Tema 1: Introducción a los Lenguajes de Marcas

1.Concepto y características generales

Los ***Lenguajes de Marcas*** (también llamados Lenguajes de Marcado) son los que ***combinan la información de un documento con marcas o anotaciones que indican la estructura o representación*** de esa información.

El lenguaje especifica las etiquetas posibles, dónde se colocan y qué significado tienen. Las etiquetas o marcas generalmente no serán visibles al usuario final, no es de su interés.

A diferencia de los lenguajes de programación no tienen funciones aritméticas ni variables.

En un documento de lenguajes de marcas encontramos: ***etiquetas***, ***elementos*** y ***atributos***.

Las ***etiquetas*** son la base de los lenguajes de marcas. Se trata de palabras concretas definidas por el lenguaje que indican cómo debe tratarse o interpretarse la información que le sigue. Las etiquetas suelen señalarse con paréntesis, corchetes, etc para distinguirlas del resto del texto. A menudo aparecen por parejas, etiqueta de inicio y etiqueta de fin.

La información que deba contener un documento escrito con un lenguaje de marcas se intercala a lo largo de todo el código entre etiquetas.

Los ***elementos*** representan las estructuras en las que se organiza el documento. Constan de etiqueta de inicio, etiqueta de fin y todo lo que haya entre ambas (***contenido***). Los elementos sin contenido se llaman ***elementos vacíos***.

Los ***atributos*** son parejas nombre-valor que se encuentran dentro de la etiqueta de inicio de un elemento e indican propiedades asociadas a los elementos.

```
<ciudadano>
      <nombre>José</nombre>
      <apellido>Saramago</apellido>
      <identidad tipo="NIF">X0000000E</identidad>
</ciudadano>
```

2. Tipos de lenguajes de marcas

Se suele diferenciar entre tres clases de lenguajes de marcado, aunque en la práctica pueden combinarse varias clases en un mismo documento.

2.1. Lenguajes de presentación

El marcado de presentación es aquel que indica el formato del texto. Este tipo de marcado es útil para maquetar la presentación de un documento para su lectura, pero resulta insuficiente para el procesamiento automático de la información. El marcado de presentación resulta más fácil de elaborar, sobre todo para cantidades pequeñas de información. Sin embargo resulta complicado de mantener o modificar, por lo que su uso se ha ido reduciendo en proyectos grandes en favor de otros tipos de marcado más estructurados.

Un ejemplo sería indicar mediante marcas que cierta palabra deber representarse en negrita o en cursiva, pero un lenguaje de presentación no indicaría el motivo por el que se hace esto.

Los lenguajes de presentación se han usado tradicionalmente por los procesadores de textos, indicando las características del texto (Wordstar o WordPerfect) hasta la aparición de los ***WYSIWYG***[1]. Estos lenguajes no suelen ser flexibles ni reusables.

2.2. Lenguajes procedimentales

También orientados a la presentación, pero permiten definir macros (secuencias de acciones).

Algunos ejemplos de marcado de procedimientos son ***nroff***, ***troff***, ***TeX***, ***LaTeX*** (muy usado cuando hay fórmulas matemáticas), ***Postcript*** (es un PDL -Page Description Language-, lenguaje de descripción de páginas, usado por muchas impresoras). Este tipo de marcado se ha usado extensivamente en aplicaciones de edición profesional, manipulados por tipógrafos calificados, ya que puede llegar a ser extremadamente complejo.

2.3. Lenguajes descriptivos

El marcado descriptivo o semántico utiliza etiquetas para describir los fragmentos de texto, pero sin especificar cómo deben ser representados, o en que orden. Los lenguajes expresamente diseñados para generar marcado descriptivo son el SGML y el XML.

En estos lenguajes, las marcas indican algo sobre la información, meta-información. Por ejemplo, estos lenguajes pueden indicar que cierta palabra es el título del documento o el nombre del autor, sin indicar cómo habrá de presentarse, ni siquiera si se debe representar.

