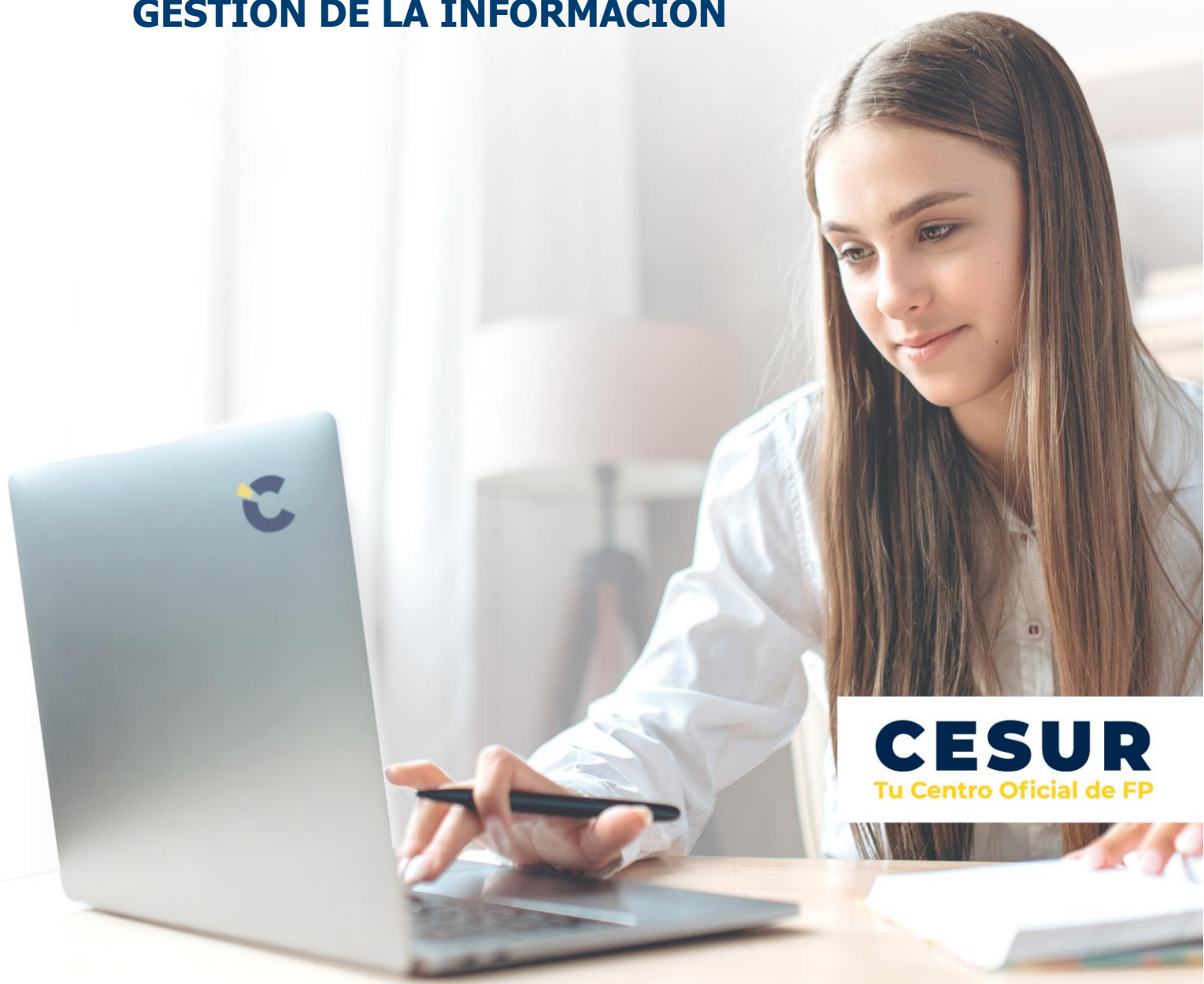


UNIDAD DIDÁCTICA 1

RECONOCIMIENTO DE LAS CARACTERÍSTICAS DE LENGUAJES DE MARCAS

**MÓDULO PROFESIONAL:
LENGUAJES DE MARCAS Y SISTEMAS DE
GESTIÓN DE LA INFORMACIÓN**



CESUR
Tu Centro Oficial de FP

Índice

RESUMEN INTRODUCTORIO	2
INTRODUCCIÓN	2
CASO INTRODUCTORIO	2
1. CLASIFICACIÓN.....	3
2. CARACTERÍSTICAS Y ÁMBITOS DE APLICACIÓN	7
3. ESTRUCTURA Y SINTAXIS	9
3.1 Estructura.....	9
3.2 Sintaxis	12
3.3 Estructura y sintaxis en XML.....	15
4. HERRAMIENTAS DE EDICIÓN	16
5. ELABORACIÓN DE DOCUMENTOS BIEN FORMADOS	18
6. UTILIZACIÓN DE ESPACIOS EN NOMBRES	23
RESUMEN FINAL	26

RESUMEN INTRODUCTORIO

En la presente unidad didáctica se estudiarán las características de lenguajes de marcas, desde su clasificación y características, hasta las herramientas de edición y cómo es su estructura y sintaxis, para permitir la elaboración de documentos bien formados.

