1. XML

XML (Extensible Markup Language, lenguaje de marcas extensible) es la última palabra de moda en Internet. También es una tecnología que madura rápidamente con múltiples aplicaciones en el mundo real, particularmente para la gestión, visualización y organización de los datos. En conjunto con su lenguaje de visualización XSL y el modelo de objeto documento DOM, es una tecnología esencial para cualquier usuario de lenguajes de marcas en la red o en la intranet.

Objetivos.

El objetivo de este tema es iniciar al alumno en los conceptos básicos de XML. Conocer los elementos que componen un documentos XML. Conocer el organismo W3C, consorcio que emite las diferentes recomendaciones relacionadas con XML.

Contenidos.

1.1. Breve historia de los lenguajes de marcas
1.2. Definición
1.3. Analizadores XML (Parsers)
1.4. HTM y XML
1.5. Jerarquías en XML
1.6. Consorcio W3C
1.7. Componentes XML
1.8. Resumen
1.9. Bibliografía del curso

1.1. Breve historia de los lenguajes de marcas

Se puede observar que el uso de los archivos binarios reporta muchas ventajas (fácil de gestionar por el ordenador, más compacto, etc.), pero los archivos de texto también tienen sus ventajas (la principal es que es un formato de intercambio universal). Quizá lo ideal sería tener un formato que combine la universalidad de los archivos de texto con la eficiencia de almacenamiento y las capacidades de formateo de los archivos binarios.

Esta idea de llegar a un formato de datos universal no es nueva. De hecho, desde que existen los ordenadores, los programadores han intentado crear modos de intercambio de información entre los diferentes programas informáticos. **SGML** (Standard Generalized Markup Language) fue uno de los primeros intentos en el que se pretendía combinar un formato de dato universalmente intercambiable con la posibilidad de almacenar los datos con información acerca de su presentación y formato. SGML fue creado para que se transformara en un estándar para el marcaje de los datos y tuvo aceptación en los grandes sistemas de gestión de documentos. Pero cuando se debe tratar con gran cantidad de datos complejos SGML se vuelve un lenguaje muy complicado.

La aplicación más conocida de **SGML** es **HTML** (HiperText Markup Language). La idea era que cualquier documento HTML (o página web) debía poder ser representada en cualquier aplicación que fuera capaz de comprender HTML (específicamente los navegadores Web). Además HTML se basa en texto, cualquiera puede crear una página HTML usando un editor de texto simple o cualquiera de los editores de páginas web existentes en el mercado. Incluso muchos procesadores de texto nos permite exportar nuestros documentos a formato HTML.

1.2. Definición

Desgraciadamente SGML, es un lenguaje tan complicado que no resulta adecuado para el intercambio de datos en la web. Y, aunque HTML ha tenido un éxito enorme, tiene un ámbito muy limitado: su objetivo es lograr la visualización de documentos en el navegador. Esto significa que podemos crear un documento HTML para visualizar la información de una persona, pero esto es todo lo que podemos hacer con ese documento. No podríamos relacionar ese documento con la información relacionada con el nombre de la persona porque HTML no tiene ninguna posibilidad de describir esta clase de información especializada.

El lenguaje de marcas XML se creó para resolver este problema. XML es un *subconjunto* de SGML, con los mismos objetivos, pero sin el grado de complejidad del lenguaje original. XML, fue diseñado para que sea totalmente compatible con SGML, lo que implica que cualquier documento que sigue las reglas de sintaxis de XML es también por definición un documento SGML. Ojo porque el caso contrario puede no ser cierto. No todos los documentos SGML son documentos XML.

