

# **XML: SU FUTURO, SU APLICABILIDAD Y SU RELACIÓN CON LAS TECNOLOGÍAS ACTUALES MÁS IMPORTANTES**

**Luis Felipe Wanumen Silva \***

Grupo de Investigación Desarrollo de Herramientas para

La creación y manipulación de contenido XML,

Adscrito al grupo METIS de la Facultad Tecnológica

## **RESUMEN**

En este artículo se pretende mostrar como la tecnología XML es más que un lenguaje de marcado extensible y como con XML se pueden llegar a crear nuevas tecnologías de comunicaciones incluyendo tecnologías para el intercambio de información gráfica, de negocios, de servicios, etc. En este artículo se puede decir que se recopilan las tecnologías más notables y prósperas derivadas de la gran tecnología XML y se analizan sus diversas aplicaciones en muchas tecnologías actuales, al mismo tiempo que se muestra la interacción de XML con los principales motores de bases de datos, con HTML, con los servicios web, y con muchas tecnologías de esta era informática.

---

\* Ingeniero de Sistemas, especialista en Ingeniería de Software de la Universidad Distrital Francisco José de Caldas. Docente de la Universidad Distrital, actualmente docente tiempo completo de la Facultad Tecnológica. Correo electrónico: lwanumen@udistrital.edu.co. Actualmente director del Grupo de Investigación Desarrollo de Herramientas para la creación y manipulación de Contenido XML.

## **PALABRAS CLAVE**

Lenguaje extensible de marcado, XML, SOAP, servicios web, Dtd, Esquema y Relax-NG, XSL-FO, XSLT, Xlink, Xpointer, CSS, P3P, MATHML , XBRL , RSS, SMIL, RDF, SVG, 3DML, WIDL, WML, VML, DSML, SGML

## **ABSTRACT**

In this article it is sought to show as the technology XML it is more than a language of marked expandable and I eat with XML they can be ended up creating new technologies of communications including technologies for the exchange of graphic information, of business, of services, etc. In this article one could say that the derived more remarkable and more prosperous technologies of the great technology XML is gathered and its diverse applications are analyzed in many current technologies, at the same time that the interaction of XML is shown with the main motors of databases, with HTML, with the services web, and with many technologies of this computer era.

## **Key Words**

Expandable language of marked, XML, SOAP, services web, Dtd, Outline and Relax-NG, XSL-FO, XSLT, Xlink, Xpointer, CSS, P3P, MATHML, XBRL, RSS, SMIL, RDF, SVG, 3DML, WIDL, WML, VML, DSML, SGML

## **INTRODUCCIÓN**

Los lenguajes de marcado, en últimas han buscado de una u otra forma definir documentos. La implementación más difundida que ayuda a hacer una primera definición de

documentos, se llama HTML, que permite definir documentos. De hecho cuando se busca en la web palabras, se puede hacer una pequeña búsqueda de las páginas con solo analizar la información contenida en la cabecera de los documentos html, que en su gran mayoría, si se es riguroso, se coloca en la cabecera del documento palabras y frases claves, que permiten de una u otra forma describir en forma muy global y también muy sencilla sobre lo que se trata el documento.

### SGML

Lenguaje, estructurado, generalizado de marcado y es el antecesor de todos los lenguajes de marcado

### HTML

Lenguaje para representar la información. Es un lenguaje de marcado, pero sin la opción de ser extensible, limitándose solamente a las ventajas del hipertexto.

### XML

Xml es el acrónimo del inglés Extensible Markup language (Lenguaje de marcado ampliable o extensible) desarrollado por el World Wide Web Consortium[11].

En la siguiente gráfica se muestra algunas tecnologías que precedieron a XML y como XML ayuda a hacer realidad el sueño de dar estructura a los documentos web, que fue uno de los grandes sueños cuando se propuso la especificación SGML que fue basada en GML:

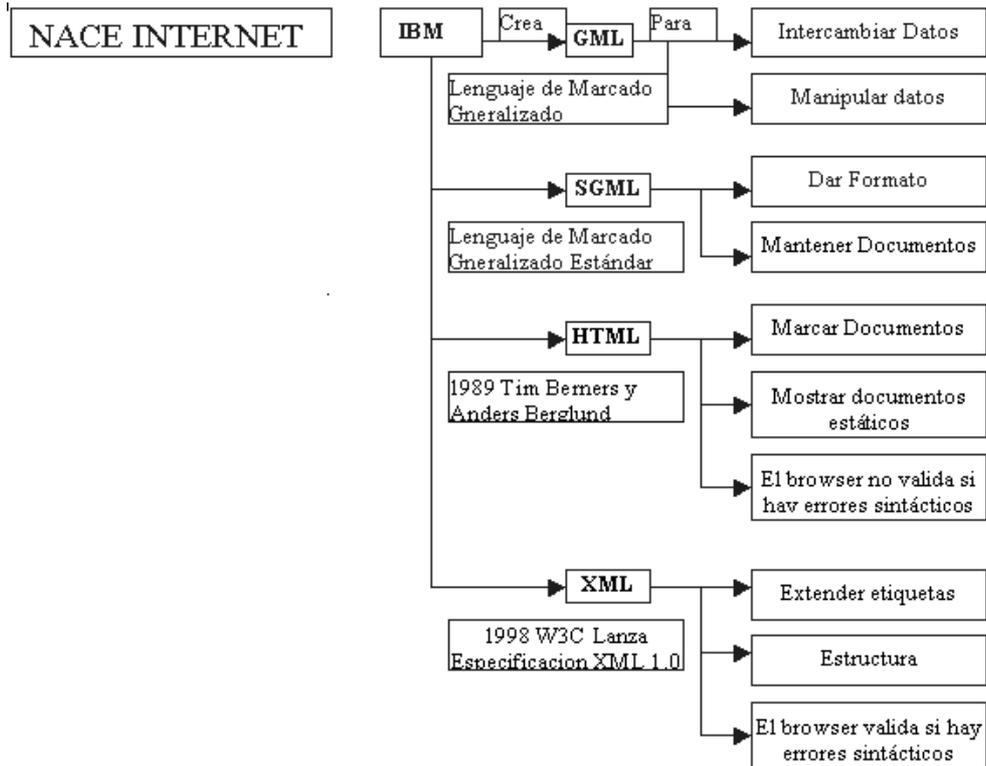


Figura 1. Elaborado por el autor, basándose en los primeros dos capítulos del libro[6]

En el presente artículo se muestran algunas tecnologías derivadas de XML y la relación entre XML y las bases de datos.

### 1. VENTAJAS DE XML

Bueno, la pregunta que se estará haciendo en este momento, es: ¿qué ventajas me trae XML?. La respuesta es simplemente bien sencilla, html ayuda a empezar a definir un documento, pero para continuar con una definición de documento de una forma más concreta, XML, es sin lugar a dudas una herramienta muy poderosa. El cuento que es necesario aclarar, es que XML, en ningún momento es el reemplazo de HTML y por eso no es posible tratarlos a los dos como rivales, sino más bien como complementarios.

Empecemos por decir que XML es un lenguaje que presenta las siguientes ventajas a la hora de realizar aplicaciones web:

- a) Es un lenguaje de etiquetado muy simple[4]
- b) Al igual que el html se basa en documentos de texto plano
- c) Nos permite introducir la extensibilidad, estructura y validación que ofrece SGML en el dominio de la web, que eran cosas que faltaban en HTML[6, pag5]
- d) XML sirve para estructurar, almacenar e intercambiar información<sup>1</sup>

Podríamos decir entonces que XML siguió a SGML, pero con la diferencia que no es tan robusto, como lo que plantea el SGML, pero a diferencia del HTML, ofrece la posibilidad de extender las marcas

## 2. VISORES Y ANALIZADORES XML

Visores de XML:

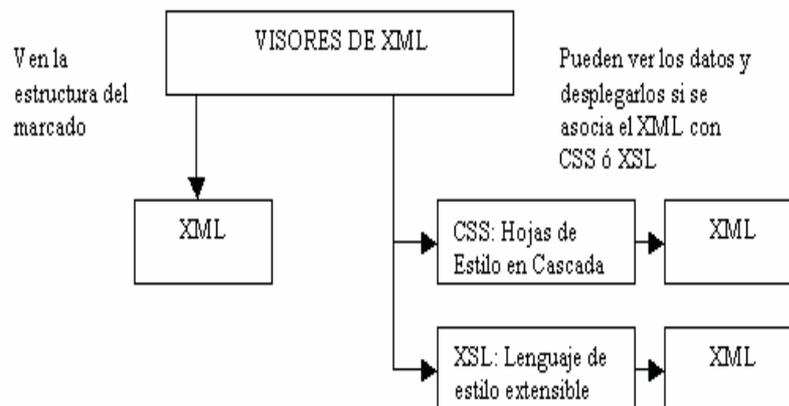


Figura 2. Visores de XML. Elaboración propia

---

<sup>1</sup> Es una de las definiciones más populares y que han sido durante mucho tiempo uno de los lemas del consorcio WWW.

