XML COMO MEDIO DE NORMALIZACIÓN Y DESARROLLO DOCUMENTAL

Antonio de la Rosa*, José A. Senso**

Resumen: El Web, como entorno de trabajo para los profesionales de la documentación, requiere la utilización de nuevas herramientas que permitan gestionar la información de forma estructurada y organizada. XML y las especificaciones que se derivan de él ofrecen una amplia gama de soluciones a los diversos problemas que atañen a nuestra disciplina, tanto para el desarrollo de software documental como para las tareas cotidianas. En este artículo se presenta brevemente la norma XML y se evalúa su posible impacto en la profesión así como las posibilidades de utilizarlo como vehículo para la creación de sistemas de información.

Palabras clave: SGML (Lenguaje Normalizado y Generalizado de Etiquetado), XML (Lenguaje de Etiquetado Extensible), XLink (Lenguaje de Enlaces XML), HyTime, Puntero X, metadatos, XML/EDI (XML/Intercambio Electrónico de Datos), sistemas de información.

Abstract: The Web, as a working environment for information science professionals, demands the exploitation of new tools. These tools are intended to allow the information management in a structured and organised way. XML and its specifications offer a wide range of solutions for the problems of our domain: either for the development of documentary software or the day-to-day tasks. In this article, the XML standard is briefly presented and its possible impact in the profesion is evaluated as well as the possibilities to use it as vehicle for the creation of information systems.

Keywords: SGML (Standard Generalized Markup Language), XML (eXtensible Markup Language), XLink (XML Linking Language), HyTime, Xpointer, metadata, XML/EDI (XML/Electronic Data Interchange), Information Systems.

1 Introducción

Si algo hemos aprendido después de 10 años utilizando el Web es la importancia de trabajar con información abierta. ¿Sigue teniendo sentido hoy en día la utilización de aplicaciones que confinan nuestros datos, nuestra información, en islas de formatos propietarios? Esta es la pregunta que tratamos de responder en este trabajo.

En la vida de cualquier organización o centro de documentación llega un momento en el que debe plantearse cuál es la mejor forma de gestionar la información que produce y consume. Durante la década de los noventa, para resolver la pregunta «¿qué método utilizar para gestionar nuestro sistema de información?» se acudía a un estu-

* Wisdom. Amsterdam. Holanda. antonio@wisdom.nl

** Facultad de Documentación. Universidad de Granada. Correo-e: jsenso@ugr.es

Recibido: 14-6-99. Segunda versión: 17-11-99.

(c) Consejo Superior de Investigaciones Científicas http://redc.revistas.csic.es
Licencia Creative Commons 3.0 España (by-nc)

dio de las ventajas y un análisis profundo de los inconvenientes que cada método traía consigo. De esta forma, las opciones se englobaban en tres grupos.

El más numeroso es el formado por los denominados «programas propietarios» (de una marca concreta), que ofrecen un conjunto de herramientas integrales (estructuración de la información, almacenamiento, recuperación...) que los convierte en la opción menos arriesgada (por no hablar de la posibilidad de obtener soporte técnico).

Un segundo grupo son los «aventureros» convencidos de las grandes posibilidades que ofrece el SGML (Standard Generalized Markup Language). Obviamente se trata de una norma para edición, pero la idea de utilizar un lenguaje que permita estructurar la información con vistas, por ejemplo, a la recuperación es, sin duda, atractiva. Pero lo cierto es que usar SGML supone un trabajo excesivamente grande para unos resultados dudosos. Por otra parte, la limitación que supone no contar con programas que faciliten la gestión de este tipo de información, unido al nulo compromiso que la mayoría de empresas creadoras de software ha demostrado por SGML, hacen que esta opción tenga pocas salidas.

Por último, encontramos una serie de empresas que han decidido apostar por sistemas basados en la idea de la multiplataforma. La aparición del HTML (Hypertext Markup Language) supuso un importante avance en este campo pero, de nuevo, nos tropezamos con el problema de la estructuración. La idea que Tim Berners Lee tenía de este lenguaje era la de una herramienta capaz de transmitir conocimiento y no datos. HTML no estructura la información de manera estricta, así que no se puede utilizar para recuperar datos de forma ordenada.

Pero tanto SGML como HTML sí representan un cambio, un primer paso, en lo que debería ser el futuro de los sistemas de gestión de información: la información abierta. Los datos deben ser independientes del sistema operativo o del programa con el que fueron creados. Esta opción, que ahora mismo está funcionando con tanto éxito en el Web, debería ser la tendencia lógica de las organizaciones.

La clave del problema no está en si la versión del programa con la que trabajamos es inferior o superior a la que se utilizó para crear el documento, o si este documento es para Macintosh, etc. La clave está en la normalización. La normalización nos permite obtener una representación del documento organizada, estructurada.

2 El punto de vista profesional

El lenguaje HTML supuso la «socialización de la información». Cualquier persona era capaz de poner contenidos en Internet utilizando un lenguaje sencillo que, con el paso del tiempo y las sucesivas versiones, se iba generalizando. A finales de la década de los noventa es difícil encontrar un procesador de textos que no permita guardar e interpretar la información en este formato.

Para un gran número de profesionales de la documentación, el Web ha sido un entorno que ha provocado preocupación y muchas horas de trabajo. Si se quiere ser competitivo es necesario utilizar los nuevos métodos de comunicación para ofrecer los productos y servicios (nuevos o clásicos) a un público con un nivel de exigencia en aumento. Al mismo tiempo, es necesario ampliar y adaptar la oferta a un terreno difícil y en constante desarrollo como es el Web.

Las posibilidades de desarrollo son prácticamente ilimitadas (diseño de sistemas de

(c) Consejo Superior de Investigaciones Científicas http://redc.revistas.csic.es
Licencia Creative Commons 3.0 España (by-nc)

información, integración de sistemas documentales, intercambio electrónico de datos, gestión de información multimedia, automatización de aplicaciones, conectividad, interactividad...), lo que no significa que todo esté solucionado. Este trabajo pretende plantear algunas de las cuestiones relativas a la documentación y su entorno de trabajo más prometedor: Internet.

