element. Se debe introducir el nombre del elemento hijo de lista\_de\_peliculas, que es pelicula.
- Añáde al elemento pelicula un hijo de tipo secuencia.
- Añáde al nodo secuencia recién creado tres hijos de tipo elemento. Sus nombres serán titulo, audiencia y comentario respectivamente.
- Selecciona el elemento pelicula con el botón derecho y activa las opciones Optional (para indicar que puede haber cero películas en la lista) y Unbounded (que puede haber un número potencialmente infinito de películas en la lista).
- Para los tres elementos al final del árbol (titulo, audiencia y comentario) se debe seleccionar el tipo xs:string en la ventana Details que aparece a la derecha. Para ello, se pulsa sobre el nodo del elemento y luego se selecciona el tipo en esa ventana (se puede seleccionar pulsando sobre el triángulo invertido de la fila "type" y eligiendo el tipo dentro de todos los disponibles o escribiendo en la caja de texto directamente).

Details	×
name	xs:QName
isRef	xs:short
minOcc	xs:string
max0cc	xs:time 🛛 💌 🔤
type	<b></b>
nillable	
block	
form	
hid	<u> </u>
Details	

- Hay que activar la opción Optional en el elemento comentario.
- El aspecto final del diagrama debería ser el siguiente:



- Una vez finalizado el diagrama, activa View->Text view y podrás ver la versión xml del Schema que acabas de crear. Éste es prácticamente idéntico al que habíamos escrito antes "a mano", excepto que hay un elemento xs:documentation anidado dentro de otro xs:annotation, insertados automáticamente por XMLSpy para añadir comentarios acerca del esquema. Puedes eliminarlos y si después cambias a la vista "Schema Design View" podrás comprobar que el comentario ha desaparecido.
- Abre el documento con XMLSpy y valídalo (F8).

#### Presentación de un XML (estilo)

- Si abres pelis2.xml en un navegador web (IE o Firefox) te mostrará el código xml, puesto que por defecto el navegador no sabe cómo debe ser mostrado cada elemento que contiene.
- El documento xml puede especificar una "hoja de estilos" que indique al navegador cómo mostrar cada elemento. Esta asociación se hace añadiendo al fichero la siguiente línea debajo de la primera (guárdalo como pelis7.xml):

```
<?xml-stylesheet type="text/css" href="estilo.css"?>
```

• Crea ahora el fichero estilo.css con el siguiente contenido:

```
lista_de_peliculas {
  background-color: #ffffff;
  width: 100%;
}
pelicula {
 display: block;
 margin-bottom: 30pt;
 margin-left: 0;
}
titulo {
  color: #FF0000;
  font-size: 20pt;
}
audiencia {
  display:block;
  color: #0000FF;
  font-size: 20pt;
}
comentario {
  display: block;
  color: green;
  margin-left: 20pt;
```

- Abre pelis7.xml con el navegador web y comprueba cómo ahora se muestra de forma muy diferente.
- Experimenta con el fichero estilo.css, dando diferente tamaño al título, diferente color a los comentarios, etc.
- Observa que en este ejemplo hemos usado un .xml que no contiene información de validación (ni DTD ni Schema) para mostrar que validación y presentación son aspectos independientes.

## HTML y XHTML

- Toma un fichero html cualquiera que hayas escrito para alguna de las sesiones previas, o para cualquier otro cometido. Mejor si es un html sencillo que no tenga objetos multimedia incrustados, como el que puedes encontrar en el apartado de material de prácticas del sitio web. Cambia su extensión a .xhml.
- Modifica la cabecera de este fichero para que sea xml (sería en concreto xhtml). Para ello el fichero debe comenzar en la forma siguiente:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Además, el tag <html> debe ser cambiado por

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">

- Tras hacer estos cambios, prueba a visualizar ese xhtml en un navegador. Posiblemente lo puedas visualizar exactamente igual que antes, pues los navegadores son muy tolerantes con los errores, pero es muy probable que tu documento no sea en realidad válido, es decir, que no cumpla exactamente con las restricciones impuestas por su DTD.
- Carga ese fichero en XMLSpy y trata de validarlo (F8).
- Si no valida, corrige todos los errores hasta convertirlo en xhtml válido.

## Conversión automática de HTML a XHTML

• Prueba el servicio on-line de conversión <u>http://www.it.uc3m.es/jaf/html2xhtml/</u> y convierte el archivo de ejemplo html del apartado anterior a xhtml 1.0.

## Validación On-line de XHTML y CSS

- Analiza el sitio web http://www.epigijon.uniovi.es/oldweb/ y valída su formato xhtml a través del servicio on-line http://validator.w3.org/.
- Valida el formato de presentación css del sitio web anterior a través del servicio on-line <u>http://jigsaw.w3.org/css-validator/</u>.
- Una vez validados los formatos habrás comprobado que se proponen códigos para insertar en el sitio web de modo que pinchando en los iconos correspondientes se pueda validar el sitio.
- Comprueba que los códigos anteriores están efectivamente insertados al final del código fuente del sitio analizado.

## SVG (Scalar Vector Graphics)

 SVG es un lenguaje (de la familia xml) para especificar gráficos bidimensionales. Al ser xml puedes "escribir" un dibujo usando el bloc de notas. Por ejemplo, el código siguiente dibuja una señal de "dirección prohibida".

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20000303 Stylable//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<!-- Dibujo simple, por JL Diaz -->
<svg xmlns="http://www.w3.org/2000/svg" version="1.1"
width="210" height="430" >
    <rect width="40" height="214" x="80" y="200"
        stroke="black" stroke-width="4" fill="grey" />
        <circle cx="100" cy="100" r="100"
        stroke="black" stroke-width="4" fill="red"/>
        <rect width="110" height="50" x="45" y="75"
        stroke="black" stroke-width="2" fill="white"/>
</svg>
```

Aun sin conocer el lenguaje SVG, es fácil comprender este código, al menos en parte. Vemos que se compone de un rectángulo gris (el palo de la señal), un círculo rojo (el disco de la señal) y un rectángulo blanco (la barra horizontal de la dirección prohibida).

- Copia el código anterior en un fichero llamado sign.svg. Abre este fichero con XMLSpy e intenta validarlo. Notarás que tarda mucho más de lo habitual. Esto es debido a que en este caso el DTD que especifica la sintaxis válida no es un archivo local, sino que debe descargarlo de Internet, de la URI que se proporciona en el Doctype.
- Si quieres ver el dibujo que resulta de ese código, necesitarás un visualizador de SVG. En la actualidad, los principales navegadores Web incorporan la capacidad de interpretar SVG y mostrar el resultado, de modo que puedes usar un navegador para verlo.

- El lenguaje SVG es mucho más potente de lo que sugiere el mínimo ejemplo anterior. Tiene soporte para transparencia, gradientes, etc. Naturalmente no está pensado para ser escrito "a mano", sino para ser generado por herramientas gráficas, como por ejemplo <u>Inkscape</u><sup>4</sup> (gratuita, de código abierto y multiplataforma). Puedes ver ejemplos de gráficos creados con esta herramienta en <u>http://www.inkscapegallery.net/en/galleries/graphics</u>.
- o Descarga alguno de los ejemplos y edítalo mediante con Notepad++

<sup>&</sup>lt;sup>4</sup> <u>http://www.inkscape.org</u>