

DAM

Linguaxes de Marcas e Sistemas de Información

UD1 **HTML**

PROFESORA | CRISTINA PUGA FDEZ

UNIDADE | UD1

TRIMESTRE | 1

MÓDULO | LMSXI

Tabla de contenido

0. Introducción a los lenguajes de marcas.....	3
0.1 Introducción.....	3
0.2 Lenguajes de marcas.....	5
0.3 Evolución de los lenguajes de marcas	8
0.4 Lenguajes para la Web.....	9
1. HTML.....	13
1.1 Introducción.....	13
1.2 HTML.....	13
1.3 XHTML.....	22
1.4 HTML5.....	24
1.5 Herramientas de diseño web.....	27
1.5.1 Editores de HTML	27
1.5.2 Herramientas de desarrollo de los navegadores	29
1.5.3 Herramientas WYSIWYG	30
ANEXO I: Materiales	31

0. Introducción a los lenguajes de marcas

0.1 Introducción

Antes de existir las bases de datos, las aplicaciones almacenaban la información en sistemas de ficheros (un conjunto de ficheros, normalmente de texto, donde se volcaba toda la información).

Estos ficheros tenían fundamentalmente un problema: que los datos se volcaban en el fichero sin almacenar información alguna acerca del contenido del mismo.

- En ficheros de almacenamiento, cuando se almacenaba mucha información hacía muy complicado saber qué información era la que se estaba almacenando, y conocer el significado de un dato concreto del fichero.
- Todos los datos se guardaban en texto plano, por lo que no era posible indicar cómo se debía mostrar cada campo a la hora de visualizarlo.

Pongamos el siguiente ejemplo:

1	Rodrigo	Vázquez	1100	762234422	3	49	1979	5	1
2	Elena	Álvarez	1200	622332122	3	49	1977	8	3
3	Marcos	Rodríguez	1300	4	49	1978	4	9	
4	Raquel	Estévez	1400	912221223		51	1975	3	1

Figura 1 – Ejemplo fichero

En un fichero como este, ¿Qué significaría el 3 señalado en rojo? ¿Por qué las filas número tres y cuatro tienen menos datos que las anteriores? ¿Qué datos faltan?

Esta falta de información acerca de los datos almacenados provoca que las aplicaciones que quieran hacer uso de ellos deban tener un conocimiento muy detallado no solo de qué datos se están almacenando sino también de qué tipo son, y cómo acceder a ellos para poder emplearlos correctamente.

Las bases de datos solucionan este problema por medio de la incorporación de metadatos (información sobre los propios datos almacenados). Para almacenar información en una base de datos, es necesario crear previamente una tabla, en la que se indicarán qué campos contiene y de qué tipo es cada uno de ellos, y al insertar datos en la tabla, cada dato se insertará en una columna concreta, con lo cual ya sabremos qué es cada uno de los datos que almacenamos:

Field	Type	Null	Key
codemp	int(11)	NO	PRI
nombre	varchar(100)	NO	
apellido	varchar(100)	NO	
salario	decimal(8,2)	NO	
telefono	char(9)	YES	
codjefe	int(11)	YES	MUL
coddepto	int(11)	NO	
anho_nac	year(4)	YES	
mes_nac	tinyint(4)	YES	
dia_nac	tinyint(4)	YES	

Figura 2 – Ejemplo definición de la tabla Empleado en una Base de Datos

codemp	nombre	apellido	salario	telefono	codjefe	coddepto	anho_nac	mes_nac	dia_nac
1	Rodrigo	Vázquez	1100.00	762234422	3	49	1979	5	1
2	Elena	Álvarez	1200.00	622332122	3	49	1977	8	3
3	Marcos	Rodríguez	1300.00	NULL	4	49	1978	4	9
4	Raquel	Estévez	1400.00	912221223	NULL	51	1975	3	1

Figura 3 – Ejemplo datos almacenados en una Base de Datos

En este caso ya sabemos que el 3 señalado en rojo es un código de jefe, y además podríamos comprobar que ese código de jefe se refiere a un código de empleado de los existentes en la tabla de empleados, podríamos mostrar el salario con el signo del dólar, las fechas con un formato concreto, etc., y sabríamos que en las filas 3 y 4 hay menos datos porque faltan un teléfono y un código de jefe, respectivamente.

Los lenguajes de marcas surgen con el objetivo de paliar esta problemática de los ficheros de texto “planos” tradicionales. El concepto de lenguaje de marcas deriva de la antigua práctica empleada por las editoriales de marcar los manuscritos con instrucciones de impresión en sus márgenes, para indicar el tipo de letra, el estilo y el tamaño que se debía dar a las distintas secciones del documento cuando se imprimiese, ya que esta es de alguna manera la filosofía que se emplea en estos lenguajes para almacenar la información: junto al contenido del documento se añaden marcas o etiquetas que proporcionan información sobre el texto, de forma que sea posible disponer de información acerca de los datos almacenados.

```
<?xml version="1.0"?>
<Empleados>
  <empleado id="1">
    <nombre>Rodrigo</nombre>
    <apellido>Vázquez</apellido>
    <salario>1100</salario>
    <telefono>762234422</telefono>
    <codjefe>3</codjefe>
    <coddepto>49</coddepto>
```

```
<fec_nac>
  <anho>1979</anho>
  <mes>5</mes>
  <dia>1</dia>
</fec_nac>
</empleado>
<empleado id="2">
  <nombre> Elena </nombre>
  <apellido> Álvarez </apellido>
  <salario>1200</salario>
  <telefono>622332122</telefono>
  <codjefe>3</codjefe>
  <coddepto>49</coddepto>
  <fec_nac>
    <anho>1977</anho>
    <mes>8</mes>
    <dia>3</dia>
  </fec_nac>
</empleado>
<empleado id="3">
  <nombre> Marcos </nombre>
  <apellido> Rodríguez </apellido>
  <salario>1300</salario>
  <codjefe>4</codjefe>
  <coddepto>49</coddepto>
  <fec_nac>
    <anho>1978</anho>
    <mes>4</mes>
    <dia>9</dia>
  </fec_nac>
</empleado>
<empleado id="4">
  <nombre> Raquel </nombre>
  <apellido> Estévez </apellido>
  <salario>1400</salario>
  <telefono>912221223</telefono>
  <coddepto>51</coddepto>
  <fec_nac>
    <anho>1975</anho>
    <mes>3</mes>
    <dia>1</dia>
  </fec_nac>
</empleado>
</empleados>
```

Figura 4 – Ejemplo fichero

0.2 Lenguajes de marcas

Un lenguaje de marcas es un lenguaje de creación de documentos basado la definición de dos

componentes:

- Un conjunto de marcas (cada marca tendrá un significado y puede ser o no obligatoria).
- Una forma de diferenciar las marcas del texto del documento.

0.2.1 Clasificación de los lenguajes de marcas

Existen tres categorías generales de clasificación de los lenguajes de marcas:

- **Lenguajes orientados a la presentación.** es el tipo de marcado usado habitualmente por los procesadores de texto: indican cómo se debe presentar el texto del documento (presentación, formato, párrafo, etc.) por medio de marcas internas que están ocultas al usuario y únicamente pueden ser manipuladas e interpretadas por medio del programa encargado de su codificación y decodificación (Word, Writer, etc.). Las instrucciones solamente pueden ser interpretadas por el programa encargado de la visualización o impresión del contenido, y por tanto no son fácilmente portables, por lo que no suelen considerarse como lenguajes de marcas como tal.
- **Lenguajes procedurales.** este tipo de marcado también está orientado a la presentación del texto, pero las marcas están incluidas en el propio texto y son visibles para el usuario, que puede modificarlas empleando cualquier editor de texto. Las marcas constituyen instrucciones para los programas que van a procesar el texto. Son de este tipo LaTeX, CSS o Postscript.
- **Lenguajes descriptivos.** en estos lenguajes se emplean las marcas para indicar qué es la información, cómo se estructura y su significado semántico (describen qué es lo que se está representando). Son de este tipo XML, HTML o JSON.