Es importante tener en cuenta que XML no es un lenguaje, sino un estándar para crear lenguajes que cumplen los criterios XML. En otras palabras, XML describe una sintaxis que se utiliza para crear nuestros propios lenguajes. Por ejemplo, supongamos que tenemos datos acerca de un nombre y queremos compartir esa información con otros. Pero también queremos utilizar esa información en un programa informático. En vez de crear un archivo de texto con el siguiente contenido:

```
<HTML>
<HEAD><TITLE>Name</TITLE></HEAD>
<BODY>
<P>John Doe</P>
</BODY>
</HTML>
```

Podemos crear un archivo XML como este:

```
<name>
        <first>John</first>
        <last>Doe</last>
</name>
```

Incluso en estos ejemplos tan simples podemos comprender por qué a los lenguajes de marcas SGML y XML se les denominan autodescriptivos. Analizando los datos se puede ver que la información está relacionada con un nombre. Un dato denominado *first* (nombre) y otro *last* (apellido). Aunque podríamos haberles asignado nombres arbitrarios, si se utiliza XML, la recomendación es siempre utilizar nombres con significado. También se puede observar que la versión XML es bastante más larga que la versión de texto simple. El uso de XML para marcar la información añadirá bytes a los datos básicos y en algunos casos aumentarán de tamaño de manera considerable. Pero cabe destacar que el tamaño de los archivos no es la preocupación principal de XML, sino que busca dar una estructura a los datos para simplificar el software que accede a los

mismos. Al fin y al cabo si el tamaño del archivo es importante a la hora de transmitir la información siempre tenemos la alternativa de comprimir los archivos.

1.3. Analizadores XML (Parsers)

Pero qué nos ofrece XML. ¿Por qué tenemos que complicarnos el trabajo creando archivos de documentos XML?¿No sería más sencillo indicar algunas reglas para un fichero con nombres y apellidos, señalando por ejemplo donde comienzan los nombres y que el apellido va tras un espacio en blanco? Además los tamaños de los archivos serían más pequeños.

Pero ahora supongamos que nos encontramos ante un nombre compuesto y ahora queremos añadir un segundo nombre: John Fitgerald Doe. Tendríamos que cambiar las reglas para indicar en el programa que en el fichero tras el primer espacio en blanco va el segundo nombre y no el apellido en los casos que existan dos espacios. En los demás aplicará la primera regla. Pero ahora llega una persona con el siguiente nombre: John Fitgerald Johansen Doe. Para una persona es simple identificar el nombre y apellido pero para el programa será un cambio lógico importante. Y aún podemos encontrarnos nuevos casos como John Fitgerald Johansen Doe Junior. Por desgracia en la mayoría de las aplicaciones el desarrollador toma una decisión, por ejemplo que las personas solo pueden tener a lo sumo dos nombres y dos apellidos. Cualquier persona fuera de este estándar o no podrá guardarse su información o esta se guardará incompleta.

El ejemplo puesto es sencillo y fácil de resolver pero nos sirve para señalar uno de los principales objetivos de XML. Cada vez que los desarrolladores cambian la estructura de los datos deben cambiar también la tecnología para extraerlos. Pero con XML existe un modo estandarizado para que obtengamos la información que necesitamos, no importa como esté estructurada. El ejemplo visto es simple y es difícil apreciar la necesidad de utilizar la tecnología XML. Pero cuando más grandes son las aplicaciones y más compleja la información que se maneja más fácilmente se aprecian las ventajas y virtudes de XML.

Si seguimos las reglas especificadas por XML, podemos estar seguros que nos resultará muy fácil acceder a nuestra información. Esto es así porque existen programas denominados analizadores (*parsers)* que son capaces de leer la sintaxis XML y obtener la información que necesitamos. Podemos utilizar estos analizadores dentro de nuestros propios programas. De esta forma nosotros no tenemos que acceder directamente a los ficheros XML lo que nos libera de parte del trabajo.

Al ser tan flexible XML está orientado para ser la base de la definición de lenguajes de intercambio de datos, especialmente para la comunicación en Internet. El lenguaje simplifica el trabajo con los datos dentro de una aplicación, como por ejemplo, cuando en una aplicación es necesario acceder a la información almacenada bajo la etiqueta <name>, pero además podemos compartir la información por la red a cualquiera incluso sin nuestro programa, puede ver nuestro archivo, o incluso ver la información con otra aplicación si antes nos hemos puesto de acuerdo en el formato del fichero XML.