Los visores son pues software especializado para mostrar archivos XML y los analizadores son software que permiten comprobar si dichos documentos XML están bien estructurados y bien formados<sup>2</sup>

Las aplicaciones XML no leen directamente el XML, sino que lo hacen a través de un procesador y el procesador no analiza directamente la sintáxis del documento XML, sino que lo hace a través del analizador XML, tal como se muestra en la siguiente figura:

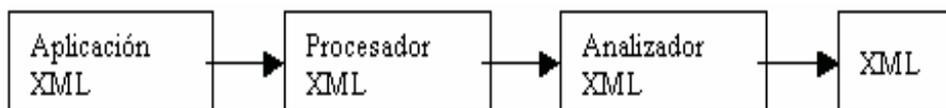


Figura 3: Explicación como las aplicaciones leen documentos XML (elaborado por el autor)

Con lo cual se está diciendo que es necesario validar un documento XML. La pregunta del millón es: ¿Cómo validar si un documento es válido o cómo sabe el analizador XML si el documento XML está bien formado?. La respuesta necesita que se aclare lo siguiente: “Recordemos que XML permite personalizar el contenido que se desea trabajar y por tanto para construir un analizador que analice cualquier tipo de documento sería complicado por cuanto el número de aplicaciones posibles es astronómicamente elevado y grande.”. La respuesta a la pregunta planteada viene dada por una tecnología inmersa en la tecnología XML y es la tecnología de definir esquemas. Gráficamente la situación anterior se puede ver de la siguiente manera:

---

<sup>2</sup> Documentos bien formados significa que sintácticamente no tienen errores y documentos válidos significa que el contenido de dicho documento es acorde con una lógica de pensamiento.

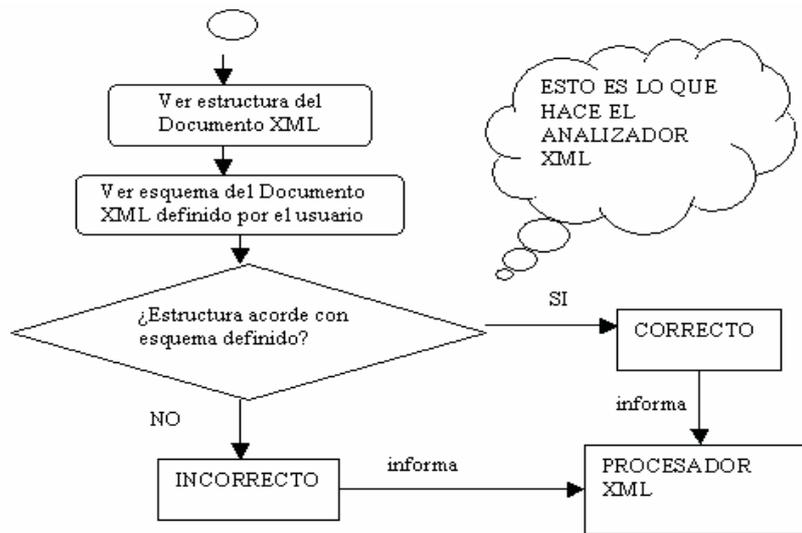


Figura 4. Elaborado por el autor, basándose en los capítulos 3,4 y5 del libro[6]

### 3. LOS MODELOS DE OBJETOS

Antes de explicar el modelo de objetos de documento que se puede usar para acceder a partes de un documento XML, es bueno conocer que este modelo es fruto del avance de otros modelos tradicionales que desde poco después de la aparición de Internet se fueron creando. En primera medida decimos que un modelo de objetos define la interfaz para los diversos aspectos del navegador y del documento que inicialmente se controlaban con JavaScript. [7 Pág 237]

El Modelo de Objetos de documento que implementa Netscape presenta algunas diferencias radicales con respecto al modelo de objetos de documento implementado por Internet Explorer, lo cual conlleva obviamente a problemas en el desarrollo de aplicaciones grandes de Internet, que tenga necesariamente clientes de ambos exploradores. Para subsanar estas

asperezas es necesario que se desarrollen aplicaciones web que cumplan el modelo de objetos de documento propuesto por el w3c, debido a que este modelo de objetos de documento es un modelo que tiende a ser el estándar por excelencia no solamente para los exploradores antes mencionados sino también para navegadores de linux como Konkeror Opera o Mozilla<sup>3</sup>

### 3.1. COMO FUNCIONA EL DOM

El modelo de objetos describe la interfaz necesaria para manipular aspectos del navegador y del documento. Es importante tener en cuenta que en los navegadores existe la posibilidad de ejecutar páginas htm que contengan programación que permita la manipulación del mismo documento como tal y también la manipulación del navegador en el que se contiene la página. En otras palabras, podemos notar que el aspecto de un documento visto desde un navegador esquemáticamente hablando es el siguiente:

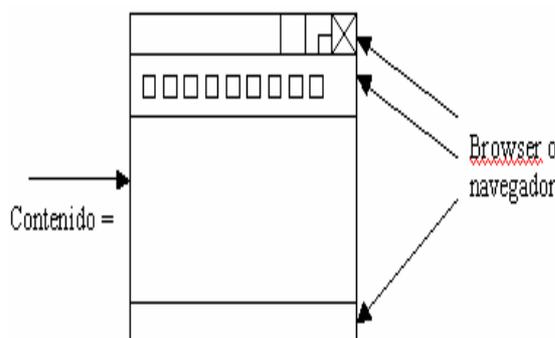


Figura 5: Elaborada por el autor

---

<sup>3</sup> Este es un resumen hecho por el autor sobre el capítulo nueve y específicamente sobre la sección “modelo tradicional de objetos de JavaScript” [7 Pág 237]

Con lo cual diremos que deberían existir formas de poder manipular por código desde una página aspectos relevantes a la presentación del documento como tal, así como aspectos relevantes del navegador. Después de analizar esta situación los desarrolladores de navegadores han incluido dos modelos uno para cada propósito. Tenemos entonces los modelos DOM y BOM, tal como lo describe la siguiente gráfica:

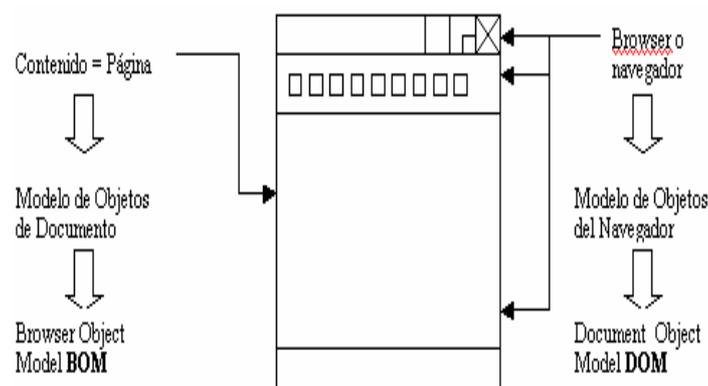


Figura 6: Elaborada por el autor

Podemos pensar que a medida que JavaScript va fortaleciéndose se van añadiendo nuevas funcionalidades a dicho lenguaje por medio de la ampliación de dicho lenguaje. Esto lleva a que haciendo un poco de memoria histórica, JavaScript haya implementado un primer modelo originalmente y después haya evolucionado dicha implementación siguiente pues un modelo de objetos más avanzado. Obviamente y como es sabido por todos JavaScript es un lenguaje que nació casi de la mano con internet y al igual que Internet, tiene muchos desórdenes, razón por la cual no se nos debe hacer extraño que JavaScript tenga algunas implementaciones distintas en los diversos navegadores.

