

Lenguajes de marcas y sistemas de gestión de información

Con acceso
al libro interactivo



Juan Carlos Moreno Pérez
Sergio Luis González Ruiz

Tercera edición

Lenguajes de marcas y sistemas de gestión de información

Juan Carlos Moreno Pérez
Sergio Luis González Ruiz

(tercera edición)



© Juan Carlos Moreno Pérez
Sergio Luis González Ruiz

© EDITORIAL SÍNTESIS, S. A.
Vallehermoso, 34. 28015 Madrid
Teléfono 91 593 20 98
www.sintesis.com

ISBN: 978-84-1357-481-3
Depósito Legal: M-7.591-2026

Impreso en España - Printed in Spain

Reservados todos los derechos. Está prohibido, bajo las sanciones penales y el resarcimiento civil previstos en las leyes, reproducir, registrar o transmitir esta publicación, íntegra o parcialmente, por cualquier sistema de recuperación y por cualquier medio, sea mecánico, electrónico, magnético, electroóptico, por fotocopia o por cualquier otro, sin la autorización previa por escrito de Editorial Síntesis, S. A.

ÍNDICE

1. Reconocimiento de las características de los lenguajes de marcas	RA1
Resultado de aprendizaje y criterios de evaluación	10
Objetivos de Desarrollo Sostenible	10
Mapa conceptual	11
Glosario	12
Punto de partida	12
1.1. Introducción	13
1.2. Clasificación y características de los lenguajes de marcas	15
1.2.1. Clasificación de los lenguajes de marcas	15
1.2.2. Características y ventajas en el tratamiento de la información	17
1.3. Identificación de ámbitos de aplicación	19
1.4. Estructura y sintaxis de un documento	20
1.4.1. Estructura jerárquica del documento	20
1.4.2. Reglas sintácticas (bien formado)	21
1.5. Herramientas de edición	22
1.5.1. Bloc de notas Windows	22
1.5.2. Gedit	23
1.5.3. Notepad++	24
1.6. Elaboración de documentos bien formados	24
1.7. Utilización de espacios de nombres	27
Ideas clave	30
Aplica lo aprendido	31
Solución del punto de partida	32
Práctica profesional	33
Autoevaluación	33

2. Utilización de lenguajes de marcas en entorno web

RA2

Resultado de aprendizaje y criterios de evaluación	34
Objetivos de Desarrollo Sostenible	34
Mapa conceptual	35
Glosario	36
Punto de partida	36
2.1. Introducción	36
2.2. Herramientas de diseño web	37
2.2.1. KompoZer	38
2.2.2. Adobe Dreamweaver CC	38
2.2.3. Bluefish	39
2.2.4. HTML Online Editor	39
2.2.5. VS Code (Visual Studio Code)	39
2.2.6. Sublime Text	40
2.3. Identificación de etiquetas y atributos de HTML	40
2.3.1. Versiones de HTML. Semejanzas y diferencias	40
2.3.2. Estructura de un documento HTML	40
2.3.3. Declaración de tipo de documento	42
2.3.4. Etiquetas y atributos HTML	44
2.3.5. HTML5	64
2.3.6. Las nuevas etiquetas HTML5	65
2.4. Hojas de estilo CSS	72
2.4.1. Versiones de CSS	74
2.4.2. Sintaxis básica	75
2.4.3. Formas de aplicar estilos a un documento HTML	86
2.4.4. CSS3	90
2.4.5. Las nuevas propiedades de CSS3	91
2.5. Validación de documentos HTML y CSS	103
2.5.1. Validación HTML	104
2.5.2. Validación CSS	105
2.6. Sindicación de contenidos (tecnologías y ámbito de aplicación)	106
2.6.1. Tecnologías y estándares de canales de contenidos	107
2.6.2. Ámbito de aplicación	113
Ideas clave	115
Ejercicios prácticos resueltos	117
Aplica lo aprendido	119
Solución del punto de partida	122
Práctica profesional	125
Autoevaluación	125

3. Acceso y manipulación de documentos web utilizando lenguajes de *script* RA3

Resultado de aprendizaje y criterios de evaluación	126
Objetivos de Desarrollo Sostenible	126
Mapa conceptual	127
Glosario	128
Punto de partida	128
3.1. Introducción	128
3.2. Los lenguajes de <i>script</i>	129
3.3. Sintaxis básica de JavaScript	131
3.4. Modificación dinámica de una página web	134
3.4.1. Acceso y modificación de nodos	136
3.4.2. Creación de nodos	137
3.4.3. Borrado de nodos	138
3.4.4. Modificación dinámica del estilo de una página web	140
Ideas clave	142
Aplica lo aprendido	143
Solución del punto de partida	144
Práctica profesional	145
Autoevaluación	145

