SESION 1: LA PÁGINA EN BLANCO

5) Ejercicio de *explorar el tema*. El procedimiento para desarrollar el modelo cubo es:
    1. *Descríbelo. ¿cómo lo ves, sientes, hueles, tocas o saboreas?*
    2. *Compáralo. ¿a que se parece o de qué se diferencia?*
    3. *Relaciónalo. ¿con qué se relaciona?*
    4. *Analízalo. ¿Cuántas partes tiene? ¿cuáles? ¿cómo funciona?*
    5. *Aplícalo. ¿cómo se utiliza? ¿para qué sirve?*
    6. *Arguméntalo. ¿qué se puede decir a favor y en contra?*

XML

XML, sigla en inglés de ***Ex**tensible **M**arkup **L**anguage* («lenguaje de marcas extensible»), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML.

XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Historia [editar]

XML proviene de un lenguaje inventado por IBM en los años setenta, llamado GML (*Generalized Markup Language*), que surgió por la necesidad que tenía la empresa de almacenar grandes cantidades de información. Este lenguaje gustó a la ISO, por lo que en 1986 trabajaron para normalizarlo, creando SGML (*Standard Generalized Markup Language*), capaz de adaptarse a un gran abanico de problemas. A partir de él se han creado otros sistemas para almacenar información.

En el año 1989 Tim Berners Lee creó la web, y junto con ella el lenguaje HTML. Este lenguaje se definió en el marco de SGML y fue de lejos la aplicación más conocida de este estándar. Los navegadores web sin embargo siempre han puesto pocas exigencias al código HTML que interpretan y así las páginas web son caóticas y no cumplen con la sintaxis. Estas páginas web dependen fuertemente de una forma específica de lidiar con los errores y las ambigüedades, lo que hace a las páginas más frágiles y a los navegadores más complejos.

Otra limitación de SGML es que cada documento pertenece a un vocabulario fijo, establecido por el DTD. No se pueden combinar elementos de diferentes vocabularios. Asimismo es imposible para un intérprete (por ejemplo un navegador) analizar el documento sin tener conocimiento de su gramática (del DTD). Por ejemplo, el navegador sabe que antes de una etiqueta ** ** debe haberse cerrado cualquier **<p>** previamente abierto. Los navegadores resolvieron esto incluyendo lógica ad hoc para el HTML, en vez de incluir un analizador

genérico. Ambas opciones, de todos modos, son muy complejas para los navegadores. Se buscó entonces definir un subconjunto del SGML que permita:

- Mezclar elementos de diferentes lenguajes. Es decir que los lenguajes sean extensibles.
- La creación de analizadores simples, sin ninguna lógica especial para cada lenguaje.
- Empezar de cero y hacer hincapié en que no se acepte nunca un documento con errores de sintaxis.

Para hacer esto XML deja de lado muchas características de SGML que estaban pensadas para facilitar la escritura manual de documentos. XML en cambio está orientado a hacer las cosas más sencillas para los programas automáticos que necesiten interpretar el documento.

Ventajas del XML [editar]

- Es extensible (una vez que un xml fue diseñado y puesto en producción, es posible extenderlo con la adición de nuevas etiquetas de modo que los antiguos consumidores puedan continuar utilizando el servicio sin complicación alguna).

- El analizador es un componente estándar, no es necesario crear un analizador específico para cada lenguaje. Esto posibilita el empleo de uno de los tantos disponibles. De esta manera se evitan *bugs* y se acelera el desarrollo de la aplicación.

- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarlo. Mejora la compatibilidad entre aplicaciones.

Estructura de un documento XML [editar]

La tecnología XML busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible. Que la información sea estructurada quiere decir que se compone de partes bien definidas, y que esas partes se componen a su vez de otras partes. Entonces se tiene un árbol de pedazos de información. Ejemplos son un tema musical, que se compone de compases, que están formados a su vez por notas. Estas partes se llaman *elementos*, y se las señala mediante etiquetas.

