

UNIVERSIDAD CATÓLICA NTRA. SRA DE LA ASUNCIÓN

TRABAJO PRÁCTICO DE TAI2

XML

(eXtensible Markup Language)

Integrantes:

- Acosta I, José Javier
- Jara L, Juan José



Índice

Introducción	1
Qué es XML?	2
Para qué sirve XML?	2
Por qué debo usar XML en vez de HTML?	3
Estructura del XML	4
DTD	4
Esquemas	4
CSS	4
XSL	4
XLL	5
Vínculos entre Documentos XML(XLink/XPointer)	5
Aplicaciones de XML	6
Desarrollo de aplicaciones con XML	6
Categorías de los documentos XML	7
Fortalezas y Debilidades del XML	8
Que es BizTalk?	8
Productos que utilizan XML	9
Categorías de Productos	9
Middleware	10
Bases de Datos XML-Enabled	11
Bases de Datos de XML Nativo	12
Servidores XML	13
Servidores de Aplicaciones XML	13
Sistemas de Administración de Contenidos	14
Motores de Consulta de XML	14
Conclusión	15
Bibliografía	16

Introducción

La mayoría de nosotros hemos oído hablar de XML (Extensible Markup Language), pero siempre surgen preguntas tales como: ¿es una versión mejorada de HTML?, ¿para que sirve?, ¿dónde se utiliza?, ¿cuáles fueron los motivos de su creación?, ¿posee soporte?, ¿qué es en realidad?

Cuestiones como las anteriores se nos presentan cada vez que conversamos con alguien y se menciona el tema, a medida que nos introduzcamos a lo largo del texto, dichas cuestiones se nos aclararan naturalmente.

El W3C (World Wide Web Consortium), se planteo el desarrollo de un nuevo estándar para subsanar las limitaciones que se presentan con el HTML (Hypertext Markup Language), que muchas empresas han tratado de hacerlo embebiendo scripts, Active X, HTML dinámico, hojas de estilo en cascada (CDD), etc, incluyendo de esta forma otras trabas para crear una arquitectura abierta tipo cliente/servidor.

La especificación XML 1.0 (eXtensible Markup Language), que parte de las amplias especificaciones del SGML (Standard Generalized Markup Language), ha sido ratificada por la W3C el 10 de febrero de 1998, e interpretado como "un sistema para definir, validar y compartir formatos de documentos en la Web". XML fue desarrollándose por el Grupo de Trabajo XML desde 1996 (en estos primeros años llamado SGML Editorial Review Board).

Como citamos anteriormente, hasta antes de la aparición de XML nos encontrábamos con la incapacidad de HTML para la manipulación de datos, de hecho, HTML esta dirigido a la presentación de información, aunque inicialmente la intención de HTML era que las etiquetas fueran capaces de marcar la información de acuerdo con su significado, sin importar cómo se mostraban en la pantalla. Lo importante era el contenido y no la forma, o sea, que era un lenguaje de marcas orientado a describir los contenidos, dejando a cada navegador la tarea de dar el formato del documento.

Hasta hace pocos años, Internet era un simple medio de acceso a texto e imágenes. No había ningún estándar establecido para la búsqueda inteligente, el intercambio de datos, la presentación adaptable ni para la personalización. Pero con la amplia difusión de las redes y el uso de las mismas con propósitos comerciales, corporativos, académicos y otros, surgió la necesidad de buscar nuevas soluciones, que permita cubrir las nuevas necesidades.

Tanto HTML como XML son "derivaciones" del SGML (Standard Generalized Markup Language), que es un estándar internacional de definición de descripciones de estructuras de diferentes tipos de documentos electrónicos.

Qué es XML?

XML fue desarrollado por un grupo de trabajo bajo los auspicios del consorcio World Wide Web (W3C) a partir de 1996. Este fue constituido en 1994 con el objetivo de desarrollar protocolos comunes para la evolución de Internet. El W3C tiene como misión la publicación para uso público de protocolos o estándares globales de uso libre.

XML es un lenguaje de marcas que ofrece un formato para la descripción de datos estructurados, el cual conserva todas las propiedades importantes del SGML, XML es un subconjunto de SGML optimizado para la red. Es decir, XML es un metalenguaje, dado que con él podemos definir nuestro propio lenguaje de presentación y, a diferencia del HTML, que se centra en la representación de la información, XML se centra en la información en sí misma.

XML es un formato basado en texto, específicamente diseñado para almacenar y transmitir datos. Un documento XML se compone de elementos XML, cada uno de los cuales consta de una etiqueta de inicio, de una etiqueta de fin y de los datos comprendidos entre ambas etiquetas. Al igual que los documentos HTML, un documento XML contiene texto anotado por etiquetas. Sin embargo, a diferencia de HTML, XML admite un conjunto ilimitado de etiquetas, no para indicar el aspecto que debe tener algo, sino lo que significa.

En un principio, no rivalizarán HTML y XML, estos se complementarán el uno al otro, anudándose ambas gramáticas. Por ejemplo: un elemento XML puede estar etiquetado como libro, autor o precio. El autor del documento es quien decide qué tipo de datos va a utilizar y qué etiquetas son las más adecuadas.

Para qué sirve XML?

Dentro de XML se pueden definir un conjunto ilimitado de etiquetas. Como sabemos, las etiquetas de HTML podemos utilizarlas para desplegar una palabra en negrita o itálicas, el XML proporciona un marco de trabajo para etiquetado de datos estructurados. Un elemento de XML puede declarar que sus datos asociados sean el precio de venta, el autor de un libro o cualquier otro elemento de datos deseado. Al irse adoptando las etiquetas XML a lo largo de una intranet de alguna organización y a lo ancho de la Internet, habrá una correspondiente habilidad para buscar y manipular datos sin importar las aplicaciones dentro de las cuales se encuentre.