1 WYSIWYG: What You See Is What You Get (Lo que ves es lo que obtienes)

Una de las virtudes del marcado descriptivo es su flexibilidad: los fragmentos de texto se etiquetan tal como son, y no tal como deben aparecer. Estos fragmentos pueden utilizarse para más usos de los previstos inicialmente. Por ejemplo, los hiperenlaces fueron diseñados en un principio para que un usuario que lee el texto los pulse. Sin embargo, los buscadores los emplean para localizar nuevas páginas con información relacionada, o para evaluar la popularidad de determinado sitio web.

El marcado descriptivo también simplifica la tarea de reformatear un texto, debido a que la información del formato está separada del propio contenido. Por ejemplo, un fragmento indicado como cursiva (texto), puede emplearse para marcar énfasis o bien para señalar palabras en otro idioma. Esta ambigüedad, presente en el marcado presentacional y en el procedimental, no puede soslayarse más que con una tediosa revisión a mano. Sin embargo, si ambos casos se hubieran diferenciado descriptivamente con etiquetas distintas, podrían representarse de manera diferente sin esfuerzo.

El marcado descriptivo está evolucionando hacia el marcado genérico. Los nuevos sistemas de marcado descriptivo estructuran los documentos en árbol, con la posibilidad de añadir referencias cruzadas. Esto permite tratarlos como bases de datos, en las que el propio almacenamiento tiene en cuenta la estructura, no como en los grandes objetos binarios como en el pasado. Estos sistemas no tienen un esquema estricto como las bases relacionales, por lo que a menudo se las considera bases semi-estructuradas.

3. Mapa de los lenguajes de marcas

Esta es una lista de los principales lenguajes de marcas ordenados por su campo de aplicación. Para ver una lista más completa puedes consultar "Lenguajes de descripción" en ***Wikipedia***.

3.1. Principales

GML --> SGML --> XML --> Dialectos XML

3.2. Documentos en general

Lenguajes descriptivos	**Lenguajes de presentación**	**Lenguajes ligeros**	**Lenguajes para manuales**
ASN.1 EBML YAML	Rich Text Format S1000D TeX HTML	BBCode Markdown ReStructuredText setext Textile Wikitexto	DocBook HelpML LinuxDoc POD Microsoft Assistance ML

3.3. Tecnologías de internet

World Wide Web	**Interfaz de usuario**	**Sindicación**	**Servicios web**
HTML	GladeXML	Atom	WSDL
XHTML	MXML (Macromedia)	RSS	XINS
Wireless ML	User Interface ML	ICE	WSCL
Handhelp ML	XAML and MyXaml	OPML y OML	WSFL
RDF	XForms	SyncML	XML-RPC
Meta Content Framework	XUL / XBL		Webml

3.4. Lenguajes especializados

- Gráficos 2D: SVG, CGM, VML, InkML.
- Gráficos 3D: VRML/X3D, STEP.
- Matemática: MathML y OpenMath.
- Música: LilyPond y MusicXML.
- Taxonomía: DITA
- Finanzas: eXtensible Bussiness Reporting Language, Financial products ML.
- Geomática: Geography ML.
- Aeronáutica: Spacecraft ML.
- Multimedia: Synchronized Multimedia Integration Language.
- Voz: VoiceXML.
- Mensajería instantánea: XMPP.
- Videojuegos: BulletML, COLLADA.

4. Historia

4.1. Orígenes

Los lenguajes de marcas se llaman así por la práctica tradicional de marcar los manuscritos con instrucciones de impresión en los márgenes. En la época de la imprenta, esta tarea ha correspondido a los marcadores, que indicaban el tipo de letra, el estilo y el tamaño, así como la corrección de errores, para que otras personas compusieran la tipografía. Esto condujo a la creación de un grupo de marcas estandarizadas. Con la introducción de las computadoras, se trasladó un concepto similar al mundo de la informática.

El concepto de lenguaje de marcas fue expuesto por vez primera por ***William W. Tunnicliffe*** en 1967, que prefería referirse a este concepto como ***codificación genérica*** (generic coding). La mayor novedad consistía en la separación entre la presentación y la estructura del texto. Tunnicliffe, dirigiría más tarde el desarrollo de un estándar al que bautizaría como ***GenCode***, destinado a la industria editorial. El editor Stanley Fish también expuso ideas similares a finales de los años 1960. Brian Reid, en su disertación de 1980 en la Carnegie Mellon University, mostró su teoría y una implementación práctica de un lenguaje descriptivo todavía en uso.