4. Definición de esquemas y vocabularios RA4

Resultado de aprendizaje y criterios de evaluación	146
Objetivos de Desarrollo Sostenible	146
Mapa conceptual	147
Glosario	148
Punto de partida	148
4.1. Introducción	148
4.2. Definición de la estructura y sintaxis de documentos XML	150
4.2.1. Instrucciones de procesamiento o prólogo	152
4.2.2. Marcas o etiquetas	153
4.2.3. Elementos	154
4.2.4. Atributos	155
4.2.5. Comentarios	156
4.2.6. Sección CDATA	157
4.3. Creación, asociación y elementos de DTD y XSD	158
4.3.1. DTD (definición de tipo de documento)	159
4.3.2. XML Schema XSD	169
4.4. Herramientas de creación y edición XML	185
4.4.1. XML Copy Editor	185
4.4.2. Altova XMLSpy	185

4.4.3. XMLSpear	186
4.4.4. XML Notepad	186
4.4.5. Online XML Editor	187
4.5. Validación XML	187
4.6. JSON y JSON Schema	190
4.6.1. Estructuras básicas JSON	190
4.6.2. Sintaxis y estructura. Reglas básicas	191
4.6.3. Herramientas para trabajar con JSON	191
4.6.4. Validación con JSON Schema	192
Ideas clave	194
Ejercicios prácticos resueltos	197
Aplica lo aprendido	200
Solución del punto de partida	204
Práctica profesional	205
Autoevaluación	205

5. Conversión y adaptación de documentos de intercambio de información **RA5**

Resultado de aprendizaje y criterios de evaluación	206
Objetivos de Desarrollo Sostenible	206
Mapa conceptual	207
Glosario	208
Punto de partida	208
5.1. Introducción	209
5.2. Técnicas de transformación de documentos XML	211
5.2.1. Diferencias entre las versiones de XSLT 1.0, 2.0 y 3.0	212
5.3. Formatos de salida y ámbitos de aplicación de XSLT	213
5.4. Descripción de la estructura y la sintaxis	213
5.4.1. Fichero XML	213
5.4.2. Hoja de estilo XSL	214
5.5. Utilización de plantillas XSL	215
5.5.1. <xsl:value-of>	216
5.5.2. <xsl:for-each>	216
5.5.3. Filtrando contenido	218
5.5.4. Ordenando el contenido	218
5.5.5. Selección avanzada IF	219
5.5.6. Selección avanzada CHOOSE	220
5.6. Utilización de herramientas de procesamiento: verificación, depuración y generación de documentación	221
5.6.1. Oxygen XML Editor	222
5.6.2. Altova Editor XSLT	222

5.6.3. Saxon	223
5.6.4. XPontus XML Editor	223
5.6.5. Python como herramienta de transformación JSON	224
5.6.6. Herramientas <i>online</i>	224
Ideas clave	226
Ejercicios prácticos resueltos	228
Aplica lo aprendido	230
Solución del punto de partida	232
Práctica profesional	234
Autoevaluación	235

6. Almacenamiento y gestión de la información de intercambio

RA6

Resultado de aprendizaje y criterios de evaluación	236
Objetivos de Desarrollo Sostenible	236
Mapa conceptual	237
Glosario	238
Punto de partida	238
6.1. Introducción	238
6.2. Sistemas de almacenamiento de información XML	240
6.2.1. Ficheros	240
6.2.2. Bases de datos XML habilitadas o <i>enabled</i>	240
6.2.3. Bases de datos XML nativas	240
6.3. Inserción y extracción de información en XML	242
6.4. Búsqueda de información en documentos XML: lenguajes de consulta y manipulación	243
6.4.1. XPath	243
6.4.2. XQuery	250
6.5. Almacenamiento XML nativo	251
6.5.1. Instalación y ejecución de una base de datos XML nativa: eXist-db	252
6.5.2. BaseX	252
6.6. Herramientas de tratamiento y almacenamiento de información en formato XML	253
6.7. Herramientas de tratamiento y almacenamiento de información en formato JSON	254
6.7.1. Instalación de un servidor JSON	255
Ideas clave	259
Ejercicios prácticos resueltos	260
Aplica lo aprendido	261
Solución del punto de partida	262
Práctica profesional	263
Autoevaluación	263