La mayoría de programas propietarios que se pueden encontrar dentro de la línea del software documental no ofrece una base común (salvo, claro está, los datos), no hay líneas compatibles de desarrollo, ni existe una tendencia decidida para aprovechar todas las posibilidades que ofrece el Web. En opinión de los autores, el entorno XML (eXtensible Markup Language) puede servir de sustrato para desarrollar sistemas de información que ofrezcan una mejor respuesta a los problemas de la documentación en el umbral de una nueva era tecnológica.

Es muy probable que el principal (y casi único) inconveniente de este lenguaje sea la novedad. Tanto los desarrolladores (ya sean casas comerciales o investigadores) como los propios profesionales de la documentación dudan de la futura implantación de las especificaciones.

3 ¿Qué sucede con HTML y SGML?

El HTML ha sido, sin lugar a dudas, uno de los mayores «culpables» del éxito de Internet. Se ha demostrado sobradamente su utilidad y validez para la transmisión de cierto tipo de información. Pero no nos engañemos: en cuanto se intenta trabajar de manera profesional con este lenguaje, sus limitaciones condicionan en exceso el producto final.

Enlaces perdidos, automatización reducida, carencia de sintaxis, nulas posibilidades de estructurar la información o escalabilidad limitada, son algunos de los problemas que el HTML no ha sabido solucionar hasta la fecha. Sin embargo XML (ya sea por sí solo o por medio de sus especificaciones) garantiza la superación de estas trabas.

En definitiva, la diferencia entre HTML y XML estriba en que el primero simplifica al máximo SGML (principalmente al limitarlo a un lenguaje de etiquetas que tienen como objetivo dar formato). Por el contrario, XML simplifica SGML, eliminando parte de sus operaciones sintácticas, pero dotando a la información de una estructura.

Al mismo tiempo, y al contrario de lo que sucede con HTML, XML permite la creación de etiquetas propias dentro de cada documento, lo que añade un nivel de precisión y detalle al que no se podría llegar nunca con HTML. Esto, unido a otras características que se irán desglosando a lo largo del artículo, hace que no sólo los datos XML sean «inteligentes»; también lo son los documentos, ya que permiten, entre otras opciones, presentar el texto dependiendo del contenido que pueda transmitir y poder contextualizar siempre la información presentada.

Como ya se ha comentado, XML es una versión reducida de SGML (orientada hacia el procesamiento de datos) que fue desarrollada para dotar al Web de las posibilidades del SGML.

En realidad, de lo que se trata es de hacer un SGML más comercial, que pueda difundirse más rápidamente. Por esta misma razón, muchas de las posibilidades de SGML se omiten en la especificación XML.

Además de lo que se expone en la tabla I, las diferencias entre ambos lenguajes son:

(c) Consejo Superior de Investigaciones Científicas http://redc.revistas.csic.es
Licencia Creative Commons 3.0 España (by-nc)

- XML (en principio) no necesita DTDs (Document Type Definition). Una DTD está formada por una serie de definiciones para tipos de elementos, atributos, entidades y notaciones. Determina qué etiquetas son «legales» dentro del documento y en qué posiciones pueden aparecer. Para que una aplicación SGML pueda procesar datos, SGML necesita los datos y su DTD. En lugar de DTD, XML mantiene punteros a la estructura de datos.
- XML requiere documentos «bien formados». A diferencia de lo que ocurre con otros lenguajes, en un documento XML siempre se necesitará una etiqueta de principio y otra de final para cada elemento.
- Entidades internas SDATA. Si un documento SGML cuenta con información especial (símbolos matemáticos, químicos, etc.) es posible definirlos con las entidades internas SDATA. XML no ofrece esta posibilidad (a cambio han surgido especificaciones como MathML o CML).
- Excepciones. Una determinada etiqueta siempre realizará la misma función en XML.
- No hay soporte para modelos de contenido AND. Con XML no se pueden enumerar los componentes de un elemento en cualquier orden, existe un orden preestablecido.

XML se ha creado para que sea posible intercambiar documentos (referenciales o de texto completo) muy estructurados a través del Web y porque las únicas alternativas viables dentro de los lenguajes de marcado no son prácticas en este sentido: HTML no es capaz de estructurar un documento y SGML puede estructurar la información, pero es demasiado complicado para implementarlo.

Posiblemente Charles Goldfarb tenga razón cuando afirma que «en el caso de producir documentos de la magnitud de Airbus o Boeing, SGML es la solución; para los demás supuestos está XML».

Tabla I

Problemas SGML y soluciones del entorno XML

Problema SGML	*Solución XML (o especificaciones)*
Ningún navegador de los más difundidos lo soporta Ni Iexplorer ni Netscape ofrecen ningún soporte para SGML. Precisamente lo que hace tan valioso a SGML es la causa de su mínima difusión: SGML ofrece tantas posibilidades que el software para editarlo o leerlo es sumamente sofisticado. Incluso las principales compañías fabricantes de herramientas para SGML (por ejemplo, Arbortext) no diseñan aplicaciones capaces de soportar el 100% de la norma. Algunos navegadores para SGML son Panorama, DINA, etc. Mientras sólo exista soporte para HTML, las grandes compañías que necesitan gestionar sus datos en SGML aplicarán sencillamente programas de conversión SGML-HTML para hacerse visibles en el WWW (aunque con la conversión su información pierda prácticamente toda la estructura y por lo tanto gran parte de su valor).	**IExplorer 5.0** IExplorer 4.0 ya podía gestionar código XML pero el 5.0, en teoría, soporta el 100% de la especificación. Aunque esto no fuera así, sólo hay que nombrar las principales firmas que están desarrollando XML (Microsoft, Netscape, Sun Microsystems, Adobe, IBM, Corel, Hewlett-Pakkard) para reforzar la idea de que las implementaciones de esta norma están aseguradas. Tomemos como ejemplo a Microsoft; además de su navegador tiene otros productos muy ligados a XML: MSXSL, parser (analizador sintáctico) para hojas de estilo, COM (Component Object Model) para orientación a objetos o CDF (Channel Definition Format) para información «push».

(c) Consejo Superior de Investigaciones Científicas http://redc.revistas.csic.es
Licencia Creative Commons 3.0 España (by-nc)

4 El «invento»

XML no ha nacido de la nada. La generación de sistemas informáticos capaces de automatizar procesos —tanto para la creación como para la edición— propios de un documento digital es uno de los campos clásicos de estudio dentro de la informática.

