

LENGUAJE DE MARCAS

Y

SISTEMAS DE GESTION

DE LA INFORMACIÓN

RA2: Utilización de lenguajes de marcas en entornos web

Tema 2: Utilización de lenguajes de marcas en entornos web

2.1 Evolución histórica

Orígenes de los lenguajes de marcas

Los lenguajes de marcas surgieron a finales de los años 60 para introducir anotaciones en documentos electrónicos. En esta época, se estandarizó el **SGML** (*Standard Generalized Markup Language*), permitiendo compartir información entre sistemas informáticos. Sin embargo, su complejidad impidió su adopción masiva.

Nacimiento de HTML y la World Wide Web

En 1980, **Tim Berners-Lee**, un físico del CERN, propuso un sistema de **hipertexto** para compartir información en redes de computadoras. Aunque el concepto de hipertexto ya existía, en informática se refiere a la posibilidad de acceder a información relacionada a través de enlaces.

Años después, Berners-Lee colaboró con el ingeniero **Robert Cailliau**, con quien desarrolló la **World Wide Web (W3)**.

Principales hitos en la evolución de HTML

- **1991:** Se presenta el primer documento con la descripción de **HTML**.
- **1993:** Primera propuesta oficial para convertir HTML en un estándar. Se introducen etiquetas para imágenes, tablas y formularios, aunque no se logra la estandarización.
- **1995:** El organismo **IETF** crea un grupo de trabajo para HTML, y el **22 de septiembre de 1995** se publica **HTML 2.0**, el primer estándar oficial.
- **1997:** La estandarización de HTML pasa a manos del **W3C (World Wide Web Consortium)**, que el **14 de enero de 1997** publica **HTML 3.2**.
- **1998:** Se lanza **HTML 4.0**, soporte para **scripts**, mejoras en accesibilidad y tablas más complejas.
- **1999:** Se publica **HTML 4.01**, que corrige errores.
- **Años 2000:** Se detiene el desarrollo de HTML en favor de **XHTML**, una versión basada en **XML**. **XHTML 1.0** se publica en **2000**, seguido de versiones experimentales **XHTML 1.1** y **2.0**.
- **2004:** Empresas como **Apple, Mozilla y Opera** critican la falta de avances en HTML y forman **WHATWG (Web Hypertext Application Technology Working Group)**.
- **2007:** Se publican borradores de **HTML 5.0**.
- **2008:** El **22 de enero** se publica el **primer borrador oficial de HTML5**.

2.2 Identificación de etiquetas y atributos de HTML

HTML (*Hypertext Markup Language*) es un lenguaje de marcas utilizado para desarrollar páginas web. A través de sus etiquetas, HTML define la estructura y el contenido de una página, permitiendo que los navegadores la interpreten y muestren correctamente.

Para desarrollar una página web se requiere:

- **Un editor de textos** para escribir el código HTML. Ejemplos: Bloc de notas, Visual Studio Code.
- **Un navegador web** para visualizar el contenido.

Publicación de una página web

Una vez creada la página, se necesita un **servidor web** para almacenarla y hacerla accesible en Internet.

- Se puede instalar un servidor propio.
- También existen proveedores de Internet que ofrecen alojamiento gratuito.

Estructura de un documento HTML

HTML organiza sus documentos en una **estructura jerárquica en forma de árbol**, donde los elementos son sus componentes principales.

Cada **elemento HTML** consta de:

- **Etiquetas** que indican el inicio y cierre del elemento.
- **Contenido**, que es la información visible dentro del elemento.
- **Atributos** opcionales, que añaden propiedades adicionales.

2.2.1 Etiquetas y atributos

Las etiquetas delimitan el elemento, estando el atributo dentro de la misma etiqueta y el contenido entre la etiqueta de apertura y la de cierre.

Etiquetas

Las etiquetas, también conocidas como marcas, definen una serie de elementos que forman el léxico del lenguaje HTML. Cada etiqueta se representa entre los signos de *menor que* (<) y *mayor que* (>).

Etiquetas cerradas:

- Se componen de una **etiqueta de apertura** y una **etiqueta de cierre**.
- La etiqueta de cierre lleva el símbolo / antes del nombre.
- <p> → Etiqueta de apertura.
- </p> → Etiqueta de cierre.

