Tema 1: Características de los lenguajes de marcas

Lenguaje de marcas

Un **lenguaje de marcas** es un modo de codificar un documento. Para esto se incorporan etiquetas, marcas o anotaciones que añaden información al texto sobre su formato de presentación o estructura. Todo lenguaje de marcas se define en un documento **DTD** o Document Type Definition en el que se establecen marcas, elementos, etiquetas, atributos, sintaxis y normas de uso.

Clasificación

* De presentación: Define el formato del texto. Ej. HTML.
* De procedimientos: Orientado a la presentación y, además, el programa debe interpretar el código en el mismo orden en el que aparece. Ej. XML.
* Descriptivo o semántico: Describe las partes de la estructura del documento sin especificar su presentación. Ej. CSS.

Agrupación por su ámbito de aplicación

* Documentación electrónica:
  + RTF (Rich Text Format o Formato de Texto Enriquecido): Desarrollado por Microsoft en 1987. Permite el intercambio de documentos de texto entre distintos procesadores de texto.
  + TeX: Crea ecuaciones matemáticas complejas.
  + Wikitexto: Crea páginas wiki en servidores separados para soportar este lenguaje.
  + DocBook: Genera documentos separando la estructura lógica del documento de su formato para su publicación en diferentes formatos.
* Tecnologías de internet:
  + HTML (Hypertext Markup Language), XHTML(eXtensible Hype-rtext Markup Language): Crea páginas web.
  + RSS: Para difusión de contenido.
* Otros lenguajes especializados:
  + MathML (Mathematical Markup Language): Expresa el formalismo matemático para que diferentes sistemas y aplicaciones puedan entenderlo.
  + VoiceXML (Voice Extended Markup Language): Intercambio de información entre un usuario y una aplicación con capacidad de reconocimiento de habla.
  + MusicXML: Intercambio de partituras entre editores diferentes.

Evolución de los sistemas de marcas

En los años 70, además de los lenguajes de programación, surgen los lenguajes orientados a gestionar información. Estos lenguajes se especializan en descripción y estructuración de la información. Son los lenguajes de marcas.

Además, también surgen otros lenguajes centrados en la representación, almacenamiento y consulta eficiente de grandes candidatos de datos. Los **lenguajes y sistemas de bases de datos**.

Los **LM** surgen originalmente como lenguajes formados por el conjunto de códigos de formato que los procesadores de texto introducen en los documentos para dirigir el proceso de presentación en una impresora. Posteriormente, se añadieron como medio de presentación a la pantalla.

* **GML** (Generalized Markup Language): Uno de los problemas más importantes en la informática era la falta de estandarización en los formatos de información. Para resolver este problema, en los 70, **IBM** le encargó a Charles F. Goldfab que crease un sistema de edición, almacenamiento y búsqueda de documentos legales. Tras un análisis previo, la conclusión fue que había que estandarizar los documentos que manejaba la empresa. Así se creó el GML que buscaba describir los documentos independientemente de la plataforma y la aplicación utilizadas.
* **SGML** (Standard Generalized Markup Language): El GML fue evolucionando hasta que en 1986 dio lugar al estándar ISO[[1]](#footnote-0) 8879 llamado **SGML**. Era complejo y necesitaba unas herramientas de software caras, por eso se usa solo en grandes aplicaciones industriales.
* **HTML** (Hypertext Markup Language): En 1989, Tim Berners-Lee, creó el World Wide Web y observó la necesidad de organizar, enlazar y compatibilizar mucha información procedente de diferentes sistemas. Para ello creó un lenguaje de descripción de documentos, el **HTML**. Este lenguaje era una combinación de dos estándares ya existentes:
  + ASCII: Formato que tiene todo procesador de textos sencillo. Permite la transferencia de datos entre diferentes ordenadores.
  + SGML: Lenguaje que estructura el texto, resaltando títulos o aplicando diferentes formatos.