7. Sistemas empresariales de gestión de la información

RA7

Resultado de aprendizaje y criterios de evaluación	264
Objetivos de Desarrollo Sostenible	264
Mapa conceptual	265
Glosario	266
Punto de partida	266
7.1. Introducción	267
7.1.1. Los CRM	268
7.2. ERP	269
7.2.1. Integración de módulos	272
7.2.2. Implantación, adaptación y configuración	278
7.2.3. Importación y exportación de información: generación de informes	279
7.3. Minería de datos	280
7.3.1. El proceso de descubrimiento de conocimiento y la minería de datos	281
7.4. OLAP	283
7.5. <i>Dashboard</i> y <i>scorecard: business intelligence</i> e informes	284
7.5.1. La evolución de los informes	284
7.6. Seguridad en sistemas de gestión empresarial	284
Ideas clave	286
Aplica lo aprendido	287
Solución del punto de partida	288
Práctica profesional	289
Autoevaluación	289

2

Utilización de lenguajes de marcas en entorno web

RESULTADO DE APRENDIZAJE Y CRITERIOS DE EVALUACIÓN

RA 2. Utiliza lenguajes de marcas para la transmisión y presentación de información a través de la web analizando la estructura de los documentos e identificando sus elementos.

- a) Se han identificado y clasificado los lenguajes de marcas relacionados con la web y sus diferentes versiones y estándares.
- b) Se ha analizado la estructura de un documento HTML e identificado las secciones que lo componen.
- c) Se ha reconocido la funcionalidad de las principales etiquetas y los atributos del lenguaje HTML.
- d) Se han establecido las semejanzas y diferencias entre las diferentes versiones de HTML.
- e) Se han utilizado herramientas en la creación de documentos web.
- f) Se han identificado las ventajas que aporta la utilización de hojas de estilo.
- g) Se han aplicado hojas de estilo.
- h) Se han validado documentos HTML y CSS.
- i) Se han identificado las tecnologías en que se basa la sindicación de contenidos.
- j) Se han reconocido los ámbitos de aplicación de la sindicación de contenidos.