Esta rama en concreto, la del tratamiento de textos, abarca un amplio espectro de programas de todo tipo (procesadores de textos sencillos, sistemas de ayuda basados en hipertexto, generación automática de bases de datos, edición, diseño...) con vistas a ser utilizados por usuarios con diferentes necesidades y formación (principiantes, profesionales, programadores...).

La evolución del procesamiento de textos se puede concretar, a grandes rasgos, en los siguiente pasos:

- Durante la década de los 60 el sistema utilizado era simple: tras generar el documento se aplica el formato deseado. Por lo general, la salida de este texto era impresa. Estos textos tenían asociada, junto al dato propiamente dicho, la descripción deseada. La descripción —llamada reproducción— era convertida por el programa en una presentación. Algunas de las notaciones utilizadas en esta época para la reproducción siguen estando hoy en día en vigencia (con modificaciones, claro está): por ejemplo en RTF (Rich Text Format) o Troff.
- Poco tiempo después de que este sistema empezara a tener éxito, apareció el «marcado»: se procedía a delimitar el texto con una serie de etiquetas o códigos.
- La proliferación de diversos formatos de archivos generó un problema: la información tenía diferentes maneras de representarse. Dentro de este contexto apareció, a finales de los 60, SGML.
- Con el surgimiento de sistemas WYSIWYG (what you see is what you get) se produjo una separación. Por un lado se desarrollaron lenguajes de etiquetado más complejos; por otro, se unió la potencia de los sistemas de reproducción a la capacidad de las nuevas interfaces. El producto final de esta opción son los actuales procesadores de texto (AmiPro, Pagemaker, Word, WordPerfect...).

Este ligero repaso deja constancia de una realidad: en un momento determinado, se produjo una separación entre presentación y estructuración.

Para gestionar información de forma eficaz es necesario contar con un sistema que permita estructurarla lógicamente, ya sea de manera tabular, arbórea... Esta estructura facilitará su recuperación, intercambio e integración.

Dentro de un modelo abierto, como es el caso del Web, es un desperdicio mantener una estructura de datos rígida.

Los sistemas de etiquetado también tienen mecanismos para facilitar la presentación: las hojas de estilo. El trabajo con ellas es muy sencillo, ya que, en lugar de aplicar directamente las órdenes de formato, se indica cómo generar reproducciones formateadas a partir de la abstracción del lenguaje de etiquetado escogido.

5 eXtensible Markup Language

XML es válido para la representación digital de documentos de cualquier tipo y con cualquier extensión. De hecho, un documento XML puede estar formado por texto (re-

(c) Consejo Superior de Investigaciones Científicas http://redc.revistas.csic.es
Licencia Creative Commons 3.0 España (by-nc)

ferencias o documento íntegro), fotografías, gráficos (ya sean vectoriales o mapas de bits).

Para generar un documento XML es necesario estructurar la información. Éste es un proceso muy sencillo dentro del campo de la documentación, ya que la mayoría de «materiales» con los que se trabaja son susceptibles de ser divididos en componentes. Así, un libro tiene título, capítulos, índices...; o un artículo resumen, apartados, subapartados, notas al pie; etc. Al mismo tiempo, cada una de estas partes tiene a su vez párrafos, frases, imágenes... XML denomina a todos estos componentes «elementos».

Como se puede observar en el ejemplo de la tabla II, un documento XML comienza siempre con una instrucción de proceso <?XML...?>. Su presencia indica el tipo de documento con el que se está trabajando (XML) y la versión en la que se ha realizado.

Tabla II

Documento XML de ejemplo

```
<?XML version=»1.0»?>


<DOCUMENTO>
    <libro fechalibro=»1975»>
            <!— La primera edición en 1970 —>
            <titulo>Organización de las bases de datos</titulo>
            <autor><apellido>Martin</apellido></autor>
            <autor><nombre>James</apellido></autor>
            <traductor><apellido>Marco</apellido>
            <traductor><nombre>Alfredo di</nombre></traductor>
  </libro>


</DOCUMENTO>
```

Ya se ha comentado previamente que XML no requiere DTD, aunque sí la admite. Debemos tener en cuenta que para algunos documentos puede resultar extremadamente útil su utilización.

El siguiente ejemplo (tabla III) es un registro bibliográfico formateado en XML. Con la finalidad de mostrar de forma clara el tipo de estructura que XML aporta a los datos —sin la necesidad de añadir sistema alguno de metadatos—, se han diseñado dos gráficos sobre este mismo documento. El primero de ellos representa las etiquetas (tabla IV), y el segundo (tabla V) el contenido (obviando el de los nodos superiores). HTML sólo permite definir el formato, y se muestra indiferente en cuanto al contenido (tabla VI). La única estructura que se atisba es la delimitada por las etiquetas <HEAD> y <BODY>, pero carece de unidad lógica con la que trabajar. Algo similar ocurriría con un registro típico de una base de datos.

Estos mismos datos se podrían haber utilizado para elaborar un registro MARC. Este sistema de intercambio de datos se encuentra tan próximo (en cuanto a concepción) a XML que podríamos afirmar que es el lenguaje de etiquetado por excelencia dentro de nuestra profesión. Pero MARC es más limitado y menos flexible que XML, por no hablar de su complejidad relativa, nula orientación a objetos y escasas posibilidades de trabajo dentro del Web.

(c) Consejo Superior de Investigaciones Científicas http://redc.revistas.csic.es
Licencia Creative Commons 3.0 España (by-nc)

Tabla III

Registro bibliográfico formateado en XML

```
<?XML version=»1.0»?>
<DOCUMENTO>
  <libro fechalibro=»1975»>
     <descripcion>
        <!— La primera edición en 1970 —>
        <titulo>Organización de las bases de datos</titulo>
        <autor>
  <apellido>Martin</apellido>
                  <nombre>James</apellido>
  </autor>
            <traductor>
  <apellido>Marco</apellido>
                  <nombre>Alfredo di</nombre>
  </traductor>
            <editorial>
            <lugar>Londres</lugar>
            <nombre>Prentice Hall PTR</nombre>
            </editorial>
            <copyright>1975</copyright>
            <deplegal>M-5487-1975</deplegal>
            <isbn>0-25-25987-4</isbn>
        <notas>
            <páginas>528</páginas>
            <ilustraciones>56 ilustraciones en blanco y negro</ilustraciones>
            <indices>Índices de contenido y acrónimos</índices>
        </notas>
     </descripción>
     <contenido>
        <resumen>
              <p>Esta obra pretende sistematizar, de alguna manera, el trabajo de diseño y gestión de
              bases de datos. Para ello se analizan los principales modelos, describiendo con especial
              énfasis el modelo entidad-relación</p>
        </resumen>
        <descriptores>
            <ul>
  <li>Bases de datos, diseño de.</li>
  <li>Bases de datos, gestión de.</li>
  <li>Bases de datos, modelos de. </li>
  <li>Modelo entidad-relación.</li>
            </ul>
            </descriptores>

     </contenido>
  </libro>
  <libro fechalibro=»1975»>
  </libro>
  <libro fechalibro=»1975»>
</DOCUMENTO>
```