El XML proporciona una representación estructural de los datos que ha probado ser ampliamente implementable y fácil de distribuir; al estar definido por el Consorcio de la World Wide Web, asegura que los datos estructurados serán uniformes e independientes de aplicaciones o compañías. Esta interoperabilidad resultante está dando el impulso de inicio a una nueva generación de aplicaciones de Web para comercio electrónico.

El lenguaje XML proporciona un estándar de datos que puede codificar el contenido, la semántica y el esquema de una amplia variedad de casos que van desde simples a complejos, por ejemplo XML puede ser utilizado para marcar lo siguiente:

- Un documento normal.
- Un registro estructurado, como un registro de citas o un pedido de compra.
- Un objeto con datos y métodos, como el formulario permanente de un objeto Java o de un control ActiveX.
- Un registro de datos, como el conjunto de resultados de una consulta.
- Metacontenido sobre un sitio Web, como el formato de definición de canal (CDF).
- Representaciones gráficas, como la interfaz de usuario de una aplicación.
- Entidades y tipos de esquema estándar.
- Todos los vínculos entre datos y personas que hay en el Web.

Una vez que los datos estén en el escritorio del cliente, pueden ser manipulados, editados, y presentados de una gran variedad de maneras, sin viajes de regreso al servidor. Los servidores se pueden convertir ahora en más escalables, debido a las menores cargas de cálculo y ancho de banda. Además, dado que los datos son intercambiados en el formato XML, pueden ser fácilmente mezclados desde diferentes fuentes.

Lo que hace muy atractivo al XML es que mantiene la separación entre la interfaz de usuario y los datos estructurados. El HTML especifica como visualizar datos en un navegador, en cambio XML define el contenido. XML solo utiliza etiquetas para describir los datos, tales como el nombre de la ciudad, temperatura y presión barométrica. Para presentar los datos en un navegador XML, este utiliza hojas de estilo tales como el Lenguaje de Estilo Extensible (XSL) y las Hojas de Estilo en Cascada (CSS). El XML separa los datos de la presentación y el proceso, permitiendo desplegar y procesar los datos tal como usted desee, al aplicar diferentes hojas de estilo y aplicaciones.

Esta separación de datos de la presentación permite una integración de datos perfecta de fuentes diversas. La información de clientes, órdenes de compra, resultados de investigaciones, pagos de facturas, registros médicos, datos de catálogo y cualquier otra información se puede convertir a XML, permitiendo a los datos ser intercambiados en línea tan fácilmente como las páginas de HTML despliegan datos hoy. Los datos codificados en XML pueden ser transmitidos sobre la Web hasta el escritorio. No es necesario retro-ajustar información en formatos propietarios almacenados en bases de datos o documentos de mainframes y, debido a que se usa el HTTP para transmitir documentos XML sobre la red, no se necesitan cambios para esta función. Los documentos XML son fáciles de crear; si está familiarizado con el HTML, puede aprender rápidamente a crear uno.

Los datos codificados en XML son auto-descriptivos, pues las etiquetas descriptivas están entremezcladas con los datos. El formato abierto y flexible utilizado por XML permite su uso en cualquier lugar donde sea necesario intercambiar y transferir información. Dado que el XML es independiente del HTML, se puede insertar código XML en documentos HTML. El W3C ha definido un formato mediante el cual se pueden encapsular en páginas HTML los datos basados en XML. Al incrustar datos XML en una página HTML, se pueden generar varias vistas a partir de los datos entregados, utilizando los datos semánticos que contiene el XML.

Por qué debo usar XML en vez de HTML?

En el punto anterior se mencionaron muchas razones pero aquí las puntualizaremos:

1. Los autores y proveedores pueden diseñar sus propios tipos de documentos, estos pueden ser hechos a la medida de una audiencia específica, esto es gracias a que los autores y diseñadores son libres de inventar sus propios elementos de marca.
2. XML puede proveer más y mejores facilidades para la presentación y el rendimiento del navegador, usando CSS y XSL.
3. XML deja de lado las complejidades de SGML a favor de un modelo más flexible, esto hace que el escribir programas para XML sea mucho más fácil que hacer lo mismo para SGML.
4. La información será más accesible y reusable, debido a que los elementos flexibles de XML pueden ser usados por cualquier programa XML en vez de estar restringido a un solo Fabricante como ha sido el caso con HTML.
5. Los archivos XML válidos son reconocidos por SGML, por lo que pueden ser usados fuera de la red también, en ambientes SGML.

Estructura del XML

DTD

El DTD (definición del tipo de documento, Document Type Definition) proporciona la gramática para una clase de documentos XML. Esta gramática contiene la definición del conjunto de etiquetas que puede contener esa clase de documentos XML, los nombres que pueden utilizarse en los elementos, dónde pueden aparecer y cómo se interrelacionan entre ellos. Se puede decir que un DTD es una definición exacta de la gramática de un documento.

Los documentos XML enviados con un DTD se reconocen como "XML válido". En este caso, un intérprete de XML podría comparar los datos entrantes con las normas definidas en el DTD para comprobar que los datos se han estructurado correctamente. Los datos enviados sin un DTD se reconocen como "bien formados". En XML no existen DTDs predefinidos, por lo que es labor del diseñador especificar su propia DTD para cada tipo de documento XML. En la especificación de XML se describe la forma de definir DTDs particularizadas para documentos XML, que pueden ser internas (cuando van incluidas junto al código XML) o externas (si se encuentran en un documento propio).