Es una versión simplificada del SGML que utiliza solo las instrucciones imprescindibles. Esto facilitó su uso generalizado para la creación de páginas web. También tiene ciertas desventajas:

* No soporta tareas de impresión y diseño.
* No es flexible, las etiquetas están limitadas.
* No permite mostrar contenido dinámico.
* La estructura y el diseño están mezclados en el documento.
* **XML** (Extensible Markup Language o Lenguaje de Etiquetas Extendido): Para solucionar estos problemas, el W3C crea, en 1998, el estándar internacional **XML**. Un lenguaje de marcas estructural que no tiene información de diseño. Las etiquetas hacen referencia al significado de los datos y no al formato. Es un metalenguaje que permite crear etiquetas propias, asignarles atributos y esquematizar de manera exacta ambos. La estructura y el diseño son independientes. Es un conjunto de estándares relacionados entre sí:
  + XSL (eXtensible Style Language): Define hojas de estilo para XML y transforma documentos.
  + XML (Linking Language): Con Xpath, Xlink y Xpointer, da aspecto a los enlaces entre documentos XML.
  + XML Namespaces: Provee contexto sobre el que aplicar las marcas en XML y diferenciarlas de otras con el mismo nombre.
  + XML Schemas: Define restricciones a XML. Las más usadas son las DTD.

| XML | SGML |
| --- | --- |
| Perfil de sgml. | Aplicación de sgml. |
| Especifica cómo se definen unas etiquetas en un documento. | Aplica unas etiquetas prefijadas a un tipo de documento. |
| Modelo de hiperenlaces complejo. | Modelo de hiperenlaces simple. |
| El navegador es una plataforma para el desarrollo de aplicaciones. | El navegador es un visor de páginas. |
| Ayuda con la compatibilidad. | Problemas de compatibilidad entre navegadores. |

| XML | SGML |
| --- | --- |
| Sencillo. Bajo coste. | Aplicación de SGML. Más cara. |
| Trabaja con documentos bien formados o validados. | Aplica etiquetas a un solo tipo de documento. |
| Se utiliza en muchas áreas. | Se utiliza en sectores muy específicos. |
| El navegador es una plataforma para el desarrollo de aplicaciones. | El navegador es un visor de páginas. |
| Compatibilidad con HTML. | Sin compatibilidad definida con HTML. |
| No usa etiquetas opcionales. |  |

Etiquetas

Los lenguajes de marcas utilizan unas **etiquetas** intercaladas en un documento de texto sin formato. Estas serán interpretadas por los intérpretes del lenguaje para ayudar al procesado del documento. Se escriben entre “<” y “>”. Generalmente se utiliza una de inicio y otra de final. En la de final se añade “/”. Según la W3C deben ir siempre en minúsculas.

Herramientas de edición

Para trabajar con XML se necesita editar y procesar los documentos.

* Editores XML: Utilizan ficheros de texto plano. Vale un procesador de texto normal para los más simples, si son más complejos se recomienda usar un editor XML para crear estructuras y etiquetas. Algunas incluyen ayudas con DTD, hojas de estilo CSS o XSL. W3C tiene un editor gratuito de HTML, XHTML, CSS y XML llamado **Amaya**.
* Procesadores XML: Para interpretar un código XML se puede usar cualquier navegador. Los procesadores XML permiten leer los documentos XML y acceder a su contenido y estructura.

Un procesador es un conjunto de módulos de software entre los que se encuentra un parser o analizador que comprueba que el documento cumple las normas establecidas. Si se centran en que el documento sea válido para trabajar con él son validadores,. Si solo exigen que el documento esté bien formado, son no validadores. Para publicar un documento XML en internet se usan procesadores XSLT que permiten generar documentos HTML a partir de XML.

Estructura y sintaxis

El XML es un lenguaje de etiquetas, creadas por el programador, que **estructuran**, **guardan** y **almacenan** la información. No representa los datos, solo organiza la estructura. Ayuda a webs y aplicaciones a guardar información. Se utiliza en la mayoría de sistemas operativos y móviles.

XML es un documento de texto compuesto por parejas de etiquetas y estructuras en árbol que describen una función en la organización del documento. Se puede editar con cualquier editor de texto e interpretar por los navegadores web. Tiene varias características:

* Compatible con protocolos HTTP y URL.
* Si se verifican las normas de XML, está conforme con SGML.
* No necesita conocimientos de programación.
* Los documentos son fáciles de crear.
* Sus documentos son fáciles de difundir.
* Es legible para humanos.
* Diseño formal y conciso.
* Es extensible, adaptable y aplicable a muchas situaciones.
* Orientado a objetos.
* Se compone de datos de marcado y datos de carácter entremezclados.

Hay varias etapas del procesado de creación de un documento XML:

* Especificación de requisitos.
* Diseño de etiquetas.
* Marcado de los documentos: Etiquetas que se añaden a un texto para estructurar el contenido para que los ordenadores los interpreten. Es todo lo que se sitúa entre los caracteres “<” y “>” o “&” y “;”.