Al igual que sucede con HTML (lenguaje con el que estamos más familiarizados), los documentos XML se componen de un contenido delimitado por una serie de etiquetas. Los diferentes tipos de etiquetas que se pueden encontrar en un documento XML son: elementos, atributos, referencias de entidad, comentarios, instrucciones de proceso, secciones CDATA y DTD.

(c) Consejo Superior de Investigaciones Científicas http://redc.revistas.csic.es
Licencia Creative Commons 3.0 España (by-nc)

Tabla IV
Árbol de etiquetas

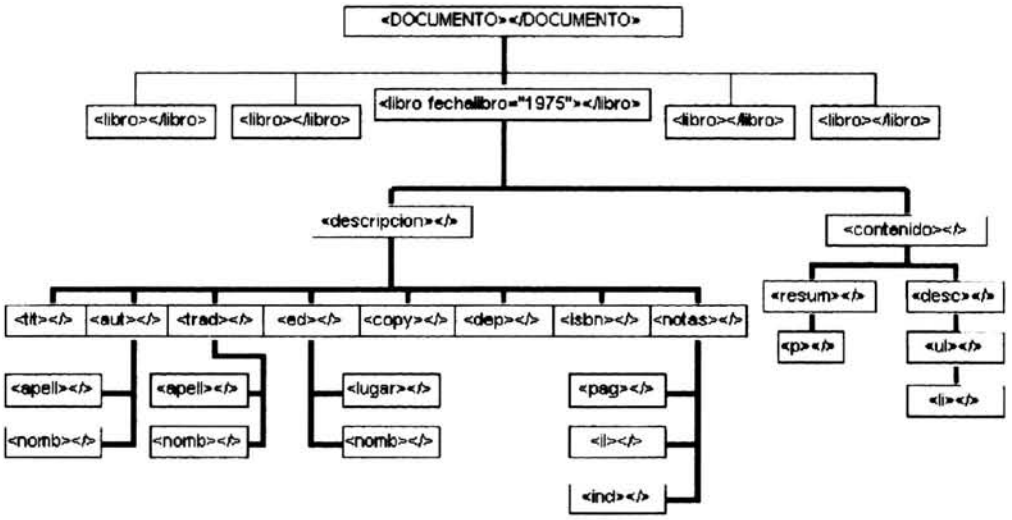
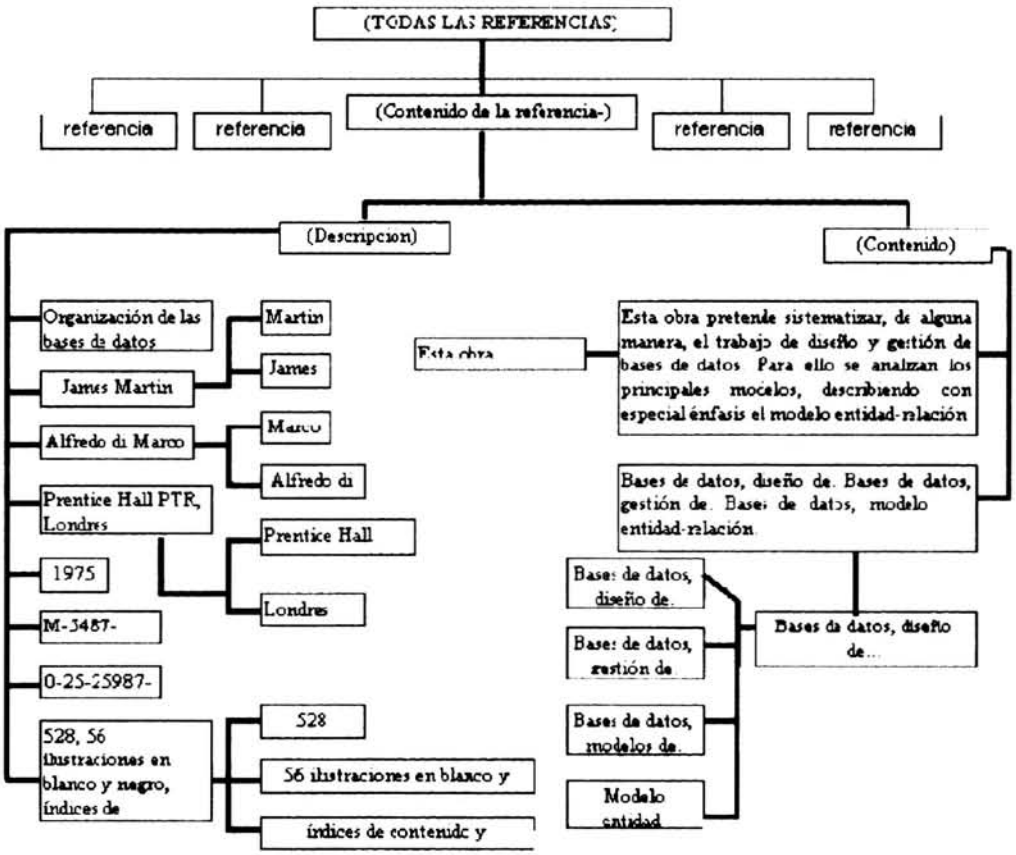


Tabla V
Árbol de contenidos



(c) Consejo Superior de Investigaciones Científicas http://redc.revistas.csic.es
Licencia Creative Commons 3.0 España (by-nc)

Tabla VI

Registro bibliográfico formateado en HTML

```
<html>
<head>
<title>Registro nº 1</title>
</head>
<body>
<p>Título: Organización de las bases de datos</p>
<p>Autor: Martin, James</p>
<p>Traductor: Marco, Alfredo di</p>
<p>Editorial: Londres: Prentice Hall PTR, 1975c</p>
<p>Depósito legal: M-5487-1975</p>
<p>ISBN: 0-25-25987-4</p>
<p>Notas: 528 páginas, 56 il. en blanco y negro, índices de contenido y acrónimos</p>
<p>Resumen: Esta obra pretende sistematizar, de alguna manera, el trabajo de diseño y gestión de bases de
datos. Para ello se analizan los principales modelos, describiendo con especial énfasis el modelo entidad-rela-
ción</p>
</body>
</html>
```

En el primer ejemplo de utilización de XML hemos comprobado la facilidad con la que se puede usar este lenguaje para hacer una descripción formal de cualquier documento. La flexibilidad de XML permite además, y simplemente añadiendo un nuevo conjunto de etiquetas, representar el contenido completo del documento (tabla VII).