Como decíamos anteriormente JavaScript ha evolucionado y sigue evolucionando, razón por la cual a continuación se muestran cuatro modelos de objetos distintos que han sido implementados por JavaScript a lo largo de su historia:

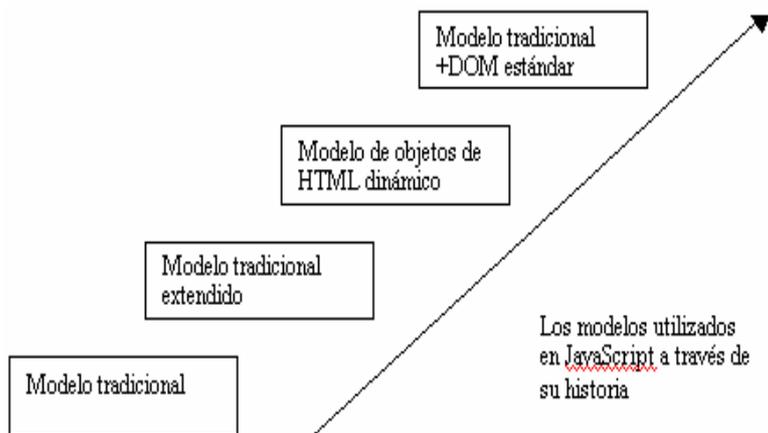


Figura 7: elaborada por el autor

### 3.2. EL MODELO DE OBJETOS TRADICIONAL

El modelo tradicional de JavaScript es obviamente el más sencillo ya que fue el primero adoptado por JavaScript y se muestra a continuación el diagrama de dicho modelo de objetos tradicional:<sup>4</sup>

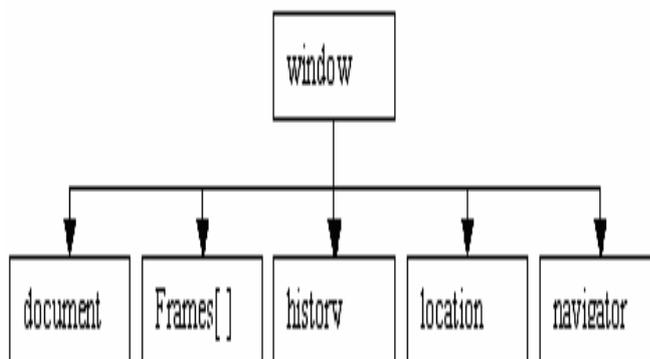


Figura 8: elaborada por el autor

<sup>4</sup> Este es un resumen hecho por el autor sobre el capítulo nueve y específicamente sobre la sección “modelo tradicional de objetos de JavaScript” [7 Pág 240-247]

El objeto document permite tener acceso a propiedades del documento y a elementos del documento. A pesar que como tal no existe un objeto en JavaScript que permita esta diferenciación, en el siguiente diagrama para ser un poco didácticos se muestra dicha relación:

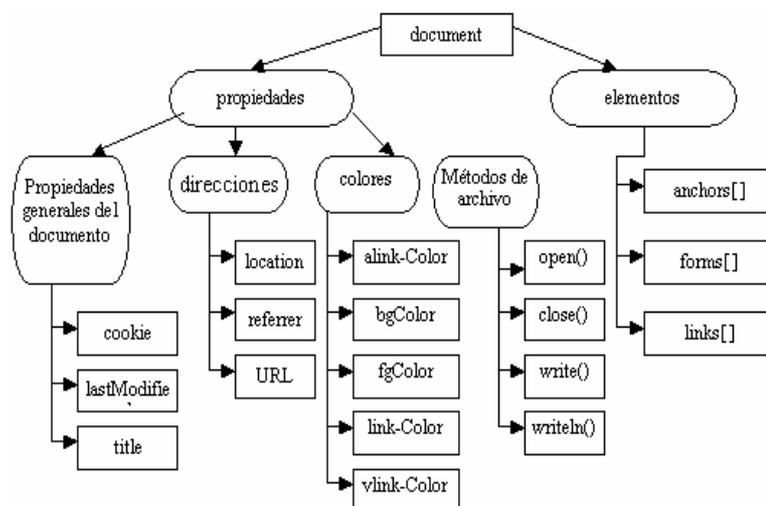


Figura 9: elaborada por el autor

En la anterior gráfica es bueno que el amigo lector / estudiante tenga en cuenta que los elementos que se encuentran en rectángulos corresponden a implementaciones de JavaScript, pero los que se encuentran en rectángulos con bordes curvos corresponden a clasificaciones hechas a manera didáctica para lograr comprender mejor el modelo.

A continuación se muestra la jerarquía de los elementos que se encuentran por abajo del elemento elements[ ] en el modelo de objetos de JavaScript.

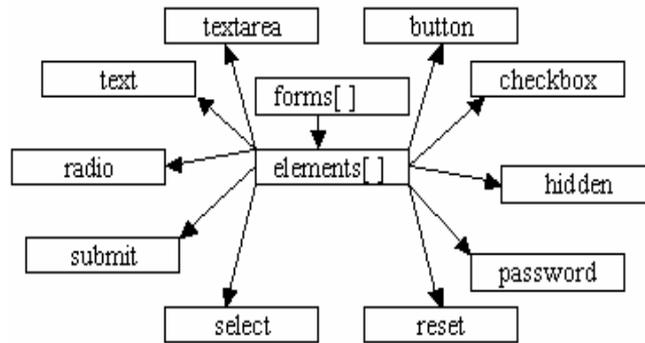


Figura 10: elaborada por el autor

### 3.3. EL MODELO DE OBJETOS DE INTERNET EXPLORER4

El modelo de objetos tal como lo hemos visto anteriormente carece de muchos elementos que ahora se trabajan en las nuevas páginas dinámicas. Es así como se tiene entonces un modelo de objetos de documento al servicio de dhtml, el cual como su nombre lo indica proporciona compatibilidad con las adiciones que trajo el dhtml al html. Con el fin de lograr hacer esta parte lo más descriptiva posible, a continuación se muestra el modelo de documento que implementa Internet Explorer 4, el cual es un modelo que trae algunas cosas nuevas:<sup>5</sup>

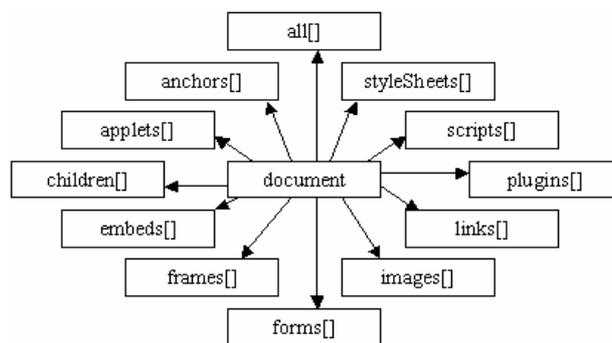


Figura 11: elaborada por el autor

<sup>5</sup> Este es un resumen hecho por el autor sobre el capítulo nueve y específicamente sobre la sección “modelo tradicional de objetos de JavaScript” [7 Pág 264-270]

Es interesante observar que la lista de modelos es más grande, pero por razones de simplicidad no se van a mostrar todos, simplemente se va a decir que a medida que los navegadores mejoran sus funcionalidades, dichos modelos se vuelven más complejos y al mismo tiempo la programación para los desarrolladores web, se vuelve más complicada por cuanto los modelos de un navegador no corresponden con los de otro navegador y a medida que pasa el tiempo se requiere que los sitios web se vean en todos los navegadores y en todas las versiones de cada navegador en forma uniforme

De otra parte es bueno notar que los modelos son una forma ideal para acceder a partes de un documento y siguen teniendo validez en documentos XML, claro que con algunas variaciones, por cuanto el modelo DOM para acceder a documentos XML presenta algunas diferencias, de lo que se trataba en esta parte era de mostrar que la historia de por qué XML se puede manipular mediante un modelo no es nueva, sino que es una idea acuñada desde la misma programación HTML.<sup>6</sup>

