CICLO ADMINISTRACIÓN SISTEMAS INFORMATICOS Y EN RED

Lenguajes de Marcas

Contenido:

Unidad 1- Concepto de Lenguajes de Marcas

Un lenguaje de marcado o lenguaje de marcas es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación.

El lenguaje de marcas más extendido es el HTML ("HyperText Markup Language", Lenguaje de marcado de hipertexto), fundamento del World Wide Web

Los lenguajes de marcado suelen confundirse con lenguajes de programación. Sin embargo, no son lo mismo, ya que el lenguaje de marcado no tiene funciones aritméticas o variables, como sí poseen los lenguajes de programación. Históricamente, el marcado se usaba y se usa en la industria editorial y de la comunicación, así como entre autores, editores e impresores.

1.- Representación de la Información

El ordenador es una máquina digital, por lo tanto sólo es capaz de representar información utilizando el sistema binario de numeración. Esto obliga a que, para poder almacenar información en un ordenador, previamente haya que codificarla en forma de números binarios.

El problema de los números binarios es que están muy alejados del ser humano; es decir, que las personas no estamos capacitadas para manejar información en binario. Nosotros usamos sistema decimal para los números y formas de representación mucho más complejas para otra información (como el texto, las imágenes, la música,…)

Sin embargo actualmente un ordenador es capaz de manejar información de todo tipo: música, imágenes, texto,…. Esto es posible porque se ha conseguido que casi cualquier tipo de información sea codificable en binario.

Los seres humanos tenemos la capacidad de diferenciar claramente lo que es un texto de una imagen, lo que es un número de una canción,… Pero en un ordenador todo es más complicado, porque todo es binario.

Desde los inicios de la informática la codificación (el paso de información humana a información digital) ha sido problemática debido a la falta de acuerdo en la representación. Pero hoy día ya tenemos numerosos estándares.

Fundamentalmente la información que un ordenador maneja son Números y Texto. Pero curiosamente a nivel formal se consideran datos binarios a cualquier tipo de información representable en el ordenador, que no es texto (imagen, sonido, vídeo,…), aunque como ya hemos comentado, en realidad toda la información que maneja un ordenador es binaria, incluido el texto.

Datos Binarios

Cualquier dato que no sea texto, se considera dato binario. Por ejemplo: música, vídeo, imagen, un archivo Excel, un programa,…
La forma de codificar ese tipo de datos a su forma binaria es muy variable. Por ejemplo en el caso de las imágenes, cada punto (píxel) de la imagen se codifica utilizando su nivel de rojo, verde y azul. De modo que una sola imagen produce millones de dígitos binarios.
En cualquier caso sea cual sea la información que estamos codificando en binario, para poder acceder a dicha información, el ordenador necesita el software que sepa como decodificar la misma, es decir saber qué significa cada dígito binario para traducirle a una forma más humana. Eso sólo es posible utilizando el mismo software con el que se codificó o bien otro software pero que sea capaz de entender la información codificada.

Texto

El texto es quizá la forma más humana de representar información. Antes de la llegada del ordenador, la información se transmitía mediante documentos o libros en papel. Esa forma de transmitir es milenaria y sigue siendo la forma más habitual de transmitir información entre humanos; incluso con la tecnología actual aplicaciones como twitter, whassap,… siguen usando el texto como formato fundamental para transmitir información.
En cuanto apareció la informática como una ciencia digital, apareció también el problema de cómo codificar texto en forma de dígitos binarios para hacerlo representable en el ordenador. La forma habitual ha sido codificar cada carácter en una serie de números binarios. De modo que por ejemplo el carácter A fuera por ejemplo 01000001 y la B el 01000010.
El problema surgió por la falta de estandarización, la letra A se podía codificar distinto en diferentes ordenadores y así nos encontrábamos con un problema al querer pasar datos de un ordenador a otro. Poco a poco aparecieron estándares para intentar que todo el hardware y software codificara los caracteres igual.

Código ASCII

El problema de la codificación de texto que hacía incompatibles los documentos de texto entre diferentes sistemas, se palió cuando se ideó en 1967 un código estándar por parte de la ANSI, la agencia de estándares norteamericana, dicho código es el llamado ASCII (American Standard Code for Information Interchange, código estándar americano para el intercambio de información). El código utiliza el alfabeto inglés (que utiliza caracteres latinos) y para codificar todos los posibles caracteres necesarios para escribir en inglés se ideó un sistema de 7 bits (con 7 bits se pueden representar 128 símbolos, suficientes para todas las letras del alfabeto inglés, en minúsculas y mayúsculas, caracteres de puntuación, símbolos especiales e incluso símbolos de control).