Sin embargo, quien es considerado el padre de los lenguajes de marcas es ***Charles Goldfarb***, investigador para la compañía IBM. Goldfarb participó en la creación del lenguaje ***GML***, y posteriormente dirigió el comité que elaboró el estándar ***SGML***, la piedra angular de los lenguajes de marcas. En cualquier caso, y a pesar de las controversias sobre su origen, es comúnmente aceptado que la idea surgió de forma independiente varias veces durante los 70, y que se generalizó en los 80.

4.2. La generalización de los lenguajes de marcas

La iniciativa que sentaría las bases de los actuales lenguajes, partiría de la empresa IBM, que buscaba nuevas soluciones para mantener grandes cantidades de documentos. El trabajo fue encomendado a Charles F. Goldfarb, que junto con Edward Mosher y Raymond Lorie, diseñó el Generalized Markup Language o ***GML*** (nótese que también son las iniciales de sus creadores). Este lenguaje heredó del proyecto GenCode la idea de que la presentación debe separarse del contenido. El marcado, por tanto, se centra en definir la estructura del texto y no su presentación visual.

El lenguaje GML fue un gran éxito y pronto se extendió a otros ámbitos, siendo adoptado por el gobierno de Estados Unidos, con lo que surgió la necesidad de estandarizarlo. En los primeros años 1980 se constituyó un comité dirigido por Goldfarb. Sharon Adler, Anders Berglund, y James D. Mason fueron también miembros de dicho comité. Se incorporaron ideas de diferentes fuentes, y participó gran cantidad de gente. Tras un largo proceso, en 1986 la Organización Internacional para la Estandarización publicaría el ***SGML*** Standard Generalized Markup Language con rango de Estándar Internacional con el código ISO 8879.

El SGML especifica la sintaxis para la inclusión de marcas en los textos, así como la sintaxis del documento que especifica qué etiquetas están permitidas y donde: el Document Type Definition o schema. Esto permitía que un autor emplease cualquier marca que quisiera, eligiendo nombres para las etiquetas que tuvieran sentido tanto por el tema del documento como por el idioma. Así, el SGML es, estrictamente hablando, un metalenguaje, del que se derivan varios lenguajes especializados. Desde finales de los 80 han aparecido nuevos lenguajes basados en SGML, como por ejemplo el TEI o el DocBook.

El SGML tuvo una gran aceptación y hoy día se emplea en campos en los que se requiere documentación a gran escala. A pesar de ello, resultó farragoso y difícil de aprender, como consecuencia de la ambición de los objetivos previstos. Su gran potencia era a la vez una ventaja y una desventaja. Por ejemplo, ciertas etiquetas podían tener sólo principio, o sólo final, o incluso ser obviadas, pensando en que los textos serían redactados a mano y que así se ahorrarían pulsaciones de teclas. Sin embargo fue un punto clave en el desarrollo de los lenguajes de marcas actuales, ya que la gran mayoría derivan de este.

4.3. La popularización: el HTML

En 1991, parecía que los editores WYSIWYG (que almacenan los documentos en formatos binarios propietarios) abarcarían casi la totalidad del procesamiento de textos, relegando al SGML a usos profesionales o industriales muy específicos. Sin embargo, la situación cambió drásticamente cuando Sir ***Tim Berners-Lee***, que había aprendido SGML de su compañero en el CERN (siglas en francés para el Consejo Europeo para la Investigación Nuclear) Anders Berglund, utilizó la sintaxis SGML para crear el ***HTML***.

Este lenguaje era similar a cualquier otro creado a partir del SGML, sin embargo resultó extraordinariamente sencillo. La flexibilidad y escalabilidad del marcado HTML fue uno de los principales factores, junto con el empleo de URLs y la distribución libre de navegadores, del éxito de la World Wide Web.