INTRODUCCIÓN

Los lenguajes de marcas son lenguajes fundamentales para estructurar todo tipo de información. En particular, el lenguaje HTML, es un lenguaje de marcado destinado a la elaboración de páginas web.

Este lenguaje, está compuesto por etiquetas, por lo que para poder crear páginas web, será necesario estudiar cada una de estas etiquetas, así como la forma de estructurarlas. Asimismo, cada una de estas etiquetas pueden personalizar su comportamiento mediante atributos, lo cual dotará de mayor potencia al lenguaje.

La importancia de crear documentos bien formados está directamente relacionada con la creación de sitios correctos tanto en estructura como en la información que van a ofrecer al usuario.

CASO INTRODUCTORIO

La empresa en la que trabajas centra su actividad en la creación de páginas y sitios web en general, y, sobre todo, a empresas que demandan de sus servicios. Como responsable del departamento de desarrollo web, ante cualquier solicitud de vuestros servicios, debes comenzar con el análisis previo de las necesidades planteadas por los clientes, y de los recursos e información que aportan para la creación de la página o sitio web corporativa.

Al finalizar el estudio de la unidad sabrás reconocer las características del lenguaje de marcas.

1. CLASIFICACIÓN

En tu empresa han recibido el encargo de realizar una página web para un cliente externo, para garantizar el uso de un lenguaje que cumpla con las necesidades planteadas y te han pedido que presentes una clasificación de los distintos lenguajes de marcas que existen en el mercado.

Los lenguajes de marcas nacen de la necesidad de aportar información sobre el contenido de un documento. Por ejemplo, si se tiene una lista de personas y se quiere saber cuáles son los nombres y apellidos, es necesario indicar de alguna forma qué parte de cada elemento de la lista se corresponde con el nombre y qué parte con cada apellido.



Lenguaje de marcas.

Fuente: <https://www.ticarte.com/contenido/que-son-los-lenguajes-de-marcas>

Para conseguirlo, se pueden emplear diversas formas de codificación de dicho documento. Una de ellas es utilizando etiquetas o marcas que añadan esa información extra que necesitamos. El lenguaje definido por estas etiquetas es lo que se denomina lenguaje de marcas.

Por supuesto, al igual que sucede con el lenguaje usado para comunicarse en cualquier idioma, será necesario establecer una serie de reglas que le den validez. No obstante, la sintaxis será mucho más sencilla que la de un idioma.

Los lenguajes de marcas más extendidos son HTML y XML. En el caso de HTML, se trata de un lenguaje de marcas orientado a la creación de páginas web. Por ello, su función fundamental es la de crear la estructura de una página y dar formato a los elementos que la componen. Además, en muchos casos añade información semántica, es decir, de significado.

HTML se creó en un principio con objetivos divulgativos. No se imaginó que la web llegaría a ser una red de ocio con carácter multimedia, de modo que, a lo largo del tiempo ha tenido que ir ampliándose en sucesivas actualizaciones para ir dando respuesta a todas las nuevas necesidades, aunque ha seguido manteniendo su estructura básica.

Dentro de la amplia oferta de lenguaje de marcas destacan algunos por sus funcionalidades y objetivos como la representación de información, serían, por ejemplo, HTML y XHTML. Encontrando otros tipos como XML más orientado al almacenamiento e intercambio de información, con una amplia variedad de dialectos o digamos variantes de los mismos.

Destacarían los siguientes:

HTML.

Lenguaje muy extendido, que tiene por objetivo ofrecer un lenguaje para diseño de páginas web que puedan ser interpretadas por un navegador. Su mantenimiento y estandarización corresponde al W3C, y, siendo actualmente su versión HTML5 la más utilizada. Cuenta con ventajas la gran cantidad de elementos de maquetación que ofrece, así como la perfecta integración de hojas de estilo en cascada (CSS).

XML.

Si existe un lenguaje de intercambio de información que destaca sobre el resto, este es XML (Extensible Markup Language). Definido como un metalenguaje por venir definido con reglas, en lugar de etiquetas y bajo la especificación del W3C que lo define como un lenguaje de propósito general. Permite la definición de dialectos de un modo específico compatibles con una gran variedad y tipos de aplicaciones.

Encontramos dialectos de XML muy interesantes como:

- SVG (Scalable Vector Graphics), para gráficos vectoriales de dos dimensiones.
- MathML (Mathematical Markup Language), para representar notaciones matemáticas.
- NewsML (News Markup Language), para representar notaciones electrónicas.
- FXML (FX Markup Language), para representar interfaces de usuario en JavaFX.
- XAML (eXtensible Application Markup Language), para definir las interfaces de usuario en aplicaciones Windows.
- RSS y Atom, para distribución de contenidos de fuentes a las que se han suscrito los usuarios con anterioridad.

XHTML.

Suele definirse como la versión XML de HTML, siguiendo las recomendaciones del W3C, surge a partir de un metalenguaje con el objetivo de constituirse en referencia en el desarrollo de páginas web, de un modo un poco continuista desde HTML, pero aportando una sintaxis más estricta y formal, lo que garantiza su validación y sencillez en el procesamiento.

JSON.

Inicialmente era el formato de texto con el que JavaScript representa en sus documentos los objetos, y el acrónimo de JavaScript Object Notacion, pero se ha ido desligando de JavaScript con el objetivo de convertirse en una alternativa a XML en lo que respecta al almacenamiento e intercambio de datos entre sistemas, sin que exista una dependencia ni de plataforma ni de lenguaje de programación.