Etiquetas abiertas (o autocontenidas):

- No requieren una etiqueta de cierre.
- Se usan para insertar elementos sin contenido de texto dentro.
- <hr>
-

-

Atributos

Los atributos modifican el significado de un elemento HTML o incluso pueden proveerle de funcionalidad. Las etiquetas pueden contener atributos si necesitan realizar alguna configuración sobre una característica determinada.

Existen atributos obligatorios y atributos opcionales, lo cual dependerá de la etiqueta HTML en la que se sitúen. Los atributos, al igual que las etiquetas, se pueden definir tanto en mayúsculas como en minúsculas, aunque los valores que se asignen a los atributos son sensibles a mayúsculas o minúsculas. Por ello, es recomendable utilizar siempre minúsculas para etiquetas y atributos. De esta forma, evitaremos confusiones.

Tipos de atributos

- **Obligatorios:** Son necesarios para que la etiqueta funcione correctamente.
- **Opcionales:** Se pueden incluir para personalizar el comportamiento o apariencia del elemento.

Ejemplo de atributo en una etiqueta	Múltiples atributos en una etiqueta
<code><p align="left">Este texto está alineado a la izquierda.</p></code>	<code></code>
Etiqueta: <code><p></code> (párrafo). Atributo: <code>align="left"</code> (alineación del texto a la izquierda).	src: Define la ubicación de la imagen. alt: Texto alternativo si la imagen no se carga. width y height: Especifican el tamaño de la imagen.

2.3 Estructura básica

- **<html>**: el elemento raíz del documento. Esta etiqueta engloba toda la página HTML.
- **<head>**: la cabecera del documento. Contiene información que no es directamente el contenido de la página. Por ejemplo, título de la página web, metadatos, estilos, código JavaScript, etc.

```
1 <html>
2   <head>
3     <title>
4       título del documento
5     </title>
6   </head>
7   <body>
8     (aquí se colocaría todo el contenido)
9   </body>
10 </html>
```

- **<title>**: el nombre que le damos al título que se asignará al documento.
- **<body>**: el cuerpo del documento. Aquí se codifica todo aquello que deseamos que se visualice en el navegador. Por ejemplo, los párrafos, las imágenes, formularios, etc.

En HTML5 mediante las etiquetas nuevas, se consigue añadir una serie de características y elementos cuya función es facilitar la tarea a los autores de la aplicación web.

HTML5 se basa principalmente en una estructuración avanzada que se encarga de definir los contenidos, agrupándolos en distintas etiquetas con un nombre asignado que se corresponde con la tarea a realizar.

- **<nav>**: enlaces de navegación.
- **<article>**: algún artículo que se haya publicado.
- **<section>**: parte correspondiente a algún artículo.
- **<aside>**: barras laterales.
- **<footer>**: pie de página.
- **<dialog>**: distintos diálogos o comentarios.

Además de estas etiquetas, HTML5 también cuenta con elementos como `<div>` y `<span>`, que se utilizan para agrupar los diferentes elementos hijos haciendo uso de atributos como: *class*, *id* o *title*. De esta manera, se pretende utilizar una misma semántica con un estilo común

2.3.1 Comentarios

Los comentarios son unas líneas que definen lo que vamos a ir realizando en cada momento, aunque no son interpretados por el navegador.

La sintaxis de los comentarios es la siguiente:

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

2. 4 Elementos de un HTML

Cabecera

La cabecera contiene información general acerca del documento, pero normalmente la información que figura dentro de la cabecera no es la destinada a visualizarse en el navegador.

Salvo el título, que sí es mostrado por el navegador, normalmente en la pestaña.

La etiqueta `<head>` es el primer descendiente de la etiqueta raíz `<html>`. La cabecera tiene una etiqueta de apertura y otra de cierre.