Una etiqueta consiste en una marca hecha en el documento, que señala una porción de éste como un elemento. Un pedazo de información con un sentido claro y definido. Las etiquetas tienen la forma **<nombre>**, donde *nombre* es el nombre del elemento que se está señalando.

Documentos XML bien formados [editar]

Los documentos denominados como "bien formados" (del inglés *well formed*) son aquellos que cumplen con todas las definiciones básicas de formato y pueden, por lo tanto, analizarse correctamente por cualquier analizador sintáctico (*parser*) que cumpla con la norma. Se separa esto del concepto de validez que se explica más adelante.

- Los **documentos han de seguir una estructura** estrictamente jerárquica con lo que respecta a las etiquetas que delimitan sus elementos. Una etiqueta debe estar correctamente incluida en otra, es decir, las etiquetas deben estar correctamente anidadas. Los elementos con contenido deben estar correctamente cerrados.

- Los documentos XML sólo permiten un elemento raíz del que todos los demás sean

parte, es decir, solo pueden tener un elemento inicial.

- Los valores atributos en XML siempre deben estar encerrados entre comillas simples o dobles.

- El XML es sensible a mayúsculas y minúsculas. Existe un conjunto de caracteres llamados espacios en blanco (espacios, tabuladores, retornos de carro, saltos de línea) que los procesadores XML tratan de forma diferente en el marcado XML.

- Es necesario asignar nombres a las estructuras, tipos de elementos, entidades, elementos particulares, etc. En XML los nombres tienen alguna característica en común.

- Las construcciones como etiquetas, referencias de entidad y declaraciones se denominan marcas; son partes del documento que el procesador XML espera entender. El resto del documento entre marcas son los datos "entendibles" por las personas.

Partes de un documento XML [editar]

Un documento XML está formado por el prólogo y por el cuerpo del documento.

Prólogo [editar]

Aunque no es obligatorio, los documentos XML pueden empezar con unas líneas que describen la versión XML, el tipo de documento y otras cosas.

El prólogo contiene:

- Una declaración XML. Es la sentencia que declara al documento como un documento XML.

- Una declaración de tipo de documento. Enlaza el documento con su DTD (definición de tipo de documento), o el DTD puede estar incluido en la propia declaración o ambas cosas al mismo tiempo.

- Uno o más comentarios e instrucciones de procesamiento.

Cuerpo [editar]

A diferencia del prólogo, el cuerpo no es opcional en un documento XML, el cuerpo debe contener un y solo un elemento raíz, característica indispensable también para que el documento esté bien formado.

Elementos [editar]

Los elementos XML pueden tener contenido (más elementos, caracteres o ambos), o bien ser elementos vacíos.

Atributos [editar]

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

Entidades predefinidas [editar]

Entidades para representar caracteres especiales para que, de esta forma, no sean interpretados como marcado en el procesador XML.

Secciones CDATA [editar]

Es una construcción en XML para especificar datos utilizando cualquier carácter sin que se interprete como marcado XML. Solo se utiliza en los atributos. No confundir con 2(#PCDATA) que es para los elementos. Permite que caracteres especiales no rompan la estructura. Ej:

<![CDATA[contenido especial: áéíóúñ&]]>

Comentarios [editar]

Comentarios a modo informativo para el programador que han de ser ignorados por el procesador.

Los comentarios en XML tienen el siguiente formato:

<!--- Esto es un comentario --->

<!-- Otro comentario -->

Validez [editar]

Que un documento esté "bien formado" solamente se refiere a su estructura sintáctica básica, es decir, que se componga de elementos, atributos y comentarios como XML especifica que se escriban. Ahora bien, cada aplicación de XML, es decir, cada lenguaje definido con esta tecnología, necesitará especificar cuál es exactamente la relación que debe verificarse entre los distintos elementos presentes en el documento.