RECUERDA

Dentro de un contexto de la programación web, el lenguaje de marcas hace referencia a la manera de codificar un documento o página web.



EJEMPLO PRÁCTICO

Ángel es el responsable dentro del departamento de informática de su empresa, de la implantación de nuevo software tanto a nivel aplicaciones como sistemas operativos, pero en este caso están planteando el cambio de la página web corporativa.

Dentro de esa nueva política de comunicación web con un sitio totalmente renovado, le has solicitado que realice una investigación sobre qué tipos de lenguajes de marcas existen para seleccionar el más adecuado en el diseño y programación del nuevo sitio web.

¿Qué lenguajes serían los que podría proponer?

Solución.

En la actualidad destacaría los siguientes:

HTML

Lenguaje muy extendido, que tiene por objetivo ofrecer un lenguaje para diseño de páginas web que puedan ser interpretadas por un navegador.

XML

Si existe un lenguaje de intercambio de información que destaca sobre el resto, este es XML (Extensible Markup Language).

XHTML

Suele definirse como la versión XML de HTML.

JSON

Inicialmente era el formato de texto con el que JavaScript representa en sus documentos los objetos, y el acrónimo de JavaScript Object Notacion, pero se ha ido desligando de JavaScript con el objetivo de convertirse en una alternativa a XML.

2. CARACTERÍSTICAS Y ÁMBITOS DE APLICACIÓN

Dando continuidad a tu labor de clasificación de los lenguajes de marcas, es necesario que amplíes la información sobre los mismos, destacando sus características y los distintos ámbitos de aplicación en los que se emplean lenguajes de marcas específicos.

La gran variedad de lenguajes de marcas, hacen que también sus características y ámbitos de aplicación sean muy numerosos, no existiendo una estructura única o común a todos ellos, por ello, se ha adoptado por norma que, al hacer referencia a un lenguaje de marcas es en el sentido de la información añadida por medio de marcas a los datos del propio documento.

Características generales.

1. Flexibilidad.
Suelen ser lenguajes que permiten su utilización en contextos muy variados.
2. Independencia.
No están sujetos a ningún software, hardware ni plataforma concreta.
3. Compatibilidad.
Este tipo de lenguajes recoge en el mismo documento las marcas y el contenido.
4. Sencillo procesamiento.
Su estructura debe permitir realizar un procesamiento automático de una manera sencilla.
5. Almacenamiento en texto plano.
El documento generado únicamente contiene caracteres de texto.

Características específicas.

Podemos ver una aproximación de las características más enfocadas en un lenguaje concreto como puede ser XML, por ser uno de los más destacados dentro del lenguaje de marcas, donde destacarían las siguientes:

- Todo elemento debe ir precedido de su correspondiente etiqueta de apertura, con el formato <nombre>, donde nombre será el correspondiente al elemento.
- Igualmente, cada elemento llevará su etiqueta de cierre con el formato </nombre>.

- Los nombres de etiquetas y propiedades no están predefinidos.
- Es un lenguaje flexible que permite decidir cuál es el nombre, orden y estructura de las etiquetas.
- Los elementos pueden contener texto, o incluso otro elemento o varios elementos.
- Existirá únicamente un elemento raíz.
- Si se emplea un atributo, éste irá compuesto por su propio nombre, seguido del signo =, para continuar con su valor entrecomillado.
- Los atributos de un elemento pueden ser desde 0 a n.

Ámbitos de aplicación.

Ante la diversidad de lenguajes, y sobre todo dialectos de XML, encontramos, entre otros, los siguientes ámbitos de aplicación del lenguaje de marcas:

- HTML Y XHTML: Creación de páginas web.
- RSS: Difusión de contenidos web.
- MathML y OpenMath: Notaciones matemáticas.
- TeX: Ecuaciones matemáticas complejas.
- DocBook: Generación de documentos separando del mismo la estructura lógica.
- Geography ML: Modelaje, transporte y almacenamiento de información geográfica.
- SMIL: Integración de audio, video, imágenes y cualquier contenido multimedia.
- VoiceXML: Capacidad de intercambio de información con reconocimiento de voz.
- MusicXML: Intercambio de partituras entre editores de partituras.
- Spacecraft ML: Aeronáutica.
- XMPP: Mensajería instantánea.
- VRML/X3D, STEP: Gráficos 3D.



VÍDEO DE INTERÉS

Visualiza una breve introducción al lenguaje de marcas, sus conceptos y ventajas.



3. ESTRUCTURA Y SINTAXIS

En tu departamento de informática de la empresa en la que trabajas, existen diferentes perfiles de programadores informáticos, como responsable del departamento, serás el encargado de confeccionar una guía en la que se especifique la estructura y sintaxis del lenguaje de marcas.

3.1 Estructura

Las marcas son señales que se colocan entre el texto, añadiéndole significado. Habitualmente, estas marcas o etiquetas consisten en un nombre encerrado entre signos de menor y mayor. Además, suelen aparecer por parejas, de manera que la primera abre un contexto y la segunda lo cierra, afectando a todo lo que queda dentro de dicho contexto.