Pero, en países con lenguas distintas del inglés, surgió el problema de que parte de los símbolos de sus alfabetos quedaban fuera del ASCII (como la letra eñe)-

Por ello se diseñaron códigos de 8 bits que añadían 128 símbolos más y así aparecieron los llamados códigos ASCII extendidos. En ellos, los 128 símbolos primeros son los mismos de la tabla ASCII original y los 128 siguientes se corresponden a símbolos extra. Así por ejemplo el sistema MS-DOS utilizaba el llamado código 437 que incluía símbolos y caracteres de otras lenguas de Europa Occidental y caracteres que permitían hacer marcos y bordes en pantallas de texto, entre otros símbolos.

Sin embargo 8 bits siguen siendo insuficientes para codificar todos los alfabetos del planeta. Por lo que cada zona usaba su propia tabla ASCII extendida. Ante el caos consiguiente, la ISO decidió normalizar dichas tablas de códigos para conseguir versiones estándares de los mismos. Lo hizo mediante las siguientes normas (cada una de las cuales definía una tabla de 256 caracteres, siempre los 128 primeros son el ASCII original)

8859-1. ASCII extendido para Europa Occidental (incluye símbolos como ñ o β)

8859-2. ASCII extendido para Europa Central y del Este (incluye símbolos como Ź o č)

8859-3. ASCII extendido para Europa del Sur (incluye símbolos como Ġ o Ï)

8859-4. ASCII extendido para Europa del Norte (incluye símbolos como ø o å)

8859-5. ASCII extendido para alfabeto cirílico (incluye símbolos como д o Ж)

8859-6. ASCII extendido para alfabeto árabe (incluye símbolos como o

7-8859ASCII extendido para alfabeto griego moderno (incluye símbolos como ф o α)

8859-8. ASCII extendido para alfabeto hebreo (incluye símbolos como o

8859-9ASCII extendido, versión de 8859-1 que incluye símbolos turcos en lugar de otros poco utilizados

8859-10. ASCII extendido, versión de 8859-4 que incluye símbolos más utilizados en las lenguas nórdicas actuales

Este problema sigue existiendo ahora de modo que en los documentos de texto hay que indicar el sistema de codificación utilizado (el caso más evidente son las páginas web), para saber cómo interpretar los códigos del archivo. Así en 8859_1 el código 245 es el carácter õ y en 8859_2 es el carácter ő

Unicode

La complicación de las tablas de código se intenta resolver gracias al sistema Unicode que ha conseguido incluir los caracteres de todas las lenguas del planeta a cambio de que cada carácter ocupe más de un byte (ocho bits). En Unicode a cada símbolo se le asigna un número (evidentemente los 128 primeros son los originales de ASCII para mantener la compatibilidad con los textos ya codificados y de hecho los 256 primeros son la tabla ISO-8859_1).

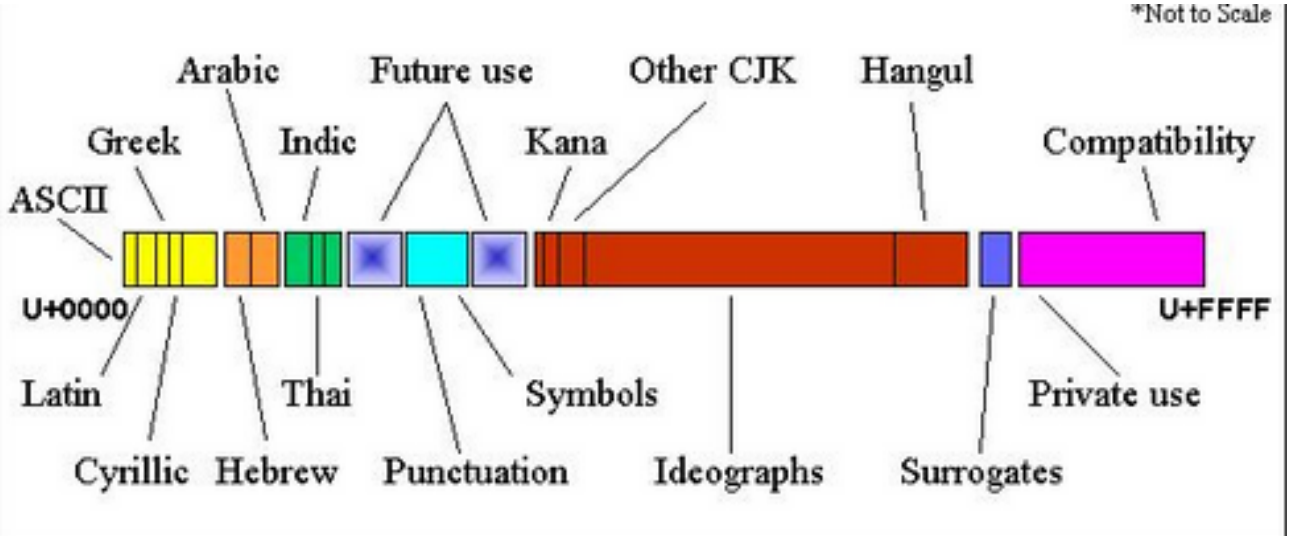
Para ello el organismo también llamado Unicode participado por numerosas e influyentes empresas informáticas y coordinado por la propia ISO, ha definido tres formas de codificar los caracteres:

UTF-8. Es la más utilizada (y la más compleja de usar para el ordenador). Utiliza para cada carácter de uno a cuatro caracteres, de forma que:

- Utilizan uno los que pertenecen al código ASCII original
- Los pertenecientes a lenguas latinas, cirílicas, griegas, árabes, hebreas y otras de Europa, Asia Menor y Egipto
- Para símbolos fuera de los alfabetos anteriores como el chino o el japonés
- Para otros símbolos: por ejemplo los matemáticos y símbolos de lenguas muertas como el fenicio o el asirio o símbolos asiáticos de uso poco frecuente.

UTF-16. Utiliza para cada carácter dos (para los dos primeros grupos del punto anterior) o cuatro caracteres (para el resto). Es más sencillo que el anterior

UTF-32. La más sencilla de todas. Cada carácter independientemente del grupo al que pertenezca ocupa 4 caracteres. No se utiliza.



2.– Archivos binarios y archivos de texto

Un archivo binario es un archivo que contiene información de cualquier tipo codificada en binario para el propósito de almacenamiento y procesamiento en ordenadores. Por ejemplo los archivos informáticos que almacenan fotografías, música así como los archivos ejecutables que contienen programas.

Muchos formatos binarios contienen partes que pueden ser interpretadas como texto. Un archivo binario que sólo contiene información de tipo textual sin información sobre el formato del mismo se dice que es un archivo de texto plano.

Ventajas de los archivos binarios

(1) Ocupan menos espacio que los archivos de texto, ya que optimizan mejor su codificación a binario (por ejemplo el número 213 ocupa un solo byte y no tres como ocurriría si fuera un texto).

(2) Son más rápidos de manipular por parte del ordenador (se parecen más al lenguaje nativo del ordenador)

(3) Permiten el acceso directo a los datos. Los archivos de texto siempre se manejan de forma secuencial, más lenta

En cierto modo permiten cifrar el contenido que de otra forma sería totalmente visible por cualquier aplicación capaz de entender textos (como el bloc de notas). Es decir los datos no son fácilmente entendibles

Sin embargo, los archivos de texto poseen otras ventajas que los hacen imprescindibles en determinadas situaciones:

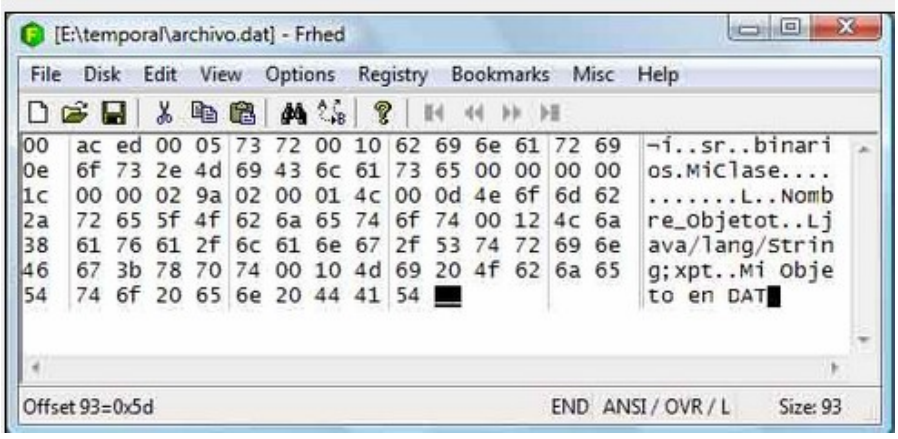
(1) Son ideales para almacenar datos para exportar e importar información a cualquier dispositivo electrónico ya que cualquier dispositivo es capaz de interpretar texto

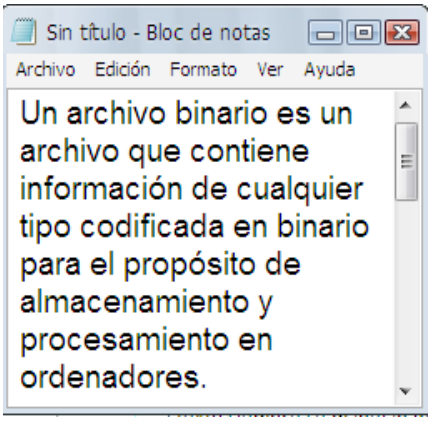
(2) Son directamente modificables, sin tener que acudir a software específico

(3) Su manipulación es más sencilla que la de los archivos binarios

(4) Son directamente transportables y entendibles por todo tipo de redes

.