Si nos centramos con detenimiento en este último ejemplo, podremos observar cómo cada elemento (artículo) genera una serie de subelementos (título, resumen, palabras clave, apartados...) que, a su vez, es posible que también hagan lo propio. De esta forma podemos ver que el documento es representado mediante una estructura arbórea, con dos elementos de documento o raíz (libro y artículo), una serie de ramas con mayor o menor alcance (título, secciones, palabras clave, apartados...) y varias hojas o elementos finales (párrafos y notas).

XML tiene en cuenta la posibilidad de que ciertos elementos necesiten incluir algún tipo de información adicional. Por ese motivo es posible utilizar atributos con el fin de describir sus propiedades.

El campo de la documentación tiene, potencialmente, más ventajas que ningún otro para implementar documentos XML. La posibilidad de organizar la información en estructuras arbóreas no es tan clara en otras disciplinas como en documentación, ya que es posible que éstas utilicen otro tipo de información (enlaces principalmente) que requiera una representación más abstracta.

En toda estructura jerárquica se distinguen dos tipos de entidades: nodos y relaciones entre ellos. Por una parte, los nodos-elemento son, o pueden ser, tratados como objetos, con todo lo que eso implica a nivel de programación. Por ejemplo, la etiqueta: <PERSONA>...</PERSONA> correspondería a un objeto del tipo PERSONA {...}. Los elementos XML anidados corresponderían a los campos del objeto. Así, los elementos <NOMBRE>, <TELEFONO> y <DIRECCION> se relacionan con los campos nombre, teléfono y dirección del objeto PERSONA.

Al mismo tiempo, y por otra parte, las relaciones entre nodos dan lugar al desarrollo de implementaciones que puedan aplicar conceptos como el de herencia o contex-

(c) Consejo Superior de Investigaciones Científicas http://redc.revistas.csic.es
Licencia Creative Commons 3.0 España (by-nc)

Tabla VII

Representación del contenido completo de un documento

```
<?XML version=»1.0»?>

<LIBRO>
<!—La primera edición en 1970 —>
<titulo>Organización de las bases de datos</titulo>
  <capítulo 1>
        <titulo>Introducción</titulo>
        <párrafo>El desarrollo de las bases de datos será
              sin duda una de las actividades...</párrafo>
        <párrafo>A medida que aumenta...</párrafo>
        <nota>Las estructuras de datos...</nota>
      </capítulo 1>
...
</LIBRO>

<ARTÍCULO>
<!—Rev. Esp. de Doc. Cient., v. 21, n. 4, 1998 —>
<título>Norma z39.50, actitud, posibilidades</titulo>
  <resumen>
        <párrafo>La tecnología de los sistemas...</párrafo>
      </resumen>
      <palabras clave>
        <párrafo>z39.50, SQL, recuperación...</párrafo>
      </palabras clave>
       <apartado 1>
         <título>Introducción: el contexto...</párrafo>
            <párrafo>Pensamos que en...</párrafo>
         </título1>
       </apartado 1>
...
</ARTÍCULO>
```

to. Los sistemas documentales ganan con esto estructura, precisión y todas las ventajas que pueda representar la orientación a objetos para el Web.

Durante el desarrollo de la explicación de las estructuras jerárquicas se ha mencionado un concepto nuevo: entidad.

Una entidad puede referirse a varias cosas: una cadena de caracteres a la que se apunte desde alguna posición en el documento mediante una «referencia de entidad»; cuando el navegador procese esa referencia mostrará la mencionada cadena de caracteres en su lugar. Por ejemplo, podemos llamar a una entidad «nombre del autor» y hacer que su contenido sea un nombre cualquiera. Cada vez que queramos que aparezca este nombre bastará con hacer una referencia a la entidad en lugar de teclearlo de nuevo.

Las entidades también permiten insertar parte de uno o varios documentos (también documentos íntegros) en el documento actual. De esta forma se pueden generar automáticamente nuevos textos aprovechando los que ya están escritos. Debemos tener en cuenta que una de las formas que puede adquirir la entidad es la de entidad externa, que permite describir la estructura física de cualquier documento o de cualquier parte de un documento.

(c) Consejo Superior de Investigaciones Científicas http://redc.revistas.csic.es
Licencia Creative Commons 3.0 España (by-nc)

6 Una DTD para XML

Una de las grandes ventajas de XML es que permite la generación de nuevas etiquetas conforme crecen las necesidades de incorporar información diferente. Sin embargo, un documento XML en el que estas etiquetas se representen de forma aleatoria carece de sentido (desde el punto de vista de la aplicación). El último de los ejemplos utilizados podría representarse así:

<!—La primera edición en 1970 —><título>Organización de las bases de datos </título><capítulo 1>título>Introducción</título><párrafo>El desarrollo de las bases de datos será sin duda una de las actividades...</párrafo><párrafo>A medida que aumenta...</párrafo><nota>Las estructuras de datos...</nota></capítulo 1>...<!—Rev. Esp. de Doc. Cient., v. 21, n. 4, 1998 —><título>Norma z39.50, actitud, posibilidades</título><resumen><párrafo>La tecnología de los sistemas... </párrafo></resumen><palabras clave><párrafo>z39.50, SQL, recuperación... </párrafo></palabras clave><apartado 1><título>Introducción: el contexto... </párrafo><párrafo>Pensamos que en...</párrafo></título1></apartado 1>...