0.2.2 Características y ventajas de los lenguajes de marcas

A continuación, comentamos algunas de las características de los lenguajes de marcas, que constituyen algunas de sus ventajas y que hicieron que sean tan utilizados en la actualidad:

- **Texto plano.** Un documento escrito con lenguajes de marcas puede ser escrito con un simple editor de textos, ya que las marcas se intercalan con el contenido, aunque podrían utilizarse programas más complejos que faciliten el trabajo.
- **Independencia lógica.** Como se trata únicamente de texto, los documentos son independientes de la plataforma, sistema operativo o programa con el que fueron creados.

- **Facilidad de procesamiento.** el marcado suele ser bastante intuitivo, y las marcas están orientadas a describir la estructura lógica del documento, por lo que el significado de las etiquetas suele ser fácil de interpretar y por tanto son fáciles de usar.
- **Flexibilidad.** La estructura de los documentos es muy flexible, lo que permite que puedan almacenar cualquier tipo de información. Además, suelen tener soporte para lenguajes de script que permiten añadir funcionalidades adicionales, no proporcionadas por el lenguaje, para un comportamiento interactivo y dinámico, por ejemplo.

0.2.3 Ámbitos de aplicación

Los principales ámbitos de aplicación de los lenguajes de marca vienen determinados en gran medida por sus características:

- **Presentación** (LaTeX, css, rtf o postscript): El primer uso para es que se concibieron los lenguajes de marcas era el poder incorporar en el propio documento información acerca de cómo se debía mostrar el contenido del documento (es decir, definir su presentación y formato). Esto era muy útil especialmente para enviar documentos a imprenta de forma que el propio creador del documento pudiese determinar su diseño final.
- **Estructuración de la información** (HTML, XHTML, XML, JSON): las marcas de este tipo de documentos no hacen referencia a cómo se mostrará su contenido, sino a cómo se estructurará, de forma que se puede organizar internamente el documento y almacenar la información con una estructura lógica que facilita su procesamiento y la interpretación de su contenido.
- **Almacenamiento** (XML, BSON): Las características de los lenguajes de marcas hacen que en determinados contextos sean una alternativa muy adecuada con respecto a las bases de datos para el almacenamiento de información:
 - Para almacenamiento de información sencilla, como ficheros de configuración: son compactos, fáciles de procesar e no requieren de un software específico para consultar su contenido).
 - Para almacenamiento de información no estructurada: su flexibilidad a la hora de representar la información los hace más adecuados para almacenar información que

no tiene una estructura fija, por lo que no es adecuada para ser almacenada en tablas de bases de datos.

- Para procesar grandes cantidades de información, se emplean sistemas distribuidos que almacenan su información en este tipo de documentos, de forma que se reparten los documentos entre los distintos nodos y éstos se procesan en paralelo, lo que hace que sean muy rápidos y fácilmente escalables.
- **Intercambio de información** (RSS, SOAP): al igual que en el caso anterior, la independencia lógica de los documentos, su compacidad y su facilidad de procesamiento hacen que sea una alternativa muy adecuada para el envío de información entre aplicaciones (servicios web, sindicación de contenidos, ...), ya que al incluir toda la información en un solo documento de texto facilita el envío, estructuración e interpretación de su contenido.
- **Áreas especializadas:** además de estos ámbitos de aplicaciones generales, existen diversos ámbitos de aplicación más específicos en los que se han desarrollado lenguajes de marcas especializados para llevar a cabo tareas determinadas:
 - gráficos (SVG, VRML/X3D), matemáticas (MathML, OpenMath)
 - finanzas (eXtensible Business Reporting Language, Financial products ML)
 - información geográfica y aeroespacial, etc.

0.3 Evolución de los lenguajes de marcas

Los primeros lenguajes de marcas existentes fueron los procedurales, que tenían como finalidad incorporar información de presentación y formato de un texto, normalmente para poder enviar los documentos a una imprenta asegurando que se respetase el diseño de estos.

A finales de los años 60, IBM crea un lenguaje de marcas (al que llamaron GML, Generalized Markup Language), en el que las marcas son descriptivas en lugar de procedurales, ya que tienen en cuenta la estructura del documento (se centra más en qué es el texto, sin especificar los procedimientos que hay que aplicarle para mostrarlo). En lugar de tener un simple esquema de etiquetas, permite definir los documentos en forma de estructura de elementos anidados.

A mediados de los 80, ANSI e ISO elaboran un estándar basado en GML, al que denominaron SGML (Standard Generalized Markup Language), que se convirtió en estándar internacional para la

definición de métodos de representación de textos en forma electrónica, independientes del dispositivo e independientes del sistema.

SGML, más que un lenguaje, es un metalenguaje para la definición y la estandarización de la estructura de los documentos, ya que define una gramática con la que se pueden crear otros lenguajes de marcas.

SGML fue diseñado originalmente para permitir el intercambio de grandes documentos con poca previsión de variación a lo largo de los años (leyes, estándares de la industria, etc), aunque fue también aplicado en la industria militar y aeroespacial, para referencias técnicas, y por la industria editorial.

SGML fue el precursor de uno de los principales lenguajes de marcas: HTML, que fue un lenguaje generalizado o descriptivo hasta su versión HTML5.

Con la aparición de XML, SGML entró en desuso ya que sus ámbitos de aplicación quedaban perfectamente cubiertos por XML.

0.4 Lenguajes para la Web

La conexión web se basa en una arquitectura cliente / servidor: se establece una conexión física entre dos equipos (el cliente y el servidor), y se utiliza un servicio (www, world wide web) que establece la conexión entre ambos empleando para ello un protocolo (http) que define las reglas que deben seguir ambos para llevar a cabo la comunicación.

- El cliente usará como herramienta un navegador web, que es una aplicación que únicamente interpreta y muestra contenido HTML (para organizar el contenido de las páginas) y CSS (para darle formato). Los navegadores actuales pueden procesar también pequeños programas definidos en javascript, además de dar soporte al contenido dinámico de HTML5.
- El servidor será el equipo donde se aloje la web, y donde estará toda la lógica de procesamiento: recibe peticiones de los clientes, ejecuta aplicaciones que procesan esas peticiones y genera páginas html que se enviarán de vuelta a los clientes.

Cuando desde un navegador web (cliente), realizamos cualquier tipo de interacción con una página web, lo que está sucediendo internamente es lo siguiente:

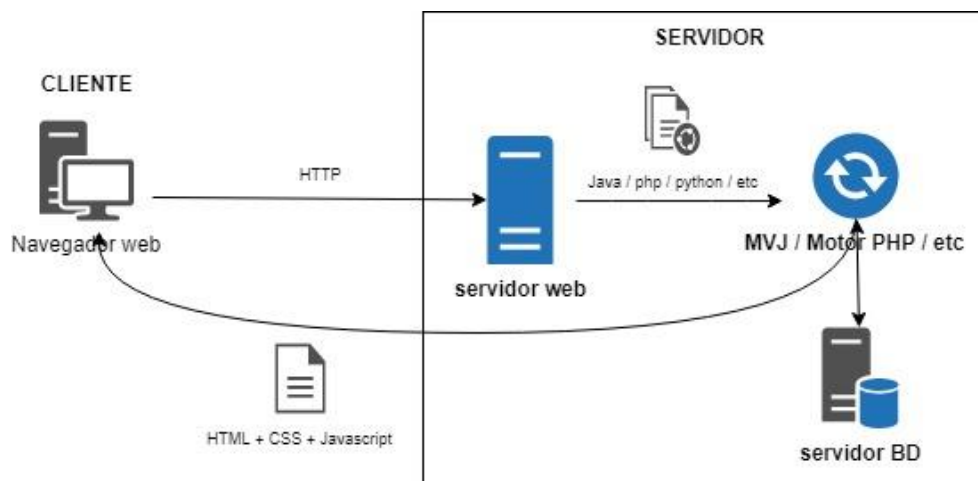


Figura 5 – Ejemplo ejecución página web

- El cliente envía una petición por medio del protocolo HTTP (hipertext transfer protocol) al servidor. En esa petición, el cliente puede enviar información al servidor, que irá en formato html o xml.
- En el servidor web se encarga de recibir las peticiones de los clientes y buscar las páginas que correspondan. Estas páginas no suelen estar en HTML, sino que normalmente están escritas en un lenguaje de programación (como Java, PHP, Python o ASP) que el propio servidor web no es capaz de procesar, por lo que se encargará de enviarlas a la aplicación que corresponda (un motor PHP, una máquina virtual de Java o de .NET, etc), que normalmente se ejecuta en el mismo equipo físico que el servidor web. El motor PHP o máquina virtual se encarga de interpretar todas las instrucciones definidas por medio de un lenguaje de programación, hacer consultas a la base de datos, y generar la página HTML, que será lo que se envíe al cliente (que puede contener información de estilo, en formato CSS, así como pequeños scripts incrustados en la propia página que se pueden ejecutar en el propio cliente, escritos en javascript o AJAX).
- El navegador web del cliente recibe la página e interpreta las etiquetas HTML y CSS para mostrar su contenido, y si el cliente ejecuta alguno de los scripts incluidos en la propia página es capaz de ejecutarlos el propio navegador sin realizar peticiones adicionales al servidor web.

Además de la comunicación entre clientes y servidores, hay ocasiones en las que los propios servidores deben comunicarse entre ellos (por ejemplo, un buscador de seguros de coche tipo rastreator: el servidor de rastreator realmente lo que hace es pedir información al cliente y con esa información realizar múltiples consultas a todos los servidores de las agencias de seguros buscando el presupuesto para las condiciones indicadas por el cliente).

En este caso el servicio que se utiliza no es www, sino que se utilizan servicios web. Los servicios web son recursos de información a los que pueden acceder las aplicaciones a través de internet. Están diseñados para permitir la comunicación de una aplicación con otra por medio del intercambio de mensajes en formato XML.

El funcionamiento básico de los servicios web es el siguiente:

- Una empresa, como proveedor, implementa su servicio y lo registra en el agente o UDDI que hace la función de listín público de servicios web (algo similar al DSN con las direcciones IP).
- El cliente o consumidor busca el servicio web que necesita a través del UDDI.
- El cliente puede acceder al servicio web del proveedor enviando la información que necesite en formato XML, y recibiendo la información del servidor también en formato XML.

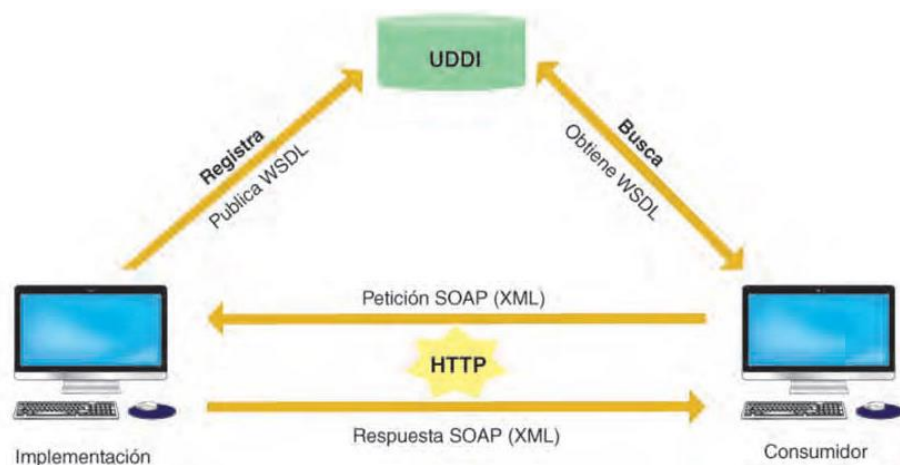


Figura 6 – Funcionamiento básico de los servicios web

Tanto en la comunicación entre usuario y servidor como en la comunicación entre aplicaciones, es importante:

- Que la información se envíe de manera compacta (tanto de servidor a cliente como de cliente a servidor), para que pueda ser recibida e interpretada conjuntamente.
- Que los datos enviados sean independientes de la plataforma, sistema operativo o incluso navegador en el que vayan a ser visualizados, para que puedan ser mostrados en cualquier entorno.

- Entre la información enviada se deberán incluir todos los metadatos necesarios para que el navegador web pueda estructurar correctamente el contenido y mostrarlo con el formato adecuado.

Los lenguajes de marcas son por tanto muy útiles para su utilización en entornos web, ya que sus características se adaptan perfectamente a estos entornos.

Los principales lenguajes de marcas usados en internet son:

- **SGML** (Standard Generalized Markup Language). En realidad, es un metalenguaje, es decir, se usa para crear otros lenguajes de marcas. Es muy potente pero también complejo de utilizar, motivo por el que apareció XML.
- **XML** (eXtensible Markup Language, lenguaje de marcado extensible): es un lenguaje derivado de SGML, desarrollado por el w3c (World Wide Web Consortium) para almacenar datos de manera legible. También es un metalenguaje, pero más sencillo de utilizar. Los documentos XML son documentos de texto cuyas etiquetas permiten representar información de manera iterativa y no estructurada, por lo que es muy utilizado para estructurar documentos, almacenar información y especialmente como estándar de intercambio de información a través de internet:
 - En servicios web, para enviar información entre servidores
 - En clientes que utilizan la tecnología AJAX (Asynchronous JavaScript and XML). Ajax permite intercambiar pequeñas cantidades de información de manera asíncrona entre el cliente y el servidor, para permitir actualizar únicamente partes concretas de una página web, sin necesidad de recargar ésta completamente desde el servidor.
- **HTML** (Hypertext Markup Language, lenguaje de marcas de hipertexto). Es un lenguaje también derivado de SGML que permite estructurar y formatear el contenido de las páginas web.
- **CSS** (Cascade Style Sheet, hojas de estilo en cascada). Lenguaje empleado para definir estilos para su utilización en páginas web. La función de CSS es definir la forma en que se mostrará el contenido de las páginas.

1. HTML

1.1 Introducción

En la década de los 70 del siglo XX, comenzaron a emplearse de forma generalizada las redes de computadoras, aunque en esa época su uso se basaba básicamente en el intercambio de correos electrónicos o chats.

En los 80 se empezó a trabajar en el intercambio de ficheros entre equipos, pero para ello era necesario establecer conexiones entre equipos diferentes y redes diferentes, por lo que se empezaron a desarrollar, consolidar y estandarizar protocolos que permitiesen llevar a cabo este tipo de comunicaciones.

En 1989, Tim Berners-Lee (que por entonces trabajaba en el CERN), propuso un sistema que permitía el acceso a ficheros ubicados en otros ordenadores de manera online, por medio del uso de redes basadas en el protocolo TCP/IP. Su objetivo inicial era permitir compartir fácilmente información entre científicos de distintas universidades e institutos de investigación de todo el mundo.

Este sistema fue el germen de la web (www, “World Wide Web”), y definía sus componentes básicos:

- La independencia del sistema.
- La facilidad de uso de los recursos de información existentes y de añadir nueva información de una manera sencilla.
- Contar con un mecanismo de transporte de los documentos entre diferentes redes (HTTP).
- Contar con un esquema de identificación para el direccionamiento de documentos de hipertexto tanto local como remoto.

Además de definir el lenguaje HTML, definió el primer navegador web que permitía utilizarlo, al que llamó World Wide Web (posteriormente denominado Nexus, ya que con www se pasó a denominar el servicio de acceso a documentos html a través de la red).

