Capítulo II

Tecnología XML

Características de XML
Ventajas de XML
Aplicaciones de XML
Estructura de un documento XML
Tipos de documentos
DTD (Definición de tipo de documento)
XML Schemas
Espacio de Nombres (NameSpaces)
XML y los navegadores
Lenguaje de enlaces XLL

XML viene de las siglas e**X**tensible **M**arkup **L**anguage (Lenguaje extensible de marcas), es un subconjunto de SGML (Lenguaje generalizado de marcas) y contiene sus características más importantes, XML se creó debido a la escasez de recursos que proporciona HTML (HTML es una aplicación de SGML).

XML es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes estructurados de etiquetas. XML da un formato para describir datos estructurados y garantizar que estos datos sean uniformes e independientes de aplicaciones o fabricantes.

Un documento XML también utiliza tags o etiquetas como SGML/HTML, sin embargo, a diferencia de HTML que ya posee DTD's (Declaraciones del tipo de documento) específicas, en XML es posible describir información general como productos, nombres, etc., denominados vocabularios.

HTML está diseñado para presentar información, pero no dice que lo que está mostrando es el título de un libro o el precio de un artículo. XML hace precisamente esto**: describe el contenido de lo que es cada etiqueta.**

XML es un estándar abierto y libre, del consorcio W3C propuesto en 1996, y la primera especificación apareció en 1998, desde entonces su uso ha tenido un crecimiento acelerado. Los miembros del W3C en lugar de desarrollar nuevas versiones de HTML, desarrollaron un nuevo lenguaje, denominado XML que aproveche las ventajas de HTML y que permita realizar muchas cosas más. No quiere decir que HTML desaparezca, además los navegadores no soportan todavía en toda su potencia XML y tecnologías asociadas. XML y HTML no rivalizarán, sino que se complementarán uno a otro ligándose ambas gramáticas.

Figura 2.1. Relación entre SGML, XML y HTML.

2.1. Características importantes de XML

Ahora explicaremos algunas de las características de XML:

- **Representación estructural de los datos**. XML permite escribir datos estructurados en un fichero de texto. XML es un conjunto de reglas, normas y convenciones para diseñar formatos de texto, produciendo ficheros fáciles de generar y de leer. XML está en formato texto, dándole ventajas de portabilidad, depuración, independencia de plataforma, e incluso de edición, pero su sintaxis es más estricta que la de HTML. XML proporciona una representación estructural de los datos que es ampliamente implementable y fácil de distribuir. El W3C, asegura que los datos estructurados serán uniformes e independientes de aplicaciones o compañías.
- **XML parece HTML pero no lo es**. XML usa marcas (etiquetas) y atributos, pero la diferencia es que en HTML cada marca y atributo está establecido mediante un significado incluyendo el aspecto que debe tener al verse en un navegador, en XML sólo se usan las marcas para delimitar fragmentos de datos, dejando la interpretación de éstos a la aplicación que los lee.
- **XML consta de algunas tecnologías**. El lenguaje extensible de hojas de estilo XSL que suministra mecanismos de presentación y selección de información. La especificación XLink, que permite crear hipervínculos avanzados a un documento XML. El Modelo de Objetos de Documento (DOM) es un conjunto estándar de funciones para manipular documentos XML. XML-Namespaces es una especificación que describe cómo puede asociarse una URL a cada etiqueta de un documento XML, otorgándoles un significado adicional.
- **XML no requiere licencias, es independiente de la plataforma, y tiene un amplio soporte**. La selección de XML como soporte de aplicaciones, significa entrar en una comunidad muy amplia de herramientas y desarrolladores. Todavía hay que utilizar herramientas de desarrollo, pero su formato es de uso estándar.
- **Es una arquitectura más abierta y extensible**. No se necesita versiones para que pueda funcionar en futuros navegadores. Extensible, se puede definir un conjunto ilimitado de etiquetas, un elemento de XML puede declarar que sus datos asociados sean ciertos elementos de datos. Este lenguaje permite agrupar una variedad de aplicaciones, desde páginas web hasta bases de datos.

- **Los datos son separados de la presentación y el proceso**. XML define el contenido de un documento a través de etiquetas que describen los datos. Para presentar los datos en un navegador, XML utiliza hojas de estilo tales como XSL o CSS (Hojas de estilo en cascada). XML separa los datos de la presentación y el proceso, permitiendo desplegar y procesar los datos aplicando diferentes hojas de estilo y aplicaciones.