Desde el punto de vista estrictamente sintáctico (es decir, analizando el texto como si se tratase de una expresión Extended Backus Naur Form, sintaxis en la que se basa XML) se trata de un documento correcto. Sin embargo, al no existir restricciones ni en la secuencia ni en la forma de anidar las etiquetas, el código fuente se convierte en un caos. Para introducir estas restricciones es necesario acudir a una «declaración». «Declaración» es el concepto; las DTDs son el medio de expresarlo.

En XML existen cuatro tipos de declaraciones que permiten que un documento comunique a la aplicación que lo analiza meta-información sobre la estructura de su contenido. Esta meta-información incluye la secuencia de etiquetas en el documento y la forma de anidarlas, los tipos de atributos, sus valores (y los que adquieran por defecto), los nombres de los ficheros externos referenciados, información sobre todo tipo de datos no XML y las entidades presentes.

El documento XML debe ser analizado por un parser (analizador sintáctico), que determinará si éste está bien formado y/o es válido. Se dice que un documento XML está bien formado cuando cada elemento que lo forma está perfectamente delimitado y especificado por sus etiquetas (de principio y de final). Un documento bien formado es solamente válido cuando atiende a las restricciones de su DTD. Así, un documento sin DTD no se considerará no válido –al no tener declaración que aplicar- ni válido.

Tanto la calificación de «bien formado» como la de «válido» son muy útiles para evitar algo que sucede con demasiada frecuencia en HTML: la mayoría de estas páginas no se ajustan a las reglas básicas de su DTD y, a pesar de eso, son procesadas por los navegadores.

Resulta obvio pensar que el parser debe haber sido diseñado consecuentemente con la DTD con la que se han realizado los documentos que se van a analizar. De esta forma se logra una gran homogeneidad de los datos, lo que facilita sobremanera el desarrollo de todo tipo de aplicaciones.

(c) Consejo Superior de Investigaciones Científicas http://redc.revistas.csic.es
Licencia Creative Commons 3.0 España (by-nc)

7 Enlaces y direccionamiento

La importancia de un mecanismo de enlaces seguro y fiable para un sistema documental es obvia. XLink (XML Linking Language), especificación derivada de XML, presenta dos grandes ventajas en este sentido:

- Es mucho más resistente a los fallos que los enlaces HTML lo cual es básico para sistemas que tienen que gestionar grandes cantidades de información interrelacionada.
- La estructura informativa sobre la que se implementa permite que los enlaces sean mucho más precisos, ya que el recurso siempre está localizado en el árbol del documento, lo que aporta información valiosa sobre su contexto. En grandes sistemas documentales este factor es de suma importancia.

El Web es un inmenso sistema hipermedia y, sin embargo, la única forma en que es posible realizar un enlace es mediante el elemento HTML <A>. Se trata de un enlace unidireccional que únicamente especifica la localización donde supuestamente se encuentra la página a la que se propone acceder. Estos enlaces no implican una representación de la información más allá del texto que se puede encontrar dentro de esta etiqueta.

Además de HTML, existen otros lenguajes y normas que utilizan los enlaces. Destacan especialmente:

- HyTime (Hypermedia Tyme based Structuring Language): es un estándar general –ISO 10744- derivado de SGML que describe, entre otras cosas, los hiperenlaces, la estructuración de medios o los soportes arquitectónicos (técnica para describir las semánticas comunes entre diversas DTDs y así poder generar documentos «híbridos»).
- TEI (Text Enconding Initiative), que utiliza un sistema de enlaces basado en la misma filosofía de trabajo que HyTime.
- Uniform Resource Names, proyecto de la Internet Engineering Task Force para la identificación de recursos.

Algunas de estas normas soportan gran parte de las características que se le presuponen a un sistema hipertextual:

- Contemplar la opción de incluir atributos en los enlaces. Esto deja abierta la posibilidad de que el lenguaje cuente con más de un tipo de enlace.
- Transclusión. Es decir, que el documento destino pueda aparecer como parte integrante del documento origen del enlace.
- Denominación independiente de la ubicación.
- Que los enlaces puedan especificarse y gestionarse desde fuera de los documentos a los que se apliquen.
- Enlaces bidireccionales.
- Enlaces agrupados, con múltiples orígenes.
- Hiperenlaces múltiples, tanto en origen como en destino.

(c) Consejo Superior de Investigaciones Científicas http://redc.revistas.csic.es
Licencia Creative Commons 3.0 España (by-nc)

Además de estas características, (que claramente no implementa HTML), se ha intentado que la especificación XLink tenga en cuenta otras circunstancias como la relación entre los enlaces (para que un enlace pueda expresar diversos tipos de relaciones), su topología, su formato (ligado claramente con su forma de actuar), su comportamiento (cómo se manifiesta un enlace en situaciones diferentes) o la sintaxis más apropiada dependiendo del tipo de localizador. Recordemos que XLink puede usar diferentes tipos de localizadores (URL, URN, URI...).

Todas estas características confieren a XLink una estructura compacta y eficaz para representar enlaces (ya estén dentro o fuera de los documentos) en parte como herencia de los sistemas en los que se basa: HyTime y TEI, y en parte por ser el resultado del análisis de los errores de otros mecanismos.

Teniendo en cuenta estos ejes de actuación, se han redefinido varios conceptos (tabla VIII).

La existencia de un enlace viene dada por un elemento que debería ser reconocido con fiabilidad por el software, con el fin de que éste se comporte de la forma predefinida. Un enlace simple podría ser del tipo:

<A XML-LINK= «SIMPLE» HREF=»http://www.cindoc.csic.es»>Cindoc</A>

En este caso el elemento de enlace es el clásico anchor <A></A>, que no resultará extraño a los que estén familiarizados con el HTML. La diferencia entre ambos sistemas radica en que los elementos del enlace XML se reconocen por el uso del atributo «XML-LINK». Los valores posibles para este atributo son: SIMPLE, EXTENDED, LOCATOR, GROUP y DOCUMENT. Cada uno de ellos caracteriza a un tipo concreto de enlace.