Cabecera <code><head></code>	
Elementos hijos	Significado
<code><title></code>	Título del documento HTML que va a aparecer en el navegador web, en la barra superior
<code><style></code>	Usado para determinar el estilo de la información del documento, es decir, información CSS que nos permite insertar una hoja de estilo en la cabecera. Mediante el atributo <i>type</i> indicamos el lenguaje del estilo
<code><link></code>	Relaciona el documento HTML con otro documento externo, como por ejemplo un archivo CSS
<code><meta></code>	Utilizado para especificar información sobre el propio documento, como el conjunto de caracteres, la descripción de la página, las palabras clave, el autor del documento... Suele llevar dos atributos: <i>name</i> y <i>content</i> , que hacen referencia al nombre de la página y a sus principales contenidos. Refresco automático.
<code><script></code>	Se usa para especificar la URL base a partir de la que partirán el resto de URL relativas del documento. Puede indicar la base de elementos como sonidos, gráficos, etc., siempre que hagan referencia a una determinada página web
<code><base></code>	Se usa para especificar la URL base a partir de la que partirán el resto de URL relativas del documento. Puede indicar la base de elementos como sonidos, gráficos, etc.,

Cuerpo del documento

Situado detrás de la cabecera `<head>` contiene todo el cuerpo correspondiente a una determinada página web, junto con los elementos propios de la página.

Atributos que se recomienda utilizar en la marca `<body>`. Son los siguientes:

- **id:** para identificar el elemento para la hoja de estilo.
- **class:** para asignar un nombre de la clase de la hoja de estilo y, de esta forma, poder mejorar su rendimiento.
- **title:** para agregar un comentario y que lo puedan visualizar los navegadores.
- **style:** aplicar un estilo a este elemento.

Podemos clasificar los elementos del cuerpo (<body>) en dos tipos:

Los **elementos de bloque** poseen las siguientes características:

- Comienzan en una nueva línea
- Ocupan todo el ancho de una línea disponible
- Por tanto, impiden que otros elementos se sitúen a su lado (horizontalmente)
- Pueden albergar en su interior otros elementos de bloque u otros elementos de línea
- Algunos elementos de bloque típicos son los siguientes:
 - <div>, <form>, <h1>, <p>, <section>, <table>...

Los **elementos de línea** tienen las siguientes características:

- Solo pueden contener en su interior datos y otros elementos en línea
- Ocupan el ancho mínimo (en horizontal) que necesitan
- Los elementos de línea más empleados son: <a>, <code>,
 - , <input>, ...

Encabezados

Para poner títulos en el contenido de un HTML podemos usar los encabezados (también llamados cabeceras).

Los encabezados que podemos usar son los siguientes:

- <h1></h1>
- <h2></h2>
- ...

Párrafos

Son una de las marcas más utilizadas. No pueden contener elementos de bloque en su interior. Todos los párrafos del documento web serán bloques delimitados por la marca *p*.

- <p> </p>: escribe un texto en forma de párrafo.
- <blockquote> </blockquote>: permite visualizar una cita con el margen izquierdo mayor, produciéndose un efecto de una sangría.
- <pre> </pre>: devuelve una copia exacta del texto respetando los espacios en blanco, tabulaciones y retorno de carro, es decir, tal y como se ha escrito.

Salto de línea

-
: representa un salto de línea entre párrafos, ya que el salto de línea del teclado es ignorado por el navegador. No necesita marca de cierre.
- <hr>: existe una forma de separar cada párrafo, utilizando una línea horizontal visible. No necesita etiqueta de cierre y podemos utilizar los siguientes atributos: *align*, *noshade*, *size*, *width*.

Estilo de texto

Son una serie de elementos HTML que proporcionan elementos de estilo a una parte del texto.

	Negrita
<i>	Cursiva
<big>	Agrandar el tamaño. Se puede introducir varios comandos para hacerlos más grande.

<code><small></code>	Pequeño. Funciona de la misma forma que <i>big</i> .
<code><s></code> o <code><strike></code>	Tachado de texto.
<code><u></code>	Subrayado.

Colores

Podemos representar los colores mediante el símbolo # seguido de tres pares de dígitos hexadecimales. El rango de colores indica la intensidad de los colores primarios (rojo, verde y azul), que en dígitos hexadecimales es #RRVVA A.

Otra forma de expresar colores en HTML es usando una notación hexadecimal más corta. Si lo preferimos, también podemos indicar el color de forma directa: *red, green, blue...*

Hipervínculos

Los hipervínculos son elementos del lenguaje HTML que permiten acceder a otro recurso. Es decir, los podemos definir como enlaces que apuntan a otro sitio web, un fichero, una imagen, etc.