1.4. HTML y XML

Lo que HTML hace para la fase de presentación XML lo hace para el intercambio de datos. Algunas tareas no pueden ser resueltas por medio de XML, así como también sucede con HMTL que no puede resolver algunas de las situaciones de presentación de la información. Muchos utilizamos *Adobe Acrobat* en nuestros equipos para visualizar documentos de una manera que HTML no puede. Pero, al igual que podemos reconocer que la mayoría de las veces HTML logra una buena visualización, la gente que trabaja con XML opina que este hace bien su trabajo a la hora de intercambiar datos entre aplicaciones.

Por supuesto existe una diferencia fundamental entre XML y HTML

- HTML está diseñado para una aplicación específica; para transmitir la información hacia los seres humanos, usualmente, de manera visual y por medio de una navegador Web.
- XML no tiene una aplicación específica; está diseñado para el uso que se precise.

HTML tiene un conjunto finito de estructuras de marcas (<P>, <UL>, <H2>, etc.), que se utilizan para crear un documento HTML correcto. En teoría, podemos estar seguros que cualquier navegador nos mostrará un documento HTML correcto porque lo único que tiene que hacer es conocer este conjunto limitado de etiquetas. Aunque en la práctica nos encontramos con documentos WEB que son bien visualizados correctamente en un navegador e incorrectamente en otros, porque difireren en la utilización de algunas de las marcas que se utilizan en HTML.

Por otra parte, si creamos un documento XML podemos estar seguros que cualquier analizador (parser) podrá extraer información del documento, pero lo que no podemos garantizar es que todas las aplicaciones *comprendan el significado de esa información*. Por lo que podemos crear documentos XML para describir información que necesitamos, pero antes de que pueda considerarse útil, debe haber aplicaciones escritas para que los comprendan. Además de las capacidades suministradas por la especificación XML básica existen varias tecnologías relacionadas algunas de las cuales serán analizadas en este curso. Estas tecnologías nos brindan capacidades adicionales, haciendo que XML sea aún más útil. Desgraciadamente algunas de estas tecnologías aún no son suficientemente maduras, lo que implica que el verdadero poder de las mismas está aún por verse.

1.5. Jerarquías en XML

En este punto vamos a analizar cómo se estructuran los datos en un documento XML. Cuando debemos tratar con grandes cantidades de información, o incluso cantidades no tan grandes. Usualmente es mejor agruparlas en subtemas relacionados en lugar de tener toda la información en un único gran grupo. Por ejemplo, este capítulo está dividido en subtemas que a su vez se subdividen en párrafos. Un formulario de impuestos se divide en subsecciones, a lo largo de páginas múltiples. Esto hace que la información resulte más comprensible y también más accesible.

Los desarrolladores de software han utilizado este paradigma durante muchos años, usando una estructura denominada *modelo de objeto*. En este modelo de objeto todala información que está siendo modelada se subdivide en varios objetos y estos, por su parte se agrupan en una jerarquía.

Por ejemplo, cuando se trabaja con HTML dinámico (DHTML) se dispone de un modelo de objetos para utilizar con documentos HTML, denominado DOM (Document Objetc Model). Esto nos permite escribir código en un documento HTML como el siguiente:

alert(document.title);

Estamos utilizando una función *alert()* para producir un mensaje que indica el título de un documento HTML. Esto se realiza accediendo a un objeto denominado *document*, este incluye entre otras muchas cosas una propiedad denominada *title* que devuelve el título del documento HMTL vigente.