Esquemas

Últimamente se está imponiendo otra forma más eficaz de definir la estructura de un documento XML, conocida como esquemas. Un esquema es una especificación formal de las normas de un documento XML, que indica qué elementos se permiten en un documento y en qué combinaciones están permitidas. Los nuevos lenguajes de esquemas, definidos en las propuestas XML-Data (Datos de XML) y DCD (Descripción del contenido del documento) enviadas por el XML-Data Working Group al W3C, proporcionan las mismas funciones que un documento DTD. No obstante, puesto que estos lenguajes de esquema son extensibles, los desarrolladores pueden ampliarlos con información adicional, como los tipos de datos, la herencia y las normas de presentación. Esto hace que los nuevos lenguajes de esquemas sean mucho más potentes que los DTD.

La expresión de esquemas dentro de XML aumenta la potencia del formato XML, pues permite que el software examine determinados datos para comprender su estructura, sin necesitar ninguna descripción previa incorporada de la estructura de los datos. Con un esquema, un autor puede definir exactamente qué nombres de elementos se permiten en un documento y, dentro de cada elemento, qué subelementos, atributos y relaciones se admiten. El autor puede importar fragmentos de otros esquemas, así como ampliar tipos a través de la herencia. Todo ello permite establecer relaciones complejas entre los elementos sin perder la simplicidad de la estructura de árbol léxico.

CSS

Las especificaciones CSS (hojas de estilo en cascada, Cascading Style Sheets) deben comprenderse como una descripción del formato en el que se desea que aparezcan las entidades definidas en un documento. Utilizar CSS con XML es similar, con la excepción de que las etiquetas son diferentes a las de HTML.

XSL

SGML tiene su propio estándar para la representación de sus documentos, el DSSSL (Document Style Semantics and Specification Language, ISO10179), que en realidad es un lenguaje de programación completo y muy potente. Por tanto, ya que XML es una versión reducida de SGML parecía lógico hacer también una versión reducida del DSSSL, llamada en este caso XSL (lenguaje de hojas de estilo extensible, Extensible Stylesheet Language). Básicamente, XSL es un lenguaje de hojas de estilos diseñado para su utilización en el Web. XSL debe representar de forma independiente a la plataforma utilizada o al medio de representación la información recogida en los documentos XML.

En cuanto a la inclusión de imágenes en las páginas XML, estas son enlaces, que pueden representarse por alguno de los tipos soportado por las especificaciones XLink y XPointer. XSL, además de permitir la descripción de la presentación física, también posibilita la ejecución de bucles, sentencias del tipo IF...THEN, selecciones por comparación, operaciones lógicas, ordenaciones de datos, utilización de plantillas, etc. El CSS muestra todos los datos que recibe mientras que el XSL puede elegir que datos mostrar.

XLL

XLL (eXtensible Linking Language): Define el modo de enlace entre diferentes enlaces. Se considera que es un subconjunto de HyTime (Hipermedia/Timed-based structuring Language o Lenguaje de estructuración Hipermedia/basado en el tiempo, ISO 10744) y sigue algunas especificaciones del TEI (Text Encoding Initiative o Iniciativa de codificación de texto). Desde marzo de 1998 el W3C trabajó en los enlaces y direccionamientos del XML. Provisionalmente se le renombró como Xlink y a partir de junio se le nombra como XLL. Este lenguaje de enlaces extensible tiene dos importantes componentes: Xlink y el Xpointer. Va más allá de los enlaces simples que sólo soporta el HTML.

Vínculos entre Documentos XML(XLink/XPointer)

Los enlaces e hipervínculos son tan importantes para los documentos XML que el W3C ha sacado las especificaciones que las controlan fuera de las descripciones del DTD, creando 2 normas: XLink y XPointer. Estas definen el modo de enlace entre diferentes documentos. Este lenguaje va más allá de los enlaces simples que sólo soporta el HTML. Esta especificación soporta por ejemplo las siguientes características:

- Denominación independiente de la ubicación.
- Enlaces que pueden ser también bidireccionales.
- Enlaces que pueden especificarse y gestionarse desde fuera del documento a los que se apliquen (esto permitirá crear en un entorno intranet/extranet un banco de datos de enlaces en los que se puede gestionar y actualizar automáticamente).
- Hiperenlaces múltiples (anillos, múltiples ventanas, etc).
- Enlaces agrupados (múltiples orígenes).

XLink define la forma en la que los documentos XML deben conectarse entre sí. XPointer describe cómo se puede apuntar a un lugar específico de un determinado documento XML. Resumiendo, XLink determina el documento al que se desea acceder y XPointer marca el lugar exacto en dicho documento. Al contrario de lo que ocurre con HTML, en XML existen dos tipos básicos de hipervínculos: simples y extendidos. Un ejemplo de un hipervínculo simple sería:

```
<AUTOR xlink:href="autores.xml#juan" xlink:show="new">
<NOMBRE>Juan Primero Segundo</NOMBRE>
</AUTOR>
```

Otro ejemplo de un hipervínculo extendido podría ser:

```
<EDITOR_AUTOR xlink:extended>
<xlink:locator href="#ana" id="editor"/>
<xlink:locator href="autores.xml#juan" id="autor"/>
<xlink:arc from="editor" to="autor" show="replace"/>
</EDITOR_AUTOR xlink:extended>
```

En el primero se puede observar la definición de un hipervínculo simple que se abre en una nueva ventana (show="new"), mientras que en el segundo se definen un hipervínculo con tres posibilidades diferentes: a una sección determinada del documento (#ana), o a un determinado lugar de otro documento (autores.xml#juan), o a una zona delimitada por dos marcadores (editor y autor). El funcionamiento de un hipervínculo simple no tiene secretos, pues es similar al que se utiliza en HTML.