El HTML es hoy día el tipo de documento más empleado en el mundo. Su sencillez era tal que cualquier persona podía escribir documentos en este formato, sin apenas necesidad de conocimientos de informática. Esta fue una de las razones de su éxito, pero también condujo a un cierto caos. El crecimiento exponencial de la web en los años 90 produjo documentos en cantidades ingentes pero mal estructurados, problema agravado aún más por la falta de respeto por los estándares, por parte de diseñadores web y fabricantes de software.

A mediados de los 90 el consorcio ***W3C*** (World Wide Web Consortium) comenzó una iniciativa para dotar a la web de un lenguaje potente que incluyera estructura semántica. En 1998 nace el ***XML*** (eXtended Markup Language), más sencillo que SGML y más potente que HTML.

4.4. La madurez: el XML

La respuesta a los problemas surgidos en torno al HTML vino de la mano del XML (eXtensible Markup Language). El XML es un meta-lenguaje que permite crear etiquetas adaptadas a las necesidades (de ahí lo de "extensible"). El estándar define cómo pueden ser esas etiquetas y qué se puede hacer con ellas. Es además especialmente estricto en cuanto a lo que está permitido y lo que no, todo documento debe cumplir dos condiciones: ser válido y estar bien formado.

El XML fue desarrollado por el World Wide Web Consortium, mediante un comité creado y dirigido por Jon Bosak. El objetivo principal era simplificar el SGML para adaptarlo a un campo muy preciso: documentos en internet.

El nuevo lenguaje se extendió con rapidez, ya que todo documento XML es a su vez SGML. Los programas y documentos creados para y con SGML podían convertirse casi automáticamente al nuevo lenguaje. El XML simplificó radicalmente la complejidad del SGML, facilitando el aprendizaje y la implementación del nuevo estándar. Se solucionaron además viejos problemas, como los surgidos de la internacionalización, y la imposibilidad de validar un documento sin schema. El acierto fundamental de este lenguaje está en que logra un equilibrio entre simplicidad y flexibilidad.

El XML fue ideado en principio para entornos semi-estructurados, como textos y publicaciones. Uno de los ejemplos más claros es el ***XHTML***, la redefinición del HTML en clave XML, con las ventajas que ello supone. Sin embargo pronto se observó que sus virtudes podían ser útiles en campos bien distintos. Los lenguajes basados en XML tienen aplicaciones incontables,

como en la transacción de datos entre servidores, intercambio de información financiera, fórmulas y reacciones químicas, y un largo etcétera.

4.5. Tendencias

Las nuevas tendencias están abandonando los documentos con estructura en árbol. Los textos de la literatura antigua suelen tener estructura de prosa o de poesía: versículos, párrafos, etc. Los documentos de referencia suelen organizarse en libros, capítulos, versos y líneas. A menudo se entremezclan unos con otros, por lo que la estructura en árbol no se ajusta a sus necesidades. Los nuevos sistemas de modelado superan estos inconvenientes, como el MECS, diseñado para la obra de Wittgenstein, o las TEI Guidelines, LMNL, y CLIX.

La Iniciativa de codificación de textos o Text Encoding Initiative (TEI) ha publicado multitud de guías para la codificación de documentos de interés en humanidades y ciencias sociales, desarrollados durante años de trabajo colaborativo internacional. Estas directrices se han empleado en innumerables proyectos de catalogación de documentos históricos, trabajos académicos, etc.

4.6. La web semántica

Los lenguajes de marcas son la herramienta fundamental en el diseño de la web semántica, aquella que no solo permite acceder a la información, sino que además define su significado, de forma que sea más fácil su procesamiento automático y se pueda reutilizar para distintas aplicaciones. Esto se consigue añadiendo datos adicionales a los documentos, por medio de dos lenguajes expresamente creados: el RDF (Resource Description Framework-Plataforma de descripción de recursos) y OWL (Web Ontology Language-Lenguaje de ontologías para la web), ambos basados en XML.