3.– Compartir Datos

Los problemas relacionados con el intercambio de información entre aplicaciones y máquinas informáticas es tan viejo como la propia informática.

El problema parte del hecho de haber realizado un determinado trabajo con un software en un ordenador concreto y después querer pasar dicho trabajo a otro software en ese u otro ordenador.

Los archivos binarios tienen la complicación de que para hacer ese proceso, el origen y el destino de los datos deben comprender cómo codificar y decodificar la información. Eso, en muchos casos, ha sido un gran problema que ha obligado a que todos los trabajadores hayan tenido que adaptarse al software de la empresa y por supuesto en toda la empresa utilizar dicho software.

En la informática actual eso es aún más problema al tener una necesidad de disponibilidad global del trabajo y además la posibilidad de ver dicho trabajo en dispositivos de todo tipo como mini ordenadores, PDA o incluso teléfonos móviles.

Por ello poco a poco han aparecido formatos binarios de archivo que han sido estándares de facto (aunque no han sido reconocidos por ningún organismo de estándares) como por ejemplo el formato documental PDF, el formato de imagen JPEG, la música MP3 o el formato MPEG de vídeo.

Pero sigue habiendo empresas que utilizan formato propio por la idea de que sus formatos de archivo están directamente relacionados con la calidad de su software es decir razonan que el software que fabrican es muy potente y necesitan un formato binario propio compatible con esa potencia. De ahí que muchas veces la opción para exportar e importar datos sea utilizar conversores, capaces de convertir los datos de un formato a otro (por ejemplo de Word a Open Office; de MP3 a MOV de Apple, etc.).

Sin embargo hay un formato de archivo que cualquier dispositivo es capaz de entender. El texto. La cuestión es que para los archivos llamados de texto, sólo son capaces de almacenar texto plano; es decir sólo texto sin indicar ningún formato o añadir información no textual.

Debido a la facilidad de ser leído con cualquier aparato, se intenta que el propio texto sirva para almacenar otros datos. Evidentemente no es posible usar texto para almacenar por ejemplo imágenes, pero sí otras cosas. Para ello dentro del archivo habrá contenido que no se interpretará como texto sin más que simplemente se debe mostrar, sino que hay texto en el archivo que se marca de manera especial haciendo que signifique otra cosa. Desde hace muchos añoso hay dos campos en los que está idea ha funcionado bien: en las bases de datos y en los procesadores de texto. Actualmente el éxito de Internet ha permitido espolear esta tecnología a otros campos.

Hay un problema con el texto, puesto que al ser formato tan universal, y ser su contenido siempre visible; es peligroso como fuente para almacenar datos confidenciales, ya que quedaría expuesto a cualquier persona.

Los datos binarios no son del todo seguros, pero como requieren del software que entienda el formato binario concreto hacen que su contenido quede menos expuesto.

4.– Lenguajes de Marcas

Como se ha comentado en el punto anterior, el problema de la exportación de datos ha puesto en entredicho a los archivos binarios como fuente para exportar e importar información.

En su lugar parece que los archivos de texto poseen menos problemas (excepto el del cifrado de su información, que queda demasiado descubierta). Por ello se ha intentado que los archivos de texto plano (archivos que sólo contienen texto y no otros datos binarios) pudieran servir para almacenar otros datos como por ejemplo detalles sobre el formato del propio texto u otras indicaciones.

Los procesadores de texto fueron el primer software en encontrarse con este dilema. Puesto que son programas que sirven para escribir texto parecía que lo lógico era que sus datos se almacenaran como texto. Pero necesitan guardar datos referidos al formato del texto, tamaño de la página, márgenes, etc. La solución clásica ha sido guardar la información de formato de forma binaria, lo que provoca los ya comentados problemas.

Algunos procesadores de texto optaron por guardar toda la información como texto, haciendo que las indicaciones de formato no se almacenen de forma binaria sino textual. Dichas indicaciones son caracteres marcados de manera especial para que así un programa adecuado pueda traducir dichos caracteres no como texto sino como operaciones que finalmente producirán mostrar el texto del documento de forma adecuada..

La idea del marcado procede del inglés marking up término con el que se referían a la técnica de marcar manuscritos con lápiz de color para hacer anotaciones como por ejemplo la tipografía a emplear en las imprentas. Este mismo término se ha utilizado para los documentos de texto que contienen comandos u anotaciones.

Las posibles anotaciones o indicaciones incluidos en los documentos de texto han dado lugar a lenguajes (entendiendo que en realidad son formatos de documento y no lenguajes en el sentido de los lenguajes de programación de aplicaciones) llamados lenguajes de marcas, lenguajes de marcado o lenguajes de etiquetas.

.