Aplicaciones de XML

Aunque no todas las herramientas relacionadas con Internet son capaces de trabajar con XML, ya que es una normativa nueva, con un editor de textos ASCII y un visualizador que incluya un intérprete de XML, ya se pueden crear y visualizar documentos XML.

Desarrollo de aplicaciones con XML

Se pueden establecer cuatro tipos de aplicaciones que impulsarán el desarrollo del XML:

- Aplicaciones que exijan que el cliente Web medie entre dos o más Bases de Datos. Se hará posible la integración de bases de datos distribuidas en los navegadores que admitan XML, pudiéndose modificar el contenido y la estructura de esta.
- Aplicaciones que intentan transferir una parte significativa de la carga del proceso del servidor al cliente Web. Esta carga hará que muchas de las funciones de modificación puedan desarrollarse desde el mismo navegador Web del cliente. El kdo más negativo es que se necesitará mayor ancho de banda y mayor potencia del procesador del equipo para poder soportar esta arquitectura de tres capas.
- Aplicaciones que precisen que el cliente Web presente diferentes versiones de los mismos datos a diferentes usuarios.
- Aplicaciones en las que agentes Web inteligentes intentan adaptar la búsqueda de información a las necesidades de los usuarios individuales. Habrá una interacción entre la información requerida y las preferencias del usuario de la aplicación. Con el XML vendrá una segunda generación de aplicaciones con una mayor precisión de la búsqueda.

Además de esto es importante destacar que XML aporta mucha potencia y flexibilidad a las aplicaciones basadas en la Web, proporcionando numerosas ventajas a los programadores y usuarios, como lo son:

- Búsquedas con más significado: Los datos se pueden etiquetar de forma exclusiva con XML, lo que permite que un cliente especifique libros escritos por Winston Churchill, en lugar de sobre Winston Churchill. Las búsquedas que utilizan los métodos actuales, por el contrario, probablemente mezclarían ambos tipos de libros. Sin XML, es necesario que la aplicación de búsqueda comprenda el esquema de cada base de datos, en el que se describe cómo se ha generado. Esto es prácticamente imposible, pues cada base de datos describe sus datos de una forma distinta. Sin embargo, con XML los libros se pueden clasificar fácilmente en categorías estándar por autor, título, ISBN u otros criterios. De esta forma, los agentes podrían buscar libros sobre Winston Churchill en estos sitios de librerías de una forma coherente.
- Programación de aplicaciones Web flexibles: Una vez encontrados los datos, el XML se puede distribuir a otras aplicaciones, objetos y servidores de nivel medio para continuar su procesamiento, o bien se puede entregar al escritorio para su visualización en un explorador. XML, junto con HTML para la presentación, las secuencias de comandos para la lógica y un modelo de objeto común para la interacción con los datos y la presentación, ofrece todas las tecnologías necesarias para el desarrollo de flexibles aplicaciones Web de tres niveles.
- Integración de datos procedentes de fuentes dispares: La capacidad de buscar en varias bases de datos no compatibles entre sí es, hoy en día, prácticamente imposible. XML permite combinar fácilmente los datos estructurados procedentes de fuentes distintas. Se pueden utilizar agentes de software para integrar los datos en un servidor de nivel medio desde bases de datos de fondo y otras aplicaciones. A continuación, dichos datos se pueden entregar a clientes u otros servidores para su agregación, procesamiento y distribución. La capacidad de ampliación y la flexibilidad de XML le permiten describir los datos contenidos en una gran variedad de aplicaciones muy diversas, desde las recopilaciones descriptivas de páginas Web hasta los registros de datos. Además, dado que los datos basados en XML son autodescriptivos, se pueden intercambiar y procesar sin necesidad de una descripción incorporada de los datos entrantes.
- Computación y manipulación locales: Después de entregarlos al cliente, los datos en formato XML se pueden analizar, editar y manipular de forma local, siendo las aplicaciones clientes quienes realizan los cálculos. Los usuarios pueden manipular los datos de diversas formas, y no limitarse a presentarlos. El Modelo de objeto de documento (DOM) de XML también permite

manipular datos con secuencias de comandos u otros lenguajes de programación. Los cálculos relativos a los datos se pueden realizar sin volver al servidor. La separación entre la interfaz de usuario que ve los datos y los propios datos permite crear, de forma natural, potentes aplicaciones para el Web que antes sólo se encontraban en bases de datos avanzadas, todo con un formato simple, flexible y abierto.

- Varias vistas de los datos: Una vez entregados los datos al escritorio, se pueden ver de varias formas. Al describir los datos estructurados de una forma simple, abierta y extensible, XML sirve de complemento para el HTML, que se utiliza ampliamente para describir las interfaces de usuario. Una vez más, mientras que el lenguaje HTML describe el aspecto de los datos, XML describe los propios datos. Dado que la presentación está separada de los datos, la definición de dichos datos en XML permite especificar varias vistas, lo que significa que los datos se pueden representar de la forma adecuada. Los datos locales se pueden presentar de una forma dinámica determinada por la configuración del cliente, las preferencias del usuario u otros criterios. CSS y XSL proporcionan mecanismos declarativos para describir una vista de los datos en particular.
- Actualizaciones granulares: Los datos se pueden actualizar de forma granular con XML, por lo que no es necesario volver a enviar un conjunto completo de datos estructurados cada vez que cambia parte de dichos datos. Sólo es preciso enviar el elemento modificado del servidor al cliente, y los datos modificados se pueden presentar sin necesidad de actualizar toda la interfaz de usuario. En la actualidad, aunque sólo cambie un elemento de los datos, es preciso volver a construir toda la página, incluso si la vista permanece igual, lo que reduce drásticamente la escalabilidad del servidor.