Para añadir comentarios se utilizan “<!” y “-->”. Pueden ir en cualquier posición menos antes del prólogo o dentro de una etiqueta.

Estos documentos se pueden formar por una parte opcional llamada prólogo y una obligatoria llamada ejemplar. El **prólogo** siempre debe ir antes que el ejemplar. Sirve para facilitar el procesado de la información del ejemplar y está dividido en:

* Declaración XML: Debe ser la primera línea del documento. Al ser opcional permite que se procesen documentos HTML y SGML como si fueran XML. El prólogo puede servir para declarar:
  + Versión del XML: Se usa la etiqueta “<?xml verizon=”1.0”?>”.
  + Condiciones de la codificación: Determina el conjunto de caracteres. “<?xml versión=“1.0” encoding=“iso-8859-1”?>”.
    - UTF-8 (Unicode): Universal.
    - ISO-8859-1 (Latin-1): Europa occidental y Latinoamérica.
    - ISO-8859-2 (Latin-2): Europa central y oriental.
    - ISO-8859-3 (Latin-3): Suroeste de Europa.
* Declaración del tipo de documento: Siempre empieza con la cadena “<!DOCTYPE Nombre\_tipo…>”.

El **ejemplar** es la parte más importante de un documento XML. Contiene sus datos reales. Se forma con elementos anidados que son los distintos bloques de información y están delimitados por una etiqueta de apertura y otra de cierre. Los nombres de estas etiquetas tienen que ser autodescriptivos. Los elementos deben seguir una serie de normas:

* Solo puede haber un elemento raíz.
* Todos los elementos tienen una etiqueta de inicio y otra de cierre.
* Si un elemento está vacío, se puede utilizar simplemente una etiqueta <elemento/>.
* No se puede cerrar un elemento que contenga otro que esté abierto.
* Los nombres de las etiquetas de inicio y cierre de un mismo elemento son idénticas.
* Puede ser cualquier cadena alfanumérica sin espacios siempre que no empiece por “:” o “xml” y variaciones.
* Los elementos no pueden contener “]]>” por incompatibilidad con el XML.
* Hay caracteres que deben sustituirse por cadenas:
  + > – &gt
  + < – &lt
  + & – &amp
  + “ – &quot
  + `– &apos
* Para utilizar caracteres especiales como “€” se deben usar expresiones &#D o &#H donde D y H son el número decimal y hexadecimal del carácter en el UNICODE. Por ejemplo, € es &#8364 o &#x20AC.

Los **atributos** permiten añadir propiedades a los elementos de un documento. No tienen jerarquía ni contienen ningún otro elemento, atributo o estructura lógica. Se definen y dan valor en la etiqueta de inicio o en una de elemento vacío. Sus valores deben ir entrecomillados. Sus nombres siguen las mismas reglas que los elementos y no pueden contener el valor “<”.

Documentos XML bien formados

Todos los documentos tienen que seguir las reglas sintácticas del W3C:

* Tener un prólogo definido con la declaración XML completa.
* Solo un elemento raíz en el que van todos los demás anidados.
* Se deben cumplir las reglas sintácticas xML para definir elementos y atributos

Utilización del espacio de nombres

Permiten definir la pertenencia de los elementos y los atributos de un documento XML al contexto de un vocabulario XML y así resolver las ambigüedades que puedan aparecer al unir documentos de más de un autor que usan la misma etiqueta para nombrar cosas diferentes. Los espacios de nombres o name spaces permiten dar un nombre único a cada elemento, indexándolos según el nombre del vocabulario adecuado. Además, se asocian a un URI que los identifica de manera única.

Las etiquetas ambiguas se sustituyen por otras en las que el nombre del elemento lleva un prefijo que determina el contexto al que pertenece la etiqueta seguido de “:”:

* <prefijo:nombre\_etiqueta></prefijo:nombre\_etiqueta>

Esta etiqueta es el **nombre cualificado**. No se pueden usar espacios, carácteres especiales ni empezarlos por un dígito. Antes de usar uno se necesita declarar el espacio de nombres, asociando un índice con el URI asignado al espacio de nombres por un atributo especial xmlns. Esto se hace entre el prólogo y el ejemplar de un documento:

* <conexion>://<direccionservidor>/<apartado1>/<apartado2>/…

1. ISO: Normas de obligado cumplimiento. [↑](#footnote-ref-0)