TeX y LaTeX

En la década de los 70 Donald Knuth (uno de los ingenieros informáticos más importantes de la historia, padre del análisis de algoritmos) creó para producir documentos científicos utilizando una tipografía y capacidades que fueran iguales en cualquier computadora, asegurando además una gran calidad en los resultados.

Para ello apoyó a **TeX** con tipografía especial (fuentes Modern Computer) y un lenguaje de definición de tipos (METAFONT). TeX ha tenido cierto éxito en la comunidad científica gracias a sus 300 comandos que permiten crear documentos con tipos de gran calidad, para ello se necesita un programa capaz de convertir el archivo TeX a un formato de impresión.

El éxito de TeX produjo numerosos derivados de los cuales el más popular es **LaTeX**.

Se trata de un lenguaje que intenta simplificar a TeX, fue definido en 1984 por Leslie Lamport, aunque después ha sido numerosas veces revisado. Al utilizar comandos de TeX y toda su estructura tipográfica, adquirió rápidamente notoriedad y sigue siendo utilizado para producir documentos con expresiones científicas, de gran calidad. La idea es que los científicos se centren en el contenido y no en la presentación


```
\documentclass{article}
% pre\'ambulo

\usepackage{lmodern}
\usepackage[T1]{fontenc}
\usepackage[spanish,activeacute]{babel}
\usepackage{mathtools}

\title{Hola Mundo}
\author{Escribe aqu\'i tu nombre}

\begin{document}
% cuerpo del documento

\maketitle

Mi primer documento en \LaTeX{}.

\end{document}
```

RTF

RTF es el acrónimo de Rich Text Format (Formato de Texto Enriquecido) un lenguaje ideado por Microsoft en 1987 para producir documentos de texto que incluyan anotaciones de formato.

Actualmente se trata de un formato aceptado como texto con formato y en ambiente Windows es muy utilizado como formato de intercambio entre distintos procesadores por su potencia. El procesador de texto Word Pad incorporado por Windows lo utiliza como formato nativo .

El siguiente ejemplo muestra el código RTF de un documento cuyo resultado es mostrar el mensaje "Esto es un **ejemplo** RTF"

```
{\rtf1\ansi\ansicpg1252\deff0\deflang5130{\fonttbl{\f0\fswiss\fcharset0 Arial;}}
{\*\generator Msftedit 5.41.15.1515;}\viewkind4\uc1\pard\f0\fs20 Esto es un \b ejemplo\b0  RTF\par
}
```

SGML

Se trata de la versión de GML que estandarizaba el lenguaje de marcado y que fue definida finalmente por ISO como estándar mundial en documentos de texto con etiquetas de marcado. La estandarización la hace el subcomité SC24 que forma parte del comité JTC1 del organismo IEC de ISO que se encarga de los estándares electrónicos e informáticos (en definitiva se trata de una norma ISO/IEC JTC1/SC24, concretamente la 8879).

Su importancia radica en que es el padre del lenguaje XML y la base sobre la que se sostiene el lenguaje HTML.

En SGML las etiquetas que contienen indicaciones para el texto se colocan entre símbolos < y >. Las etiquetas se cierran con el signo /. Es decir las reglas fundamentales de los lenguajes de etiquetas actuales ya las había definido SGML.

En realidad (como XML) no es un lenguaje con unas etiquetas concretas, sino que se trata de un lenguaje que sirve para definir lenguajes de etiquetas; o más exactamente es un lenguaje de marcado que sirve para definir formatos de documentos de texto con marcas. Entre los formatos definidos mediante SGML, sin duda HTML es el más popular.

Postscript

Se trata de un lenguaje de descripción de páginas. De hecho es el más popular. Permite crear documentos en los que se dan indicaciones potentísimas sobre como mostrar información en el dispositivo final. Se inició su desarrollo en 1976 por John Warnock y dos años más tarde se continuo con la empresa Xerox, hasta que en 1985 el propio Warnock funda Adobe Systems y desde esa empresa se continua su desarrollo.

Es en realidad todo un lenguaje de programación que indica la forma en que se debe mostrar la información que puede incluir texto y el tipo de letra del mismo, píxeles individuales y formas vectoriales (líneas, curvas). Sus posibilidades son muy amplias.

El concepto PostScript se diferenció, fundamentalmente, por utilizar un lenguaje de programación completo, para describir una imagen de impresión. Imagen que más tarde sería impresa en una impresora láser o algún otro dispositivo de salida de gran calidad, en lugar de una serie de secuencias de escapes de bajo nivel (en esto se parece a Emacs, que explotó un concepto interno parecido con respecto a las tareas de edición).