2.2. Ventajas de XML

- **XML proporciona una descripción más exacta del contenido del documento al habilitar códigos extensibles.-** Los implementadores de XML pueden definir sus propios códigos para describir el contenido de un documento. Esta capacidad es una ventaja que XML tiene sobre HTML, ya que en HTML sólo hay un juego fijo de códigos de entre los que el desarrollador puede elegir.
- **XML permite validar el contenido del documento con una gramática estandarizada.-** Es la DTD (Definición del tipo de Documento), la que define el contenido y la estructura del documento. La gramática describe los códigos válidos, el orden de los códigos, las reglas de anidación, los atributos y otros elementos del contenido de un documento XML.
- **XML hace que sea más fácil intercambiar documentos entre usuarios y aplicaciones.-** El número de usuarios potenciales se amplía con el uso de las redes de gran tamaño, como Internet. Y XML es el mejor formato para los documentos fuente, porque permite entregar el contenido en el formato de salida más apropiado por ejemplo, HTML, PDF (Portable Document Format) y PostScript y para los formatos de las aplicaciones (EDI o intercambio electrónico de datos).
- **XML separa la estructura de los documentos del contenido y de la presentación.-** Separar el contenido de los documentos de su estructura es importante, cuando el contenido de los documentos web debe generarse dinámicamente mediante programas. Las hojas de estilo XSL controlan la presentación de los documentos XML, éstas pueden estar incorporadas en un archivo aparte. El hecho de poner los controles de presentación en un archivo aparte del contenido permite a los implementadores de XML crear múltiples vistas (formatos) de un documento XML sin cambiar el documento en sí.

- **XML mejora la respuesta del usuario, la carga de red y la carga de servidor.-** Las implementaciones de XML podrán hacer que el servidor web envíe al cliente una sola vez un documento XML y sus hojas de estilo asociadas. Cada hoja de estilo proporcionará una vista distinta de los datos del documento. El usuario selecciona la hoja de estilo que se ha de aplicar. El hecho de cambiar de una vista (hoja de estilo) a la siguiente no implicaría enviar otra petición al servidor.
- **Elimina muchas de las complejidades de SGML.-** En favor de la flexibilidad del modelo, con lo que la escritura de programas para manejar XML será más sencilla que hacer el mismo trabajo en SGML. Los archivos XML válidos son válidos también en SGML.
- **XML dará soporte a la búsqueda avanzada.-** Será más fácil encontrar las cosas en un documento XML debido a que se conoce la estructura y el significado del contenido del documento. Los documentos XML deben estar bien estructurados y los códigos deben anidarse en un determinado orden, para facilitar el análisis y manipulación de los documentos. Las gramáticas definen la estructura, el contenido de los documentos XML y esto permitirá realizar búsquedas más eficaces. Los motores de búsqueda devolverán respuestas más adecuadas y precisas, ya que la codificación del contenido en XML consigue que la estructura de información resulte más accesible.
- **XML da soporte al enlace avanzado entre documentos.-** En HTML se cuenta con enlaces unidireccionales. Por el contrario, XML da soporte a dos tipos de enlace avanzado: XLink y XPointer. Los atributos de XLink podrán proporcionar información adicional sobre el enlace y el documento destino. Los enlaces XPointer podrán hacer referencia a cualquier parte de un documento XML, incluso a las que no tengan ningún identificador.

2.3. Aplicaciones de XML

Nombraremos algunas de las importantes aplicaciones de XML, entre ellas constan ciertas aplicaciones XML ya definidas para un uso particular, y se citan algunas otras aplicaciones propuestas.

- **Base de datos**: XML es capaz de interactuar con sistemas que no suelen estar relacionados con la documentación, como las bases de datos. XML posibilita establecer una conexión entre el servidor web y la base de datos, constituyendo el

sitio web en una consulta de los datos en la base de datos, viabilizando el intercambio de información estructurada con enlaces que se encuentran en bases de datos.

- **Negocios en la web**: Internet está evolucionando hacia una red de negocios, en la cual el comercio electrónico esta tomando gran importancia. Como ha nacido una nueva generación de software de servidor web que permite añadir un servidor de nivel intermedio, que permite extraer de una sola vez de la base de datos remota la información que se ajusta al perfil del cliente en documentos XML, finalizando después de esto la conexión a servidores remotos, realizando así una entrega eficaz, rápida y confidencial de los datos al cliente.
- **EDI**: XML es una solución al problema EDI (Intercambio Electrónico de Datos) que sirve para las transacciones entre compañías y proveedores, aportando el formato de mensajes estructurado y flexible que combinados con Internet y la conectividad de bases de datos, pasa EDI de ser una serie práctica de interconexiones únicas a ser un proveedor web.
- **Traducción**: En el proceso de traducción para controlar al contenido (nuevo, reutilizado, repasado, traducido, revisado y aprobado), se ha combinado XML y los contenidos, logrando simplificar los procesos, evitar repetir tareas ya realizadas y controlar los costes. XML divide la información en componentes de información más pequeños, para que sea más fácil precisar los cambios, traducir sólo la nueva información y actualizar de forma automática la información reutilizada en el documento.
- **Buscadores inteligentes**: Debido a que la información en los documentos XML está etiquetada por su significado de forma precisa y con DTDs estandarizados para distintas aplicaciones se podrán programar buscadores web que recuperen información sobre un producto de cualquier website en el mundo sabiendo que todos tendrán el mismo formato de datos (DTD), aunque no tengan necesariamente la misma representación gráfica. XML permitirá que los buscadores sean más eficaces, ya que el usar XML implica que las propias páginas web definirán su contenido, de esta forma, los buscadores que usen XML podrán especificar que la información solicitada se refiere a su autor, por ejemplo, de manera que sólo aparecerán las páginas relacionadas con él.