Categorías de los documentos XML.

Los documentos XML pertenecen a dos categorías: "*basados en datos*" y "*basados en documentos*".

- Los documentos XML "*basados en datos*" son en los que XML es usado como un transporte de datos. Estos son por ejemplo órdenes de compra, registros de pacientes y datos científicos.
- Los documentos XML "*basados en documentos*" son en los que XML es usado para representar documentos, como un manual de usuario, páginas estáticas, folletos de marketing. Este último tipo de documento se caracteriza por su estructura irregular.

Para grabar y recuperar datos en un documento "basados en datos", se necesitará una Base de datos, como puede ser una Base de Datos relacional o una orientada a objetos.

Para grabar y recuperar datos en un documento "basados en documentos", se necesita una Base de Datos de XML o un Sistema de Administración de Contenidos. Ambos están diseñados para almacenar fragmentos del contenido, como procedimientos, capítulos, y glosarios, y pueden incluir metadatos, como nombre del autor, fecha de revisión, etc. Un Sistema de Administración de Contenidos generalmente tiene funcionalidades adicionales, como editores, controladores de versiones, etc [Bourret R.,XML].

Fortalezas y Debilidades del XML

La meta fundamental del XML es hacer la cooperación y la interoperabilidad más fáciles entre módulos que pertenecen a diferentes sistemas, e incluso a diferentes organizaciones. Entonces, ¿por qué el XML se ha convertido en tecnología del alto nivel tan rápidamente? La respuesta es simple: XML es texto simple, pero es también elegante. Un lenguaje de marcas hace fácil identificar (y extraer en un tiempo posterior) segmentos específicos de información con significados especiales. Usar texto simple y mantener el contenido universalmente interpretable son dos condiciones para implementar una tecnología exitosamente entre sistemas heterogéneos. Una cadena del texto se puede transferir y se puede manejar fácilmente en cualquier plataforma destino. Sin embargo, si usted solo planea definir un protocolo basado en texto para hacer que sus módulos se comuniquen, usted no necesita necesariamente XML. El ASCII, de hecho, ha existido durante mucho tiempo. La diferencia recae en las etiquetas XML que se utilizan para delimitar segmentos de información.

La principal fortaleza del XML es también, actualmente, su debilidad principal. El XML es demasiado genérico para ser utilizado sin definir externamente la sintaxis exacta de un documento y cómo puede localizarse y extraerse cada fragmento de datos intercambiados. Tal estandarización es más simple alcanzar dentro de una sola aplicación corporativa, incluso si atraviesa varias plataformas heterogéneas de hardware y de software. Conseguir el mismo resultado en un contexto más amplio requiere un enorme esfuerzo para llegar a algunas definiciones estándares. Este proceso puede tomar años en ser completado [MSDN Latinoamérica].

Desgraciadamente, cuando se trata de integrar sistemas de diferentes organizaciones, especialmente de una manera abierta, el XML muestra algunas limitaciones. El XML trata de la descripción de los datos, pero solamente es útil cuando la gente está de acuerdo en la manera que los datos deben ser descritos. Una factura, por ejemplo, es lógicamente el mismo tipo de documento para cualquier proceso que lo manipule. Pero diferentes procesos pueden convertirla en formatos binarios diferentes. A pesar del formato de almacenamiento físico, el documento necesita ser transmitido y ser recibido en un formato comúnmente reconocido disponible en todas las plataformas. Por supuesto, el XML se puede utilizar como este formato.

Imagine el enorme costo que supondría para los consumidores y las empresas si cada empresa definiera su propia forma de publicar la información. Incluso con la Web, los costos asociados a la configuración y el mantenimiento de un sitio Web superan la capacidad de algunas empresas. Al no haber ningún límite al número de empresas que podrían publicar esta información, la falta de estándares que definan el modo de publicarla de una forma segura y controlada tendría como consecuencia miles y miles de implementaciones, enfoques de exploración de la Web y profundidad del contenido distintos.

Por este motivo, en la actualidad se están definiendo esquemas por grupos sectoriales con similares intereses, de forma que existirán esquemas estándares avalados por asociaciones de empresas y organismos que garanticen que cualquier usuario que las adopte como suyas, trabaje con las mismas etiquetas. Como ejemplos de estos esquemas estándares tenemos: CDF - Channel Definition Format (define canales para enviar información periódica a los clientes), CML - Chemical Markup Language (define información del sector químico), MathML - Mathematical Markup Language (define datos matemáticos), SMIL - Synchronized Multimedia Integration Language (define presentaciones de recursos multimedia).

Que es BizTalk?

BizTalk (<http://www.biztalk.org/>) es una iniciativa mundial dirigida a crear una base de datos de formatos o esquemas de documentos en XML. Importantes compañías se han unido ya a BizTalk, que ofrece un gran número de esquemas de XML a los cuales las compañías interesadas puedan referirse. Un documento de BizTalk es un documento XML que proporciona las etiquetas de un cierto vocabulario y sigue las reglas que la organización ha definido para este tipo de documento. Este sitio crecerá hasta convertirse en un portal para localizar, administrar, publicar y obtener información sobre XML, XSL y los modelos de información utilizados en miles de aplicaciones.