También implementó, notablemente, la composición de imágenes. Estas imágenes se describían como un conjunto de:

Líneas horizontales

Píxeles al vuelo

Descripciones por curvas de Bezier

Tipos de letra (mal llamados fuentes) de alta calidad a baja resolución1 (e.g. 300 puntos por pulgada).

```
%!
% Ejemplo de código
  /Courier findfont
  20 scalefont
  setfont
% Courier tamaño 20
  newpath
  72 72 moveto
% Cursor en la esquina
% inferior izquierda (72,72)
  (¡Hola mundo!) show
  showpage
```

PostScript

HTML

Tim Bernes Lee utilizó SGML para definir un nuevo lenguaje de etiquetas que llamó Hypertext Markup Language (lenguaje de marcado de hipertexto) para crear documentos transportables a través de Internet en los que fuera posible el hipertexto; es decir la posibilidad que determinadas palabras marcadas de forma especial permitieran abrir un documento relacionado con ellas.
A pesar de tardar en ser aceptado, HTML fue un éxito rotundo y la causa indudable del éxito de Internet. Hoy en día casi todo en Internet se ve a través de documentos HTML, que popularmente se denominan páginas web.
Inicialmente estos documentos se veían con ayuda de intérpretes de texto (como por ejemplo el Lynx de Unix) que simplemente coloreaban el texto y remarcaban el hipertexto. Después el software se mejoró y aparecieron navegadores con capacidad más gráfica para mostrar formatos más avanzados y visuales.

5.- XML

Se trata de un subconjunto de SGML ideado para mejorar el propio SGML y con él definir lenguajes de marcado con sintaxis más estricta, pero más entendibles. Su popularidad le ha convertido en el lenguaje de marcado más importante de la actualidad y en el formato de documentos para exportación e importación más exitoso.

XML es un lenguaje de marcas que se ha estandarizado y se ha convertido en uno de los formatos más populares para intercambiar información.

Se trata de un formato de archivos de texto con marcado que deriva del original SGML, pero que le ha superado añadiendo otro tipo de reglas y de forma de trabajar.

La realidad es que XML siempre ha estado muy ligado al éxito de HTML. Se planteó por los problemas crecientes que se fueron observando en las páginas web.

HTML tiene estos problemas como formato de intercambio de información:

La mayoría de etiquetas HTML no son semánticas; es decir, no sirven para decir el tipo de contenido que tenemos sino para indicar el formato. Por ejemplo la etiqueta H1 sí es semántica ya que indica que el texto que contiene es un encabezado de nivel principal. Mientras que la etiqueta HTML clásica font sirve para colorear o cambiar el tipo de letra, sin indicar qué tipo de texto tenemos (se aplica a cualquiera)

HTML es un lenguaje rígido, no es extensible. Es decir no podemos añadir etiquetas ya que ningún navegador las reconocerá. Cada vez que se decide añadir hay que cambiar el estándar y los navegadores se deben de adaptar a los cambios.

Requiere de arreglos “extraños” para añadir potencia y funcionalidad, por lo que los diseñadores tienden a incrustar dentro del código HTML código de lenguajes como PHP o Javascript que dificultan su legibilidad y comprensión.

Por ello al crear XML se plantearon estos objetivos:

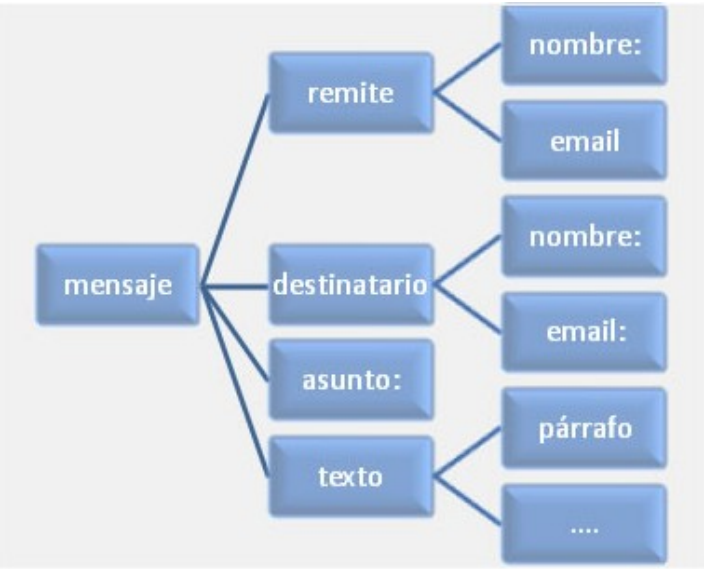
(1) Debía de ser similar a HTML (de hecho se basa en el lenguaje SGML base para el formato HTML)

(2) Debía de ser extensible, es decir que sea posible añadir nuevas etiquetas sin problemas. Esa es la base del lenguaje XML.

(3) Debía de tener unas reglas concisas y fáciles, además de estrictas.

(4) Debía de ser fácil de implantar en todo tipo de sistemas. XML nace con una vocación multiplataforma, como base de intercambio de información entre sistemas de toda índole.