En resumen:

- La idea de introducir un marcado en un documento electrónico viene heredada de la corrección manual de manuscritos
- En la década de los 60' se empieza a desarrollar la idea de separar presentación y estructura.
- Desde IBM se impulsa la creación del lenguaje GML, que resultó ser la semilla de una versión posterior estandarizada: SGML.
- La potencia de SGML implica una dificultad en su aprendizaje y uso.
- El HTML se crea a partir del SGML.
- XML surge como respuesta al desorden que supuso el rápido crecimiento del HTML.

5. Organizaciones desarrolladoras

ISO (International Organization for Standarization, Organización Internacional para la Estandarización). La ISO establece normas internacionales de fabricación, comercio y comunicación para todas las ramas (salvo la eléctrica y electrónica). Agrupa los institutos de normas nacionales de 163 países. Sus normas son voluntarias y para acceder a ellas hay que comprarlas.

Esta organización es la que publicó en 1986 el SGML (código ISO 8879)

W3C (World Wide Web Consortium) creado en 1994 por Tim Berners-Lee. Su función principal es tutelar el crecimiento y organización de la web. Su primer trabajo fue normalizar el HTML, lenguaje de marcas con el que se escriben las páginas web. En 1998, publicó el XML.

6. Conceptos Básicos

Texto: Llamamos texto a cualquier combinación de caracteres que puede incluir imágenes, tablas, etc

Hipertexto: Texto que incluye hiperenlaces.

Hiperenlace: Parte interactiva de un texto que permite desplazarse hasta otro documento o hasta otra parte concreta de este documento. Se suelen usar indistintamente los términos enlace, hiperenlace, vínculo, hipervínculo, link o hyperlink, si bien estos dos últimos son términos del inglés.

HTTP: (Hyper Text Transfer Protocol) Protocolo de Transferencia para Hipertexto. Es el protocolo (conjunto de reglas que han de conocer emisor y receptor) para la transmisión de hipertexto entre dos equipos informáticos.

HTML: (HyperText Markup Language) Lenguaje de Marcado para Hipertexto. Es un lenguaje que se utiliza para describir documentos de hipertexto.

Fichero de texto: (o texto plano) Fichero que almacena una secuencia ordenada de caracteres sin ningún formato. Además de caracteres imprimibles, sólo puede incluir algunos caracteres de control, como el de retorno de carro o el que señala el final del fichero.

El código de los lenguajes de marcas se escribe en texto plano. Esto facilita la interpretación del código por parte de un posible lector humano y supone una ventaja evidente respecto a los sistemas de archivos binarios, que requieren siempre de un programa intermediario para trabajar con ellos. Un documento escrito con lenguajes de marcas puede ser editado por un usuario con un sencillo editor de textos (como ***gedit*** en Linux o el ***Bloc de Notas*** de Windows), sin perjuicio de que se puedan utilizar programas más sofisticados que faciliten el trabajo.

Al tratarse solamente de texto, los documentos son independientes de la plataforma, sistema operativo o programa con el que fueron creados. Esta fue una de las premisas de los creadores de GML en lo años 70, para no añadir restricciones innecesarias al intercambio de información. Es una de las razones fundamentales de la gran aceptación que han tenido en el pasado y del excelente futuro que se les augura.

Editor de textos: Aplicación informática que permite crear y editar ficheros de texto. Ejemplos: **gedit** para Linux, o el **Bloc de Notas** incluido en Windows. No confundir con un procesador de textos.

Navegador o Cliente web: Aplicación que permite visualizar un documento de hipertexto.

Agente de usuario: Aplicación que permite representar un documento descrito con HTML. Un navegador sería un agente de usuario, pero este concepto es mucho más amplio e incluye aplicaciones que representen el documento en sistema Braille, mediante voz, etc

7. El lenguaje SGML

SGML (Standard Generalized Markup Language, 1986): es un metalenguaje que permite definir lenguajes de marcado.