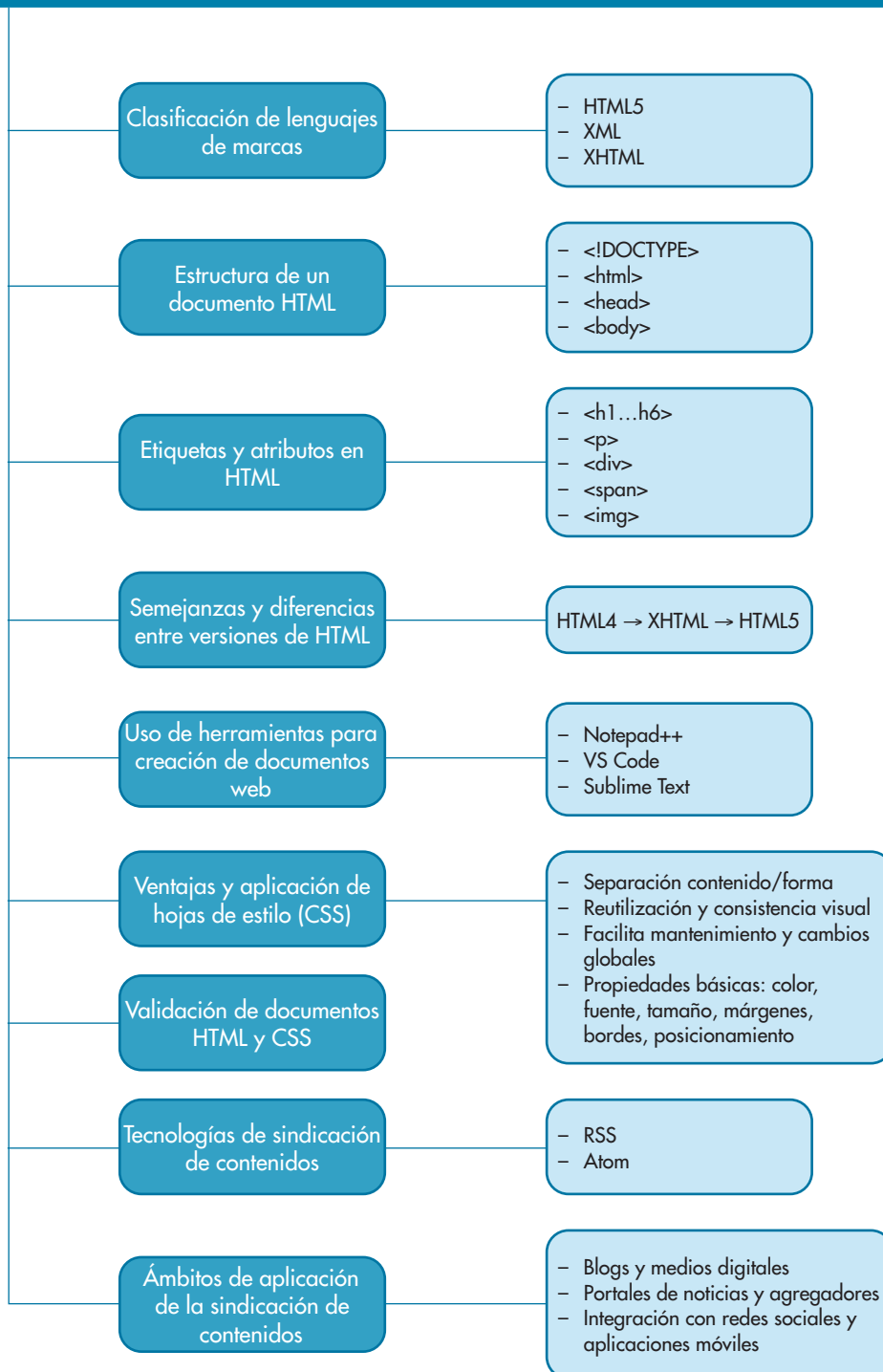
Objetivos de Desarrollo Sostenible

En este capítulo se van a trabajar los ODS 7 y 14.



MAPA CONCEPTUAL

UTILIZACIÓN DE LENGUAJES DE MARCAS EN ENTORNO WEB





GLOSARIO

- AJAX.** Sigla de *asynchronous JavaScript and XML*. Técnica de programación web de tecnología asíncrona, creada en el 2005 por Jesse James Garrett, que permite obtener respuestas en segundo plano que vienen del servidor sin tener que recargar la página completa.
- CSS.** Sigla inglesa que significa “hojas de estilo en cascada”. Lenguaje de diseño gráfico que se emplea para establecer el diseño visual de los documentos web.
- Emmet.** Herramienta de desarrollo web que utiliza abreviaturas para escribir código HTML y CSS de manera más rápida y eficiente.
- OS/2.** Sistema operativo desarrollado por IBM, aunque en sus comienzos también participó Microsoft, cuyas siglas se interpretan como “sistema operativo de segunda generación”.
- SASS.** Sigla de *syntactically awesome style sheets*. Metalenguaje de CSS evolucionado a un lenguaje de script, SassScript.



PUNTO DE PARTIDA

Una pequeña empresa local necesita una página web básica para presentar sus servicios, mostrar información de contacto y publicar noticias o novedades. Recadero, alumno en prácticas, recibe el encargo de crear esta web utilizando HTML y CSS. Para ello, debe saber estructurar correctamente el contenido (encabezados, secciones, menús, imágenes, párrafos), aplicar estilos coherentes y asegurarse de que el código cumple los estándares web.

Tras haber trabajado los contenidos del módulo, ¿será capaz Recadero de analizar la estructura de un documento HTML, utilizar las etiquetas adecuadas, aplicar hojas de estilo y emplear herramientas que aseguren una presentación y un funcionamiento correctos del sitio web?

2.1. Introducción

La transmisión y presentación de información a través de la web se basa en el uso de lenguajes de marcas capaces de estructurar, describir y dar formato a los contenidos, siendo HTML y CSS los pilares fundamentales para construir páginas web, definir su estructura y controlar su apariencia.

Comprender sus elementos, versiones, etiquetas y herramientas asociadas permite crear documentos correctos, accesibles y compatibles con los estándares actuales. Además, tecnologías como la sindicación de contenidos facilitan la distribución automatizada de información en distintos entornos digitales, ampliando el alcance y el dinamismo de los sitios web.

Cuando se usan los navegadores para consultar las distintas web, se aprecia una serie de páginas que contienen elementos como enlaces, texto, imágenes, vídeos, etc. Si se accede al código fuente de la página, se observa que el HTML (*hypertext markup language*) es el componente básico de la web, junto con otras tecnologías entre las que destacan CSS, AJAX, JavaScript, etc.