- **RDF (Resource Description Framework)**: Esquema de descripción de recursos. Una de las aplicaciones más importantes que permitirá describir los datos de cada documento y definir las relaciones que hay entre los datos XML. Tratará de los metadatos que podrían considerarse como los "META del XML". RDF posee la capacidad de describir los contenidos y sus relaciones en una biblioteca digital o sede Web. Permitirá el acceso a una parte concreta del documento y se facilitará el intercambio de los datos.
- **OSD (Open Software Description Format)**: Formato abierto de descripción de software. Desarrollo de software en múltiples plataformas. Las etiquetas XML con las que está descrito definen los componentes, la versión que es la plataforma en la que ha sido creado, la relación con otros componentes, etc. Esto hará que se simplifique el proceso de instalación para el usuario y permitir también un fácil uso de las actualizaciones.
- **CML (Chemical Markup Language)**: Lenguaje de marcas para química. Describe entre otras fórmulas, las estructuras moleculares y cristalinas, los análisis de espectros y otros objetos de interés para químicos.
- **MathML (Mathematical Markup Language)**: Lenguaje de marcas para matemáticas. Apto para codificar signos matemáticos, símbolos científicos, etc. El MathML es un lenguaje de bajo nivel que permite comunicar máquina a máquina datos estructurados como información de bases de datos. Este lenguaje utiliza dos series de códigos progresivos: el primero presenta los signos matemáticos en series crecientes, y el segundo transmite el significado semántico de las expresiones matemáticas, lo que posibilita la codificación de símbolos y signos.
- **OFX (Open Financial Exchange)**: Intercambio financiero abierto. A través de software de gestión financiera se podrá conectar con el banco para gestionar las cuentas y extraer información de la cuenta bancaria. Esto se hará a través de unos protocolos seguros.
- **TEI (Text Encoding Initiative)**: Iniciativa que partió de diversas asociaciones profesionales en los campos de humanidades. Trata de establecer etiquetas que propicien la descripción de textos científicos y literarios.
- **WML (Wireless Markup Language):** En telefonía móvil. WML es utilizado para escribir las páginas que se visualizan en los móviles.

- **HRMML (Human Resource Management Markup Language)**: Intercambio de perfiles de puestos de trabajo y currículo.
- **BSML (Bioinformatic Sequence Markup Language)**: Codificación y representación de información de secuencias de ADN, ARN y proteínas.

2.4. Estructura de un documento XML

Cualquier aplicación que trabaje con XML necesita un módulo software llamado procesador XML; su función es leer documentos y proporcionar acceso a su contenido y estructura, para poder realizar esto la aplicación debe informar al procesador XML cómo se encuentra almacenada la información a través de una DTD (declaración del tipo del documento), que proporciona la gramática del documento XML, ésta define los nombres de los elementos, dónde pueden aparecer y cómo se interrelacionan entre ellos. XML, permite a los procesadores analizar automáticamente un documento e identificar dónde viene cada elemento y cómo se relacionan entre ellos, para que otras aplicaciones puedan utilizarlos.

Un documento XML tiene dos estructuras, una lógica y otra física, las estructuras lógica y física deben encajar de manera adecuada. La estructura física y lógica de un documento XML es análoga a una base de datos relacional, que emplea tablas y campos para la estructura lógica y archivos binarios para la estructura física.

2.4.1. Estructura lógica

Lógicamente, el documento está compuesto de declaraciones, elementos, comentarios, referencias a caracteres e instrucciones de procesamiento, todos los cuales están indicados por una marca explícita. Cada documento XML contiene uno o más elementos, cuyos límites están delimitados por etiquetas de comienzo y de final o, elementos vacíos. Cada elemento tiene un tipo denominado identificador genérico y puede tener un conjunto de especificaciones de atributos. Cada especificación de atributo tiene un nombre y un valor.

2.4.2. Estructura física

Físicamente, el documento está compuesto por una o más unidades de almacenamiento virtual llamadas entidades, una entidad puede hacer referencia a otra entidad. Cada documento XML comienza con una entidad documento (raíz), que es el punto de inicio para el procesador XML y contiene el documento completo. Todas las entidades tienen

contenido y todas ellas excepto la entidad documento, están identificadas por un nombre. Las entidades pueden ser analizadas o sin analizar (procesadas o sin procesar).

El contenido de una entidad analizada se conoce también como texto de reemplazo, y es parte integrante del documento. Las entidades no analizadas son recursos (*como los enlaces*) cuyo contenido puede o no ser texto, y si es texto que no sea XML.

2.5. Tipos de documentos XML

Los documentos XML se dividen en dos grupos, documentos bien formados y documentos válidos, este es un aspecto importante de este lenguaje, así que se debe entender bien la diferencia.

2.5.1. Documentos bien formados

Los documentos bien formados son los que cumplen las especificaciones del lenguaje respecto a las reglas sintácticas sin estar sujetos a los elementos fijados en una DTD. Los documentos XML deben tener una estructura jerárquica estricta y los documentos bien formados deben cumplirla.