Pero además de XML-LINK, existen otros atributos que pueden adosarse al ele-

Tabla VIII
Términos relacionados con XLink

Término	*Definición*
Recurso	Unidad de información o servicio que participa en un enlace: archivos, imágenes, documentos, programas, resultados de una búsqueda, etc.
Elemento de enlace	El enlace físico, lo que define a un enlace y describe sus características. Por ejemplo, en HTML todo lo que hay entre las etiquetas <A HREF.....> y </A>.
Localizador	La parte del elemento de enlace que referencia el recurso.
Título	Encabezamiento asociado con el recurso que lo identifica.
Activar un enlace	La acción de usar un enlace o, lo que es lo mismo, de acceder a un recurso. Un enlace puede ser activado por un usuario o mediante un programa.
Enlace multidireccional	Un enlace que puede ser activado desde cualquiera de los recursos que enlaza.
Enlace in-line	Enlace en el cual el propio contenido del elemento de enlace funciona como recurso del enlace.
Enlace out of line	Enlace en el cual el contenido del elemento de enlace no funciona como recurso. Este tipo de enlaces sólo son útiles si tenemos en cuenta el concepto *Grupos de enlaces*.

(c) Consejo Superior de Investigaciones Científicas http://redc.revistas.csic.es
Licencia Creative Commons 3.0 España (by-nc)

mento de enlace, indicando cada uno un comportamiento diferente: ROLE, HREF, TITLE, SHOW, INLINE, CONTENT-ROLE, ACTUATE, BEHAVIOR y STEPS. Estos atributos se pueden asociar con el elemento de enlace XML de dos formas: declarando explícitamente esta asociación (es decir, escribiendo todo el enlace) o aprovechando la capacidad de XML para declarar valores de los atributos por defecto.

Punteros extendidos. Punteros X

Básicamente, las direcciones XLink están especificadas en forma de URI (Uniform Resource Identifier), «versión» ampliada del famoso URL (Uniform Resource Locator). La ventaja que presenta este sistema de direccionamiento con respecto a otros es que su sintaxis no depende del tipo de recurso al que se envíe. Formalmente, URI especifica una localización (similar a una dirección postal) de manera jerárquica.

Este localizador se divide en cinco partes:

- Protocolo. Define el sistema que utilizará el navegador para llegar al recurso determinado. Entre los protocolos más conocidos se encuentran: gopher, ftp, http...
- Nombre del equipo remoto.
- Ruta de datos.
- Consulta.
- Identificador de fragmento.

XLink enriquece este sistema añadiendo la posibilidad de utilizar punteros (XPointers) o, lo que es lo mismo, una serie de términos de localización que permiten identificar partes diferentes dentro de un mismo documento (por ejemplo, párrafos). Sin la utilización de este mecanismo, XML estaría limitado a enlazar documentos enteros, como ocurre en la actualidad con HTML (y, para ser sinceros, con la mayoría de sistemas propietarios).

Los punteros tienen diversos instrumentos para iniciar el proceso de búsqueda de un párrafo determinado a lo largo del documento XML. Sin duda alguna, el mecanismo que demuestra más «inteligencia» (e innovación) es aquel que lo realiza directamente en el árbol de elementos del documento. Como ya hemos apuntado, XML estructura la información de manera que pueda ser tratada como un árbol con raíz (libro), ramas primarias (capítulos), ramas secundarias (secciones) y hojas (párrafos).

En definitiva, XLink logra aumentar espectacularmente los sistemas de enlace hipertextual, permitiendo construir documentos nuevos y vivos, que citen o referencien la última versión, no sólo de un documento, sino también de un párrafo en concreto.

8 Presente y futuro

Hablar del futuro de una especificación que tiene tantas posibilidades es como intentar poner vallas al océano. Las alternativas de trabajo con XML o con cualquiera de sus «derivados» (SOX, VML, WIDL, XSL...) son ilimitadas.

Los proyectos que se están realizando en la actualidad demuestran que este lenguaje es válido para ser utilizado en multitud de campos. Vamos a centrarnos en aquéllos que atañen directa o indirectamente a la documentación.

(c) Consejo Superior de Investigaciones Científicas http://redc.revistas.csic.es
Licencia Creative Commons 3.0 España (by-nc)

En lo que respecta a cuestiones generales de aplicación, la introducción de XML ha significado un cambio en la manera de gestionar procesos en varios campos:

- XML permite una mayor estructuración de los documentos, lo que repercute en una mayor integración con otros datos a la hora de la edición de bases de datos. Pero esta especificación no es sólo positiva en el aspecto de creación, también lo es para el intercambio.
  Hasta ahora, cuando era necesario compartir información entre diversas bases de datos, se recurría a formatos simples (por lo general separando los campos por algún tipo de indicador, excepto en formatos más «avanzados», como es el caso de MARC, de EDI y de otros «propietarios»). Esto puede ser válido cuando se está trabajado con texto simple, pero resulta un verdadero problema cuando se pretende gestionar información orientada a objetos.
  XML facilita el intercambio de estructuras de datos, por lo que nunca se perderán objetos (ni sus atributos y herencias) durante el proceso de integración de datos nuevos.
- Después de conocer esto, no es de extrañar que empresas como Oracle o Informix hayan formado equipos de trabajo para el estudio de XML dentro de sus departamentos de bases de datos corporativas para Internet.
- La integración de esta especificación dentro de un sistema de comercio electrónico permite la normalización de gran parte de los procesos que tienen lugar en la cadena de comercialización. Al mismo tiempo facilita la salida final hacia Internet permitiendo, además, la utilización de sistemas que verifiquen la integridad (servidores seguros).
  La Unión Europea se ha dado cuenta rápidamente de la facilidad con la que EDI (Electronic Data Interchange) se puede integrar con XML, y está financiando varios proyectos que pretenden realizar una DTD de XML para EDI. Entre ellos destaca el European XML/EDI Pilot Project.
  http://www.cenorm.be/isss/workshop/ec/xmledi/isss-xml.html
- Si utilizamos un sistema de metadatos probado y estable (ya sea RDF –Resource Description Format-, TEI, Dublin Core..., o un mecanismo generado a partir de XML) para describir el contenido de un documento que, a su vez, se ha realizado en XML siguiendo una DTD bien formada, obtendremos un sistema de información robusto, que facilitará tareas clásicas en la gestión de la información: edición y recuperación especialmente. Estamos uniendo la potencia en la recuperación de los metadatos con la versatilidad y flexibilidad en la representación de la información que ofrece XML.
  Al menos así lo han visto en el College of Law de la Universidad de Cincinati, donde han codificado todos sus archivos siguiendo una DTD propia y añadiendo una descripción utilizando TEI.
  http://www.law.uc.edu/CETL/proc.html
  La aportación española a este nuevo método de edición corre a cargo del Grupo Anaya. Dentro del entorno URN han logrado generar documentos XML descritos con RDF (utilizando, además, una API –Application Programming Interface- para introducir esta información) que se puede visualizar desde un navegador creado por ellos que, además, muestra las relaciones por medio de un mapa de conceptos.