(5) Debía ser fácil de leer por los humanos y fácil crear procesadores XML software (llamados parsers)



Lenguajes basados en XML

RSS. (Really Simple Syndication, aunque hay otras interpretaciones de los acrónimos) Para producir contenidos sindicables, se utiliza fundamentalmente para producir noticias. Es una de las aplicaciones XML más utilizada

Atom. Formato semejante al anterior, pensado también para distribuir información desde una web. Muchas veces complementa a RSS

ePUB. Formato de libro digital que se ha convertido en un estándar de facto por la gran implantación que está teniendo en todos los dispositivos de lectura digitales. En realidad es un archivo comprimido (ZIP) que contiene tres documentos XML que son los que especifican la estructura y contenido del documento.

DITA. Darwin Information Typing Architecture, se utiliza para producir documentos técnicos y está siendo cada vez más popular. Tiene capacidad para incluir numerosos metadatos y para poder adaptarse a las necesidades de las entidades que utilicen el lenguaje. Permite dividir en temas reutilizables los documentos.

MathML. Pensado para representar documentos con expresiones matemáticas.

ODF u OD. Open Document for Office Applications. Es un formato abierto que pretende ser un estándar como formato de intercambio entre documentos de oficina (hojas de cálculo, procesadores de texto, presentaciones, diagramas, …). Su popularidad aumentó notablemente cuando fue adoptado por el software Open Office, actualmente perteneciente a la empresa Oracle.

OSDF. Open Software Description Format, basado en especificaciones de las empresas Marimba y Microsoft, sirve para describir la tecnología y los componentes con los que se ha desarrollado un determinado software, a fin de facilitar el proceso de instalación.

OOXML. Office Open XML. Formato rival del anterior, propiedad de Microsoft y utilizado como formato XML para el software Microsoft Office. Pretende también ser un estándar.

RDF, Resource Description Format, Sirve para desarrollar documentos que describan recursos. Se trata de un proyecto ya antiguo para definir modelos de metadatos. Se basa en los modelos conceptuales como el modelo entidad/relación o los diagramas de clases, aunque actualmente se utiliza fundamentalmente para describir recursos web.

SMIL. Synchronized Multimedia Integration Language, Lenguaje sincronizado de integración multimedia. Utilizado para producir presentaciones de TV en la web, fundamentalmente.

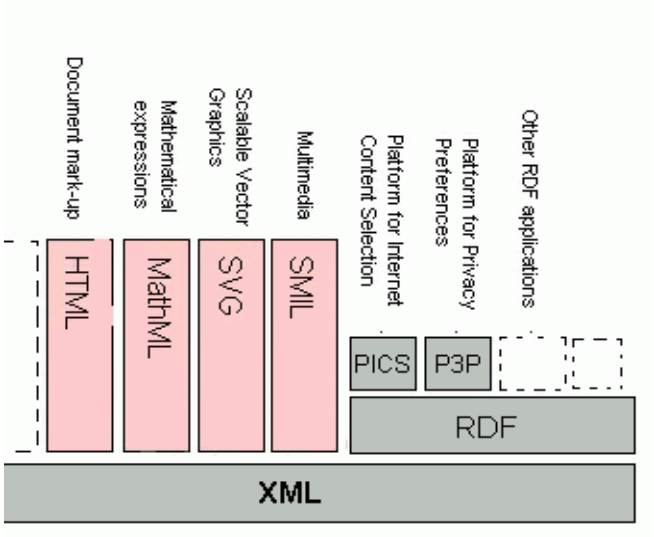
SOAP. Simple Object Access Protocol. Protocolo estándar de comunicación entre objetos utilizado para comunicar con Servicios Web.

SVG. Scalable Vector Graphics, gráficos de vectores escalables. Permite definir imágenes vectoriales pensadas para ser publicadas en una página web.

VoiceXML. Se utiliza para representar diálogos vocales.

WSDL. Web Services Description Language. Lenguaje para definir Servicios Web.

XHTML. Versión del lenguaje de creación de páginas web, HTML, que es compatible con las normas XML.



Usos de XML

1.– Contenido web

XML podría pasar a reemplazar a HTML como el formato en el que se escriben las páginas web. Es de hecho una de sus vocaciones ya que ofrece una sintaxis más rígida (que es ventajosa ya que facilita su aprendizaje), una escalabilidad (que permite que jamás se quede obsoleto el lenguaje) y una serie de tecnologías relacionadas más poderosas.

La razón del fracaso de HTML como estándar se debe a que cada navegador impone etiquetas propias, es decir que no hay un estándar real. Con XML no es posible esta situación, ya que la propia naturaleza del lenguaje no la hace posible ya que todo XML tendrá un documento de validación conocido por los navegadores que especifica exactamente qué elementos y de qué forma se pueden utilizar en el documento.