#### **4. TECNOLOGÍAS RELACIONADAS CON XML**

Según el [www](http://www), existen algunas tecnologías disponibles de XML como: Xsl, Xpath, Xlink, Xpointer, Xql[4], pero la verdad otros autores hablan de tecnologías como Dom XML (un subconjunto de xsl), la recomendación 1.0 de espacios de nombres XML y esquemas XML avanzados[6 pag 16] e incluso sitios de renombre en el campo de las tecnologías hablan de

---

<sup>6</sup> Recordemos que la programación HTML se acuña cuando surge un lenguaje de Script como JavaScript, el cual fue el primer lenguaje de Script para la web, aunque después surgieron otros como VBScript, éste no tuvo tanta acogida como el primero, debido a su incompatibilidad con navegadores como Netscape

tecnologías relativas a XML y las clasifican en: Esquemas, hojas de estilo y programación. En los esquemas hablan de Dtd, Esquema y Relax-NG, en las hojas de estilos hablan de XSL-FO, XSLT, Xlink, Xpointer y CSS[6]<sup>7</sup>

A parte de las anteriores hay varios textos que tratan el tema de aplicaciones simplificadas de XML para usos específicos y entre ellas tenemos las siguientes: P3P, MATHML (para trabajar con fórmulas matemáticas), XBRL (lenguaje extensible de reportaje de negocios)<sup>8</sup>, RSS[11], SMIL(Lenguaje sincronizado de integración multimedia)[6 pag 652], RDF[1 Pág. 2]<sup>9</sup> (Formato universal para lograr intercambiar datos en la web), SVG (Scalable Vector Graphics)<sup>10</sup> 3DML(Lenguaje de modelado de realidad virtual), WIDL, WML, VML, SOAP[3 Cap 7]<sup>11</sup> y DSML 1.0, Directory Services Markup Language (basado en XML, específico para definir directorios)

A continuación se da una descripción global de algunas de éstas tecnologías de XML:

#### P3P:

Permite a los usuarios establecer preferencias que rigen el modo en el que la información se va a compartir con sitios web. La idea es que un sitio web proporcione unas normas de privacidad para que el usuario las valore y decida en que grado desea compartir información con el sitio. Al menos una parte de este proceso puede ser automatizado a través de un vocabulario basado en XML que anuncie normas de intimidad [6 pag 770]

---

<sup>7</sup> Aunque hablan en otras palabras, si mencionan exactamente dichas tecnologías y muestran enlaces internos y externos en donde el amigo lector / estudiante puede consultar más sobre cada uno.

<sup>8</sup> XBRL es el acrónimo de eXtensible Business Reporting Language

<sup>9</sup> RDF (Resource Description Framework)

<sup>10</sup> Es un lenguaje para describir gráficos bidimensionales en XML

<sup>11</sup> En este libro se tratan estas últimas tecnologías como derivados de Xml

SMIL:

Es un lenguaje útil para integrar objetos multimedia y provee formas para integrar medios y sincronizarlos [6 pag 652]<sup>12</sup>

MATHML

Es un vocabulario XML diseñado para modelar contenido matemático[6 pag 750]

RSS

Sumario enriquecido de un sitio (Rich Site Summary)

Algunas personas creen que xml es un simple lenguaje de marcado que no va a llegar tan lejos y que sus repercusiones no serán más grandes de lo que en la actualidad se ve, pero la verdad es que xml revolucionará el mundo hasta de los programas basados en reglas y como si fuera poco es muy posible que el aspecto del diseño de datos para que las interfaces de lenguaje natural puedan manipular dichos datos<sup>13</sup>, sea revolucionado con XML

## **5. ¿PUEDO CREAR UNA NUEVA TECNOLOGÍA BASADA EN XML?**

A pesar que la traducción estricta sobre el significado de las siglas XML es: Lenguaje Extensible de marcado, se puede asegurar que es más que un lenguaje e incluso algunos

---

<sup>12</sup> Si desea profundizar sobre SMIL puede consultar <http://www.w3.org/TR/smil-boston>

<sup>13</sup> Beth Breidenbach arquitecto de productos para Getronics, trabaja en este tipo de aplicaciones actualmente y es revisor técnico del libro c++ xml guía avanzada, editorial prentice may. Año 2002

autores defienden la idea basada en el concepto que XML no es un lenguaje en sí mismo, sino un sistema para definir lenguajes de propósito específico [5 Pág 405]

En el presente texto, se pretende mostrar algunos elementos que podrían dar una luz sobre la posibilidad que tienen los desarrolladores que conozcan bien XML para desarrollar nuevas tecnologías basadas en XML. Supongamos que usted desea hacer una aplicación que intercambie información con otra aplicación, pero que sus datos sean de tipo matemático, entonces tiene dos opciones:

1. Inventarse un formato sin la ayuda de XML para capturar los datos que el usuario digite y procesarlos de acuerdo como requiera la aplicación. Esta solución obviamente traería un problema y es que todos deben conocer el formato en el que se deben ingresar los datos matemáticos y si no los conocen, tendríamos entonces que capacitar a dichas personas para que conozcan como ingresar información al sistema. Es importante notar que ésta solución traería otro inconveniente y es el de la escalabilidad[18]<sup>14</sup>, debido a que en el momento en el que nuevos usuarios o por no ir tan lejos, nuevas aplicaciones quieran interactuar como usuarios de dicho aplicativo, tendríamos problemas de comunicación. Esto no quiere decir que esto no se podría hacer, simplemente quiere decir que sería necesario construir rutinas adicionales que se encarguen de interpretar de quien provienen los datos con el fin que los transforme para lograrlos comprender la otra aplicación. Ahora bien, imagínese el problema si se quieren adicionar cuatro usuarios que tienen formas diferentes de hablar, tendríamos un problema más grande y exponencialmente el problema se vería reflejado en las siguientes figuras:

---

<sup>14</sup> De Wikipedia, la enciclopedia libre. Dirección se encuentra en la infografía

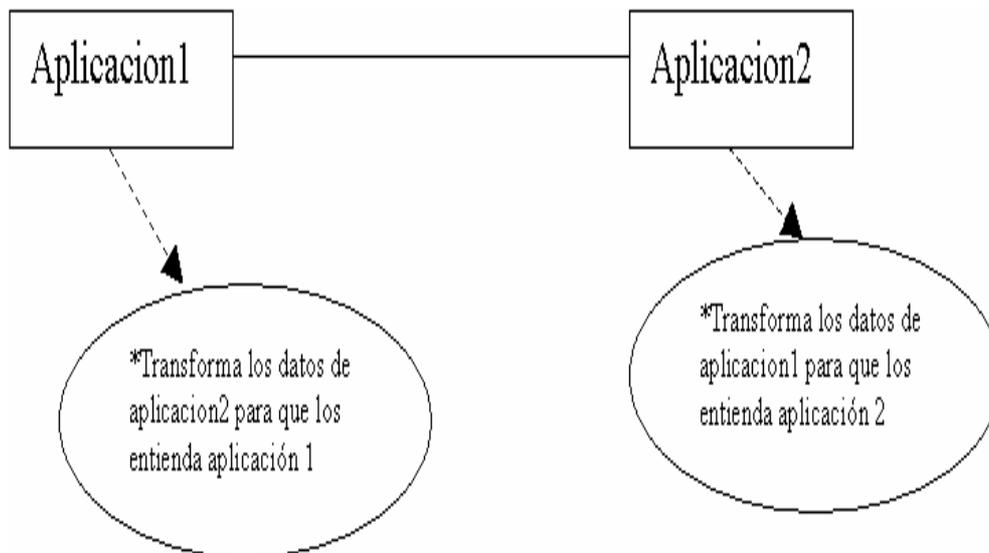


Figura 12: elaborada por el autor

En el grfico anterior, se muestra que si las dos aplicaciones son de empresas distintas y si dichas empresas no usan a XML como estndar para intercambio de informaci3n, tendramos que construir un m3dulo en la aplicaci3n que transforme los datos de la aplicaci3n 2 para que los entienda la aplicaci3n 1 y en forma anloga, tendramos que construir un m3dulo en el aplicativo 2 que transforme los datos de la aplicaci3n 1 para que los entienda el aplicativo 1. A continuaci3n se muestra que 3ste tipo de soluci3n presenta algunos inconvenientes cuando el nmero de aplicaciones que interactan crece.

