

Lenguajes de marcas y sistemas de gestión de la información (LMSGI)

Observaciones sobre los contenidos de la UD1 de la plataforma a distancia: *Reconocimiento de las características de lenguajes de marcas*

1 Editores para XML

En los contenidos de la unidad se menciona el editor [Amaya](#) parece estar estancado desde 2012. En realidad, al ser texto, os podría servir cualquier editor, pero existen herramientas que os facilitarán la tarea.

En este [vídeo](#) os explican cómo utilizar el editor Notepad++ con el plugin XML. Aunque el vídeo al inicio menciona que hay solo algunas versiones compatibles, yo he probado con la versión 7.9 de 64 bits y tanto el Plugin Manager y como el plugin XML Tools funciona correctamente en Windows 10.

De todas formas, si estáis familiarizados con IDEs (entornos de desarrollo integrados) como [Eclipse](#), [NetBeans](#), etc., aunque son herramientas mucho más potentes más orientadas a la programación, por supuesto también os facilitarán la tarea.

Por último, existen herramientas más especializadas como [Oxygen XML Editor](#), que aunque siendo de pago, cuenta con una versión de prueba de 30 días y es muy útil para la edición de XML.

Para esta unidad os basta el Notepad++ con el plugin de XML Tools y podéis escoger la herramienta con la que os sintáis más cómodos.

2 Verificar si un XML está bien formado

- El plugin XML Tools os ayudará a comprobar la sintaxis, aunque es posible que no detecte todos los errores.
- Existen validadores *online* para XML. Por ejemplo, el de [W3Schools](#): en el apartado de “*Syntax-Check Your XML*” introducís el documento XML y os permite chequear la sintaxis y os indica en qué línea encuentra un error con un mensaje en inglés.

3 Erratas

En el ejercicio propuesto en el apartado **5.2.1 Atributos**, las respuestas que os ofrecen como correctas no son exactas:

Cales son os erros do seguinte documento XML?

```
<?XML version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE biblioteca >
<biblioteca>
  <exemplar tipo_ejem='libro' titulo='XML práctico' editorial='Edicions Eni'>
    <tipo> <libro isbn='978-2-7460-4958-1' edicion= paxinas='347'></libro> </tipo>
    <autor nome='Sebastien Lecomte'></autor>
    <autor nome='Thierry Boulanger'></autor>
    <autor nomee='Angel Belinchon Rúa' funcion='tradutor'></autor>
    <prestado lector='Pepe Pérez'>
      <data_pres dia='13' mes='mar' ano='2009'></data_pres>
      <data_devol/>
    </prestado>
  </exemplar>
</biblioteca>
```

- ☒ a) Utiliza maiúsculas na definición da versión XML.
- ☐ b) Non utiliza o código de caracteres adecuado.
- ☐ c) Os valores dos atributos non están entre comiñas **dobres**.
- ☒ d) Hai algún atributo baleiro.
- ☐ e) A etiqueta <data_devol/> non se pecha.
- ☐ f) Úsanse maiúsculas nos datos do documento.

Figura 1: Respuestas realmente correctas a la pregunta

El World Wide Web Consortium ([W3C](http://www.w3.org/)) es una comunidad internacional que desarrolla estándares que aseguran el crecimiento de la Web a largo plazo. En cuanto al código de caracteres, en el siguiente enlace de [W3C](http://www.w3.org/) se aclara:

Se podrá utilizar cualquier codificación de caracteres para su documento siempre que se haya declarado correctamente y que los caracteres que representa el set conformen un set secundario (un subconjunto) del repertorio Unicode (sería extremadamente inusual encontrar uno que no fuera un set secundario (un subconjunto)).

Para ver las diferencias entre conjunto de caracteres, conjunto de caracteres codificados y codificaciones (o códigos de caracteres), os dejo el siguiente recurso de [W3C](#) (lo podéis traducir al español en Chrome con botón derecho > “Traducir a español”)

UTF-8 es perfectamente válido como codificación.

3.1 Standalone

En el apartado **5.1 O prólogo**, con respecto al atributo *standalone* se debería decir que tendrá el valor “yes” en el caso de que el documento no precise de otro documento para su interpretación (sea independiente) y tendrá el valor “no” en caso de que sí lo necesite (y no sea independiente).