- Especifica la sintaxis para la inclusión de marcas en los textos, así como la sintaxis del documento que especifica qué etiquetas están permitidas y dónde: el Document Type Definition.
- La definición de la estructura y el contenido de un tipo de documento se realiza por medio de su DTD (Document Type Definition)

Ventajas de SGML:

- Reutilización de los datos
- Integridad y mayor control sobre los datos
- Portable
- Flexible
- Perdurabilidad de la información

Inconvenientes de SGML:

- Alta complejidad

Ejemplo de SGML

```
<EMail>
    <remitente>
        <persona>
            <nombre> Karen </nombre>
            <apellido> Lemone </apellido>
        </persona>
    </remitente>
    <destinatario>
        <persona>
            <lista_distribution> cs525@cs.com </lista_distribution>
        </persona>
    </destinatario>
    <contenido>¿no es sencillo?</contenido>
</Email>
```

8. Introducción a HTML

HTML: lenguaje de marcado definido en SGML

- Origen: 1989 en el Laboratorio Europeo de Física de Partículas (CERN)
- Objetivo inicial: presentar información estática.

Jugó un papel fundamental en el crecimiento de Internet.

Presenta limitaciones relacionadas con:

- Tratamiento de información dinámica.
- No es un metalenguaje, por lo que dispone de un número fijo de etiquetas.
- Su vocabulario es muy limitado.

Ventajas de HTML:

- Es muy simple y sencillo de aprender y usar.
- No requiere herramientas especiales.
- Está muy difundido.

Inconvenientes de HTML:

- Carecer de chequeo sintáctico.
- Carecer de estructura lógica.
- Estar orientado fundamentalmente a la representación de los datos y no a su estructura.
- Carecer de una semántica estándar.
- No ser adecuado para el intercambio de datos.
- No ser extensible.
- No permitir la reutilización de la información.

Ejemplo de HTML

```
<html>
    <head>
        <meta http-equiv="content-type" content="text/html" charset="ISO-
8859-1"/>
        <meta name="generator" content="Adobe GoLive 5"/>
        <title>Archivo L&eacute;ame de Adobe Acrobat para Windows</title>
    </head>
    <body bgcolor="#ffffff">
        <p></p>
        <p>
        <b><font size="+1">21 de septiembre de 2019</font></b>
        </p>
        <p><b><font size="+2">Archivo L&eacute;ame de Adobe Acrobat para
        Windows</font></b></p>
        <p>Bienvenido al archivo L&eacute;ame de Adobe&reg; Acrobat&reg;
5.0.5. Si lo desea, puede acceder al <a href="http://www.adobe.com/
supportservice/">soporte t&eacute;cnico</a><br/> </p>
        <p> Este archivo está dividido en los siguientes apartados:<br/>
        </p>
    </body>
</html>
```

9. Introducción a XML

XML (Extensible Markup Language): es una forma restringida de SGML optimizada para su utilización en Internet. Tiene su origen: 1996 World Wide Web Consorium (W3C)

Objetivos iniciales:

- Lenguaje estructurado, extensible y que se pueda validar.
- Permite la transmisión de información realmente estructurada.

Características de XML:

- Es un subconjunto de SGML (toma el 80% de sus ventajas y le resta el 20% de complejidad).
- Es simple de usar y se basa en etiquetas de texto.
- Es una tecnología madura puesto que se basa en SGML.
- Soporta Unicode.
- Se orienta a los datos, su semántica y no a la representación.
- Se está convirtiendo en el lenguaje de Bases de Datos de la Web.
- Permite un fácil intercambio de información entre aplicaciones.
- Al tratarse de un metalenguaje tiene un vocabulario extensible:
  - Permite definir lenguajes de marcado por medio de DTD's (Document Type Definition) o de XML-Schemas
  - Sirve para representar datos estructurados en un fichero de texto.
  - Usa etiquetas para delimitar los datos pero deja su interpretación a la aplicación que lee el código XML.