El nombre que se le haya dado a la etiqueta será el que le dé un significado u otro. Por ejemplo, **<p>** será indicativo de párrafo y **** significará negrita. Además, se permite el anidamiento de etiquetas, es decir, pueden aparecer unas dentro del contexto limitado por otras.

A estas mismas etiquetas, se les puede añadir una barra (/) tras el símbolo de menor, convirtiéndolas así en etiquetas de cierre. En los ejemplos anteriores, serían **</p>** y ****. A partir de estas etiquetas de cierre, las correspondientes etiquetas de apertura pierden vigencia, dejando así de afectar al texto:

****Esto aparecerá en negrita.**** Esto ya no.

Un archivo HTML estará compuesto por un conjunto de etiquetas, que se utilizan para definir la forma o estilo que se quiere aplicar a cada parte del documento. Será el navegador el que se encargue de interpretar todas estas etiquetas y mostrar el contenido con el formato definido por ellas.

Es importante tener en cuenta que HTML no es un lenguaje de programación, ya que su función no es la de crear un programa, ni cuenta con las estructuras básicas de los lenguajes de programación. Por tanto, no se debe usar la palabra programar para referirse a la creación de páginas web. No obstante, es posible combinar HTML con otros lenguajes como javascript, que sí es un lenguaje de programación.



Estructura documento HTML.

Fuente: <https://studentplace98.blogspot.com/2019/04/EstructuraBasica.html>

Todo documento HTML ha de estar delimitado por las etiquetas **<html>** y **</html>**.

Dentro de este documento, se pueden asimismo distinguir dos partes principales:

- La cabecera, delimitada por **<head>** y **</head>**, donde se colocarán etiquetas de índole informativo como, por ejemplo, el título de nuestra página. En ese caso, el título irá delimitado por las etiquetas **<title>** y **</title>**
- El cuerpo, flanqueado por las etiquetas **<body>** y **</body>**, que será donde se coloque todo el contenido de la página.

El resultado es un documento con la siguiente estructura:

```
<html>
<head>
  <title>Cesur</title>
  <!--Etiquetas y contenidos de la cabecera.
  Datos que no aparecen en la página pero que son importantes
  para catalogarla: Título, palabras clave,...-->
</head>
<body>
  <!-- Etiquetas y contenidos del cuerpo.
  Parte del documento que será mostrada por el navegador: Texto e
  imágenes-->
</body>
</html>
```

Este ejemplo, puede servir también para ver cómo incluir comentarios en el archivo HTML. Tan sólo se trata de escribir el texto que interese entre las etiquetas `<!--` y `-->`. Su función es la de poder añadir texto que no será interpretado por los navegadores, y por tanto no aparecerá por pantalla.

Como se ha comentado anteriormente, HTML ha ido evolucionando con el tiempo, dando lugar a diferentes versiones, cada una con soporte para unas u otras etiquetas, si bien, la estructura que se acaba de presentar es común para todas. No obstante, resulta importante para el navegador conocer qué versión de HTML usa la página que está leyendo, ya que ello podrá determinar si la sintaxis del documento es válida o no. Para eso, se debe usar el elemento DOCTYPE.

Las declaraciones DOCTYPE no son más que enlaces a documentos de definición (DTD), que se estudiarán más adelante, cuyo cometido es definir los elementos válidos del lenguaje y la relación entre ellos. Deben aparecer al principio del documento, antes de su estructura, con la sintaxis siguiente: `<!DOCTYPE ...>`

Para la versión 4 de HTML, que ha sido la que ha tenido mayor vigencia en el tiempo y la más extendida hasta la aparición de la versión 5, se pueden encontrar las siguientes declaraciones DOCTYPE:

- Estricta: se trata de la declaración por defecto, que no permite usar elementos del lenguaje desfasados.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 //EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
```

- Transitional: para facilitar la transición entre versiones anteriores y esta, permite usar elementos obsoletos.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 transitional//EN"
"http://www.w3.org/TR/REC-html40/loose.dtd">
```

- Frameset: una variante de transitional pero para usar páginas con marcos.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN"
"http://www.w3.org/TR/REC-html40/frameset.dtd">
```

A continuación, se muestra cómo quedaría una página, con estos primeros conceptos que se han presentado:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 //EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<html>
<head>
  <title>Hola Mundo</title>
</head>
<body>
  <p>
    <b>Bienvenido,</b>
  </p>
  <p>
    Esta es la página <b>Hola Mundo</b>.
  </p>
</body>
</html>
```

Bienvenido,

Esta es la página **Hola Mundo**.

3.2 Sintaxis

Un archivo HTML no es más que un conjunto de elementos, cada uno con un nombre (etiqueta), un contenido y unas características (atributos). Al ser un lenguaje de marcas, son las etiquetas (tags) las que se usan para delimitar el contenido de estos elementos.

Las etiquetas pueden ser de 3 tipos:

- De apertura: <xxx>
- De cierre: </xxx>
- Autocerradas. Son como una combinación de las dos anteriores, para elementos que no tienen contenido: <xxx />

Aunque los nombres de las etiquetas se pueden escribir tanto en mayúsculas como en minúsculas, como convención se usa la escritura en minúsculas.