Microsoft® pretende establecer por medio de BizTalk un marco para el comercio electrónico de empresa a consumidor. En declaraciones de Bill Gates, "para conseguir introducir a más de un millón de empresas en el comercio electrónico, necesitamos ayudar a que sus programas de software hablen un lenguaje de negociación de un modo consistente. Hasta ahora ha sido extremadamente difícil para las empresas conducir sus negocios sobre Internet por la ausencia de un vocabulario simple que soporte los procesos de negocio. BizTalk proporciona un marco que acelerará la adopción del comercio electrónico al proveer de un lenguaje que facilite que los programas de software corran en cualquier plataforma o tecnología" [Marketing y Comercio Electrónico; XML].

Productos que utilizan XML

El número de productos para utilizar XML con Bases de Datos está creciendo a una gran velocidad. Nuevos productos entran al mercado de forma constante.

Categorías de Productos

Para realizar esta división de los productos que están vinculados a la tecnología XML se tomó en cuenta cuales son las características de los mismos, las funciones que cumplen y como interactúan con las Bases de Datos. Cabe destacar que a veces las diferencias entre algunas de estas categorías es un poco arbitraria, ya que los productos ofrecen funcionalidades muy variadas que implica que se superpongan entre las categorías. Se puede establecer que siete categorías de productos:

- **Middleware**: Software que es invocado desde nuestras aplicaciones para transferir datos entre documentos XML y Bases de Datos. Orientados a aplicaciones "basados en datos".
- **Bases de Datos XML-Enabled**: Bases de Datos que pueden transferir datos entre documentos XML y ellas mismas. Orientados principalmente para aplicaciones "basados en datos".
- **Bases de Datos de XML Nativo**: Base de Datos que almacenan XML en su forma "nativa", generalmente tanto texto indexado como alguna variante del DOM. Son tanto para aplicaciones "basados en datos" o "basados en documentos".
- **Servidores XML**: Plataforma que brinda servicio de datos (en forma de documentos XML) desde y hacia aplicaciones distribuidas, como e-commerce. Los documentos XML deben ser fáciles de crear y aplicaciones business to business. Orientados principalmente a aplicaciones "basados en datos".
- **Servidores de Aplicaciones XML**: Servidores de aplicaciones Web que brindan servicio de XML a los navegadores. Son tanto para aplicaciones "basados en datos" o "basados en documentos".
- **Sistemas de Administración de Contenidos**: Sistemas para administrar documentos legibles e incluyen soporte para editar, controlar las versiones, crear nuevos documentos a partir de documentos existentes. Principalmente destinados a aplicaciones "basados en documentos".
- **Motores de consulta de XML**: Aplicaciones que permiten realizar consultas sobre documentos XML.

Middleware

Es software que es usado en aplicaciones "basadas en datos" para transferir datos entre documentos XML y Bases de Datos. Están escritos en diferentes lenguajes, pero la mayoría de estos utilizan ODBC, JDBC, o OLEDB. Aunque algunos de estos pueden enviar datos a través de Internet, la mayoría de estos necesitan ser usados con un servidor Web si se necesita acceso remoto a los datos. Dentro de esta categoría de productos podemos encontrar los siguientes productos:

Producto	Desarrollador
ADO®	Microsoft®
Allora®	HiT Software®
ASP2XML®	Stonebroom®
Attunity Connect®	Attunity Ltd.®
DatabaseDom®	IBM®
DataCraft®	IBM®
DB2XML®	Volker Turau®
DB/XML Vision®	DATA Mirror Corp.®
IConnector®	Infoteria®
idx -xmnesia®	IDEALX®
InterAccess®	XML Software Corporation®
ODBC2XML®	Intelligent Systems Researc®
ODBC Socket Server®	Team FXML®
PerXML Smart Transformation System®	PerCurrence®
XML Lightweight Extractor (XLE)®	IBM®
XML SQL Utility for Java and XSQL®	Oracle®

ADO. Desarrollador: Microsoft

ADO provee de un camino para transferir datos entre documentos XML y una Base de Datos, usando Recordsets como objetos intermedios. Si el Recordset tiene pendiente updates, deletes, o inserts, estos son especificados por banderas específicas de ADO.

XML SQL Utility for Java y XSQL Servlet. Desarrollador: Oracle

XML SQL Utility for Java es un conjunto de clases java para transferir datos entre Bases de Datos relacionales y documentos XML. Cuando se transfiere datos desde la Base de Datos a un documento XML, el usuario debe proveer tanto una sentencia SELECT o un conjunto de resultados JDBC. Cuando la transferencia es en el otro sentido, el usuario provee un documento XML o un documento DOM.

Bases de Datos XML-Enabled

Este tipo de Bases de Datos son (usualmente relacionales) que brindan la posibilidad de transferir datos entre documentos XML y ellas mismas. Estas Bases de Datos son generalmente diseñadas para almacenar y recuperar documentos "basados en datos". Esto es porque los datos son transferidos desde y hacia tablas definidas por el usuario, en lugar de tablas especialmente diseñadas para modelar el documento XML. Sin embargo, muchas de estas pueden almacenar documentos "basados en documentos" en una sola columna.