Un documento XML está bien formado si sigue las siguientes reglas básicas.

- Hay exactamente un elemento, llamado raíz o elemento documento, de forma que ninguna parte del mismo aparecerá en el contenido de ningún otro elemento y dentro de este deben estar todos los demás elementos.
- Todo elemento deberá estar delimitado por una etiqueta de inicio y su correspondiente etiqueta de cierre, o una sola etiqueta vacía.
- Todos los elementos deben encontrarse anidados correctamente, los documentos XML deben tener una estructura jerárquica con lo que respecta a las etiquetas que delimitan sus elementos. Esto significa que los elementos deben estar correctamente estructurados y que los elementos no se pueden solapar entre ellos. Además los elementos con contenido deben estar correctamente cerrados.
- Todas las entidades (externa o interna) que se usen deberán declararse en la DTD.
- Todos los valores de los atributos deben ir entrecomillados.

2.5.2. Documentos XML válidos

Un documento XML válido además de estar bien formado, debe seguir una estructura y una semántica determinada por una DTD. Evidentemente, el tener una DTD no basta para que un documento sea válido, el documento también debe adherirse a la DTD, o sea sostener el modelo de contenido que se declare en la DTD y usar los tipos de atributos apropiados.

La información que se encuentra correctamente estructurada debe tener una utilidad, para lo cual se vio la necesidad de validar el documento. Es decir, puede ser que un documento esté perfectamente estructurado y, sin embargo no cuente con la información clave. La DTD permite al procesador saber las características del documento y, así, poder verificarlo.

Un documento XML se dice que es válido si sigue las siguientes reglas básicas.

- El documento debe estar bien formado o bien construido.
- El nombre del elemento raíz del documento, debe coincidir con el nombre de la declaración del tipo de documento.
- El documento debe tener una DTD que declare todos los elementos, atributos y entidades que se usen en el documento. Puede ser una DTD interna o externa.
- El documento debe adherirse a la gramática que establezca la DTD.

En general, se dice que todos los documentos válidos están bien construidos, pero no todos los documentos bien construidos son documentos válidos.

La creación de un documento XML válido puede parecer a primera vista, un proceso innecesario, sin embargo, si lo que se quiere es que el documento se adapte a una estructura específica, o a un conjunto de estándares, la inclusión de una DTD, que defina la estructura, permitirá a los procesadores de XML verificar si el documento se adapta a la misma. Es decir, la DTD proporciona al procesador la estructura, de forma que, al comprobar la validez del documento, pueda forzar dicha estructura deseada y garantizar que el documento cumpla con los estándares requeridos. La creación de documentos válidos es especialmente útil para garantizar la uniformidad entre un grupo de documentos similares.

2.6. Definición del tipo de documento (DTD)

Una DTD (Document Type Definition) es una especie de definición de la gramática del documento, es un conjunto de reglas sintácticas para definir los elementos que componen un documento XML y la relación entre ellos. Crear una definición del tipo de documento (DTD) es como crear un propio lenguaje de marcado, para una aplicación específica. La DTD permite realizar la validación de documentos, es decir establecer si el documento es válido o si sólo está bien construido. Una declaración del tipo de documento indica el nombre del tipo de elemento y el contenido permitido del mismo, especificando a menudo el orden en que pueden existir los elementos hijos. La DTD, describe toda la estructura lógica del documento, esto significa que las declaraciones de tipo de elemento indican los tipos de elementos que contiene el documento, el orden de dichos elementos y la especificación del contenido de los mismos.

Cuando se procesa cualquier información formateada mediante XML, primero se comprueba si está bien formada, y luego, ver si incluye una referencia a una DTD y comprobar que sigue sus reglas gramaticales. Las funciones de una DTD son:

- Especificar el elemento raíz del documento.
- Definir elementos, atributos y entidades específicas del documento (DTD internas).
- Identificar una DTD externa en el documento.

Las declaraciones de marcado definen la estructura lógica y física del tipo de documento.

La inclusión de una DTD y la comprobación de validez son especialmente importantes si los documentos van a ser procesados por algún software específico que necesita una estructura de documento determinada. La mayoría de las aplicaciones XML, como MathML, consisten en una DTD estándar que todos los usuarios de la aplicación incluirán en sus documentos XML, de forma que la comprobación de la validez de los documentos pueda garantizar que se adapten a la estructura de la aplicación y que serán reconocidos por cualquier software diseñado para esa aplicación.

2.6.1. DTDs Internas y DTDs Externas

Una DTD de un documento se puede describir utilizando una DTD interna, DTD externa o ambas. Para usar las DTD internas deben declararse en el documento XML y las DTD externas deben hacerse referencia en el mismo.

La DTD externa es la que más se utiliza en la validación de documentos. En una DTD externa se debe colocar declaraciones de marcado de documentos si:

- Está creando un documento o documentos que deben ser válidos.
- Está creando múltiples documentos de la misma clase.
- Se desea usar una DTD existente.
- Se quiere hacer que los documentos fueran lo más conciso posible.