3.2 Elementos

En el apartado **5.2 O exemplar. Os elementos se menciona** que no se permiten comillas dobles, apóstrofes y mayor que “>” en el contenido de los elementos y esto no es cierto. Lo veréis con claridad en la tarea de la unidad.

3.3 Espacios de nombres

En el apartado **7.Utilización de espazos de nomes en XML** el documento con esas declaraciones no está bien formado.

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes"?>
<!DOCTYPE miembros>
<alumnos xmlns:alumnos="http://ASIR/alumnos">
  <profesores xmlns:profes="http://ASIR/profesores">
    <asistentes>
      <alumnos:nombre>pepe</alumnos:nombre>
      <profes:nombre>raquel</profes:nombre>
    </asistentes>
  </profesores>
</alumnos>
```

Para corregirlo hay que añadir la declaración del espacio de nombres al elemento raíz:

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes"?>
<!DOCTYPE miembros>

<asistentes xmlns:alumnos="http://ASIR/alumnos"
  xmlns:profes="http://ASIR/profesores">
  <alumnos:nombre>pepe</alumnos:nombre>
  <profes:nombre>raquel</profes:nombre>
</asistentes>
```

Podéis consultar cómo declarar y cuál es el ámbito de uso de los espacios de nombres [aquí](#)

4 ¿Cuándo usar elementos y cuándo atributos?

Cuando se diseña un documento XML es común cuestionarse si crear un elemento o un atributo. No hay una regla que nos dé una solución indiscutible al respecto, pero os resumo aquí unos consejos de la serie [“Principios de diseño XML”](#):

En él se recomienda usar elementos si:

- consideras que la información en cuestión es parte del material esencial que se expresa o comunica en el XML
- la información se expresa en forma estructurada, especialmente si la estructura puede sufrir modificaciones
- la información está destinada a ser leída y comprendida por una persona
- necesitas que el valor que te genera duda (sobre si usar elemento o atributo) sea modificado por otro atributo, puesto que un atributo no debe modificar a otro atributo

Y en él se recomiendan usar atributos si:

- consideras que la información es periférica a la comunicación principal, o está destinada exclusivamente a ayudar a las aplicaciones a procesar la comunicación principal. Por ejemplo, el uso de identificadores:

```
<mail id="0352">
  <to>You</to>
  <from>Me</from>
  <heading>My message</heading>
  <body>Dear friend, ...</body>
</mail>
```

- la información se expresa como un ente atómico y tiene una estructura fija, por ejemplo, las fechas expresadas en ISO-8601

YYYY-MM-DD	or	YYYYMMDD
YYYY-MM		(but not YYYYMM)
--MM-DD	or	--MMDD ^[1]

Figura 2: [ISO 8601](#) estándar par fechas de calendario

Sin embargo, no sería incorrecto expresarla como un elemento:

```

<mail id="0352">
  <to>You</to>
  <from>Me</from>
  <heading>My message</heading>
  <body>Dear friend, ...</body>
  <date>2020-10-15</date>
</mail>

<mail id="0352" date="2020-10-15">
  <to>You</to>
  <from>Me</from>
  <heading>My message</heading>
  <body>Dear friend, ...</body>
</mail>

```

En este caso mi opción personal sería utilizar la fecha como elemento porque me resulta más legible y forma parte de la información principal.

En este mismo debate, si tomamos el ejemplo de establecer el tamaño de una página, si pensamos en tamaños de página reglados por la norma DIN A0, A2, etc. podríamos usar un atributo *size*:

```
<page size="a4">
```

Size es un atributo que modifica a página, tiene unos valores predefinidos. Pero tampoco sería incorrecto expresarlo como elemento:

```

<page>
  <size>a4</size>
</page>

```

Personalmente, encuentro más intuitivo el uso del atributo, pero insisto que el uso de un elemento no es incorrecto.

Sin embargo, si se usase un tamaño arbitrario en diferentes unidades (mm, cm, etc), habría que descomponer más la información. Por ejemplo:

```

<page>
  <size unit="cm">
    <width>15</width>
    <height>25</height>
  </size>
</page>

```

Con el atributo *unit* indicamos las unidades de las dimensiones anchura y altura. Aporta información sobre el tamaño y no es susceptible de descomponerse en el futuro en otros elementos.

También se podría usar como un elemento más:

```

<page>
  <size>
    <unit>cm</unit>
    <width>15</width>

```

```
        <height>25</height>  
</size>  
</page>
```