`<a></a>`

Disponemos de otro atributo `target` (opcional) que hará referencia al destino en el que se mostrará la información disponible en esa dirección a la que nos lleva. Enlace en una pestaña o página nueva: `_blank` como valor del atributo `target`, misma pestaña o ventana: `_self`.

Anclas y vínculos internos

Los vínculos internos permiten acceder a un sitio concreto dentro de una página web. Aunque antes debemos establecer un ancla, que es el punto fijo de posición al que accederemos tras un vínculo interno.

Rutas Absolutas y relativas

Rutas absolutas son aquellas que enlazan con páginas, cuya dirección absoluta se indica en el atributo `href` del comando `a`. Suelen ser páginas web externas a nuestro proyecto.

`<a href = "http://www.google.es">`

Las **relativas** se corresponden con aquellos enlaces cuya dirección relativa se indica en el atributo `href` del comando `a`. Suelen ser enlaces a páginas internas del mismo proyecto.

`<a href = "/pagina2/pagina2.html">`

Imágenes

Las imágenes pueden estar en diferentes formatos, como pueden ser los mapas de bits (archivos PNG, GIF, JPEG), documentos vectoriales de una página (archivos PDF, XML), mapas de bits animados, gráficos de bits animados, etc.

Gracias al elemento `<img>` podemos representar una determinada imagen, ayudados por su atributo (obligatorio) `src`. En este caso, indicamos:

- La dirección válida en la que está la imagen que queremos visualizar.
- Una ruta relativa, si es que la imagen está en alguna parte local.
- Una URL, si se refiere a una imagen externa que se encuentra almacenada en una página web diferente

El atributo **alt** nos permite indicar un texto alternativo que sea capaz de representar el contenido de una imagen. Podemos utilizar esta forma en caso de que el navegador no pueda visualizar o descargar las distintas imágenes.

Las etiquetas **<figure>** y **<figcaption>** son novedosas para HTML5. Nos ofrecen la posibilidad de agrupar una imagen junto con su información o leyenda.

Incorporación de imágenes

Si queremos que una imagen se muestre en una web, en primer lugar, necesitamos declarar una etiqueta ****, que no necesita etiqueta de cierre. Por ejemplo:

```

```

Etiqueta	Atributo	Valor	Significado
img	src	URL	Indica la URL de la imagen.
	alt	Texto	Define un texto alternativo por si no se encontrara la imagen deseada
	Aling	Top, middle, bottom, left, right, center	Alinea la imagen respecto al texto, tanto en sentido horizontal como en sentido vertical.
	Border	Número	Pone un borde o marco a la imagen. Se expresa en píxeles.
	Height	Número %	Especifica la altura que debe tener la imagen. Se expresa en píxeles o porcentaje.
	Width	Número %	Especifica el ancho que debe tener la imagen. Se expresa en píxeles o porcentaje.
	hspace	Número	Especifica en píxeles la separación horizontal entre el texto y la imagen.
	vspace	Número	Especifica en píxeles la separación vertical entre el texto y la imagen.

Uso de mapas sensibles

Los mapas de imagen nos permiten definir distintas zonas clicables dentro de una imagen.

Las diferentes zonas que se definen en una imagen se van creando mediante rectángulos, círculos y polígonos. A la hora de crear un mapa de imagen:

- En primer lugar, insertamos la imagen original mediante la etiqueta ****.
- A continuación, utilizamos la etiqueta **<map>** para definir las distintas zonas o regiones de la imagen. Definiremos cada una de estas zonas con la etiqueta **<area>**.

<map>	Mapa de imagen
Atributos comunes	Básicos, i18n y eventos
Atributos específicos	<i>Name</i> = "texto". Nombre que identifica de forma única al mapa definido (es obligatorio indicar un nombre único)
Tipo de elemento	Bloque y en línea
Descripción	Se emplea para definir mapas de imagen



Listas

Las listas en HTML permiten agrupar un conjunto de elementos o ítems relacionados en un documento.

Listas ordenadas: sirven para representar los elementos de una lista de manera ordenada, usando una numeración incremental...