HTML es un lenguaje de marcado que se usa para la creación de páginas web y tuvo su origen en 1991. Hoy en día, se encuentra disponible el primer documento que se publicó.



SABÍAS QUE...

HTML se define como un lenguaje de etiquetas.

Para poder crear cualquier documento, es necesario disponer de un editor de texto plano. Existen herramientas de escritorio gratuitas y comerciales que facilitan este trabajo. Si no se desea instalar ni usar ninguna herramienta de escritorio, existen herramientas *online* que permiten crear y editar este tipo de documentos.

De HTML, se estudiará su estructura y sintaxis, así como sus elementos principales (listas, tablas, formato de texto, párrafos, cabeceras, formularios, etc.).

Elaborar una web sin un diseño atractivo es algo impensable. En este punto, es cuando entran en juego las hojas de estilo en cascada (CSS), que describen la manera de visualizar una página web en la pantalla.

La clasificación de los lenguajes de marcas web se pueden realizar según distintos criterios:

- Según su propósito:
 - Presentación de contenido: HTML, XHTML, SVG, MathML.
 - Estructura e intercambio de datos: XML.
 - Sindicación de contenidos: RSS, Atom.
- Según su origen o modelo sintáctico:
 - Basados en SGML: HTML clásico.
 - Híbridos o derivados modernos: HTML5.
 - Basados en XML: XHTML, SVG, MathML, RSS, Atom.
- Según su ámbito de aplicación:
 - Web general: HTML, HTML5, XHTML.
 - Datos y backend: XML.
 - Gráficos y diseño: SVG.
 - Ciencia y educación: MathML.

Los lenguajes de marcas web evolucionan mediante versiones y estándares publicados por organismos, como el W3C, que se verán a lo largo del presente capítulo

2.2. Herramientas de diseño web

En este apartado, se dará un repaso a las herramientas de diseño web más utilizadas por los desarrolladores.

Algunos editores son superiores a un simple editor de texto como el Bloc de notas de Windows, porque incluyen funcionalidades específicas para la programación:

- *Resaltado de sintaxis*: cambian el color del texto según su función (etiquetas HTML, propiedades CSS, variables JavaScript), haciendo que el código sea mucho más fácil de leer para detectar errores visualmente.
- *Autocompletado*: sugieren nombres de etiquetas, propiedades o funciones mientras escribes, lo que acelera el proceso de codificación y reduce la posibilidad de errores tipográficos.
- *Plegado de código*: permite ocultar secciones de código (como una función o un bloque <div> grande) para centrarte en otras áreas, mejorando la navegación en archivos grandes.
- *Temas y personalización*: ofrecen la posibilidad de cambiar el esquema de color, las fuentes y la apariencia general para adaptarse a las preferencias del usuario.
- *Extensiones/plug-ins (especialmente vs. code y sublime text)*: permiten añadir funcionalidades específicas.

2.2.1. KompoZer

Software para el diseño web de código abierto, de fácil uso, que permite la administración de un sitio FTP, así como personalizar la barra de herramientas. Trae integrado un editor de CSS y un validador HTML. Disponible en varios idiomas, permite la descarga para sistemas operativos Windows, Linux, OS/2 y MacOSX. Este proyecto lleva años sin recibir actualizaciones y se considera discontinuado.

2.2.2. Adobe Dreamweaver CC

Software comercial específico para diseñar y desarrollar sitios web basados en estándares con una interfaz moderna.

Entre sus características, destacan las siguientes: permite la previsualización en varios dispositivos; arquitectura 64 bits; compatible con PHP 7, JavaScript, GIT y CEF, y trae un espacio de trabajo para desarrolladores que es compatible con procesadores de CSS como SASS, LESS y SCSS.

Fue creado inicialmente por Macromedia, y la empresa que lo proporciona actualmente es Adobe Systems.



RECURSO WEB

A través de estos códigos QR podrás descargar la versión de prueba de Adobe Dreamweaver CC (QR1) y Bluefish (QR2), y usar HTML Online Editor (QR3).



QR1



QR2



QR3

2.2.3. Bluefish

Bluefish es un editor web *open source* multiplataforma con licencia GNU. Aunque las primeras versiones estaban plagadas de fallos y su funcionalidad era muy justa, las versiones actuales lo hacen una alternativa fiable a las plataformas de pago.

2.2.4. HTML Online Editor

Programa de editor HTML WYSIWYG gratuito que permite editar código fuente en línea sin descargar ninguna aplicación. Permite trabajar con documentos HTML, CSS y JS.