Al colocar declaraciones de marcado en una DTD externa se establece una separación lógica entre las declaraciones de marcado de un documento y el verdadero contenido de éste. La DTD externa es valiosa al crear múltiples documentos válidos de la misma clase.

La DTD interna debe tener copias de declaraciones de marcado en cada documento. Ventajas de colocar declaraciones de marcado de documento en una DTD interna:

- No tiene importancia para que los documentos sean válidos.
- Si está creando un solo documento.
- Si desea minimizar el coste asociado con los documentos.

Si los documentos no van a ser válidos no es necesario usar DTD externa ni DTD interna, pero los documentos tienen que estar bien construidos. Una DTD interna es más eficaz que una DTD externa ya que la interna forma parte del archivo del documento físico.

2.6.2. Tipos de declaraciones que se realizan en una DTD [LIB002]

Una DTD puede contener los siguientes tipos de declaraciones de marcas:

- Declaraciones de los tipos de elementos: Definen los tipos de elementos que puede contener el documento, así como el contenido y ordenación de los elementos.
- Declaraciones de lista de atributos: Cada declaración de lista de atributos define los nombres de los atributos que podrán utilizarse con un determinado elemento, así como los tipos de datos y valores predeterminados de estos atributos. Los atributos son las características de formato con que se estructura un documento.

Los elementos pueden tener atributos, que son una manera de incorporar características o propiedades a los elementos de un documento.

- Declaraciones de entidades: Se puede utilizar entidades para almacenar bloques de texto frecuentemente utilizados o para incorporar datos que no sean de XML.
- Declaraciones de notación: Una notación describe un formato de datos, o identifica el programa empleado para procesar un formato determinado.

2.7. Esquemas XML (XML Schemas)

Un problema que presentan los DTDs es que no siguen una sintaxis XML, sino una propia. Un grupo de desarrolladores han propuesto una alternativa a los DTDs, llamada esquemas (schemas). Un esquema es un conjunto de reglas que sirven para forzar la estructura y la articulación del conjunto de documentos XML.

Un esquema XML es algo similar a un DTD, es decir, que define qué elementos puede contener un documento XML, cómo están organizados, y qué atributos y de qué tipo pueden tener sus elementos. Un esquema XML además permite asociar los tipos de datos con elementos facilitando que los procesadores XML lleven a cabo la validación del contenido; en una DTD el contenido del elemento está muy limitado a las cadenas mientras que en un esquema XML se puede establecer el tipo de datos de los elementos a tipos muy específicos, como enteros y fechas.

A continuación se indica las ventajas de los Esquemas XML en comparación con las DTD:

- XML Schema se basa en XML y no en una sintaxis especializada.
- XML Schema puede ser analizado sintácticamente y manipulado como cualquier otro documento XML.
- XML Schema soporta una serie de tipos de datos (int, float, bolean, date, etc)
- XML Schema presenta un modelo de datos abierto, lo que permite ampliar los vocabularios y establecer relaciones de herencia entre los elementos sin invalidar los documentos.
- XML Schema soporta la integración de los espacios de nombres, lo cual permite asociar nodos individuales de un documento con las declaraciones de tipos de un esquema.

- XML Schema soporta grupos de atributos, lo que permite lógicamente combinar atributos.

2.7.1. Diferencias entre DTDs y Esquemas XML

Veamos unas diferencias entre los esquemas y los DTDs:

- Las DTDs están basadas en una sintaxis específica, mientras que los esquemas XML están basados en XML.
- Las DTD tratan los datos como cadenas o cadenas enumeradas y los esquemas XML soportan tipos de datos (int, float, boolean, date, etc)
- Los DTD emplean un modelo de contenido cerrado que no permite la ampliación. Los esquemas XML soportan el modelo de contenido abierto y la integración del espacio de nombres.
- Las DTD son compactas mientras que los esquemas no son compactos, pero son fáciles de entender.
- Las DTD soportan agrupación a través de entidades de parámetros, esto es una limitación ya que el procesador XML no sabe nada acerca del agrupamiento. Los esquemas XML soportan los grupos de atributos que permiten combinar atributos lógicamente.
- Las DTD no pueden dejar de existir solo por el hecho de que los esquemas XML representan una solución más potente al modelado de datos. Aunque los esquemas XML tienen mucho más que ofrecer en términos de poder y flexibilidad, las DTD han existido mucho más tiempo, tienen más aceptación y describen muy bien la estructura de los documentos XML.

Antes de decidir entre las DTD y los esquemas XML, se deben evaluar las necesidades y las herramientas que se van a utilizar. Si necesita una solución a corto plazo mejor decidirse por las DTD, si se cuenta con tiempo y deseo de utilizar las características avanzadas se recomienda los esquemas XML.

2.8. Espacios de Nombres de XML (XML Namespaces) [WWW003]

Los espacios de nombres de XML ofrecen la posibilidad de crear nombres en un documento XML que estén identificados por medio de un Identificador uniforme de recursos (URI – Uniform Resource Identifier). Utilizando espacios de nombres en un documento XML, se pueden identificar de forma única sus elementos y atributos.