Por otro lado, cada uno de los elementos puede tener una serie de propiedades que se denominan atributos, y se declaran en la etiqueta de apertura de dicho elemento, asignándoles un valor entre comillas (simples o dobles), de la forma: atributo="valor". Todos los atributos deben escribirse separados tanto de la etiqueta como de otros atributos.

Los atributos admitidos y el valor que se le puede dar a cada uno de ellos vendrán determinado por la especificación de la versión de HTML que se esté usando, siendo posibles valores de tipo textual o numérico. Éstos últimos, pueden aparecer en diferentes bases (decimal, hexadecimal...) y con diferentes unidades (sin unidades, en píxeles, en porcentajes...).

Por tanto, la sintaxis de un elemento es la siguiente:

```
<nombre_elemento atributo1="valor1" atributo2="valor2" ...>
    contenido_elemento
</nombre_elemento>
```

Por ejemplo, como se vio anteriormente, el elemento principal en un documento html es el delimitado por <html> y </html>. Su contenido es el propio contenido del documento, compuesto a su vez por los elementos <head> y <body>.

El principal atributo de <html> es lang, que identifica el idioma en el que está nuestra página. Por tanto, para una página en español, se escribiría:

```
<html lang="es">
```

Cada etiqueta puede tener unos posibles atributos, según viene dado en la especificación de la versión de HTML correspondiente. Eso sí, no todos los atributos admitidos en versiones anteriores, se admiten en las posteriores, como es el caso de los atributos relativos a la presentación, como se verá más adelante.

Se debe tener en cuenta que cada etiqueta tiene sus propios requerimientos:

- Algunas necesitan aparecer cerradas, como `<a>`.
- Otras pueden tener etiqueta de cierre o no (como `<p>`, que no siempre requiere `</p>`).
- Otras no llevan nunca etiqueta de cierre (como ``, la cual jamás lleva ``).

No obstante, es importante saber que la validez de estas etiquetas cambia según la versión de HTML o XHTML que se esté usando, lo que puede inducir a error en algunos casos.

Para no equivocarse, conviene saber:

- Elementos vacíos (aquellos que no tienen etiqueta de cierre, como `
`): en HTML podían cerrarse o no. En XHTML deben cerrarse siempre (`
`). Sin embargo, en HTML5 no se usan elementos autocerrados, por lo que actualmente se escribe `
`. De todos modos, `
` no produce error (en ocasiones se usa para mantener compatibilidad con XHTML), ya que HTML5 ignora la barra de autocierre en las siguientes etiquetas: `area`, `base`, `br`, `col`, `embed`, `hr`, `img`, `input`, `link`, `meta`, `param`, `source`, `track`, `wbr`.
- Elementos no vacíos: mientras que en XHTML deben cerrarse siempre, en HTML5 hay muchos que se permite que se escriban sin cerrar. Por ejemplo, `<p>`. Esto permite ahorrar código, si bien, habitualmente no resulta recomendable, ya que el código así escrito pierde legibilidad.
- Atributos vacíos: en XHTML no estaban permitidos, mientras que en HTML5 sí (su valor por defecto es una cadena vacía).
- Atributos con valores sin comillas: en XHTML no era válido, necesitando especificar comillas (dobles o simples), mientras que en HTML5 son válidos. No obstante, es recomendable usar siempre comillas para mantener un estilo uniforme y evitar errores cuando el valor del atributo contenga ciertos caracteres como el espacio.

3.3 Estructura y sintaxis en XML

Como sabemos XML no representa datos en sí, simplemente afecta a la organización de la estructura. Proporciona como ventaja el minimizar tiempos en los desarrollos y aumentar la potencia de webs y aplicaciones en el guardado de información, constituyéndose en uno de los formatos más utilizados por sistemas operativos y dispositivos en general, sobre todo móviles.

Es un paso más allá de HTML pero no deja de ser un documento de texto, con estructura de árbol y sus correspondientes etiquetas y descripciones de funciones, con posibilidad de edición mediante cualquier tipo de editor de textos, y poder ser interpretado por los distintos navegadores web.

El documento consta de un prólogo donde se define la información de procesado de la información y que estará dividida en dos partes: La declaración, donde se indica la versión de XML, y la otra parte es la declaración del tipo de documento.

A continuación encontramos los elementos del documento, que son los datos, estructurados en forma secuencial y con posibilidad de estar anidados.

Importante también son los atributos, como la forma de asignar propiedades a los elementos.



VÍDEO DE INTERÉS

En este vídeo se explica qué es y para qué sirve HTML.

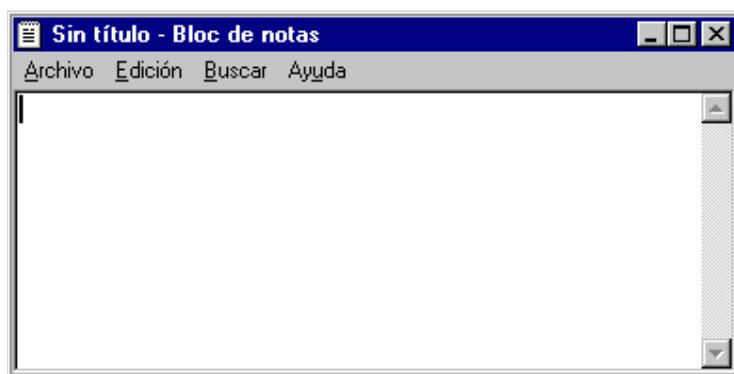


4. HERRAMIENTAS DE EDICIÓN

A la hora de realizar la página web que os han encargado en la empresa, es un requisito previo al desarrollo del código, conocer qué tipos de herramientas de edición del mismo existen y pueden ser utilizadas para facilitar la tarea de programación. Eres la persona encargada de recopilar qué herramientas se pueden utilizar para sentar las bases a los programadores.