1.2 HTML

HTML es un lenguaje de marcas empleado para definir el contenido y la estructura de las páginas

web.

Es un lenguaje descriptivo, ya que define la forma en que se organizan y estructuran sus contenidos, aunque también permite definir aspectos del formato que se empleará para mostrar los datos.

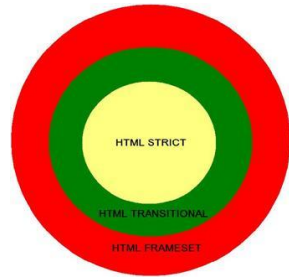
1.2.1 Versiones¹

En 1991, una vez desarrollados tanto el lenguaje como el navegador web, el código y las especificaciones tanto de HTML como del navegador se compartieron en Internet, lo que facilitó que rápidamente comenzasen a aparecer otros navegadores² y nuevas etiquetas para el lenguaje. Para evitar la proliferación de diferentes versiones surgió la necesidad de crear un estándar para que tanto los programadores como los navegadores pudieran basarse en unas mismas normas para escribir HTML, por lo que en 1995 se definió el primer estándar oficial del lenguaje, al que se denominó HTML 2.0. Cada versión de HTML establece unas normas respecto a cuáles son las etiquetas válidas y cómo se deben escribir.

Estándar	Año	Características
HTML 2.0	1995	Estándar inicial (no existe versión 1.0). La declaración de elementos de estructura (html, head y body) es opcional. Sin soporte para tablas.
HTML 3.2	1997	Simplificación del HTML 3.0 propuesto por IETF. Su elaboración por parte de compañías como Netscape, Microsoft, IBM o Sun, dio lugar al nacimiento del W3C. Incluye soporte para applets, mayor versatilidad en la disposición del texto y las imágenes, y añade las tablas.
HTML 4.0	1997	Estandariza los marcos, las hojas de estilo CSS y los scripts, incluye más soporte multimedia y mejoras de accesibilidad, entre otras cosas.

¹ https://www.elconfidencial.com/multimedia/album/tecnologia/2016-03-17/webs-viejunas_1169669#0

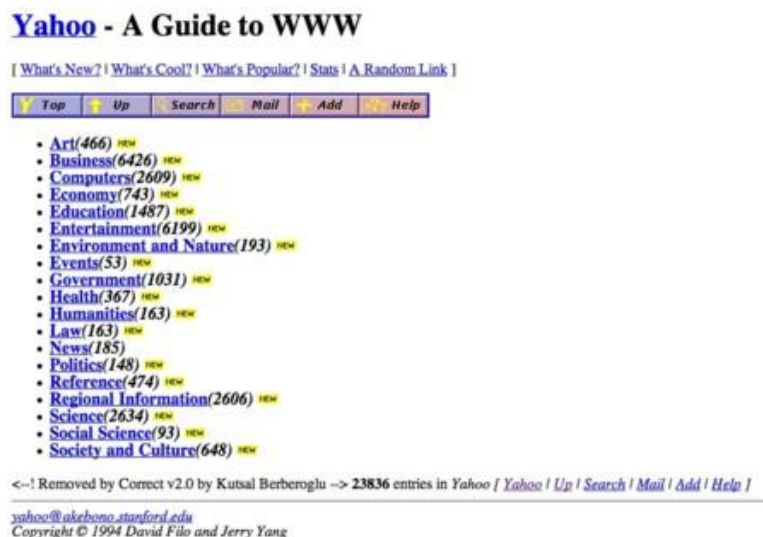
² <http://www.evolutionoftheweb.com/?hl=es>

HTML 4.01	1999	Fue la más utilizada hasta la aparición de HTML5 en 2014. Compatible con la mayoría de navegadores. Dispone de tres subestándares: Strict, Transitional, y Frameset.	
HTML 4.01 Strict		No se aceptan etiquetas obsoletas (deprecated) de estándares anteriores.	
HTML 4.01 Transitional		Permite el uso de todo tipo de etiquetas, aunque sean obsoletas.	
HTML 4.01 Frameset		Da soporte a frames.	
XHTML 1.0	2000	Surge como una combinación de XML y HTML.	
XHTML 1.1	2001	Elimina el soporte para marcos y elementos obsoletos, siendo más o menos equivalente a HTML 4.01 Strict.	
HTML 5.0	2014	Incluye muchas novedades, entre las que destacan nuevas etiquetas y mayor soporte e integración de contenidos multimedia. HTML5 implicaba una revisión por parte de los navegadores, ya que incorpora nuevas etiquetas y funcionalidades, pero supuso una gran evolución respecto a la versión anterior, por lo que rápidamente se ha ido generalizando su uso.	
XHTML 5.0	2014	Definido conjuntamente con el anterior	

Además de HTML, el propio servicio www ha evolucionado considerablemente, dando lugar a varias versiones:

- **web 1.0:** las primeras versiones de la web únicamente permitían páginas estáticas: el servidor alojaba páginas HTML ya definidas y cuando recibía solicitudes por parte de los clientes les proporcionaba directamente dichas páginas. Para crear una página web era necesario definirla directamente usando HTML, por lo que muy pocos usuarios podían crear páginas web. No

contienen contenido semántico (información acerca de la propia web), por lo que no son muy valoradas por los buscadores.



3

Página de Yahoo en 1994

- **web 2.0:** las páginas en el servidor ya no se crean en HTML, sino que se crean páginas dinámicas por medio de un lenguaje de programación y en función de la solicitud que lleve a cabo el cliente creará una página HTML de una forma u otra (consultando información almacenada en bases de datos). Se crean gestores de contenidos (wordpress, Moodle, joomla, etc) y sitios web con blogs, wikis, foros, etc, que permiten a los usuarios introducir sus propios contenidos de manera sencilla empleando un editor de texto, por lo que se generaliza la creación de contenidos y se multiplican exponencialmente el número de páginas disponibles en Internet. Disponen de información semántica acerca de la web, por lo que pueden realizarse búsquedas de sus vídeos, de sus imágenes, acerca de sus contenidos, etc. Las páginas pueden intercambiar información entre sí por medio de XML y RSS (Really Simple Syndication).

³ <https://www.genbeta.com/web/la-compra-de-yahoo-asi-ha-sido-la-evolucion-visual-de-este-historico-de-internet>



Página de Yahoo en 2004

1.2.2 Estructura de una página web

Los documentos HTML tienen una estructura bien definida, creada por medio de elementos. La plantilla básica para crear un documento HTML sería la siguiente:

```
<html>
  <head>
    <title>Titulo de la página</title>
  </head>
  <body>
    CONTENIDO DEL DOCUMENTO
  </body>
</html>
```

- En la primera línea se puede incluir una declaración DOCTYPE `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`). En HTML no es obligatorio indicar esta línea, pero sí lo es en XHTML.
- Dentro de la etiqueta `<html>` se incluyen las dos secciones principales que tiene un documento HTML: la cabecera y el cuerpo. Todo documento debe contener una etiqueta de inicio de documento al principio, `<html>` y una etiqueta de fin de documento, `</html>` al final del documento. Contiene un único elemento HEAD y un único elemento BODY.

- La cabecera contiene el título y otra información complementaria sobre el documento (metadatos, hojas de estilo, scripts, etc). Las etiquetas `<head>` y `</head>` marcan el inicio y el final de la cabecera.
- En el cuerpo es donde va el contenido del documento con las marcas asociadas a su estructura. Las etiquetas `<body>` y `</body>` marcan el inicio y fin del cuerpo. El elemento body delimita el contenido del documento y sus atributos se utilizan para efectuar cambios que afecten al documento entero como por ejemplo establecer imágenes o colores de fondo.