XML también agrupa la información en jerarquías. Los elementos de nuestros documentos se ordenan por medio de relación **padre/hijo** o **hermano/hermano**. Estos componentes se denominan (tal y como veremos en el próximo tema) *elementos.*

Consideremos nuestro ejemplo <name>, mostrado jerárquicamente:

- <name>
    - <first>
        - "John"
    - <midle>
        - "Fitzjerald Johansen"
    - <last>
        - "Doe"

<name> es padre de <first>, que, por su parte, es hijo de <name>. <first>, <midle> y <last> son todos hermanos (es decir, son todos hijos de <name>). Observar también que el texto es también hijo de un elemento. Por ejemplo "John" es hijo de <first>.

Esta estructura también se denomina árbol. Todas las partes del árbol que contiene hijos se denominan *ramas*, mientras que las partes que no tienen hijos se denominan *hojas*. Dependiendo del contenido de los elementos se pueden clasificar en:

- Elementos con contenido de elemento. Aquellos que poseen hijos que son a su vez otros elementos. En nuestro ejemplo <name>
- Elementos de contenido simple. Aquellos que solo contienen texto. En nuestro ejemplo <first>, <midle> y <last>.
- Elementos de contenido combinado. Cuando poseen tanto texto como otros elementos. En nuestro ejemplo no hay ningún caso. Pero, por ejemplo: '<doc> <parent>this is some text in my element</parent></doc>'. En este ejemplo <parent> tiene tres hijos:
    - Un hijo de texto "this is some"
    - Un hijo
    - Otro hijo de texto "in my element"

Las relaciones también se pueden definir haciendo la analogía del árbol un poco más compleja. <doc> es el *ancestro* de ; es un *descendiente* de <doc>.

Después que se comprendan las relaciones jerárquicas entre los elementos (y el texto que contiene), se tendrá una idea más completa de la naturaleza de XML. También estará mejor preparado para trabajar con algunas de las otras tecnologías relacionadas con XML, que hacen amplio uso de este paradigma.

1.6. Consorcio W3C

Una de las razones por las que HTML y XML han resultado ser tan buenas ideas es porque son **estándares**. Esto significa que todos pueden seguir esos estándares y las soluciones que se desarrollen permitirán la interoperabilidad. Por lo tanto, ¿quién crea los estándares?

En 1994 se formó W3C (World Wide Web Consortium), con el objetivo, tal y como se expresa en su sitio http: //www.w3c.org, "de lograr que la web desarrolle todo su potencial por medio de la creación de protocolos comunes que promuevan su evolución y que aseguren la interoperabilidad". Al reconocer la necesidad de los estándares, W3C produce recomendaciones que construyen la estructura básica de la web. W3C las denomina recomendaciones, en lugar de estándares, porque los desarrolladores y fabricantes tienen la libertad de seguir esas recomendaciones que brindan esa interoperabilidad.

Su contribución más conocida para la web es, por supuesto, la recomendación HTML; cuando un navegador web señala que se cumple con la versión 3.2 o 4.01 de las recomendaciones HTML, se están refiriendo a la recomendación desarrollada bajo la autoridad de W3C.

La razón por la que las especificaciones de W3C se implementan de manera tan extendida se debe a que la creación de estos estándares es un proceso abierto: cualquier empresa o individuo puede ser miembro de W3C, y ser miembro significa que esas empresas o individuos pueden formar parte del proceso de creación de los estándares. Esto significa que los navegadores web como Netscape Navigator e Interne Explorer de Microsoft muy probablemente implementarán la misma versión del estándar HTML, ya que ambas empresas, Microsoft y Netscape, estuvieron involucradas en la evolución del estándar.

Debido a los objetivos de interoperabilidad de XML, W3C es un buen sitio para desarrollar los estándares relacionados con esta tecnología. Las tecnologías que trataremos en este curso se basan en estándares de W3C: la recomendación XML, la recomendación XSLT, la recomendación XPATH etc.