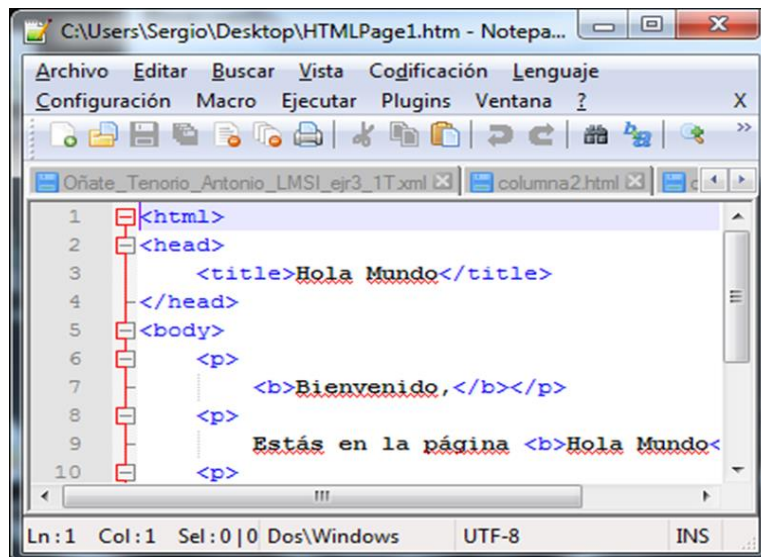
Antes de empezar a crear las primeras páginas, es necesario saber con qué herramienta se puede hacer. Este tipo de lenguaje se adapta tanto a los entornos de desarrollo específicos como a editores de textos.

Cualquier sistema operativo dispone de editores de texto de forma nativa o disponibles de forma gratuita. Un claro ejemplo es el bloc de notas de Windows, disponible en todas sus versiones.



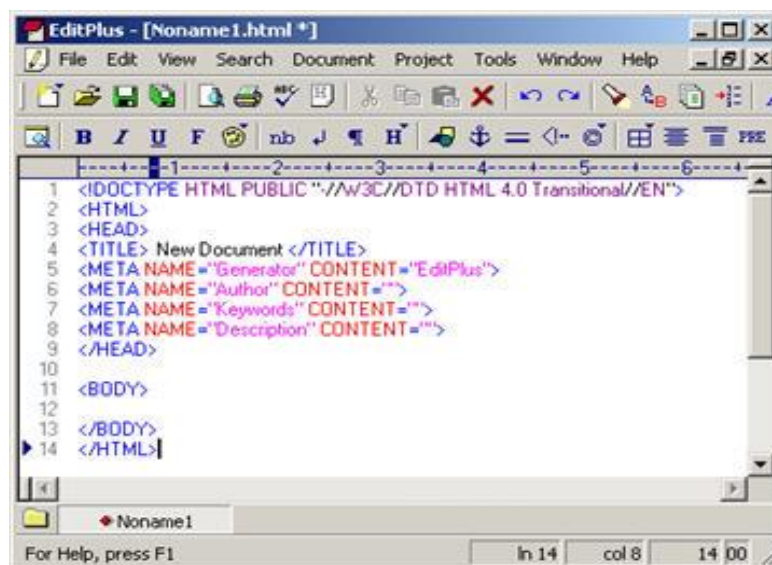
Los usuarios de Macintosh pueden usar un editor de texto llamado **SimpleText**, que además permite colorear la edición del texto, dando puntos de referencia a las etiquetas.

También se pueden encontrar diversos editores más avanzados, igualmente gratuitos, que proporcionan mayor comodidad a la hora de trabajar, como control de sangrías, búsquedas y reemplazos avanzados, etc. Notepad++ es un buen ejemplo de ello: un editor de texto potente y repleto de opciones. A medio camino entre un bloc de notas mejorado y un entorno de desarrollo ligero.



```
1 <html>
2 <head>
3   <title>Hola Mundo</title>
4 </head>
5 <body>
6   <p>
7     <b>Bienvenido,</b></p>
8   <p>
9     Estás en la página <b>Hola Mundo</b>
10  </p>
```

No obstante, existen editores de código mucho más potentes e indicados para el diseño de páginas web con html que incorporan barras de herramientas que generan automáticamente algunas de las etiquetas del código HTML, con lo que así se puede evitar escribirlas. Es el caso, por ejemplo, del editor **Edit Plus**.



```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
2 <HTML>
3 <HEAD>
4 <TITLE> New Document </TITLE>
5 <META NAME="Generator" CONTENT="EditPlus">
6 <META NAME="Author" CONTENT="">
7 <META NAME="Keywords" CONTENT="">
8 <META NAME="Description" CONTENT="">
9 </HEAD>
10 <BODY>
11 <P>
12   <b>Bienvenido,</b>
13 </P>
14 <P>
15   Estás en la página <b>Hola Mundo</b>
16 </P>
17 </BODY>
18 </HTML>
```

Bluefish es un editor HTML multiplataforma, que se puede descargar de la siguiente página Web: <http://bluefish.openoffice.nl/index.html>



Además, quien esté acostumbrado a manejar un IDE para programar, también puede usarlo para trabajar con HTML, ya que los más populares como Eclipse, NetBeans e IntelliJ tienen soporte para editar HTML.