Muchas de las Bases de Datos pueden publicar documentos en el Web, esto implica que la línea que separa entre las Bases de Datos XML-Enabled y los Servidores de XML es confusa, pero cabe destacar que estos últimos están diseñados primariamente para construir aplicaciones basadas en el Web. Con más claridad se puede diferenciar las Bases de Datos XML-Enabled y los Servidores de Aplicaciones XML, los cuales no puede recibir datos como documentos XML. Dentro de este tipo de Bases de Datos, encontramos los productos:

Producto	Desarrollador
DB2 XML Extender and DB2 Text Extender®	IBM®
Informix®	Informix®
Microsoft SQL Server 2000®	Microsoft®
Oracle 8i 9i®	Oracle®

DB2 XML Extender y DB2 Text Extender. Desarrollador: IBM

El DB2 XML permite transferir datos entre documentos XML y DB2. Los DTDs de XML son mapeados a esquemas (y viceversa) con el lenguaje XML-based Data Access Definition (DAD). Una herramienta visual es provista para construir documentos DAD, de forma de mapear elementos y atributos a las tablas y columnas. Las aplicaciones son procedimientos almacenados, que invocan el Extender, quien luego guarda o trae datos basados en el documento DAD. El XML Extender administra documentos DAD y DTDs en sus propias tablas, evitando que las aplicaciones tengan que hacer esto y optimizando el acceso. Además, el DB2 Text Extender contiene una variedad de tecnologías para realizar búsquedas.

Informix. Desarrollador: Informix

Informix soporta XML a través de su traductor "Object Translator" y el producto "Web DataBlade." El Object Translator genera código, incluyendo la capacidad de que los objetos transfieran sus datos desde y hacia la Base de Datos. También soporta funcionalidades como transacciones y tipos de bloqueos pesimistas y optimistas. El soporte de XML está provisto a través de métodos generados que pueden transferir datos entre objetos y documentos XML. Los usuarios deben mapear los documentos XML a un árbol de objetos de datos con una herramienta visual. El Web DataBlade es una aplicación que crea Documentos XML desde templates conteniendo sentencias embebidas de SQL.

Microsoft SQL Server 2000. Desarrollador: Microsoft

Microsoft SQL Server 2000 soporta XML de tres formas distintas: La cláusula FOR XML en sentencias SELECT, consultas XPath que utilizan esquemas XML-Data, y la función OpenXML en procedimientos almacenados. Los esquemas XML-Data, también llamados "esquemas de mapeo", contienen atributos extras que mapean elementos y atributos a tablas y columnas. Esto modela un documento XML como un árbol de objetos de datos y son usados para consultar la Base de Datos usando un subset de XPath. Existe una herramienta gráfica para construir esquemas de mapeo.

La función OpenXML es usada para extraer cualquier parte de un documento XML como una tabla y puede ser usado como una tabla en la mayoría de los lugares donde estas son utilizadas, como por

ejemplo en la cláusula FROM de una sentencia SELECT. Esto se puede utilizar en forma conjunta con una sentencia INSERT para transferir datos de un documento XML a la Base de Datos. Una expresión XPath identifica el elemento o atributo que representa una fila de datos. Inserts, updates, y deletes son realizados a través de documentos XML llamados "update grams". Estos contienen los datos de antes y después (ambos en caso de un update), solamente el después en el caso de un insert, y sólo el antes en caso de un delete.

Oracle 8i, 9i Desarrollador: Oracle

Oracle 8i puede almacenar documentos XML de tres maneras distintas: en el Internet File System (iFS), usando el utilitario para Java XML SQL, y como un objeto binario que puede ser buscado utilizando el Oracle Intermedia XML Search. Oracle 8i también incluye un número de otras herramientas relacionadas con XML, la más interesante de las cuales es el XML Class Generator, quien puede generar clases Java desde un DTD.

Con iFS, una o más definición de tipos de documentos definen como se van a mapear un documento XML como un árbol de objetos de datos específicos. iFS usa ese mapeo de documentos tanto para construir tablas en cuales el documento XML puede ser almacenado, y para transferir datos entre documentos XML y la Base de Datos. Oracle Intermedia XML Search es un utilitario que puede automáticamente indexar y buscar documentos XML y fragmentos de estos, de cualquier tamaño hasta 4 Gigabytes.

Oracle 9i incluye Base de Datos con soporte nativo de XML, e introduce un nuevo tipo de objeto de datos (XMLType).

Bases de Datos de XML Nativo

Dentro de las Bases de Datos de XML Nativo encontramos dos tipos de categorías:

- **Almacenamiento basado en texto** Guarda el documento entero en forma de texto. Una estrategia simple es guardar documento como un BLOB (Binary Large Object) en una Base de Datos relacional o como un documento en un File system que brinde búsquedas sobre documentos.
- **Almacenamiento basado en el Modelo** Grabar un modelo binario del documento en un repositorio de datos existente. Por ejemplo, estos deben mapear el DOM a tablas relacionales como los elementos, atributos o entidades.

Las Bases de Datos de XML Nativo difieren de las Base de Datos XML-Enabled en tres formas principales:

- Las Bases de Datos de XML Nativo pueden preservar la estructura física, como también los comentarios, DTDs, etc. Mientras que las Bases de Datos XML-Enabled pueden hacer esto en teoría, pero no es realidad en la práctica.
- Las Base de Datos de XML Nativo pueden almacenar documentos XML sin conocer su esquema (DTD).
- La única conexión con los datos de una Base de Datos de XML Nativo es XML y las tecnologías relacionadas como XPath, DOM u otras API. En el otro sentido, las Base de Datos XML-enabled, ofrecen acceso directo a los datos como por ejemplo por ODBC.