2.– Intercambio de información entre aplicaciones

El hecho de que XML almacene información mediante documentos de texto plano, facilita que se utilice como estándar, ya que no se requiere software especial para leer su contenido: es texto y es entendible por cualquier software.

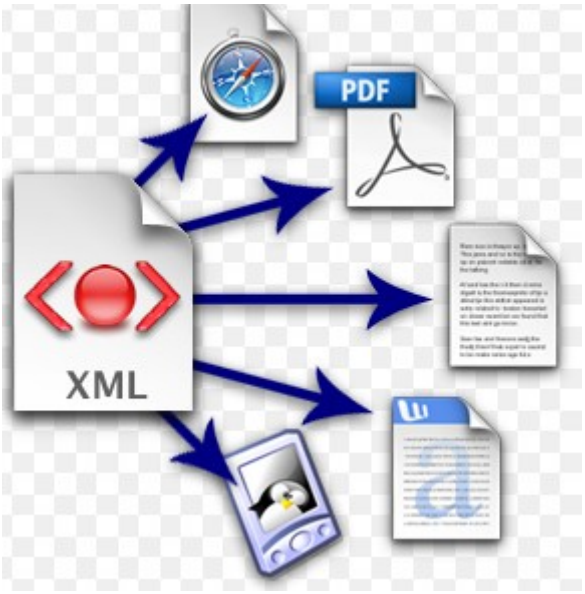
3.– Computación distribuida

Se trata de la posibilidad de utilizar XML para intercambiar información entre diferentes computadoras a través de las redes. Las ventajas de XML están relacionadas con el hecho de que con él se crean documentos inocuos (no pueden contener código maligno como virus o espías), con lo que la seguridad de esos sistemas es total.

Es decir un documento XML no puede contener virus o software espía. Las máquinas enrutadoras de las redes no tienen ningún problema en encaminar archivos XML al ser texto. Entienden que no es una amenaza.

4.– Información empresarial

XML es un formato que tiene cada vez más importancia para generar documentos empresariales por la facilidad de estructurar los datos de la forma más apropiada para la empresa. Un documento XML se parece mucho a una pequeña base de datos, con la ventaja de que es muy fácil darle formato de salida por pantalla o impresión.



Tecnologías XML

XML posee un gran número de tecnologías para dar funcionalidad, presentación o integración con otros lenguajes. Las más importantes son:

DTD. Document Type Definition, definición de tipo de documento. Es un lenguaje que permite especificar documentos cuyas reglas han de cumplir los documentos XML a los que se asocien. Es decir, permite crear documentos de validación para archivos XML.

XML Schema. La función que cumple esta tecnología es la misma que la anterior, la diferencia está en que los documentos XML Schema poseen una sintaxis 100% XML, por lo que es un formato orientado a suplir al anterior.

Relax NG. Otro formato de definición de validaciones para documentos XML. Es una alternativa a las dos anteriores y la que tiene un formato más sencillo.

Namespacing, espacios de nombres. Permite conseguir nombres de elementos que carecen de ambigüedad: es decir nombres únicos dentro de los documentos XML.

XPath. Lenguaje de consulta que permite seleccionar o acceder a partes de un documento XML.

XQuery. Permite consultar datos de los documentos XML, manejándole como si fuera una base de datos.

En principio XML se puede escribir desde cualquier editor de texto plano (como el bloc de notas de Windows o el editor vi de Linux). Pero es más interesante hacerlo con un editor que reconozca el lenguaje y que además marque los errores en el mismo.

De hecho el software necesario es el siguiente:

(1) Un editor de texto plano para escribir el código XML. Bastaría un editor como el bloc de notas de Windows o el clásico vi de Linux; o las opciones de editores capaces de colorear el código como emacs, Notepad++ o SublimeText.

(2) Un analizador sintáctico o parser, programa capaz de entender y validar el lenguaje XML. Apache Xerces es quizá el más popular validador de documentos XML.

(3) Un procesador XML que sea capaz de producir un resultado visual sobre el documento XML. Un simple navegador puede hacer esta función, pero cuando se aplican formatos visuales sobre el documento XML (como los creados mediante XSL) entonces hace falta un software especial que convierta los datos a la forma final visible por el usuario. Apache Xalan y Saxon son los dos procesadores más conocidos

IES MIGUEL ROMERO ESTEO
C/MARTIN CARRIÓN S/N
29006 MALAGA

Teléfono: 951298668
Fax: 951298670

Estos apuntes forman parte del curso "Lenguajes de Marcas y Sistemas de Gestión de la Información" que forma parte del currículum del ciclo superior de "Administración de Sistemas Informáticos en Red".

El temario completo se encuentra en la web del instituto Miguel Romero Esteo.

IES MIGUEL ROMERO ESTEO

www.romeroesteo.es

JOSE RODRIGUEZ—2014