Un problema ms grave sucede si se requiere aadir una tercera aplicaci3n. Veamos:

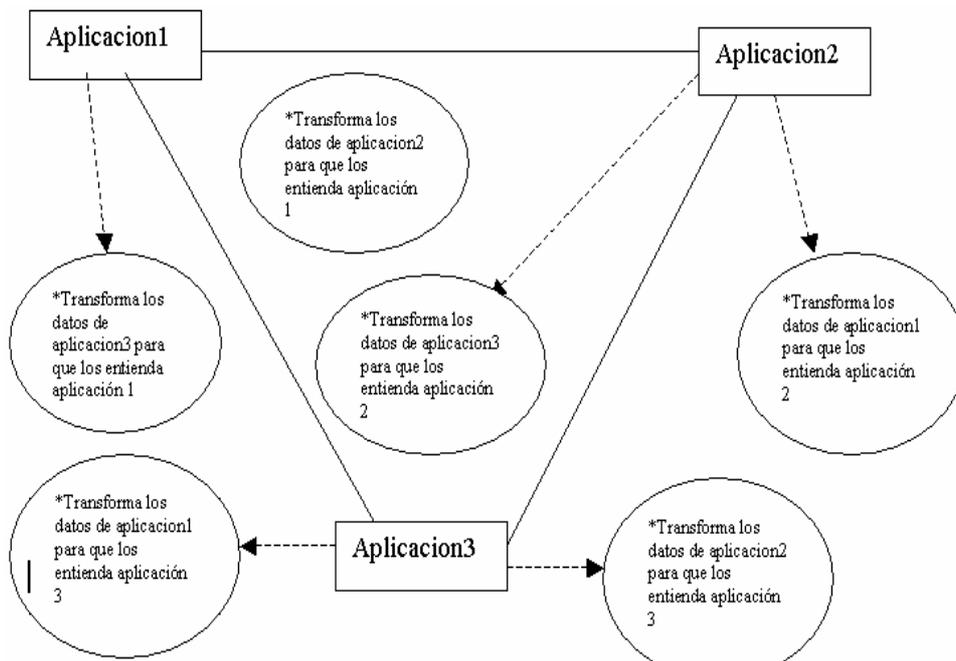


Figura 13: elaborada por el autor

Con lo cual queda claro que el número de rutinas que es necesario construir para aceptar más clientes a una aplicación, cuando las aplicaciones se intercomunican son exponenciales. Tendríamos pues la siguiente tabla:

Tabla 1.

Número de aplicativos conectados	Número de módulos que es necesario desarrollar para tener una perfecta comunicación
1	0
2	2
3	6
4	12
5	20

Y como conclusión decimos que cuando tengamos N aplicativos conectados, tendremos que desarrollar  $N*(N-1)$  módulos para tener nuestros sistemas perfectamente comunicados. Esto obviamente también genera una serie de traumas en el rendimiento de las aplicaciones,

porque el gran módulo de intercambio de información estaría compuesto por una serie de módulos para lograr que ésta aplicación se comuniquen con las demás aplicaciones. Esto obviamente llevará a que la escalabilidad, es decir el número de clientes de una aplicación no pueda crecer rápidamente. Otra consecuencia funesta de escoger esta solución para intercambiar información es la de la incompatibilidad[9], y el coste económico elevado entre otras desventajas.<sup>15</sup>

2. Inventarse un formato para intercambiar información, basado en XML. Esta opción es bastante buena debido a que garantiza que las salidas de una aplicación de otra compañía que también use XML, sean perfectamente compatibles con las entradas de nuestra aplicación y viceversa, las salidas de nuestro aplicativo sean compatibles con las entradas del otro aplicativo. La pregunta que se nos viene en este momento es: ¿No será tortuoso para otros usuarios tener que generar salidas en XML para que sean tratadas como insumos de nuestras aplicaciones?. La respuesta es bien sencilla, sería tortuoso si no existieran herramientas que permitan la transformación de datos en datos XML, pero la verdad, en la actualidad están surgiendo una serie de herramientas que le permiten convertir por decir algo un documento html a un documento XML, pero a pesar de esto, podríamos pensar que ésta solución no es tan buena, debido a que dichos sistemas todavía no es que hagan muy bien el trabajo. Entonces, tenemos la otra posibilidad, decirles a las bases de datos remotas que nos entreguen la información en XML, cosa que ahora ya todos los motores de bases de datos más importantes del momento lo hacen (bueno, todavía no incluyo en esta lista a mysql). Pues bien, con todo lo expuesto hasta el momento queda claro que por estos medios

---

<sup>15</sup> En la página 18 del libro citado en [9], Adolfo Vásquez plantea otras estas desventajas y las explica de otra manera.

es posible que una aplicación entregue documentos XML a otra para lograr una perfecta comunicación entre los dos, pero viene entonces otra pregunta: ¿Los usuarios no experimentados cómo hacen para desarrollar documentos XML y pasárselos a otra aplicación?. La respuesta a ésta última pregunta es bien interesante y tiene que ver con el amplio desarrollo actual de múltiples herramientas que prometen generar y revisar los guiones. Obviamente éstas herramientas son lo suficientemente claras y fáciles de manejar como para lograr que los usuarios actuales puedan crear sus propios documentos. Bueno, pero a pesar de todas estas maravillas, muchas de éstas herramientas presentan algunos inconvenientes, pero se supone que cuando se arreglen todas éstas dificultades tendremos a XML más arraigado al desarrollo de múltiples aplicaciones.

De todas formas es interesante notar que para crear una nueva tecnología basada en XML se requiere que dicho lenguaje XML sirva para crear lenguajes de extensión<sup>16</sup>. La pregunta que nos viene a la cabeza es: ¿Es posible crear lenguajes de extensión con las tecnologías existentes?. La respuesta a esta pregunta es que sí y esto es cierto desde mucho antes que se popularizara XML, esto comenzó a ser posible desde que se habló de xhtml (que fue una forma de introducirnos a XML e incluso se desarrollaron en su tiempo múltiples aplicaciones que convertían xhtml a XML y viceversa). Bueno, y ¿cómo es posible que por medio de xhtml pueda crear nuevos lenguajes de marcas?. Tenemos entonces que la respuesta a ésta última pregunta se comprende cuando se menciona que xhtml permite la modularización[14]<sup>17</sup>. Es tanto así que se pueden crear módulos propios y de declaración a partir de xhtml.

---

<sup>16</sup> Los lenguajes de extensión son la base para interoperar entre aplicaciones de propósito específico[14]

<sup>17</sup> Xhtml se puede subdividir y esto le permite extenderse a si mismo en módulos xhtml

Con todo lo dicho anteriormente se puede observar que a partir de xhtml, se puede extender y se podría crear un lenguaje extensible. Ahora la pregunta es: ¿si se podía extender xhtml y crear lenguajes extensibles a partir de éste, por qué no se podrá hacer esto mismo con XML que es mucho más avanzado que xhtml?. La respuesta a ésta pregunta es un rotundo sí, al decir que por supuesto que es posible extender y crear un lenguaje extensible a partir de XML, y esto lo han dicho innumerables autores entre los que se cuenta Fabio arciniegas, quien escribió lo siguiente: “¿por qué usar XML como lenguaje de extensión?. Algunas plataformas ya definen mecanismos para que los programas expongan sus interfaces con el fin que los lenguajes de script puedan manipularlos. Además la sintaxis XML no es tan concisa como en otros lenguajes, así que ¿por qué crear lenguajes de extensión basados en XML?. La respuesta es doble. Primero desde la perspectiva del usuario, escribir guiones en XML puede ser mucho más agradable y seguro que en Lisp, Visual Basic o algún otro lenguaje de script. El usuario ya tiene las herramientas para hacerlo, conoce las convenciones y la sintaxis subyacente y puede usar una gran cantidad de herramientas gratuitas disponibles para generar y revisar los guiones. Segundo desde la perspectiva del desarrollador, usar XML conlleva un compromiso de portabilidad, transparencia, la posibilidad de implementar fácilmente extensiones de enlace multiplataforma sin confiar en ninguna plataforma o middleware particular (como CORBA o COM), robustez y muchas oportunidades para reutilizar código. (Por ejemplo, ya no se necesitará escribir un analizador sintáctico de bajo nivel para el lenguaje. Para ese fin ya tenemos SAX)”[2 Pág 240]