Esta relación entre elementos se especifica en un documento externo o definición (expresada como DTD *(Document Type Definition = Definición de Tipo de Documento)* o como XSchema). Crear una definición equivale a crear un nuevo lenguaje de marcado, para una aplicación específica.

Document type definition (DTD) [editar]

La DTD define los tipos de elementos, atributos y entidades permitidas, y puede expresar algunas limitaciones para combinarlos. Los documentos XML que se ajustan a su DTD son denominados válidos.

Declaraciones tipo elemento [editar]

Los elementos deben ajustarse a un tipo de documento declarado en una DTD para que el documento sea considerado como válido.

Modelos de contenido [editar]

Un modelo de contenido es un patrón que establece los subelementos aceptados, y el orden en que se aceptan.

Declaraciones de lista de atributos [editar]

Los atributos se usan para añadir información adicional a los elementos de un documento.

Tipos de atributos [editar]

- Atributos CDATA y NMTOKEN
- Atributos enumerados y notaciones
- Atributos ID e IDREF

Declaración de entidades [editar]

XML hace referencia a objetos que no deben ser analizados sintácticamente según las reglas XML, mediante el uso de entidades. Las entidades pueden ser:

- Internas o externas
- Analizadas o no analizadas
- Generales o parametrizadas

Espacios de nombres [editar]

Los espacios de nombres XML permiten separar semánticamente los elementos que forman un documento XML.

XML Schemas (XSD) [editar]

Un Schema es algo similar a un DTD. Define qué elementos puede contener un documento XML, cómo están organizados y qué atributos y de qué tipo pueden tener sus elementos.

Ventajas de los Schemas frente a los DTDs [editar]

- Usan sintaxis de XML, al contrario que los DTDs.
- Permiten especificar los tipos de datos.
- Son extensibles.

Herramientas para trabajar con documentos XML [editar]

De hecho cualquier procesador de texto, que sea capaz de producir archivos txt es capaz de generar XML, aunque en los entornos de desarrollo como Eclipse o Visual Studio, se facilita, ya que reconoce los formatos y ayuda a generar un XML bien formado.

Lenguajes creados usando XML [editar]

Extensible Stylesheet Language (XSL) [editar]

EL Lenguaje de Hoja de Estilo Extensible (*e**X**tensible **S**tylesheet **L**anguage*, XSL) es una familia de lenguajes que permiten describir como los archivos codificados en xml serán formateados (para mostrarlos) o transformados. Hay tres lenguajes en esta familia: XSL Transformations (XSLT), XSL Formatting Objects (XSL-FO)y XML Path Language.

Lenguaje de enlace XML (XLINK) [editar]

XLink es una aplicación XML que intenta superar las limitaciones que tienen los enlaces de hipertexto en HTML. Es una especificación que todavía está en desarrollo.

Otras tecnologías [editar]

- Hojas de estilo
  - XSL-FO
  - XSLT
  - XLink
  - XPointer
  - XSL
  - hojas de estilo en cascada (CSS)
  - XLT (*X*ML representation of *L*exicons and *T*erminologies)
- Programación
  - JDOM
  - SAX
  - STAX
  - VTD-XML
- Consulta de datos
  - XQuery
  - Xpath
- Seguridad
  - Xades (XML Advanced Electronic Signatures)

Hay quien opina que XML es demasiado pesado para algunas aplicaciones y difícil de editar con un editor de texto simple. Por ello merece la pena mencionar algunas alternativas más ligeras y simples. Los lenguajes de marcas ligeros:

- Simple Outline XML: es un XML simplificado que se puede convertir sin problemas en XML *completo*.
- YAML y OGDL. Estos dos son ficheros de solo texto que no están emparentados con XML como el SOX, antes comentado. Más Información
- BBCode. Éste tiene un uso muy restringido para dar formato nada más.
- Slip

También hay por lo menos un lenguaje basado en XML en formato binario, llamado EBML.

Fuente: Wikipedia http://es.wikipedia.org/wiki/XML