Como ya adelantamos, la etiqueta **** tiene disponibles los siguientes atributos:

Apartado 1

Apartado 2

- **start** = número: permite seleccionar el número que deseemos que sea el primero de la lista
- **type** = tipo: permite seleccionar el tipo de numeración que deseemos

...

En HTML5, podemos diferenciar entre los siguientes tipos:

- 1: expresa números desde 1, 2, 3, etc.
- a: expresa letras minúsculas a partir de a, b, c, etc.
- A: expresa letras mayúsculas a partir de A, B, C, etc.
- i: expresa números romanos desde i, ii, iii, etc.
- I: expresa números romanos en mayúscula desde I, II, III, etc.

Listas no ordenadas: ofrecen la posibilidad de poder representar los diferentes elementos de una lista sin enumerar, independientemente del lugar en el que se vayan a almacenar.

Apartado 1

La directiva **** tiene disponibles los siguientes parámetros:

Apartado 2

- **Type** = tipo: nos permite seleccionar el tipo viñetas que deseemos.

...

En HTML5, podemos diferenciar entre los siguientes tipos:

- **Disc**: expresa una viñeta en forma de disco.
- **circle**: expresa una viñeta en forma de círculo.
- **square**: expresa una viñeta en forma de cuadrado

Listas de definición o glosario: ofrecen la posibilidad de representar los diferentes elementos de un diccionario formado por su término y su definición. En las listas de definición, se utilizan las directivas:

<dl>

<dt>Coche</dt>

<dd>Vehículo de cuatro ruedas</dd>

<dt>Moto</dt>

<dd>Vehículo de dos ruedas</dd>

...

</dl>

Agrupación de contenidos

Esta marca sirve para agrupar varios elementos dentro de un bloque y permite asignarle un identificador.

<div></div>

Tablas

Son elementos de HTML que nos ofrecen la posibilidad de representar datos de una o varias dimensiones (matriz) con la información distribuida en filas y columnas.

La etiqueta básica de HTML para incluir una tabla en un documento es la siguiente:

<table>... </table>

Filas, columnas y celdas

- La etiqueta **<tr>** hace referencia a *fila* (*r* de *row*, en inglés).
- La etiqueta **<th>** hace referencia a las cabeceras de las columnas (*h* de *head*, en inglés).
- La etiqueta **<td>** hace referencia a los datos que pondremos en cada celda (*d* de *data*, en inglés).

Combinación de celdas

Es posible combinar un conjunto de celdas de una determinada tabla, que van a afectar a su definición:

- **<colgroup>**: representa las columnas de una tabla.
- **</col>**: indica el número de columnas que tiene una tabla.

Elementos que nos permiten referirnos a las diferentes partes de una tabla en el lenguaje HTML:

- **<thead>**: cabecera.
- **<tbody>**: bloques de filas.
- **<tfoot>**: pie.

Atributos de tabla

Hasta el momento, y para versiones anteriores a HTML5, se han podido utilizar distintos atributos, como *align* y *bgcolor*. Sin embargo, desde la aparición de HTML5, se soportan todos estos atributos:

- **summary**: realiza el resumen y la estructura de la tabla para facilitar la tarea de los buscadores.
- **width**: indica el ancho total de la tabla en distintas unidades. Píxel (px) o porcentaje (%).
- **frame**: indica el nombre del marco en el que se puede visualizar la tabla.
- **rules**: especifica las líneas de división que serán visibles.
- **border**: determina el ancho del borde anterior.
- **cellspacing**: especifica el espacio existente entre las distintas celdas.
- **cellpadding**: especifica el espacio existente entre el borde de las celdas y el contenido.

<caption>

...

</caption>

Lo usamos para poner un título a la tabla, es decir, la información que pongamos entre las etiquetas aparecerá justo arriba de la tabla, no en el interior, sino encima de la tabla.

Formularios

Utilizaremos los formularios cuando necesitemos recoger información específica de un objeto en cuestión.

Los formularios están compuestos, entre otros, por diferentes elementos como las cajas de texto (*textbox*), las casillas de verificación (*checkbox*), las casillas de opción (*radio*, *button*), las listas desplegables y los subformularios.