1.7. Componentes XML

La "comunicación de la información" es un tema muy amplio, y por lo tanto, no es posible que una única especificación pueda abarcar toda la materia. Por esta razón existen varias especificaciones interrelacionadas que funcionan en conjunto para formar la familia de tecnología XML, y cada especificación cubre un aspecto diferente de la comunicación de la información. Aquí presento una lista de las más importantes:

- **XML 1.0 y 1.1**: son las especificaciones base a partir de las que se construye la familia XML. Describe la sintaxis que deben seguir los documentos XML, las reglas que los analizadores XML tienen que cumplir y todo lo que haga falta para leer o escribir un documento XML (también define DTD, aunque a veces se trata como una tecnología diferente.
- Para poder componer nuestras propias estructuras y nombres de elementos para nuestros documentos, las **DTD** y los **esquemas** nos brindan diferentes métodos para crear plantillas para nuestros tipos de documentos. Podemos lograr que otros documentos, producidos por otros desarrolladores, utilicen estas plantillas y obtener así documentos compatibles con nuestra definición.
- Espacios de nombres, **Namespaces**, es un medio para distinguir entre un vocabulario XML

y otro, lo que nos permite crear documentos más consistentes mediante el uso de múltiples vocabularios dentro de un mismo documento.

- **XPath** describe un lenguaje de consulta para el direccionamiento de partes de un documento XML. Esto permite que las aplicaciones puedan requerir una parte específica de un documento XML, en lugar de tener que tratar siempre con todo el documento completo. Por ejemplo podemos utilizar XPath para obtener todos los apellidos que aparezcan en un documento.
- Tal y como vimos anteriormente, en algunos casos nos puede interesar visualizar nuestros documentos XML. En los caos más simples podemos usar **CSS** (Cascading Style Sheets) para definir la presentación de nuestros documentos. En los casos más complejos podemos utilizar **XSL** (Extensible Style Sheet Languaje) que está compuesto por **XSLT** que puede transformar nuestros documentos de un tipo a otro, y **Formating Objects** (Objetos de formateo), que se responsabilizan de la información.
- Para suministrar un medio para que las aplicaciones más tradicionales puedan tener un interfaz con los documentos XML, existe un modelo de objeto documento **DOM**. Existe un medio alternativo para tener una interfaz con documentos XML desde el código de un programa: la API para XML de SAX.

1.8. Resumen

En este capítulo hemos realizado una revisión general de XML y ya tenemos una idea conceptual de su utilidad. Hemos visto las ventajas de los archivos de texto y de los archivos binarios y del modo que XML combina las ventajas de ambos pero eliminando la mayor parte de las desventajas. Hemos visto la flexibilidad que tenemos para crear los datos en el formato que precisemos.

XML es un subconjunto de una tecnología muy probada, SGML, existen muchos años de experiencia detrás del estándar. Existen otras tecnologías, que a su vez utilizan XML, y que nos permiten crear aplicaciones tan complejas o simples como sea necesario.

Gran parte del poder que se obtiene de XML proviene del modo rígido que debe seguir la escritura del documento. En el capítulo siguiente, analizaremos las reglas para la creación de documentos XML bien formados.

1.9. Bibliografía del curso

1.9.1. Material utilizado para la preparación del curso

- XML (Curso de Iniciación). David Hunter. ISBN 84-96097-45-5. INFORBOOK'S, S.L.
- Curso de XML. Gregorio Martín e Isabel Martín. ISBN 84-205-4245-8. Prentice Hall.
- http://www.w3c.es. Consorcio Internacional W3C. En este sitio podemos encontrar las distintas recomendaciones que el consorcio va creando así como las nuevas versiones.
- http://www.w3schools.com/ En esta web se pueden encontrar tutoriales e información sobre todos los estándares (recomendaciones) que van apareciendo desde W3C. A la hora de trabajar con XML (y con otras recomendaciones) es una Web imprescindible. Guía rápida y ejemplos de todos los elementos que componen cada una de las recomendaciones.

1.9.2. Otra bibliografía

- XML Imprescindible. Harold, ElliotteRusty u Scott Means, W. Anaya Multimedia-Anaya Interactiva
- http://www.programacion.net/tutoriales/XML/ Donde podemos encontrar diversos tutoriales sobre XML.