Ventajas:

– Numerosas tecnologías asociadas:

- XML (Estructura de los datos)
- XSL= XSLT+XSL-FO's + XPath (hojas de estilo)
- XLL = XLink + XPointer+ Xpath (hiperenlaces)
- XQL (consultas a bases de datos)
- DOM (Document Object Model)
- SAX (Simple Api for XML)
- …

Ejemplo de documento XML:

Archivo ejemplo.xml:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE Edit_Mensaje SYSTEM "Edit_Mensaje.dtd">

<!-- Ejemplo de un documento XML -->
<Edit_Mensaje>
    <Mensaje>
        <Remitente>
            <Nombre>Nombre del remitente</Nombre>
            <Mail> Correo del remitente </Mail>
        </Remitente>
        <Destinatario>
            <Nombre>Nombre del destinatario</Nombre>
            <Mail>Correo del destinatario</Mail>
        </Destinatario>
        <Texto>
            <Asunto>
                Este es mi documento con una estructura muy sencilla
                no contiene atributos ni entidades...
            </Asunto>
            <Parrafo>
                Este es mi documento con una estructura muy sencilla
                no contiene atributos ni entidades...
            </Parrafo>
        </Texto>
    </Mensaje>
</Edit_Mensaje>
```

Como podemos observar el ejemplo es un archivo xml en el que se almacena la información de un correo electrónico.

10. Conceptos previos interesantes

Unicode : el alfabeto de los documentos XML.

El Estándar **Unicode** es un estándar de codificación de caracteres diseñado para facilitar el tratamiento informático, transmisión y visualización de textos de múltiples lenguajes y disciplinas técnicas además de textos clásicos de lenguas muertas. El término Unicode proviene de los tres objetivos perseguidos: universalidad, uniformidad y unicidad.

UNICODE PROPORCIONA UN NÚMERO ÚNICO A CADA CARÁCTER INDEPENDIENTEMENTE DE LA PLATAFORMA, EL PROGRAMA O EL IDIOMA.

Unicode especifica un nombre e identificador numérico único para cada carácter o símbolo, el *code point* o punto de código, además de otras informaciones necesarias para su uso correcto: direccionalidad, capitalización y otros atributos. Unicode trata los caracteres alfabéticos, ideográficos y símbolos de forma equivalente, lo que significa que se pueden mezclar en un mismo texto sin la introducción de marcas o caracteres de control.

Este estándar es mantenido por el Unicode Technical Committee (UTC), integrado en el Unicode Consortium, del que forman parte con distinto grado de implicación empresas como: Microsoft, Apple, Adobe, IBM, Oracle, SAP, Google, instituciones como la Universidad de Berkeley, y profesionales y académicos a título individual. El Unicode Consortium mantiene estrecha relación con ISO/IEC, con la que mantiene un acuerdo desde 1991 con el objetivo de mantener la sincronización entre sus estándares que contienen los mismos caracteres y puntos de código.

Formas de codificación

Unicode define tres formas de codificación bajo el nombre UTF o Formato de Transformación Unicode (Unicode Transformation Format):[8]

UTF-8: codificación orientada a byte con símbolos de longitud variable. Aquí dejo el siguiente enlace donde podrás conseguir rápida y cómodamente el código de cualquier carácter Unicode.

UTF-16: codificación de 16 bits de longitud variable optimizada para la representación del plano básico multilingüe (BMP).

UTF-32: codificación de 32 bits de longitud fija, y la más sencilla de las tres.

Las formas de codificación se limitan a describir el modo en que se representan los puntos de código en formato inteligible por la máquina.

El uso de Unicode es un requisito impuesto a XML para la codificación de los caracteres de sus textos. Este sistema de codificación nació con el objetivo, perfectamente alcanzado, de poder procesar la totalidad de los caracteres de los lenguajes actualmente existentes en el mundo. Al poner esta exigencia XML aseguraba la universalidad buscada para la Web.

Unicode supera situaciones propias de determinados sistemas de codificación, como por ejemplo que el conjunto ISO88597; incluya letras griegas, mientras que el conjunto ISO88591 no lo hace.

Señalemos que en XML puede usarse cualquier carácter de Unicode, a excepción de los que tienen algún significado especial para el lenguaje y que son concretamente : & (utilizado para dar un texto que reemplaza a una entidad), < , > , (usadas para las etiquetas) ' , “ (reservados para delimitar valores de atributos) para cuyo uso, como veremos, XML usa un mecanismo basado en la referencia de entidades.