Propiedades de los formularios

Para la sintaxis básica de un formulario utilizamos la siguiente etiqueta:

```
<form>
```

...

```
</form>
```

La etiqueta *form* consta de varios atributos bastante importantes. Podemos encontrar los siguientes:

- **action:** indica el lugar al que se envían los datos.
- **method:** indica el método de transferencia de los datos en el servidor web. Los valores que puede tomar este atributo pueden ser:
 - **post:** para el envío de los datos al usuario de forma codificada.
 - **get:** para el envío de los datos a una dirección web.
- **target:** se utiliza para indicar o especificar dónde vamos a mostrar la respuesta del formulario. Los diferentes valores que puede tomar este atributo son:
 - **"_blank":** se muestra en una ventana nueva.
 - **"_self":** se muestra en la misma ventana del **formulario**.
 - **"_parent":** se muestra en la ventana padre, es decir, la que precede al formulario.
- **name:** identifica el formulario mediante el nombre. De esta forma, podemos llamarlo desde otro archivo. Recomendamos que el nombre de formulario sea único para facilitar su tratamiento.

Elementos de los formularios

Dentro de un formulario podemos tener distintos elementos que van a componer todo el diseño y contenido del formulario.

- **fieldset:** Con este elemento podemos agrupar un conjunto de elementos bajo un mismo nombre. `<fieldset> ...</fieldset>`
- **input:** Nos permite crear varios tipos de elementos dentro de un formulario, dependiendo del valor que asignemos al atributo *type*. `<input>...</input>`
- **textarea.** Podemos definir un campo de texto de grandes dimensiones para que el usuario tenga la posibilidad de escribir todo lo deseado. `<textarea>... </textarea>`

Los atributos de este elemento que podemos incluir en la etiqueta de apertura son los siguientes:

- **name:** identifica el nombre del área de texto.
- **cols:** número de caracteres que puede contener cada línea.
- **rows:** número de líneas del área de texto.
- **readonly:** para impedir que el usuario pueda editar este campo.

button: Es la forma que tenemos de insertar un botón en el formulario. `<button type="..." > ... </button>`

select. Permite definir una lista desplegable con varios ítems, es decir, con varias opciones a elegir.

```
<select name = "..." id = "..." >
```

```
<option value = "..." > ...</option>
```

```
<option value = "..." > ... </option>
```

...

```
</select>
```

Elemento label. Con este elemento relacionamos un texto o título a un determinado campo del formulario. `<label for="...">... </label>`

Marcos

Definición de marcos

Los marcos ofrecen la posibilidad de mostrar varios archivos HTML en la misma ventana del navegador.

También tenemos la opción de que estos *frames* interactúen. Cuando presionamos sobre un enlace en un *frame* para acceder a él, podemos cargar una página en otro frame diferente.

Uso de marcos

En este ejemplo podemos ver la estructura de un documento HTML que realiza la función de esquema de una página web, dividida en dos columnas. La primera, ocuparía un 20% del total de la página. La segunda, el resto.

En el primer marco se puede visualizar el documento *pagina2. html* y en el segundo marco lo enlazamos a la *página3. html*. Por tanto, para hacer uso de los marcos, vamos a tener que cargar tres archivos HTML diferentes: el principal y los dos enlaces correspondientes a los marcos.

```
<html>
<head>
<title>Prueba de frames</title>
</head>
<frameset cols="20%,80%">
<framesrc="pagina2.html">
<framesrc="pagina3.html">
</frameset>
<p>El navegador no soporta frames</p>
</html>
```

Objetos multimedia

<iframe></iframe>: Inserta un marco en un documento. Suele utilizarse para insertar publicidad o páginas de colaboración.

<object></object> Inserta un objeto en un documento. El tipo del objeto viene determinado por el atributo, pudiendo ser: imágenes, aplicaciones de Java, animaciones u otros documentos HTML.

Además de este atributo, también podemos detallar, en los atributos **height** y **width**, tanto el ancho como la altura del objeto.

Y, para finalizar, en el atributo **data**, podemos determinar la ruta del objeto a visualizar, bien sea de audio o de vídeo.

<param>: Este comando no tiene etiqueta de cierre y permite inicializar las variables objeto.

2.5 XHTML y sus diferencias con HTML

XHTML es una derivación del lenguaje de marcas XML. Podríamos decir que XHTML es una versión de HTML bajo la sintaxis de XML.