El funcionamiento es muy intuitivo. En el panel derecho, el usuario va introduciendo las líneas de código HTML y, de forma concurrente, el resultado aparecerá en el panel izquierdo. También puede trabajarse sobre el documento de forma visual, y el código HTML asociado aparecerá en el panel derecho.

HTML Online Editor

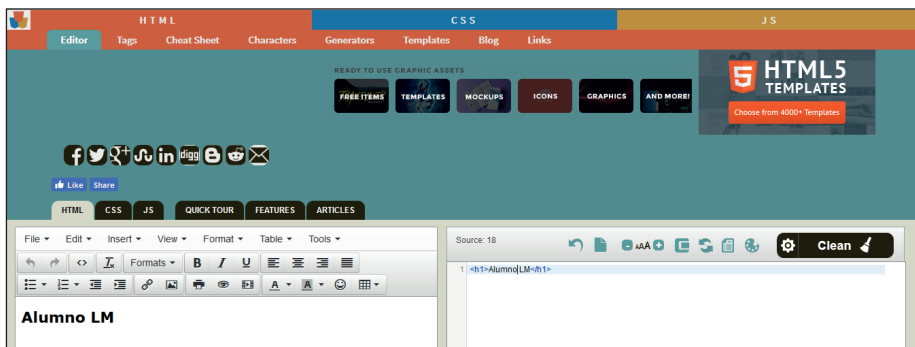


Figura 2.1
Interfax de HTML Online Editor

2.2.5. VS Code (Visual Studio Code)

VS Code es un editor de código fuente gratuito, ligero y multiplataforma desarrollado por Microsoft. Es ampliamente usado para desarrollo web, incluyendo HTML, CSS, JavaScript y otros lenguajes. Algunas de las ventajas de VS Code para HTML y CSS son las siguientes:

- Autocompletado inteligente.
- Resaltado de sintaxis (*syntax highlighting*).
- Emmet integrado.
- Previsualización en vivo.
- Gestión de proyectos y archivos.
- Extensiones para desarrollo web.
- Control de versiones integrado.

2.2.6. Sublime Text

Sublime Text es un editor de código fuente ligero, rápido y multiplataforma, muy popular entre los desarrolladores web por su simplicidad y su eficiencia. Aunque no es completamente gratuito, es ampliamente utilizado para HTML, CSS, JavaScript y otros lenguajes.

2.3. Identificación de etiquetas y atributos de HTML

En este apartado del capítulo, se describe la estructura del lenguaje HTML y sus etiquetas, aspecto básico en la programación de páginas web. HTML es un lenguaje de marcado que va a ser la base del módulo Lenguajes de Marcas y Sistemas de Gestión de Información.

2.3.1. Versiones de HTML. Semejanzas y diferencias

Las versiones de HTML están definidas por la comunidad internacional W3C (*World Wide Web Consortium*). HTML nace en el año 1991, desarrollado por el científico Tim Berners-Lee (conocido como el *padre de la web*), y era denominado *HTML Tags*.

- *HTML 2.0*. En 1995, el Internet Engineering Task Force (IETF) creó el primer estándar oficial, y, a partir del año siguiente, el W3C se encargó de publicar los estándares de HTML. Se centra en texto, listas, enlaces y formularios muy básicos.
- *HTML 3.2*. A principios de 1997, aparece esta versión, publicada por el W3C (incluyendo *applets* de Java).
- *HTML 4.0*. El 24 de abril de 1998 se publica esta versión, que presenta como novedades las siguientes: tablas complejas, mejoras en los formularios, hojas de estilo, añadir *script* y mejora de accesibilidad.
- *HTML 4.01*. A finales de 1999, surge HTML 4.01 como actualización y revisión de HTML 4.0, aunque no aporta novedades relevantes.
- *HTML 5*. Publicada a finales del 2014, establece nuevos elementos, algunos similares a los anteriores, pero con un significado semántico: `<header>`, `<nav>`, `<article>`, `<section>`, `<footer>`.

2.3.2. Estructura de un documento HTML

La estructura principal de una página web está compuesta por una serie de elementos dentro de un documento de texto plano que dan un formato al contenido. Dicho contenido es interpretado por el navegador, que muestra el resultado final.

A) Estructura

Un documento HTML está compuesto por etiquetas y secciones jerárquicas, que definen la estructura y el contenido de la página web.