XML Namespaces proporciona una forma de asignar nombres únicos a elementos XML. La técnica que se emplea para garantizar la singularidad es inteligente e implica el uso de URI (Identificadores Uniformes de Recursos). Los URI suelen hacer referencia a los recursos físicos en Internet, están garantizados que sean únicos. Así, un espacio de nombres es el nombre de un URI.

El concepto de "Namespaces" surge de la necesidad para combinar diferentes *vocabularios* en XML, es una forma de *agrupar* distintos elementos para poder ser utilizados en un solo documento, esto significa que por ejemplo, se está trabajando con s proveedores | clientes y estos facilitan su información en XML, debido a que están en la misma industria es muy probable que utilicen elementos en común como <monitor> , <software> ...etc. si desarrolla un programa que haga uso de estos elementos comunes es mediante *Namespaces* que se elimina la ambigüedad que pueda surgir en el programa.

2.8.1. Declarar espacios de nombres

Los espacios de nombres se declaran en los documentos XML y no en las DTD, por medio de una declaración de espacios de nombres, que adopta esta forma:

xmlns: *prefix* = *"NameSpace"*

La palabra clave xmlns es lo que alerta a un procesador XML de que se está declarando un espacio de nombres. La parte prefix permite establecer un prefijo que servirá como referencia del espacio de nombres a lo largo del alcance del elemento en el que se declara el espacio de nombres, prefix es opcional.

La porción NameSpace es donde se identifica el propio espacio de nombres, esta porción identifica un URI que garantiza la singularidad en los elementos y atributos que se utilizan en el ámbito de la declaración de espacios de nombres.

Hay dos soluciones distintas para declarar espacios de nombres:

. Declaración predeterminada. El espacio de nombres se declara sin un prefijo, y se hace referencia a todos los nombres de atributos y elementos que haya en su ámbito con nombres no cualificados (*no incluyen prefijo y están asociados con un espacio de nombres predeterminado o no están asociados con ninguno*).

. Declaración explícita. El espacio de nombres se declara con un prefijo, y todos los nombres de elementos y atributos que estén asociados con el espacio de nombres deberán usar el prefijo como parte de sus nombres cualificados (*incluyen el prefijo y la porción local del nombre*).

2.8.2. Utilizar los espacios de nombres en los esquemas

Los esquemas están asociados con los espacios de nombres de varias formas. Una es a través de un URN (*es el modo en que los espacios de nombres de tipos de datos XMLSchema están asociados con sus respectivos esquemas*). También puede hacer referencia a un esquema especificando el nombre del archivo de esquema como un URI de una declaración de esquemas.

Los espacios de nombres resuelven un problema muy importante inherente a la misma naturaleza de XML, que es el conflicto de los nombres en los vocabularios personalizados. Sin los espacios de nombres, sería imposible garantizar la singularidad entre los vocabularios XML desarrollados y utilizados en la Web. Podríamos crear un vocabulario con un elemento llamado idproducto que sirviera como identificador de un producto que se tiene a la venta en una aplicación basada en la Web, si integráramos otro vocabulario XML en la aplicación que usará también un elemento llamado idproducto, la aplicación no podría distinguir entre los dos. Los espacios de nombres resuelven este problema proporcionando un mecanismo que establece la singularidad entre los elementos y los atributos de los vocabularios XML.

2.9. XML y los navegadores

Es importante saber que nos depara el futuro y qué podremos obtener de XML como lenguaje de publicación en Internet. Para ello hace falta conocer de que forma soportan los navegadores el nuevo lenguaje.

2.9.1. Tipos de Navegadores

El navegador es el programa que solicita y muestra en la pantalla del ordenador personal los documentos que residen en los servidores remotos de toda la World Wide Web. Tenemos dos tipos de navegadores:

2.9.1.1. Navegadores de acceso alternativo.

Estos navegadores se caracterizan por presentar la información de páginas web de manera distinta a la convencional, lo hace en formato sólo texto o mediante audio y síntesis de voz. Los navegadores de este tipo son: Lynx, Net-Tarner, PwWeb Speak, Web-On-call Óbice browser, Web-Tv, entre otros.

2.9.1.2. Navegadores de acceso estándar

Los más populares son Microsoft Explorer y Netscape Navigator, aunque existen muchos más como Opera, Mozilla, AOL etc. Suelen ser de uso gratuito y pueden descargarse desde los sitios web de las casas fabricantes o desde colecciones de freeware y shareware.

El problema que aparece es la necesidad de estar al día con los últimos estándares y posibilidades de la web que van apareciendo permanentemente, en la actualidad los navegadores deben incluir de algún modo un parser (analizador) para XML y un motor que acepte las hojas de estilo (CSS o XSL), y aquí empiezan a verse diferencias.