PARA SABER MÁS

Para ampliar información sobre la configuración de Notepad++ accede a este enlace:



5. ELABORACIÓN DE DOCUMENTOS BIEN FORMADOS

Después de tener claras las distintas cuestiones sobre el tipo, características, estructura, etc., del lenguaje de marcas a utilizar, resulta de interés conocer cómo elaborar un documento bien formado que recoja el código. Serás la persona encargada de dar las directrices a seguir, después de recopilar la información y presentarla al resto de compañeros.

En la elaboración de documentos bien formados, es necesario tener en cuenta una serie de recomendaciones para que las páginas ofrezcan al usuario una buena experiencia de navegación, así como para evitar errores de diseño y que resulte fácil mantenerlas.

En la elaboración de documentos bien formados, uno de los aspectos principales a tener en cuenta es la sintaxis que utilizamos en lenguajes de marcas, como es el caso de XML, donde destacamos las siguientes relacionadas con:

- 1) Etiquetas:
 - Utilización correcta de etiquetas en los elementos.
 - Cuidar el formato de la propia etiqueta.
 - Selección correcta de nombres de elementos.
 - Deben estar correctamente anidadas.

2) Elementos y atributos:

- Asignación de atributos para cada elemento de acuerdo a las normas de escritura.
- Los valores de los atributos deben ir entrecomillados con comillas dobles o simples.
- El elemento debe contar con una etiqueta de apertura y de cierre.
- La separación de los atributos será con un espacio en blanco.
- Debe existir un elemento raíz.
- Todo elemento debe tener un elemento padre.

3) Procesamiento:

- Las instrucciones para el proceso deben ser escritas correctamente.
- La primera línea escrita debe contener la declaración XML.
- En la utilización de CDATA, sus secciones y comentarios ser escritos correctamente.

No debemos olvidar que además de cumplir con una sintaxis correcta debe recoger las especificaciones que se recojan en el DTD (Definición de tipo de documento) o esquema, de modo que cumpliendo ambos requisitos tendremos un documento XML bien formado y válido a la hora de su procesamiento. Pudiendo utilizar otros métodos como XML Schema o RELAX NG (Regular Language for XML Next Generation).

Si nos centramos en la estructura interna correcta de un documento XML, veremos que viene determinada por una serie de elementos y sus posibles atributos, pudiendo también carecer de estos. Destacamos lo relacionado con ellos a continuación.

- Elementos.

Forman parte de las etiquetas que utilizamos, pudiendo diferenciar entre:

- Obligatorios.
 - Identificador o etiqueta de cabecera, donde se define el comienzo del elemento.
 - Etiqueta de cierre, que define el final del elemento.
- Opcionales.
 - Atributos del elemento.
 - Cuerpo o contenido.

- Atributos.

Los elementos pueden tener o no asignados uno o varios atributos que, en realidad, son propiedades que definen más al propio elemento sobre el que se asigna, pudiendo contener un número ilimitado de atributos.

El formato vendría definido de la siguiente manera:

```
<Elemento atributo1="valor1" atributo2="valor2" atributoN="valorN">
```

En XML existen una serie de atributos que se reservan para acciones concretas, entre los que tenemos:

Xml:Space	Mantiene en el contenido la representación de los espacios en blanco
Xml:Link	Utilizado por el procesador XLink, indica que el elemento al que pertenece el atributo en un enlace a otro elemento
Xml:Attribute	Se utiliza para prevenir conflictos con otros atributos
Xml:lang	Clasifica el contenido por el idioma en que está escrito

Como hemos comentado, añaden propiedades a los elementos de un documento XML, no permiten una organización jerárquica ni ser anidados.

Se definen y dan valor dentro de las etiquetas de inicio y fin del elemento, detrás del nombre del elemento con un espacio de separación, llevando el valor precedido del signo =, al que le sigue el valor del atributo entrecomillado (dobles o sencillas), por ejemplo:

```
<cliente nombre="Luis Pérez García"></cliente>
```

Donde el elemento sería cliente y el atributo nombre, con el valor Luis Pérez García.



¿SABÍAS QUE...?

Los documentos XML deben cumplir las reglas sintácticas de las recomendaciones del W3C, relativas a:

- Prólogo con la declaración XML
- Elemento raíz único.
- Correcta definición de elementos y atributos.



ENLACE DE INTERÉS

Conoce más sobre los conceptos básicos de XML y claves para un documento bien formado en este enlace:





EJEMPLO PRÁCTICO

Elisa trabaja como programadora en una empresa de desarrollo web donde reciben encargos de empresas externas para la creación de portales, sitios y páginas web.

Conscientes de la importancia de la elaboración correcta de los documentos XML que contienen el código fuente, tiene encargado elaborar unas pautas a seguir en el proceso de creación de documentos para que el resto del equipo tengan un documento de referencia en lo que constituyen documentos bien formados.