Los navegadores también permiten ir inspeccionando el código conforme aparece el resultado. La estructura principal de una página web es la siguiente:

```
<!DOCTYPE html>
<html>
  <head>
    <title>LM</title>
  </head>
  <body>
    <!-- contenido -->
  </body>
</html>
```

La etiqueta primera de un documento es `<html>`, que define el inicio del fichero, por lo que `</html>` es la última que tiene que escribirse. Después, se encuentra la etiqueta `<head>`, donde se especifica la cabecera del documento y que puede anidar dentro otras como:

- `<title>`. Título que se muestra en la barra de título del explorador.
- `<meta>`. Sirve para el uso de metadatos y para posicionar la web en los buscadores.
- `<style>`. Se utiliza en la definición de estilos.
- `<link>`. Vincula las hojas de estilo como recurso externo.



RECUERDA

Pueden escribirse las etiquetas en mayúsculas o minúsculas, ya que el navegador las interpretará igual. Una etiqueta `<HTML>` será interpretada de igual manera por el navegador que si se escribe en minúsculas `<html>`.

Por último, se aprecia la cabecera `<body>` o cuerpo, que es donde se define el contenido de una página web visible en los navegadores. Contiene todo el contenido visible de la página web: textos, imágenes, enlaces, menús, vídeos, etc.

Dentro del `<body>` se suelen usar etiquetas semánticas (HTML5) para organizar la información:

- `<header>`: encabezado de la página o sección.
- `<nav>`: menús de navegación.
- `<section>`: secciones de contenido relacionadas.
- `<article>`: contenido independiente, como un artículo o una noticia.
- `<aside>`: información complementaria.
- `<footer>`: pie de página con información adicional o legal.

En el ejemplo anterior (estructura principal de una página web), se aprecia un comentario, entre las etiquetas `<!--` y `-->`. El uso de comentarios en este tipo de documentos es muy importante, ya que ayuda a organizar el código y a ocultar texto que ya no se usa.



SABÍAS QUE...

En muchos documentos HTML, puede observarse que la primera línea es la declaración del tipo de documento:

```
<!DOCTYPE html...>
```

B) Elementos HTML

La estructura básica de HTML se denomina *elemento* y está compuesto por dos propiedades: atributos y contenido. Por lo general, tiene una etiqueta de inicio (que puede contener un atributo con su valor) y otra de cierre.

Existen algunas etiquetas sin contenidos o vacías que no necesitan de etiqueta de cierre como, por ejemplo, `
`.

En la figura 2.2, se presenta un ejemplo de elemento HTML.

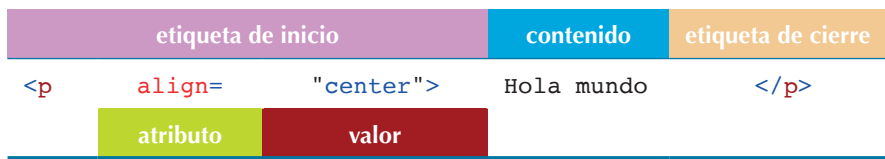


Figura 2.2
Elemento HTML.

Algunas de las etiquetas HTML que se utilizan en una página web aparecen reflejadas en la figura 2.3.



Figura 2.3
Etiquetas HTML que se usan en una página.

2.3.3. Declaración de tipo de documento

La declaración de tipo de documento (DTD) añade reglas sintácticas que permiten definir el documento, con sus reglas y atributos, para de esta manera comprobar si es válido. Entre la información que puede añadirse destaca la versión HTML.

Debe declararse al principio del documento.

Por ejemplo, para HTML 4.01, existen tres DTD diferentes, que incluyen los siguientes campos:

1. *Estricto*. Solo pueden usarse etiquetas HTML 4.01, y no permite marcado de presentación.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

2. *Transicional*. Permite todos los elementos HTML, incluyendo los antiguos, desaconsejados o en desuso.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