2.9.2. Tabla de Comparación de Navegadores

Es importante considerar que “parte del futuro del intercambio de datos está en XML, con lo que el no ofrecer soporte para él es arriesgado”.[WWW004]

El siguiente gráfico muestra qué soporta cada uno de los navegadores más importantes del mercado:

Claves	
soportado	X
algo soportado	s
no soportado	

Windows													
Navegadores	java	frames	tables	plug -ins	font size	font color	java script	style sheets	gif89	dhtml	I- Frames	Table color	XML
Explorer 6.0	s	X	X	X	X	X	X	X	X	X	X	X	X
Explorer 5.5	X	X	X	X	X	X	X	X	X	X	X	X	X
Explorer 5.0	X	X	X	X	X	X	X	X	X	X	X	X	s
Explorer 4.0	X	X	X	X	X	X	X	X	X	X	X	X	
Netscape 7.0	X	X	X	X	X	X	X	X	X	X	X	X	X
Netscape 6.1	X	X	X	X	X	X	X	X	X	X	X	X	X
Netscape 6.0	X	X	X	X	X	X	X	X	X	X	X	X	X
Navigator 4.7	X	X	X	X	X	X	X	X	X	X		X	
Navigator 4.5	X	X	X	X	X	X	X	X	X	X		X	
Mosaic 3.0		X	X		X								
Mozilla 1.31	X	X	X	X	X	X	X	X	X	X	X	X	X
Mozilla 1.0	X	X	X	X	X	X	X	X	X	X	X	X	X
Opera 7.x	X	X	X	X	X	X	X	X	X	X	X	X	X
Opera 6.0	X	X	X	X	X	X	X	X	X	X	X	X	X
Opera 5.11	X	X	X	X	X	X	X	X	X	X	X	X	X
Opera 4.02	X	X	X	s	X	X	X	X	X		X	X	X
Opera 3.60		X	X	s	X	X	X	X	X			X	
Lynx		X	X										

Tabla 2.1. Navegadores para Windows [WWW004]

Linux													
Navegadores	java	frames	tables	plug -ins	font size	font color	java script	style sheets	gif89	dhtml	I- Frames	Table color	XML
Netscape 7.0	X	X	X	X	X	X	X	X	X	X	X	X	X
Netscape 6.1	X	X	X	X	X	X	X	X	X	X	X	X	X
Netscape 6.2	X	X	X	X	X	X	X	X	X	X	X	X	X
Netscape 6.0	X	X	X	X	X	X	X	X	X	X	X	X	X
Navigator 4.7	X	X	X	X	X	X	X	X	X	s		X	
Mozilla 1.3.1	X	X	X	X	X	X	X	X	X	X	X	X	X
Mozilla 1.0	X	X	X	X	X	X	X	X	X	X	X	X	X
Galeon	X	X	X	X	X	X	X	X	X	X	X	X	X
Konqueror	X	X	X	X	X	X	X	s	X	s		X	X
Amaya 5		X	X		X	X		X	X	s		X	X
Opera 7	X	X	X	X	X	X	X	X	X	X	X	X	X
Opera 6.x	X	X	X	X	X	X	X	X	X	X	X	X	X
Opera 5	X	X	X	X	X	X	X	X	X	X	X	X	X
Lynx		X	X										

Tabla 2.2. Navegadores para Linux [WWW004]

2.10. XLL (XML Linking Language) Lenguaje de enlaces extensible

Las habilidades de enlazado de los sistemas XML son mayores que las de HTML. Los vínculos HTML, siempre están basados en dos recursos: el origen y el destino, vinculados en una sola dirección. Podría pensar que el botón Atrás de un navegador web califica los vínculos HTML como bidireccionales pero está es sólo una característica del navegador que mantiene una lista de páginas para que el usuario pueda volver atrás.

El elemento que se utiliza para establecer vínculos en HTML es el elemento <a> que identifica el recurso de destino de un vínculo HTML por medio del atributo href que contiene un URL. Los vínculos HTML pueden vincularse con documentos enteros o con un fragmento de documento.

La solución que da XML a la vinculación va mucho mas allá de las limitaciones de HTML, seria bueno si los vínculos fueran bidireccionales, estos tratarían los recursos como orígenes y destinos. Seria muy útil tener vínculos que hicieran referencia a múltiples recursos de destino. Esto evitaría que los desarrolladores web tuvieran que duplicar el contenido con el único propósito de proporcionar orígenes de vínculos.

XLL está orientado a cambiar las conexiones fundamentales que hay entre los recursos de la Web. Para dar una idea de lo innovadora que es la tecnología consideremos que XLL admite la creación de vínculos que residen fuera de los documentos que vinculan. Esto puede resultar muy útil cuando no se puedan modificar los otros documentos. Esta capacidad de crear este tipo de vínculos probablemente creará un nuevo tipo de negocio web: el almacén de vínculos, que sería una base de datos de vínculos que describirían conexiones útiles entre los recursos de la Web.

XLL consta de tres tecnologías y son:

- XPath. Un lenguaje que se emplea para dirigirse a partes de un documento XML.
- XPointer. Un lenguaje que se emplea para dirigirse a las estructuras internas de los documentos XML.
- XLink. Un lenguaje XML que específica las construcciones de la vinculación avanzada de los documentos XML.