(c) Consejo Superior de Investigaciones Científicas http://redc.revistas.csic.es
Licencia Creative Commons 3.0 España (by-nc)

http://www.anaya.es
Existen procesos a la hora de generar gran cantidad de documentos electrónicos que se pueden simplificar al máximo utilizando XML como puente entre pasos. De esta forma, no sólo se ganará en rapidez sino que además se abaratarán los costes de la producción final.
La editorial holandesa Samsom Publishers ha desarrollado un sistema de edición (denominado PRISMA) que logra alcanzar estos objetivos.
http://www.samsom.nl

Está claro que la mayoría de proyectos pasan por el estudio y posterior creación de una DTD que facilite el tratamiento de la información. Entre las últimas DTDs creadas para XML destacan:

- La realizada por la oficina de patentes de los Estados Unidos (U. S. Patent and Trademark Office) para facilitar la publicación de este tipo de datos. Destaca sobremanera el exhaustivo mecanismo de descripción de las patentes.
  http://www.uspto.gov
- EAD/DTD, que es nombre de la DTD desarrollada por la Biblioteca del Congreso con el fin de dar soporte XML al sistema de metadatos EAD (Encoded Archival Description) para la descripción de documentos de archivo.
  http://lcweb.loc.gov/rr/ead/eadhome.html
- DocBook, DTD generada y auspiciada por varios organismos e instituciones (especialmente destaca la etapa con el Davenport Group. En la actualidad su desarrollo corresponde a Oasis) para representar el contenido de documentación técnica (sobre todo informática).
  http://www.oreilly.com/davenport
- La desarrollada por MSR Consortium para facilitar el intercambio de información en ingeniería y telecomunicaciones. Se trata de una DTD muy compleja, ya que no sólo tiene que gestionar texto. Además contempla la posibilidad de incluir fórmulas matemáticas. Para ello se está teniendo en cuenta una especificación de XML: MathML.
  http://www.msr-wg.de

9 Conclusiones

Tal y como se mencionó en la primera parte del trabajo, el principal problema que tendrá que afrontar XML es su novedad. Hay pocas aplicaciones (y las que existen no están demasiado difundidas), y es reducido el número de compañías que han apostado por una migración total a este nuevo sistema.

Por si fuera poco, la aparición de XML ha contribuido a aumentar la gran confusión que existe hoy en día dentro de los modelos de implementación de bases de datos, sistemas integrados y navegadores.

Entre las cosas que se deberían tener ya claras está la idea de que XML (o cualquiera de sus especificaciones) suministra un sistema flexible, potente y con vocación Web; que facilita la implantación de una nueva forma de trabajar que (a medio plazo) ahorra costes, tiempo y facilita la integración con otros sistemas de información; y, por

(c) Consejo Superior de Investigaciones Científicas http://redc.revistas.csic.es
Licencia Creative Commons 3.0 España (by-nc)

último, que permite tratar los datos como objetos, aportando un valor añadido mayor del que se puede lograr con cualquiera de los programas existentes hoy en día en el mercado.

El hecho de que en este trabajo no se mencione con más exhaustividad el resto de especificaciones XML no significa que no tengan una salida documental. Todas ellas constituyen un entorno de desarrollo y trabajo común. Donde no llegue XML es muy posible que sí lo hagan VML, SMIL, MathML...

Si esta profesión ha asumido ya que Internet es un campo más de acción (y no un sitio donde hay que «estar», por si acaso) que permite ampliar las perspectivas de trabajo y de negocio, entonces tiene a su disposición una lista interminable de posibilidades de aplicación de este nuevo entorno. El impacto de XML puede ser decisivo –lo comprobaremos en menos de un año- en todas aquellas disciplinas que gestionen información y/o conocimiento. En opinión de los autores, la Documentación no puede permitirse el lujo de permanecer al margen.

10 Bibliografía

BLAKE, P. Taking HTML to the next level: XML allows you to define your own language. *Digital Publishing Strategies,* 1997, vol. 2, p. 14-15.

BOOCH, G. *Object-Oriented Analysis and Design with Applications.* Redwood City, Californina; Benjamin-Cummings Publishing Co., 1991.

BURNARD, L. SGML on the Web: too little too soon, or too much too late? *Computers and Texts,* 1997, vol. 15, n. 1, p. 12-15.

DOSS, G. M. *CORBA developer's guide with XML.* Plano; Wordware Pub., 1999.

FLYNN, P. *Understanding SGML and XML tools: practical programs for handling structured text.* Boston; Kluwer Academic Publishers, 1998.

GOLDFARB, C. y PRESCOD, P. *XML handbook.* Upper Saddle River; Prentice Hall, 1998.

GOLDFARB, C. *The SGML Handbook: The Annotated Full Text of ISO 8879 - Standard Generalized Markup Language.* Oxford [etc.]; Clarendon Press, 1990.

GRAHAM, I. S.; QUIN, L. *XML specification guide.* New York; Wiley, 1999.

HOLZNER, S. *XML complete.* New York; McGraw Hill, 1998.

JACSÓ, P. RealGood Multimedia on the Web: RealNewtworks' RealSystem G2 adds to its impressive array of tools. *Information Today,* 1998, vol. 15, n. 8, pp. 46-7.

KHARE, R.; RIFKIN, A. The origin of (document) species. *Computer Networks and ISDN Systems,* 1998, vol 30, n. 1, p. 389-97.

KOPPEN, E. y NEUMANN, G. A practical approach towards active hyperlinked documents. *Computer Networks and ISDN Systems,* 1998, vol 30, n. 1, p. 251-8.

KRISTENSEN, A. Template resolution in XML/HTML. *Computer Networks and ISDN Systems,* 1998, vol. 30, n. 1, p. 239-49.

(c) Consejo Superior de Investigaciones Científicas http://redc.revistas.csic.es
Licencia Creative Commons 3.0 España (by-nc)