1.2.3 Etiquetas y atributos

Los documentos se estructuran por medio de elementos, pares de etiquetas de inicio y cierre:

```
<title>Título de la página</titulo>
```

- Algunos elementos tienen etiquetas de cierre opcionales (`<p>`) y otros tienen una única etiqueta de apertura (`
`) o de apertura y cierre conjuntamente (``).
- Las etiquetas deben estar anidadas, no cruzadas.
- Los nombres de elementos no son sensibles al uso de mayúsculas o minúsculas pero sí podría serlo el valor de los atributos.
- Los navegadores ignoran los elementos o atributos desconocidos.
- Los navegadores colapsan los espacios en blanco a un único espacio, esto incluye a los tabuladores, saltos de línea y retorno de carro.

Los elementos pueden tener atributos:

```

```

El valor de los atributos debe ir entre comillas.

En la página se pueden incluir también comentarios, que serán ignorados por el navegador. Los comentarios se incluyen entre los símbolos `<!--` y `-->`

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

Los símbolos de `>` y `<` en HTML son caracteres especiales que se utilizan para definir las etiquetas. Si queremos incluir en el texto de una página un símbolo de comparación tendremos que usar el carácter de escape `&` seguido del código que corresponda al símbolo que queramos usar. Esto se

utiliza también a veces para evitar problemas con los acentos. Algunos de los códigos más utilizados⁴:

Carácter	Entidad HTML	Carácter	Entidad HTML
á	á	À	Á
é	é	É	É
í	í	Í	Í
ó	ó	Ó	Ó
ú	ú	Ú	Ú
ü	ü	Ü	Ü
ñ	ñ	Ñ	Ñ
<	<	&	&
>	>	"	"e;
Espacio	 		

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Documento de ejemplo</title>
  </head>
  <body>
    <p>Us&eacute; un car&aacute;cter especial</p>
  </body>
</html>
```

1.2.3.1 Elementos de la cabecera

Etiqueta	Descripción
<base>	Proporciona una ruta base para hacer referencia a otras páginas web.
<link>	Permite especificar una relación entre el documento actual y otros recursos.
<meta>	Permite incorporar metadatos a la página (palabras clave acerca de su contenido)
<script>	Para definir funciones de script de cliente que se usarán en la página. Tiene un atributo obligatorio, type, para indicar el lenguaje de script utilizado (normalmente "text/javascript")
<style>	Permite incluir hojas de estilo en las que se definen los estilos que se emplearán en la página para dar formato a sus contenidos. Tiene un atributo obligatorio: type (para especificar el tipo de contenido del lenguaje de estilo, normalmente "text/css").

⁴ <https://ascii.cl/es/codigos-html.htm>

<code><title></code>	Para indicar el título de la página. Cuando se visualiza el documento, su título aparecerá en la barra del navegador. Solo puede haber un título en cada página.
----------------------------	--

1.2.3.2 Elementos del body

Etiqueta	Descripción
<code><h1><h2>...<h6></code>	Encabezados (de mayor a menor tamaño)
<code><p></code>	Delimita el contenido de un párrafo de texto.
<code>
</code>	Introduce un salto de línea.
<code><hr></code>	Inserta una línea horizontal.
<code><pre></code>	Texto preformateado, muestra el texto que contiene tal y como está escrito, manteniendo los espacios en blanco.
<code><i></code>	Muestra el texto en cursiva.
<code></code> y <code></code>	Muestra el texto en negrita. Strong añade contenido semántico.
<code></code>	Imágenes. Tiene cuatro atributos importantes: height y width (alto y ancho de la imagen), src (la ruta donde está la imagen a mostrar) y alt (texto alternativo que se mostrará cuando no se encuentre la imagen o mientras ésta no se cargue),
<code><a></code>	Enlaces. Tiene los atributos "name" (el nombre que se asignará a ese enlace) y "href" (la dirección a la que apuntará el enlace)
<code><table></code>	Define una tabla
<code><th></code> y <code><tr></code>	Definen respectivamente la fila cabecera (no obligatoria) y el resto de filas de una tabla
<code><td></code>	Define una celda de una tabla. Contiene dos atributos: colspan y rowspan para indicar el número de columnas o filas que abarcará esa celda (lo que permite fusionar celdas)
<code></code> y <code></code>	Permite crear listas ordenadas (ol) o desordenadas (ul). Las listas ordenadas tienen un atributo "type", que permite indicar qué caracteres se usarán para la numeración (1,a,etc) y un atributo "start" que permite definir en qué número comenzarán a numerarse los elementos de la lista.
<code></code>	Marca cada uno de los elementos de una lista.


<div>	Se utiliza para estructurar una página en secciones o divisiones
	Permite agrupar un texto sin separarlo del texto anterior (se suele usar para asignar estilos)
<iframe>	Permite insertar una página web dentro de otra

Prácticamente todos los elementos, independientemente del tipo que sean, contienen los siguientes atributos:

- “style” permite definir características del formato del elemento. Definiendo el estilo como un atributo de un elemento, ese estilo solo se aplicará a ese elemento concreto, mientras que si lo definimos empleando el elemento “style” de la cabecera, ese estilo se aplicará a todos los elementos de ese tipo que se incluyan en la página.
- “class” permite definir elementos de la misma clase a los que se asignará el mismo estilo.
- “id” permite asignar un nombre o identificador único a cada elemento de la página para poder hacer referencia a él (se usar sobre todo al definir contenido dinámico con JavaScript para poder realizar cambios sobre un elemento concreto de la página)

1.2.3.3 Formularios

Etiqueta	Descripción	
<form>	Se emplea para insertar un formulario en la página	
	Atributo	Descripción
	action = "url"	Indica la URL que se encarga de procesar los datos del formulario
	method = "POST" o "GET"	Indica el método HTTP utilizado para enviar el formulario
<input>	Se usa para insertar un control en un formulario	
	Atributo	Descripción
	type = "text password checkbox radio submit reset file hidden image button"	Indica el tipo de control que se emplea en el formulario
	text	<p>Nome</p> <input type="text"/>
	password	<p>Contraseña</p> <input type="password"/>
	checkbox	<p>Postos de trabajo buscados</p> <p><input type="checkbox"/> Dirección <input type="checkbox"/> Técnico <input type="checkbox"/> Empleado</p>
	radio	<p>Sexo</p> <p><input checked="" type="radio"/> Home <input type="radio"/> Muller</p>

	submit	<div>Procurar</div> <div>Botón de envío del formulario al servidor.</div>
	reset	<div>Borrar datos do formulario</div> <div>Botón que devuelve el formulario a su estado inicial.</div>
	file	<div>Escolla arquivo...<div>Seleccionar archivo</div>Ningún archivo seleccionado</div>
	hidden	Se utiliza para añadir información oculta en el formulario.
	image	<div></div> <div>Botón personalizado con una imagen.</div>
	button	<div>Gardar Cambios</div> <div>Botón.</div>
	name = "texto"	Asigna un nome al control (imprescindible para que el servidor pueda procesar el formulario).
	value = "texto"	Valor inicial del control.
alt = "texto"	Descripción del control	
<textarea>	Se utiliza para incluir un área de texto en un formulario.	
	Atributo	Descripción
	rows = "número"	Número de filas que mostrará el área de texto.
	cols = "número"	Número de columnas que mostrará el área de texto.
<select>	Se utiliza para incluir una lista desplegable en un formulario.	
	Atributo	Descripción
	size = "número"	Número de filas de la lista que se muestran (por defecto, una).
	multiple = "multiple"	Si se incluye, permite seleccionar más de un elemento.
<option>	Se utiliza para definir cada elemento de una lista desplegable.	
	Atributo	Descripción
	selected = "selected"	Indica si el elemento aparece seleccionado por defecto al cargarse la página.
	value = "texto"	Valor que se envía al servidor cuando el usuario elige dicha opción.

1.3 XHTML

XHTML (eXtensible Hyper Text Markup Language) es una adaptación de HTML para definir HTML con formato de XML. Mantiene prácticamente todas las etiquetas de HTML pero con algunas restricciones propias de XML.

XHTML es más restrictivo que HTML, pero es compatible con prácticamente todos los navegadores.

1.3.1 Estructura básica

La estructura de una página XHTML es similar a la de una página HTML pero con dos variaciones:

- Es obligatorio incluir la etiqueta DOCTYPE para indicar la DTD que se usará para validar el documento. Una DTD (Document type definition) es un documento utilizado en XML para definir las etiquetas que se pueden utilizar, los atributos de cada etiqueta y el tipo de datos de los valores que puede tomar cada atributo. Para comprobar si una página cumple con todas las normas definidas en su DTD (lo que se denomina validar el documento), se pueden utilizar validadores online como el del w3c: <http://validator.w3.org>.

En XHTML5, la etiqueta doctype que se utiliza es la siguiente:

```
<!DOCTYPE html>
```

- En la etiqueta HTML es necesario indicar el atributo xmlns, que determina el espacio de nombres asociado al documento. Un espacio de nombres en un documento XML permite diferenciar las etiquetas y atributos que pertenecen a cada lenguaje. En XHTML5 el xmlns es "<http://www.w3.org/1999/xhtml>".

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Ejemplo de página XHTML</title>
  </head>
  <body>
    <p>Hola mundo</p>
  </body>
</html>
```

1.3.2 Diferencias con HTML

- En la estructura:
 - La etiqueta XHTML DOCTYPE es obligatoria, para indicar la dtd que usará el documento.
 - El atributo xmlns del elemento html es obligatorio
 - Los elementos "html", "head", "title" y "body" son obligatorios.
- En los elementos:
 - Todos los documentos deben estar correctamente anidados
 - Todos los elementos deben estar siempre cerrados
 - Los elementos se definen siempre en minúsculas

- Los documentos xhtml deben tener un elemento raíz
- Desaparecen las etiquetas de HTML y <center>
- El aspecto del documento se debe definir siempre por medio de css.
- En los atributos
 - Los nombres de los atributos deben estar en minúsculas
 - Los valores de los atributos deben ser definidos siempre entre comillas (incluso los números)
 - No es posible minimizar atributos (checked="checked")

1.3.3 Ventajas de XHTML

- XHTML es totalmente compatible con navegadores antiguos: la información se visualiza, aunque puede que sin formato.
- Facilita el mantenimiento: todas las etiquetas están en minúsculas, los valores entrecomillados y el documento está correctamente estructurado, lo que evita errores y facilita el mantenimiento de la página.
- A los documentos XHTML se les pueden aplicar transformaciones XSLT, que permiten la conversión de un documento XML en otro con una nomenclatura o formato diferente (RSS, PDF, etc). Esto permite que un mismo documento pueda adoptar diseños totalmente distintos en diferentes aparatos, o incluso distintos diseños para un mismo medio de visualización.
- XHTML es un formato abierto y compatible con los nuevos estándares que está desarrollando el W3C. Esto facilita que los documentos puedan ser transformados directamente a distintas especificaciones aplicándoles una XSLT.
- Los documentos escritos conforme a XHTML 1.0 pueden potencialmente presentar mejor rendimiento en las actuales herramientas web que aquellos escritos conforme a HTML.

1.4 HTML5

La última versión importante publicada de HTML es la versión 5, en la que se actualizan tanto el estándar de HTML como el de XHTML. En esta versión se implementa de forma nativa muchas

utilidades que anteriormente era necesario realizar con Javascript. No es que se incorpore programación en el HTML, sino que se incorporan nuevas etiquetas y serán los navegadores los que tendrán que incorporar la funcionalidad necesaria para interpretar esas etiquetas (es decir, es muy dependiente de que los navegadores lo interpreten bien, por eso las versiones antiguas de los navegadores no son compatibles con HTML5).

El gran impulso para HTML5 ha sido que Apple decidiera que sus dispositivos no visualizasen flash, con lo cual HTML pasó a asumir la funcionalidad que aportaba flash, especialmente en cuanto a la visualización del contenido multimedia.

Entre las principales novedades destacan:

- Nuevas etiquetas de estructura de la página, orientadas a CMS (content manager system, sistema de gestión de contenidos):

Etiqueta	Descripción	
<header>	Define la cabecera de la página.	<header>
<nav>	Define la sección de menús de navegación (enlaces a otras partes de la página o a otras páginas)	<nav>
<footer>	Define el pie de página o sección.	<footer>
<section>	Define una sección general dentro de un documento (un grupo de contenido con una temática común).	<section>
<article>	Define un artículo: una composición completa, autocontenida en un documento que puede ser distribuida o reutilizada de forma independiente.	<article>
<aside>	Define un contenido de la página distinto del que lo rodea. Se utiliza, por ejemplo, para contenido publicitario, otros elementos de navegación, etc.	<aside>

- Otras etiquetas nuevas, especialmente para elementos gráficos y multimedia

Tipo	Descripción	
<datalist>	En los <input> de tipo text, email, tel y url permite desplegar un listado con diferentes valores asociados a ese input para que el usuario seleccione uno de ellos.	Estudiante a cualificar
<svg>	Se utiliza para definir los gráficos en una web	21456987-F 78415963-G 78159357-X
<canvas>	Se utiliza para definir un lienzo que contendrá gráficos.	Nuno Dantas Antia Dapena Unio Fernandez
<embed> y <object>	Definen un contenedor para una aplicación externa (plug-in), con lo que permiten incluir un documento dentro de otro.	
<audio> y <video>	Especifican un modo normalizado de incluir audio y vídeo en una página web, sin necesidad de aplicaciones externas (plug-ins).	
	Atributo	Descripción

	src	Permite indicar la ruta del archivo.
	autoplay	Indica que el archivo debe reproducirse de forma automática al cargarse la página.
	controls	Indica que se deben mostrar los controles para iniciar, detener y pausar el archivo.
	loop	Indica que el archivo debe volver a reproducirse otra vez cuando acabe.
	autobuffer	Indica que el archivo debe descargarse completamente antes de empezar a reproducirse.
	width	Indica el ancho (en píxeles) de la pantalla en la que se reproduce el vídeo.
	height	Indica la altura (en píxeles) de la pantalla en la que se reproduce el vídeo.
<source>		Permite especificar distintos archivos con distintos formatos por si el navegador que usa el usuario no permite algún formato concreto

- Nuevos valores para el atributo type del elemento <input> de los formularios. Estos nuevos tipos de entradas incorporan ciertas validaciones de datos sin recurrir a Javascript.

Tipo	Descripción
email	Permite introducir un mail.
url	Permitir introducir una URL.
tel	Permitir introducir un teléfono.
search	Campo de búsqueda de información.
number	Permite introducir un número. Se pueden usar atributos adicionales para definir el máximo, mínimo y el intervalo entre un valor y el siguiente.
range	Permitir introducir un número entre un máximo y un mínimo mediante una barra de desplazamiento.
color	Permite seleccionar un color.
date	Permite seleccionar una fecha en un calendario.
month	Permite seleccionar un mes en un calendario.
week	Permite seleccionar una semana en un calendario.
time	Permite seleccionar una hora.
datetime	Permite seleccionar fecha y hora internacional.
datetime-local	Permite seleccionar fecha y hora local.

- Nuevos atributos

Atributo	Descripción
placeholder	Muestra un texto que aparecerá por defecto en las etiquetas <input> y que desaparece cuando se selecciona el elemento ("Introduzca aquí su teléfono").
autofocus	Indica el elemento del formulario que estará seleccionado por defecto al cargar la página. Solo puede haber uno por página (sino tendría en cuenta solo el primero).
autocomplete	Indica si en este elemento está o no activa la opción de autocompletado.
required	Indica que el campo es obligatorio. Si no se indicase un valor para el campo, mostraría un error al enviar el formulario.
form	Indica a que formulario pertenece un campo, lo que permite colocar elementos de formulario fuera de las etiquetas <form>
max	Indica el valor máximo que puede tomar un campo numérico.
min	Indica el valor mínimo que puede tomar un campo numérico.
step	Indica el intervalo entre distintos valores en un campo numérico.