2.10.1. XPath

XPath es un lenguaje no XML (no se describe por medio de elementos y atributos) que se usa para dirigirse a partes de un documento XML. XPath proporciona una vía abstracta para dirigirse a partes de documentos XML y forma la base del direccionamiento de documentos en Xpointer y XSLT (Transformaciones XSL). La sintaxis que emplea XPath está diseñada para ser utilizada en URI y valores de atributos XML. Esta es la razón por la que XPath no se implementa como vocabulario XML.

XPath trata el documento como un árbol de nodos, los nodos de un documento XML pueden dividirse en nodos de elementos, nodos de atributos y nodos de texto. A continuación se exponen los tipos de nodos que pueden aparecer en un árbol XPath:

• Nodos raíz.

• Nodos de elementos.

• Nodos de texto.

• Nodos de atributos.

• Nodos de espacios de nombres.

• Nodos de instrucciones de procesamiento

• Nodos de comentarios.

El recorrido que se lleva a cabo en un documento XML para llegar a un nodo determinado se lo hace por medio de expresiones. Cuando se evalúan las expresiones XPath, acaban siendo un objeto de datos de uno de los tipos siguientes:

• node-set. Un conjunto de nodos.

• boolean. Un valor true-false

• number. Un número de coma flotante

• string. Una cadena de texto

El modo en que se evalúa una expresión XPath depende completamente del contexto de la expresión que viene determinada por Xpointer o XSLT, que a su vez determina como se evalúa la expresión.[LIB002]

2.10.2. XPointer

XPointer es el componente de la tecnología XLL que se emplea para dirigirse a partes (estructuras internas) de documentos XML, soporta una sintaxis concisa que le permite recorrer la estructura de árbol de nodos de un documento XML para hacer referencia a elementos, atributos y cadenas de texto. XPointer no es necesario en lo relativo a vínculos que hacen referencia a documentos XML completos.

A continuación se ve una lista de parámetros de diseño, tomada de la especificación Xpointer, que ilustra las directrices utilizadas por el W3C para crear XPointer:

1. Los XPointer deben estar dirigidos a documentos XML.
2. Los XPointer deben poder ser usados de forma clara en Internet.
3. Los XPointer deben poder ser usados de forma clara en los URI.
4. El diseño XPointer debe ser preparado rápidamente.
5. El diseño XPointer debe ser formal y conciso
6. La sintaxis XPointer debe ser compacta y legible.
7. Los XPointer deben ser optimizados para su utilización.
8. Debe ser factible implementar XPointer.

En HTML se dirige a un punto general de un documento por medio de la etiqueta de anclaje y el elemento name. XPointer es más específico en su direccionamiento, permitiéndole dirigirse a un elemento que tenga un determinado valor dentro de una lista de elementos de un determinado tipo.

XPointer se implementa como lenguaje conciso que se usa para construir expresiones, estas expresiones se adjuntan al final de los URI, con un símbolo (#). Una expresión XPointer consta de rutas de ubicación que tienen una sintaxis muy parecida a las llamadas de función de los lenguajes de programación tradicionales. [LIB003]

2.10.3. XLink

XLink es un vocabulario XML que añade posibilidades de vinculación avanzadas al metalenguaje XML. XLink está diseñado para soportar vínculos sencillos, de estilo HTML unidireccional, así como una serie de vínculos extendidos que ofrecen formas de vincular documentos. Por medio de Xlink, se puede especificar la forma de atravesar un vínculo.

Por ejemplo, se puede crear un vínculo que se active automáticamente tan pronto como se carga un documento. Esto podría parecer extraño hasta que se considera que el documento vinculado puede abrirse en el contexto de un documento actual, formando así un documento compuesto. XLink se implementa como vocabulario XML.

A continuación vemos una lista de estos parámetros de diseño que creó el W3C como base para la creación de XLink.:

1. XLink debe poder ser utilizado en Internet de una forma clara.
2. XLink debe poder ser utilizado por una serie de dominios de utilización de vínculos y por clases de software de aplicación de vinculación.
3. XLink debe soportar las construcciones de vinculación HTML.
4. El lenguaje de expresión XLink debe ser XML.
5. El diseño XLink debe ser formal, conciso e ilustrativo.
6. Se debe poder leer y escribir en los XLinks.
7. Los XLinks pueden residir dentro o fuera de los documentos donde residan los recursos implicados.
8. XLink debe representar la estructura abstracta y la importancia de los vínculos.
9. Debe ser posible implementar Xlink, independientemente de los medios empleados para la presentación de los vínculos.
10. XLink debe ser informado de la existencia de sistemas y estándares de hipermedios.

La construcción que se emplea para especificar vínculos en XLink es el elemento de vinculación, que afirma la existencia de un vínculo y que describe sus características. No existen elementos de vinculación predefinidos en el lenguaje XLink. Para evitar el problema de HTML de vincularse a un elemento fijo, XLink define atributos de vinculación estándar que se pueden asociar con cualquier elemento. Un elemento de vinculación utiliza una construcción llamada localizador para conectar los recursos que están implicados en un vínculo. Tanto en HTML como en XML, el atributo ref sirve como localizador de los vínculos, en XML los vínculos describen completamente los recursos implicados, incluso cuando un recurso de destino sea un fragmento de documento.