3. *Marcos o frames*. Es similar al anterior, pero usa elementos para *frames*. Se ha quedado obsoleto.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
```

Si se analiza cada uno de los campos de una declaración, se observa que consisten en:

- *!DOCTYPE*: se usa esta cabecera para la declaración de tipo. Es la manera que tiene el navegador de saber la versión especificada.
- *HTML*: informa de cuál es el elemento raíz del documento.
- *PUBLIC*: presenta dos posibles valores para este campo, *PUBLIC* o *SYSTEM*, donde el primero indica que la declaración de tipo de documento está disponible de manera pública.
- *"-//W3C//DTD HTML 4.01 Transitional//EN"*: es el identificador público formal, que informa de la organización y contiene información diversa separada por los caracteres de la barra doble (*//*):
 - *-*: presenta dos posibles valores (*-* es una organización no registrada por la ISO, mientras que *+* es una organización registrada por la ISO).
 - *W3C*: es la organización responsable.
 - *DTD HTML 4.01 Transitional*: indica el tipo de documento que está declarándose.
 - *EN*: indica el idioma de la DTD.
- *"http://www.w3.org/TR/html4/loose.dtd">*: indica la URL de la versión del DTD.

2.3.4. Etiquetas y atributos HTML

Los atributos son elementos compuestos por dos partes claramente diferenciadas (el nombre del atributo y su valor correspondiente) y se añaden en la etiqueta de inicio de un elemento HTML.

La manera de asignar un valor a un atributo es mediante el signo igual (=). El valor puede ponerse con comillas dobles, simples o sin comillas en HTML.

A) Atributo bgcolor

Atributo para signar color usado en algunas etiquetas. La manera de asignar el color es muy diversa: en hexadecimal, mediante el nombre del color y en RGB.

- *Hexadecimal*: el valor es “#**xx****yy****zz**”, siendo ese grupo de letras 3 agrupaciones numéricas con valores hexadecimales (0 1 2 3 4 5 6 7 8 9 A B C D E F). Los valores de baja intensidad se representan con 00, donde FF son los de mayor intensidad.

El color negro se representa con el código #000000, mientras que el blanco es #FFFFFF.

- **xx** es la intensidad de rojo
- **yy** es la intensidad de verde.
- **zz** es la intensidad de azul.

```
<html>
<head>
  <title>LM</title>
</head>
<body bgcolor="#FF0022">
  colores
</body>
</html>
```

Resultado en el navegador 



ACTIVIDAD RESUELTA 2.1

Asigna valores hexadecimales para los colores rojo, verde y azul.

Solución 

- *Por nombre: red, yellow, blue, brown, silver, gray, black, white, etc.*

```
<html>
<head>
  <title>LM</title>
</head>
<body bgcolor="blue">
  colores
</body>
</html>
```

Resultado en el navegador 

- *RGB: son colores aditivos o primarios y están basados en la combinación de píxeles de colores rojo (RED), verde (GREEN) y azul (BLUE).*
 - Los colores se codifican con un número.
 - Los valores para cada color van desde 0 a 255; por lo tanto, hay 256 valores posibles para cada valor.
 - Con la unión de los tres colores en su máximo valor, se obtiene el color blanco, y con la ausencia de todos, el negro (todos a cero). El blanco será (255,255,255) y el negro será (0,0,0).
 - Permite representar $256 \times 256 \times 256$ combinaciones diferentes de colores, con lo que hay 16 millones de colores.
 - Debe tenerse en cuenta que puede que no todos los navegadores admitan esta codificación de colores.

Si se mezclan los tres colores, se obtienen las siguientes combinaciones de colores secundarios:

- Verde + azul = cian.
- Rojo + azul = magenta.
- Rojo + verde = amarillo.
- Rojo + azul + verde = blanco.

```
<html>
<head>
  <title>LM</title>
</head>
<body bgcolor="rgb(0,0,255)">
  colores
</body>
</html>
```

Resultado en el navegador 



ACTIVIDAD PROPUESTA 2.1

Señala la respuesta correcta a las siguientes preguntas:

- a) ¿Con qué valores para RGB se obtiene el color blanco?
- V F a) (255,255,255).
 V F b) (0,0,0).
- b) ¿Con qué valores para RGB se obtiene el color negro?
- V F a) (255,255,255).
 V F b) (0,0,0).
- c) ¿Cuál es el valor máximo de cada color?
- V F a) 255.
 V F b) 256.
- d) ¿Qué color es (255,0,0)?
- V F a) Rojo.
 V F b) Verde.
- e) ¿Qué color es (0,255,0)?
- V F a) Rojo.
 V F b) Verde.
- f) ¿Qué color es (0,0,255)?
- V F a) Rojo.
 V F b) Azul.
- g) ¿Qué color es (128,0,0)?
- V F a) Rojo claro.
 V F b) Rojo oscuro.
- h) ¿Qué color es (0,128,0)?
- V F a) Verde claro.
 V F b) Verde oscuro.
- i) ¿Qué color es (0,0,128)?
- V F a) Azul claro.
 V F b) Azul oscuro.

Verificar