¿Qué pautas o recomendaciones deberían incluirse para la elaboración de documentos bien formados?

Solución.

Las recomendaciones o pautas a tener en cuenta en la elaboración de documentos bien formados serían las siguientes:

Etiquetas:

- Utilización correcta de etiquetas en los elementos.
- Cuidar el formato de la propia etiqueta.
- Selección correcta de nombres de elementos.
- Deben estar correctamente anidadas.
- Elementos y atributos:
 - Asignación de atributos para cada elemento de acuerdo a las normas de escritura.
 - Los valores de los atributos deben ir entrecomillados con comillas dobles o simples.
 - El elemento debe contar con una etiqueta de apertura y de cierre.
 - La separación de los atributos será con un espacio en blanco.
 - Debe existir un elemento raíz.
 - Todo elemento debe tener un elemento padre.

Procesamiento:

- Las instrucciones para el proceso deben ser escritas correctamente.
- La primera línea escrita debe contener la declaración XML.
- En la utilización de CDATA, sus secciones y comentarios ser escritos correctamente.

6. UTILIZACIÓN DE ESPACIOS EN NOMBRES

En la escritura del código para la página web que os han solicitado, como requisito imprescindible, se debe tener en cuenta la corrección del mismo, con el fin de evitar errores de sintaxis, por ejemplo. Por tu experiencia en el cargo de responsable del departamento, eres consciente del fallo común y lo delicado que resulta la utilización de espacios en los nombres, por lo que vas a preparar la forma de trabajar con ellos y en especial el uso de blancos.

Todos los espacios en blanco, tabuladores y saltos de línea, son considerados por el intérprete HTML como un espacio simple, ya que de esta forma se puede escribir el código ordenadamente y con sangrías.

Por tanto, si hace falta poner más de un espacio en blanco en medio de un texto, se debe usar el carácter especial equivalente al espacio en blanco visto en la sección anterior:

Por ejemplo:

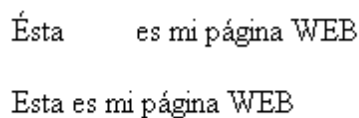
```
<!DOCTYPE html>
<html>
<head>
  <title>espacios</title>
</head>
<body>
  <p> Esta          es mi página WEB</p>
  <p>Esta es mi página WEB</p>
</body>
</html>
```

Esta es mi página WEB

Esta es mi página WEB

Por lo que, para que se vean las dos frases de forma diferente, como en el propio archivo html, habrá que escribir:

```
<!DOCTYPE html>
<html>
<head>
    <title>espacios</title>
</head>
<body>
    <p> Ésta&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& es mi
página WEB</p>
    <p>Esta es mi página WEB</p>
</body>
</html>
```



Para conseguir saltos de línea, se debe usar la etiqueta
 (no lleva etiqueta de cierre). No obstante, se debe tener en cuenta que otros elementos pueden generar por sí mismos saltos de línea, como los párrafos y encabezados.

Por el contrario, para conseguir tabulaciones, no se emplean etiquetas ni códigos especiales, sino que se definen márgenes, como se estudiará más adelante.

En el caso de XML, utilizar los espacios en nombres continúa siendo conflictivo e importante, sobre todo teniendo en cuenta que es la manera de poder nombrar los elementos de forma adecuada, pudiendo relacionarse con una identificación también única conocida como identificador URI, mediante la definición de prefijos que permiten calificar tanto los nombres de los elementos como los atributos en los datos XML.

En este caso el espacio de nombres (namespace) es el identificador que permite la resolución de problemas al nombrar elementos o atributos en XML por tener un nombre coincidente.

Permite relacionar cada uno de los elementos con su correspondiente espacio de nombres, resolviendo cualquier posible confusión entre elementos o atributos.

La forma de declarar un espacio de nombres en XML será utilizando el atributo xmlns seguido del nombre del espacio y una url, de tal modo que entraríamos una instrucción del siguiente tipo:

```
xmlns:nombre_del_espacio='http://nombre_de_url';
```



ENLACE DE INTERÉS

En este enlace puedes ver cómo administrar espacios de nombres en un documento XML



RESUMEN FINAL

En esta unidad se ha visto el lenguaje de marcas, comenzando con su clasificación y destacando como los más extendidos HTML y XML, continuando con sus principales características, tanto generales destacando flexibilidad, independencia, compatibilidad, entre otras, como específicas relativas a su forma de escritura, por ejemplo. Sin olvidar sus ámbitos de aplicación en áreas como las matemáticas, multimedia, mensajería instantánea, etc.

Aspecto importante es su estructura y sintaxis, con el uso de etiquetas de apertura y cierre que deben contener todos los documentos, y sus partes principales como cabecera (head) y cuerpo (body).

En la edición del código, es interesante conocer las distintas opciones que existen a la escritura directa por parte del programador, disponiendo de herramientas que facilitan la labor, desde el simple bloc de notas a herramientas más completas como puede ser Notepad++ o Bluefish.

Relacionado con la estructura y sintaxis está la elaboración de documentos bien formados, donde deberemos tener especial cuidado con las etiquetas, los elementos y atributos y el procesamiento y finalizamos la unidad con un apartado dedicado a la utilización de espacios en los nombres y en los textos con las etiquetas a usar.