Ahora para finalizar nuestro raciocinio, podemos pensar que si ya hemos comprobado que a partir de un lenguaje extensible podemos crear una nueva tecnología y además hemos

comprobado que hay autores que afirman que a partir de XML se puede crear un lenguaje extensible, podemos concluir que a partir de XML se pueden crear tecnologías nuevas. De hecho son muchos los elementos que entran a apoyar la idea que a partir de un lenguaje extensible se pueden crear nuevas tecnologías y si no: ¿qué son las nuevas tecnologías basadas en XML?. Nos queda entonces solucionar otra duda: ¿el lenguaje extensible que se cree y las nuevas tecnologías que se creen serán interoperables y tendrán todos los beneficios que tienen actualmente las aplicaciones para XML?. La respuesta es otro rotundo sí, puesto que los lenguajes extensibles que se creen a partir de XML, son lenguajes que siguen siendo lenguajes XML, pero con características especiales y particulares que se adaptan a una determinada situación, a un entorno y a una problemática concreta, pero que siguen siendo soluciones XML.

Entonces comprendemos con todo esto que en verdad con XML podemos hacer muchas cosas. Pues tanto es así, que tenemos incluso metalenguajes que nos servirán para definir posibles lenguajes, y si esto no fuera así, ¿que sentido tendría hablar de la web semántica, de ontologías, de MDA y de otros temas de actualidad en el ámbito de las tecnologías?.

## **6. ¿CUÁNDO USAR XML?**

La pregunta que a todas las personas que comienzan a comprender que surgió una nueva tecnología llamada “XML”, es la siguiente: ¿Debo usar XML siempre?. La respuesta es que no todas las veces XML ofrece la mejor posibilidad para almacenar los datos y se podría pensar que cuando la información está bastante estructurada y se tiene la certeza que su

distribución es regular, podemos afirmar que cabría la posibilidad de usar XML no es tan buena como la de usar un motor de bases de datos relacional, obviamente suponiendo que existen otra serie de factores que favorecen ésta toma de decisión, como por ejemplo el hecho que no se requiera interoperar con otros sistemas y que tampoco se requiera cambiar la estructura rígida de almacenamiento de los datos.

Todo lo anterior nos deja entrever que en verdad XML se irá imponiendo como una forma para almacenar información, debido a que dicha información a pesar de ser almacenada en forma estructurada, no es tan rígida como sí lo es en los motores de bases de datos relacionales

## **7. ¿XLINK Y XPOINTER MEJORARÁN LOS VÍNCULOS ACTUALES?**

Xlink y Xpointer son tecnologías que están demostrando su utilidad en estos tiempos modernos. Decimos que los vínculos html que conocemos son bastante buenos para muchas tareas, pero se podrían haber diseñado de mejor manera, de tal suerte que pudieran permitir mayor flexibilidad al momento de navegar por la web. Comencemos comprendiendo lo que hacen los vínculos html, podemos afirmar que dichos vínculos están asociados con un origen y un destino, con lo cual el origen es el recurso en donde se activó el vínculo y el destino es el recurso al cual se desea llegar cuando se llegue a dicho vínculo. La pregunta es entonces: ¿cómo se pueden mejorar los actuales vínculos?. La respuesta a ésta pregunta no es tan sencilla, por cuanto las mejoras que se le podrían hacer a los vínculos actuales no es tan clara para la mayoría de personas y usuario de la web. En primera medida podremos

decir que los vínculos html son unidireccionales, es decir que los vínculos no me permiten ir al recurso en el que se encontraba el navegante antes de activarlo. De otra parte a pesar que los navegadores web tienen el soporte de atrás mediante un botón que permite la navegación atrás, es importante comprender que éste soporte es algo adicional de los navegadores, pero en sí el vínculo no tiene forma de saberlo. Otra limitación de los actuales vínculos es que el destino de un vínculo es único y no es posible asociar un vínculo con varios recursos de destino, lo cual haría que la web fuera mucho más interactiva y más fácil de desarrollar.<sup>18</sup>

## **8. BASES DE DATOS Y XML**

Hasta ahora se ha mencionado como con XML podemos crear documentos bastante estructurados, pero para todos es claro que los motores de bases de datos también almacenan información estructurada, la pregunta es: ¿cuál es la diferencia?. La respuesta que hemos tratado de mencionar en el presente documento está dada por el hecho que la información en XML a pesar de ser estructurada puede ser irregular<sup>19</sup>

---

<sup>18</sup> En el capítulo 19 los autores de XML al descubierto tratan el tema xlink y xpointer con más detalle y muestran no sólo éstos aspectos, sino otros aspectos interesantes de éstas nuevas tecnologías de vinculación de documentos.

<sup>19</sup> En la página 454, en el capítulo 22 se habla sobre la utilización de patrones XSL y XQL y se deja en claro que xml puede almacenar información irregular.[6]

## 8.1. SQL SERVER 2000 Y XML

En SQL Server 2000 se trabaja también el lenguaje XML, y existen algunas cuestiones bien particulares en la forma como lo trabaja Microsoft, entre una de ellas podemos mencionar que para la creación de esquemas de un XML, se usa un lenguaje XML simplificado (XDR) que es una forma muy particular de Microsoft para hacer lo mismo que hacen las DTD<sup>20</sup>, aunque también se admite el uso de las mismas

Si dispone de esquemas XDR públicos (como los esquemas Microsoft BizTalk<sup>21</sup>), puede realizar alguna de las siguientes operaciones:

a. Escribir la consulta del modo FOR XML EXPLICIT para que los datos generados sean válidos con el esquema XDR público; sin embargo, escribir consultas FOR XML EXPLICIT puede ser algo complicado.

b. Haga una copia privada del esquema XDR público. Luego agregue anotaciones a esta copia privada para así generar un esquema de asignación. Puede especificar consultas XPath con el esquema de asignación. En consecuencia, la consulta genera los datos en el espacio de nombres del esquema público. Crear esquemas anotados y especificar consultas XPath con ellos es un proceso mucho más sencillo que escribir consultas complejas FOR XML EXPLICIT. La siguiente ilustración muestra el proceso.

---

<sup>20</sup> Una DTD es un archivo XML muy particular que se usa para validar la estructura de otro archivo XML

<sup>21</sup> Microsoft BizTalk Framework es un intento de definir un formato XML estándar para objetos comerciales comunes, como Contactos, Pedidos y Reuniones. Podrá encontrar copias de estos esquemas comerciales en <http://biztalk.org>. Este tipo de información se encuentra también en el sitio de SQL Server de Microsoft

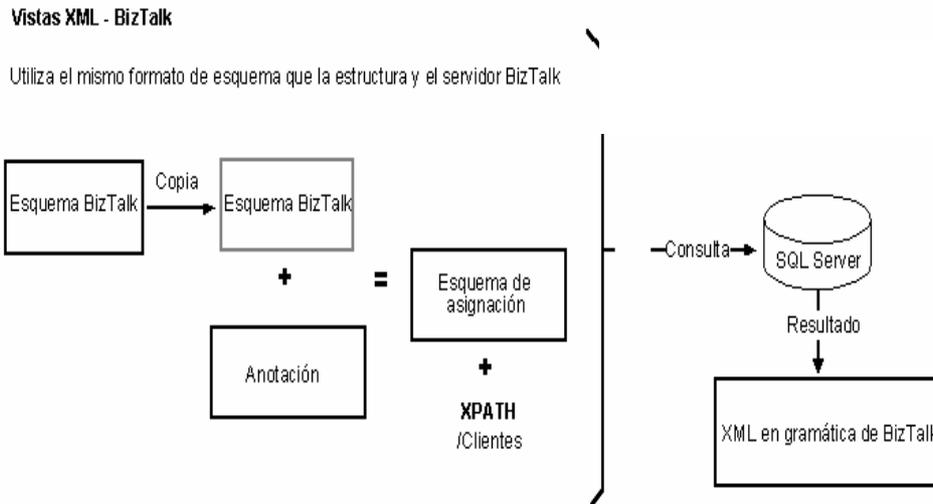


Figura 14: Grafica de la ayuda de SQL Server 2000. Vistas XML con BizTalk en SQL Server<sup>22</sup>

Con el anterior gráfico es posible deducir que Microsoft brinda soporte no sólo a las cosas básicas de XML como las DTD, los esquemas, sino también a cosas que incluso java no ha brindado soporte y es precisamente el soporte a XPATH, pero de todas formas es importante anotar que la versión de Java j2sdk1.5 promete dar soporte a éste tipo de tecnologías de XML.