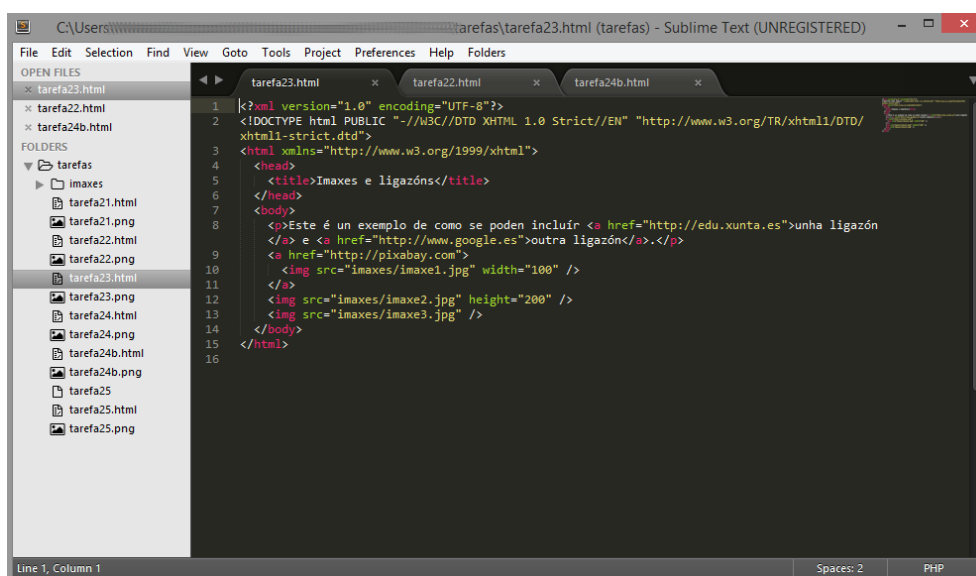
- Permite gestionar los elementos de la página: hacer drag and drop, manejar el foco, ...

- Integra elementos de geolocalización
- Permite trabajar fuera de línea y almacenar datos en local (creando una “mini base de datos” que se guardará en local)

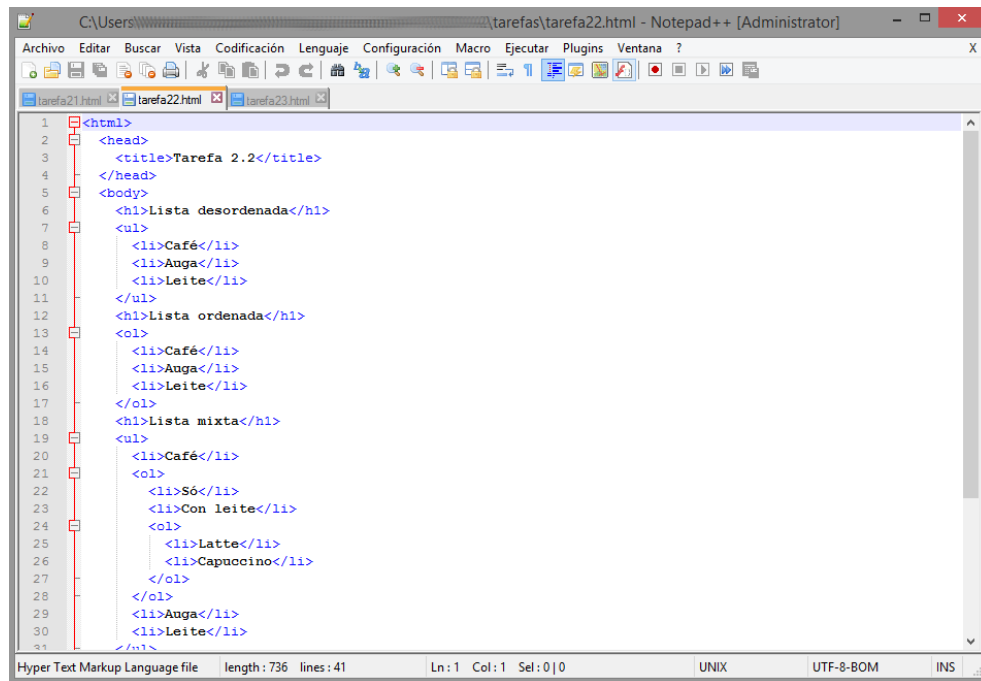
1.5 Herramientas de diseño web

1.5.1 Editores de HTML

- **Sublime Text:** editor de código multiplataforma. Es software propietario, pero puede usarse de forma gratuita (se mostrará un mensaje cada cierto número de usos). Es un editor muy popular para el diseño web ya que ofrece muchas características útiles como el minimapa, multi selección, multi cursor, distintas disposiciones de pantalla, soporte nativo para infinidad de lenguajes, configuración del resaltado de la sintaxis, búsqueda dinámica, auto completado y marcado de llaves y paréntesis, configuración total de atajos de teclado, etc. Además ofrece una API (interface de programación de aplicaciones) en Python que permite la creación de nuevos complementos personalizados para ampliar sus funcionalidades.



- **Notepad++:** es un editor de texto y código de software libre para Windows que soporta, entre otras, las siguientes funcionalidades: auto completado, marcadores, resaltado de sintaxis, resaltado de llaves, búsqueda por expresiones regulares, configuración de atajos de teclado, modo columna, etc.



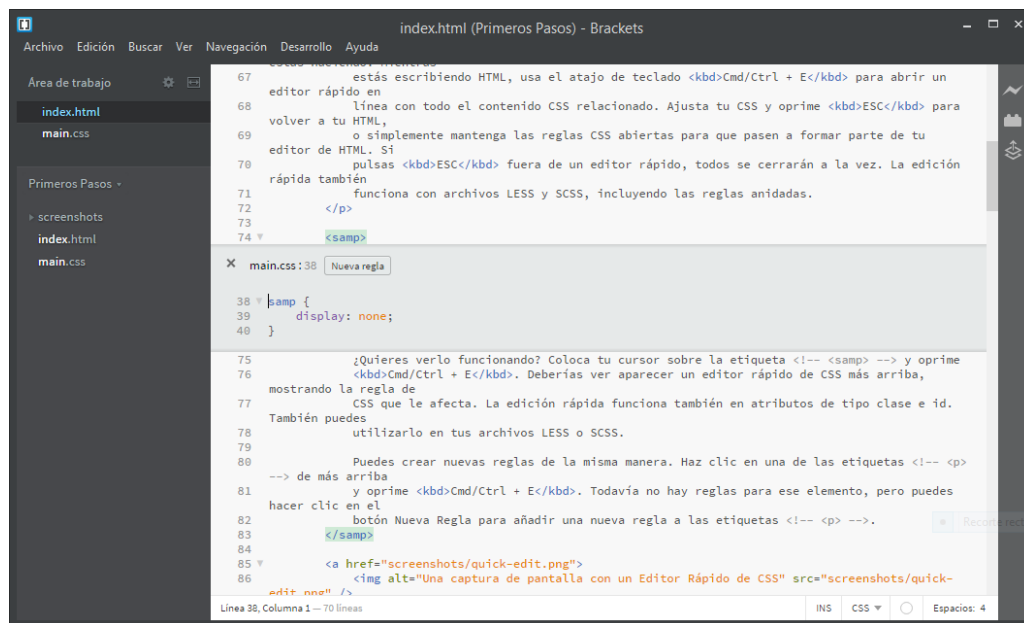
The screenshot shows the Notepad++ editor window titled 'tarefas\tarefa22.html - Notepad++ [Administrator]'. The editor contains the following HTML code:

```
1 <html>
2 <head>
3   <title>Tarefa 2.2</title>
4 </head>
5 <body>
6   <h1>Lista desordenada</h1>
7   <ul>
8     <li>Café</li>
9     <li>Anga</li>
10    <li>Leite</li>
11  </ul>
12  <h1>Lista ordenada</h1>
13  <ol>
14    <li>Café</li>
15    <li>Anga</li>
16    <li>Leite</li>
17  </ol>
18  <h1>Lista mixta</h1>
19  <ul>
20    <li>Café</li>
21    <ol>
22      <li>Só</li>
23      <li>Con leite</li>
24    </ol>
25    <li>Latte</li>
26    <li>Capuccino</li>
27  </ul>
28  <ol>
29    <li>Anga</li>
30    <li>Leite</li>
31  </ol>
```

The status bar at the bottom indicates: 'Hyper Text Markup Language file', 'length: 736', 'lines: 41', 'Ln: 1 Col: 1 Sel: 0 | 0', 'UNIX', 'UTF-8-BOM', and 'INS'.