Dentro de esta categoría podemos encontrar los siguientes productos:

Producto	Desarrollador
4Suite - 4Suite Server®	FourThought®
Birdstep DataBase Engine®	Birdstep®
DbXML®	The dbXML Group®
EXcelon®	eXcelon Corp.®
Lore®	Stanford University®
Virtuoso®	OpenLink Software®

Servidores XML

Los Servidores XML son plataformas para recibir y enviar datos entre aplicaciones distribuidas, como e-commerce y business-to-business. En la gran mayoría de los casos los datos son transferidos en formato de documentos XML del tipo "basados en datos". Los servidores XML generalmente contienen ambientes de desarrollo de aplicaciones y proveen acceso a los datos de distintos tipos.

La línea divisoria entre los Servidores XML, las Bases de Datos XML-Enabled y las Bases de Datos de XML nativo es confusa. Los sistemas que se encuentran en la categoría de Servidores XML son los que se muestran como plataformas para construir aplicaciones de e-commerce y otras aplicaciones distribuidas, en lugar de estrictamente ser una Base de Datos, que obviamente pueden ser usados para desarrollar estas aplicaciones. Dentro de esta categoría podemos encontrar los siguientes productos:

Producto	Desarrollador
Castor®	Exolab.org®
MatriXML®	MojoTechnology®
Rhythmyx Integrator®	Percussion Software®
Total-e-Business®	HP/Bluestone®
Versant enJin®	Versant Corp®
XA-BizDoc Server®	XAware Inc®
XML Portal Server (XPS)®	Sequoia Software Co rp.®

Servidores de Aplicaciones XML

Los Servidores de Aplicaciones XML son servidores de aplicaciones Web que son capaces de dar servicio a cualquier tipo de documento XML. Estos utiliza sentencias SQL embebidas en un lenguaje de scripts para obtener datos y generar dinámicamente documentos XML.

La división entre Servidores XML y Servidores de Aplicaciones XML está poco definida; los sistemas que se pueden catalogar como Servidores de Aplicaciones XML tienden a ser dirigidos por lenguajes de scripting o lenguajes de marcas personalizados. Sin embargo, muchos Sistemas de Administración de Contenidos pueden también publicar documentos XML al Web, estos son fundamentalmente diferentes en que son diseñados para administrar documentos "basados en documentos", mientras que los Servidores de Aplicaciones XML sirven tanto para documentos "basados en datos" (páginas dinámicas) como "basados en documentos" (páginas estáticas). Dentro de esta categoría podemos encontrar los siguientes productos:

Producto	Desarrollador
AxKit®	AxKit.com Ltd.®

Cache®	InterSystems Corp®
Cocoon®	Apache.org®
ColdFusion®	Allaire®
Lasso®	Blue World Communications®
Noetix Web Query®	Aris Software Inc.®
Zope®	Digital Creations®

Sistemas de Administración de Contenidos

Los Sistemas de Administración de Contenidos sirven para almacenar, recuperar e integrar documentos XML. Estos incluyen características como editores, controladores de versiones y control de accesos a usuarios concurrentes. Estas ofrecen funcionalidades en el manejo de documentos, como por ejemplo controlador de versiones. La mayoría de los Sistemas de Administración de Contenidos pueden publicar XML en el Web, pero son primariamente diseñados para administrar documentos "basados en documentos".

Producto	Desarrollador
Astoria®	Chrystal Software®
BladeRunner®	Interleaf®
Documentum®	Documentum®
Dynabase®	eBusiness Technologies®
Epic®	Arbortext®
Frontier®	UserLand Software®
GEM Content Management System®	X.Systems®
GroveMinder®	TechnoTeacher®
Hynet Directive®	Hynet Technologies®
Information Manager®	Interleaf®
POET Content Management Suite®	Sorman®
Prowler®	Infozone®
Target 2000®	Progressive Information Technologies®

Motores de Consulta de XML

Motores de consulta XML son programas que pueden realizar consultas sobre un Documento XML. Actualmente, existen una gran variedad de lenguajes: XQL, XPath, XML-QL, Quilt, XQuery, etc. Se espera que en el futuro el W3C logre estandarizar estos lenguajes en: XPath y XQuery.

Por más información consultar <http://www.w3.org/XML/Query>.



Conclusión

XML abre la posibilidad de migrar las aplicaciones empresariales “tradicionales” a la www, ya que estas implican la manipulación o la transferencia de datos desde y hacia Bases de Datos.

Hemos visto también que la mayor ventaja XML también es su desventaja, ya que podría generarse una explosión de “estándares” de estructuras, y al final resultaría difícil nuevamente alcanzar la meta de manipulación y transferencia de datos.

Aparentemente este estándar, marca una nueva etapa en la www, como lo hizo en su momento HTML, ya que las aplicaciones desarrolladas tenderían a ser “más inteligentes”, interactuando con otras aplicaciones.

Podríamos decir que aquellas empresas que no adopten este tipo de estándares, les resultara cada vez más difícil interactuar con el resto del mundo, ya que es a lo que tiende la www.

Cabe destacar que como es un metalenguaje, es posible su utilización en otras áreas, no limitándose su aplicación solamente a la web.



Bibliografía

<http://www.w3.org/XML/>
<http://www.ucc.ie/xml/>
<http://msdn.microsoft.com/xml/>
<http://www.ibium.com/alf/xml/index.asp>
<http://www.programacion.com/html/xml/>
http://www.csclub.uwaterloo.ca/u/relander/XML/Wave/xml_mw.html
<http://www.xml.com/xml/pub>
<http://www.lavariabile.com/art/xml/axml003/axml003.asp#Intro>
<http://www.monografias.com/ixml>
http://www.ramon.org/xml/articulos/intro_xml-html.htm