Es importante mencionar que el soporte a XML en las bases de datos lo ha dado Microsoft a partir del lanzamiento de su motor SQL Server 2000 y a partir de éste motor las otras versiones del manejador de bases de datos seguirán dando mayor soporte a ésta tecnología, según palabras de Bill

<sup>22</sup> Para obtener éste gráfico se entra a la ayuda de Libros en pantalla de SQL Server 2000, se busca en índice la palabra "XML" y se elige la opción "Consulte también XDR y el ítem "Crear vistas XML mediante esquemas XDR anotados""

## 8.2. XML Y LA ARQUITECTURA DE TRES NIVELES EN MICROSOFT

Si XML sirve para comunicar aplicaciones, podemos preguntar: ¿el mismo servidor IIS de Microsoft se puede comunicar con otras aplicaciones usando tecnología XML?. Obviamente si la respuesta a la anterior pregunta fuera negativa estaríamos diciendo que Microsoft pregona una cosa, pero internamente sus aplicaciones no implementan dicha tecnología, pero bueno, afortunadamente la respuesta a la pregunta anterior es totalmente afirmativa y decimos que las peticiones válidas entre el servidor IIS y los clientes con un browser pueden ser HTML ó XML. De otra parte las consultas XML(que pueden ser OPENXML ó FOR XML) en conjunto con plantillas, esquemas y archivos XSL son recibidas por la dll “SQLXMLX.DLL” cuando el SQLOLEDB detecta que son consultas XML y dicha dll es la que la ejecuta y devuelve el resultado nuevamente a “SQLOLEDB”. Los archivos de plantilla, los archivos de esquema XML simplificado (XDR) y los archivos de lenguaje de hojas de estilo ampliable (XSL) residen en el servidor IIS. Las consultas XPath y los esquemas XDR se controlan en el servidor IIS. Las consultas XPath se convierten en comandos SQL mediante Sqlxmlx.dll

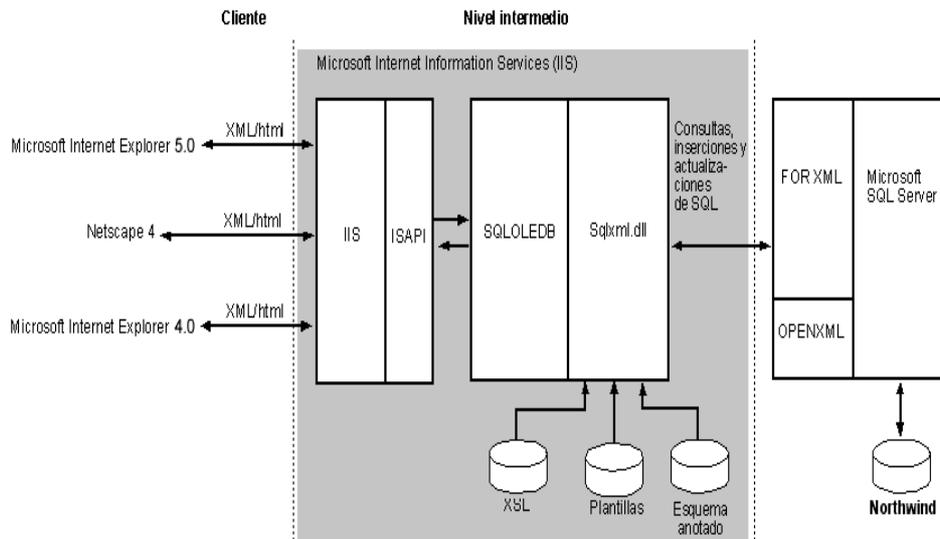


Figura 15: elaborada por el autor. Gráfico obtenido de la ayuda de SQL Server 2000<sup>23</sup>

### 8.3. ORACLE Y XML

Oracle a partir de su versión 8i utiliza xml para analizar y procesar los archivos en su iFS, (Internet File System) es decir en su sistema de archivos de internet<sup>24</sup> el cual proporciona una integración de datos relacionales con datos no relacionales. En realidad la base de datos con el nuevo iFS convierte el contenido de los archivos conocidos en archivos XML utilizando un filtro de archivo, posteriormente se puede almacenar el XML en una tabla de la base de datos. Cuando se haga una consulta del documento, ORACLE es capaz de usar un traductor el cual convierte el documento XML en el formato que el usuario desee, bien sea HTML, XML, etc[16 Pág 261]

<sup>23</sup> Para obtener éste gráfico se entra a la ayuda de Libros en pantalla de SQL Server 2000, se busca en índice la palabra “XML” y se elige la opción “Obtener acceso al servidor SQL Server en la URL”

<sup>24</sup> Con el nuevo sistema iFS de archivos de Oracle se pueden almacenar las páginas web, los documentos, las hojas de cálculo, los archivos de procesamiento de texto, etc dentro de la base de datos.

## 9. XML Y LAS TECNOLOGÍAS INALÁMBRICAS

El lenguaje de marcado XML cuando se usa para tecnologías inalámbricas adquiere unos aspectos interesantes y concretos, lo cual ha dado origen al WML(Wireless Markup Language), el cual está elaborado específicamente para desarrolladores de aplicaciones que usen tecnología WAP (Wireless Application Protocol). En el siguiente cuadro se muestran algunas de las analogías que existen entre HTML y WML con el fin de tener un referente de las equivalencias entre las tecnologías que se usan en un lenguaje y en el otro.

ITEM DE COMPARACION	TECNOLOGÍA HTML	TECNOLOGÍA WML
ETIQUETAS	Etiquetas preestablecidas	Admite etiquetas no preestablecidas
APLICABILIDAD	Aplicable a servidores web y a computadores	Aplicable a teléfonos móviles o los PDA(Personal Digital Assistant)
Protocolo relacionado	http	WAP
Manejo de hiperenlaces	Proporciona información por medio de hiperenlaces	Proporciona información por medio de hiperenlaces
Lenguajes de Script relacionados	JavaScript VBScript	WMLScript
Forma de crearlos	Se pueden crear en un editor de textos plano	Se pueden crear en un editor de textos plano
Aplicación para visualizarlos	Es suficiente un navegador o browser.	Para visualizarlo en un computador es necesario tener instalado un simulador o emulador de dispositivo inalámbrico como teléfono móvil o PDA
Presencia de estructuración de formación al visualizarlos	Algunos navegadores no son estrictos en chequear que estén bien formados para ser visualizados	Deben estar bien formados para ser visualizados

Existencia de herramientas para su edición y elaboración	Existen herramientas propietarias y libres para su edición	Existen herramientas propietarias y libres para su edición
Organización que establece los estándares	El consorcio W3C establece los estándares para tecnologías de protocolo HTTP.	El WAP forum establecía los estándares <a href="http://www.wapforum.org">http://www.wapforum.org</a> , ahora aunque sigue siendo el mismo ha cambiado su nombre por OMA (Open Mobile Alliance) y se puede encontrar en el sitio web <a href="http://www.openmobilealliance.org">http://www.openmobilealliance.org</a>
Elementos básicos del documento	Principalmente HEAD y BODY	HEAD, TEMPLATE Y CARD
Extensión que tienen los script desarrollados para esta tecnología	js	wmls
Extensiones válidas	Htm,html	Wml

Tabla 2: Comparación entre HTML y XML<sup>25</sup>

Como se decía anteriormente wml está basado en xml, lo que quiere decir que se parece en muchas cosas a dicho lenguaje extensible de marcado, pero obviamente debe haber diferencias o similitudes, de las cuales algunas se muestran en la siguiente tabla comparativa:

Tabla 3. Comparación entre XML y WML

ITEM DE COMPARACION	TECNOLOGÍA XML	TECNOLOGÍA WML
Admisión de DTD	Si	Si
Visualizador	Browser	WMLBrowser
Tecnologías que puede soportar directamente	DTD, CSS, XML SCHEMA, XPATH, XSL,XSLT-T, XSL-FO	DTD, CSS

<sup>25</sup> Esta tabla se obtuvo después de leer varios libros entre los que vale la pena destacar[8], el cual dedica un capítulo interesante a exponer algunos conceptos de la tecnología WML

## 10. SERVICIOS WEB XML

Una pregunta que muchos desarrolladores de software se hacen es: ¿Cómo hacer para lograr interoperabilidad entre aplicaciones?. Es decir, ¿cómo lograr que una aplicación web solicite alguna cosa a otra y ésta última le conteste y de esta forma establecer una comunicación a nivel de aplicaciones?. La respuesta a la pregunta es que hoy en día con los desarrollos tecnológicos es posible desarrollar éste tipo de aplicaciones, pero se requiere que la comunicación se realice por medio de unos protocolos muy especiales y basados en XML. Como conclusión podemos decir que una aplicación “A” que permita que otras aplicaciones se comuniquen con la misma por medio de protocolos basados en XML y al mismo tiempo que dicha aplicación “A” responda efectiva y concretamente a las necesidades funcionales que demandaba otra aplicación se puede decir que es un servicio web.

Surgen entonces otros interrogantes: ¿Qué lenguaje se debe usar para crear éste tipo de aplicaciones? ¿Qué protocolos se pueden usar para lograr la comunicación entre aplicaciones?. La respuesta a la primera pregunta es: “muchos”, entre ellos podemos contar php, java, phytón, etc, en tanto que la respuesta a la segunda pregunta es: SOAP[15], y XML-RPC[10] son los protocolos más estándar basados en XML para la comunicación entre servicios web.<sup>26</sup>

---

<sup>26</sup> En la siguiente dirección electrónica [http://weblog.masukomi.org/writings/xml-rpc\\_vs\\_soap.htm](http://weblog.masukomi.org/writings/xml-rpc_vs_soap.htm) se puede encontrar una discusión que ayuda al programador a decidir cuándo usar SOAP y cuando usar XMLRPC

En términos sencillos podemos decir que SOAP es más complejo que XML-RPC, pero de todas formas existen algunas diferencias entre el uno y el otro, las cuales se muestran a continuación:

<b>CARACTERÍSTICA ANALIZAR</b>	<b>A</b>	<b>PROTOCOLO XML- RPC</b>	<b>PROTOCOLO SOAP</b>
Complejidad		Más sencillo	Más complejo
Soporte de Servicios web		Los soporta algunos	Los soporta todos
Tiempo de aprendizaje		Corto	Mediano y largo

Tabla 4: Comparativa entre los protocolos XML-RPC y SOAP<sup>27</sup>

## **DEFINICIONES**

### SOA

Estándar para construir servicios discretos, potencialmente a través de múltiples máquinas que pueden ser combinadas para construir aplicaciones que reduzcan costos de integración[12]<sup>28</sup>

### Escalabilidad<sup>29</sup>

Es la capacidad de un sistema informático de adaptarse a un número de usuarios cada vez mayor, sin perder calidad en los servicios. En general, se podría definir como la capacidad del sistema informático de cambiar su tamaño o configuración para adaptarse a las circunstancias cambiantes. Por ejemplo, una empresa que establece una red de usuarios por

---

<sup>27</sup> Esta tabla fue elaborada por el autor, después de observar la documentación de los sitios oficiales que hablan de XML-RPC y SOAP

<sup>28</sup> Este sitio está en inglés pero la traducción de esta frase fue hecha por el autor Luis Felipe W.S.

<sup>29</sup> De Wikipedia, la enciclopedia libre. Dirección se encuentra en la infografía

Internet, no solamente quiere que su sistema informático tenga capacidad para acoger a los actuales clientes, sino también a los clientes que pueda tener en el futuro y, también, que pueda cambiar su configuración si es necesario [13]

## **Conclusiones**

1. Es posible en la actualidad crear herramientas y tecnologías basadas en XML y se podría decir que es el momento tecnológicamente hablando coyuntural para que los países subdesarrollados entremos en la era de ser desarrolladores de tecnologías y no consumidores de tecnologías.
2. XML es más que un lenguaje y será el estándar por excelencia en la nueva era de Internet, siendo obligatorio su estudio y se conocimiento para el desarrollo de aplicaciones computacionales e inalámbricas
3. XML trae consigo el nacimiento de otras tecnologías como los servicios web, la web semántica y otras tecnologías que podrán ser comprendidas si se comprenden las ventajas de la tecnología XML.
4. Actualmente ya existen muchas herramientas para el manejo y generación de contenidos XML, ninguna de ellas es perfecta, debido a lo nuevo que es esta tecnología, pero aún así, ya son muchos los sistemas que se basan en mencionada tecnología.

5. Las bases de datos y XML estarán cada vez más de la mano y gracias a XML será posible en un futuro no muy lejano acceder a repositorios relacionales, orientados a objetos, de red, jerárquicos y cualquier otro sistema por medio de un lenguaje estándar sin importar el tipo de almacenamiento, todo esto con la ayuda de los servicios web.

### **Referencias Bibliográficas**

[1] Aqueveque Fernandez, Juan Pablo. Sindicación Xml, Rdf, Rss y Atom. 2004-09-22

Fecha Descarga: 6 septiembre de 2005

[2] Arciniegas, Fabio. C++ XML, Guía Avanzada. Prentice may.2002

[3] González Moreno, Oscar. XML. Editorial Anaya Multimedia. ISBN: 84-415-1108-X.

Fecha Publicación: Febrero 2001.<sup>30</sup>

[4] Guía Breve de Tecnologías XML. Última modificación: 10/02/2005. Copyright © 1994-2005 W3C®

[5] Hanna, Phil. Manual de Referencia JSP. Incluye un caso práctico de una aplicación web completa. Editorial McGraw-Hill Osborne Media.2002

[6] Morrison Et Al, Michael, Xml al descubierto. Editorial Prentice may, Madrid 2000  
ISBN: 84-205-2964-8

[7] Powell, Thomas y Schneider,Fritz. Manual de Referencia JavaScript. Incluye tratamiento detallado del estándar DOM del W3C. Editorial McGraw-Hill Osborne Media.2002

---

<sup>30</sup> Este libro se puede comprar en: <http://www.agapea.com/XML-n10218i.htm>

- [8] Rodríguez Zamora, Alonso. Publicación en Internet y tecnología XML. Editorial Alfaomega Ra-Ma.2002
- [9] Vásquez, Adolfo. XML, Navegar en Internet.. Editorial: Alfaomega Rama. Diciembre 2001
- [10] William G, Oracle 8/8i Edición Especial. Page, Jr. Et Al. Editorial Prentice Hall.2001

### **Infografía**

- [11] <http://es.wikipedia.org/wiki/XML>
- [12] <http://community.java.net/java-ws-xml/>. Fecha 6 septiembre de 2005
- [13] <http://es.wikipedia.org/wiki/Escalabilidad>
- [14] [http://www.sidar.org/recur/desdi/traduc/es/xhtml/tutxhtml/tutorial\\_xhtml\\_castellano.html](http://www.sidar.org/recur/desdi/traduc/es/xhtml/tutxhtml/tutorial_xhtml_castellano.html) . Fecha: 7 de septiembre de 2005. Última actualización: Fecha: 2001/08/0119:26:28.  
Editor: Shane McCarron, Applied Testing and Technology Copyright ©2001 W3C® (MIT, INRIA, Keio)
- [15] <http://www.w3.org/TR/SOAP/>
- <http://www.abcdatos.com/webmasters/tutorial/15917.html>. Modularización de xhtml
- <http://www.debian.org/~nick/xml.html>
- <http://www.w3.org/XML/>
- <http://www.w3c.es/divulgacion/guiasbreves/TecnologiasXML>
- <http://es.selfhtml.org/introduccion/tecnologias/xml.htm>
- <http://www.maestrosdelweb.com/editorial/sindicacion/>