- **Brackets:** es un editor de código escrito en HTML, CSS y JavaScript enfocado al desarrollo web. Es software libre multiplataforma y fue creado por Adobe, aunque es mantenido actualmente en GitHub.

Presenta una serie de ventajas para su utilización en desarrollo de aplicaciones web, como el acceso a cada regla CSS a partir de la etiqueta HTML, clase o identificador; vista previa dinámica en el navegador Chrome para los cambios en HTML y CSS; visor de colores, gradientes e imágenes integrado; así como otras características ya habituales en los editores de código (resaltado de sintaxis, resaltado de llaves, etc.).



1.5.2 Herramientas de desarrollo de los navegadores

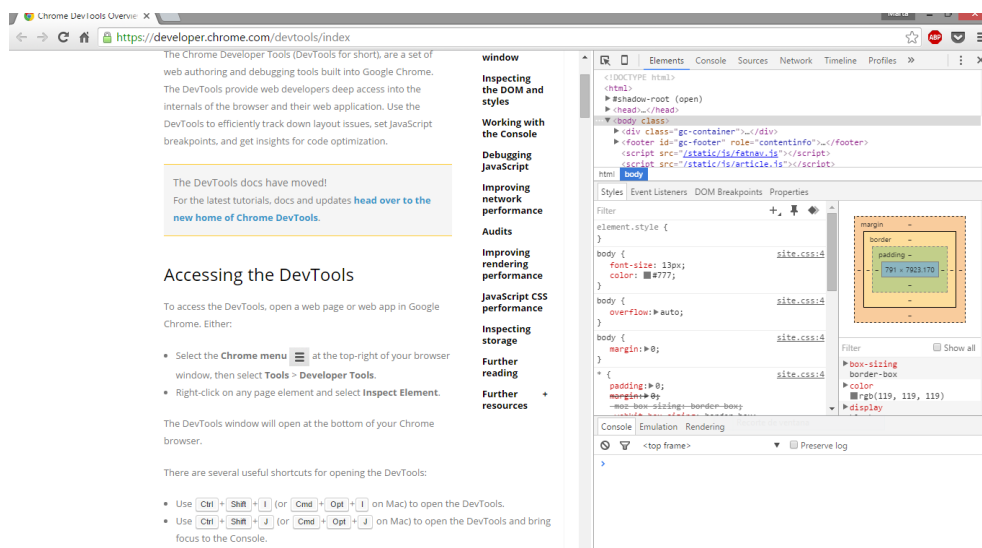
Los propios navegadores web también disponen de sus propias herramientas para la edición de código HTML de forma ágil, y en tiempo real.

Las herramientas de desarrollo de Chrome⁵ (DevTools) son un conjunto de herramientas de auditoría y depuración dentro del propio navegador que permiten a los desarrolladores tener acceso al interior del navegador y de su aplicación web. Su uso eficiente permite resolver asuntos de formato, manipular elementos DOM, estilos CSS, JavaScript y optimizar el código, entre otras cosas.

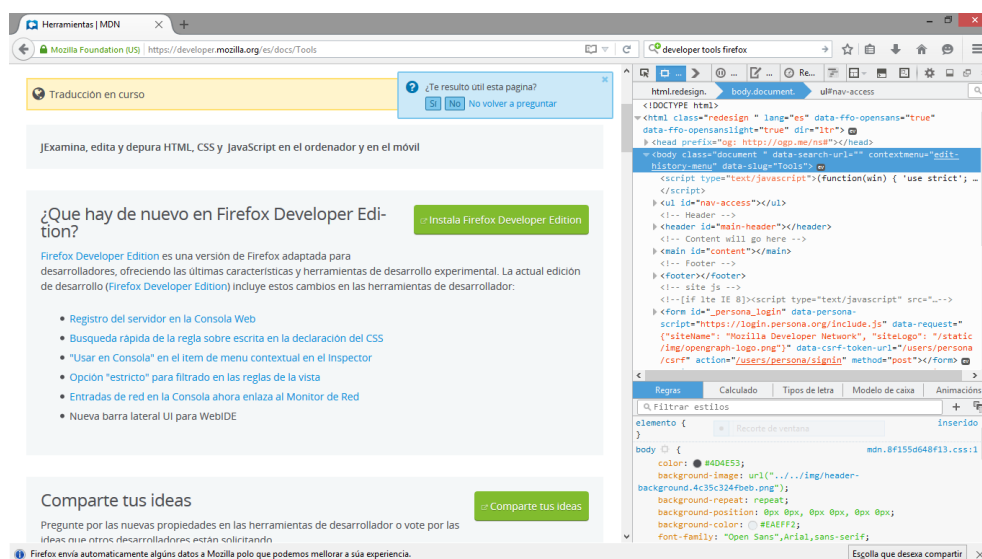
Se puede acceder a estas herramientas pulsando la tecla F12, a través del menú del navegador, o haciendo clic derecho en cualquier elemento de la página y después en "Inspeccionar elemento".

Una vez activadas, encontramos varias solapas, con los elementos, consola, archivos fuente, etc, donde podemos modificar el código HTML, los estilos CSS o probar el código JavaScript.

⁵ <https://developers.google.com/web/tools/chrome-devtools>



Firefox⁶ también dispone de herramientas para el desarrollo web.



1.5.3 Herramientas WYSIWYG

Para publicar páginas web no es necesario disponer de conocimientos de HTML. Hay editores, los llamados WYSIWYG (What You See Is What You Get, lo que ves es o que obtienes), que permiten definir directamente el código que se quiera publicar, dándole formato, insertando enlaces o elementos multimedia, etc. Normalmente estos editores son herramientas web, y se utilizan

⁶ <https://developer.mozilla.org/es/docs/Tools>

normalmente en gestores de contenidos, blogs o redes sociales para que los usuarios puedan realizar publicaciones de forma sencilla.

Su principal desventaja es que son mucho más limitados que si insertásemos el código directamente (habrá muchas etiquetas que no se puedan usar, y atributos que no podamos definir), con lo cual el diseño normalmente suele ser bastante simple.

ANEXO I: Materiales

I. Textos de apoyo o de referencia

- J. M. CASTRO RAMOS E J. R. RODRÍGUEZ SÁNCHEZ. Lenguajes de marcas y sistemas de gestión de información. Editorial Garceta. Ed. 2012
- FRANCISCO JAVIER SANCHEZ ZURDO, PABLO TOHARIA RABASCO e LAURA RAYA GONZÁLEZ. Lenguajes de marcas y sistemas de gestión de información. Editorial Ra-Ma. Ed. 2011
- ELENA TORRES MARÍA. Lenguaje de Marcas y Sistemas de Información
<http://jmlenguajesdemarcas.blogspot.com.es/>
- [https://manuais.iessanclemente.net/index.php/\(X\)HTML](https://manuais.iessanclemente.net/index.php/(X)HTML)
- <https://www.w3schools.com/html/default.asp>

II. Recursos didácticos

- Apuntes en el aula virtual.
- Ordenador personal, con navegador web y conexión a internet.
- Software para elaboración de documentos de texto.