Los documentos XHTML son documentos bien formados que cumplen todas las normas sintácticas y estructurales de los documentos XML, algo que no ocurre con HTML.

Las diferencias más notables entre XHTML y HTML son:

La etiqueta **<!DOCTYPE...>** y el espacio de nombres son obligatorios en XHTML, en HTML no.

- También son obligatorios los elementos `<html>` `<head>` `<title>` y `<body>`.
- En XHTML los elementos deben estar estrictamente bien anidados.

- Toda etiqueta que se abra en XHTML debe ser cerrada, siempre.
- Los elementos y atributos en XHTML deben estar siempre en minúsculas.
- Los valores de los atributos en XHTML deben estar siempre entre comillas.

2.6 Versiones de HTML y XHTML

Debido al rápido crecimiento de la web y a cómo van evolucionando las versiones HTML, aparece la necesidad de estandarizarlo para que autores y navegadores reconozcan el tipo de versión HTML que pueden utilizar.

HTML llegó a convertirse en estándar en 1995 y, con el paso de los años, ha experimentado un desarrollo constante. Hace algunos años, la versión de HTML recomendada por el W3C era HTML 4.01 y ya llegó a desarrollarse la versión HTML5.

- **HTML 4.01 Strict (*Strict DTD*)**: es la más restrictiva de todas, ya que no permite la utilización de etiquetas que estén anticuadas.
- **HTML 4.01 Transitional (*Transitional DTD*)**.
- **HTML 4.01 Frameset (*Frameset DTD*)**.

HTML 1	Inicios a partir de 1991
HTML 2	Noviembre 1995
HTML 3	Enero 1997
HTML 4	Diciembre 1997
HTML 5	Octubre 2014

XHTML

HTML fue adoptado rápidamente por la comunidad, pero el crecimiento descontrolado de nuevas etiquetas causó problemas de interpretación en diferentes navegadores. Esta falta de estandarización hizo que HTML fuera más complejo con el tiempo.

Para resolver estos problemas, apareció **XML** (Extensible Markup Language), que ofreció una alternativa al caos generado por HTML, aunque **XML** se orienta más al intercambio de información que a la presentación visual de los datos.

Para combinar lo mejor de ambos, se creó **XHTML** (eXtensible HyperText Markup Language), una versión de HTML con las características de presentación de HTML pero con las reglas de **XML**.

Características de XHTML:

1. **Origen**: XHTML es una versión de **HTML 4.0** con las características de **XML**.

Versiones:

- En 2000 aparece **XHTML 1.0**.
- En 2001 se lanza **XHTML 1.1**, con otros borradores como **XHTML 1.2** y **XHTML 2.0**, aunque nunca llegaron a estandarizarse.
- Actualmente, se está trabajando en **XHTML 5.0**.

Diferencias entre HTML y XHTML

La principal diferencia es que **HTML** proviene de **SGML** (Standard Generalized Markup Language), mientras que **XHTML** proviene de **XML**. Los documentos XHTML deben cumplir las reglas estrictas de XML, como:

- **Un único elemento raíz**: <html>.
- **Instrucción de procesamiento XML**:
- **Declaración DOCTYPE** para la validación del documento:
- **Etiquetas en minúscula**: XML diferencia entre mayúsculas y minúsculas, por lo que todas las etiquetas deben ir en minúsculas.
- **Comillas dobles**: Los valores de los atributos siempre deben ir entre comillas dobles.

2.7 Herramientas de Diseño de Web

2.7.1 Editores Simples

El sistema operativo Windows suministra la herramienta Bloc de Notas, así como la de NotePad.

Haremos uso de esta herramienta básica para escribir un código en HTML, con la salvedad de almacenarlo con la extensión que lo identifica: **.html**.

2.7.1 Editores Avanzados

Los editores de texto permiten, mediante distintos colores podemos diferenciar cada elemento de una parte del código en lenguaje HTML.

Entre los ejemplos más conocidos, destacamos NotePad++ y Sublime Text, editores muy utilizados por la comunidad desarrolladora de código, herramientas de código libre. Por otra parte, y aunque sea de licencia de pago, podemos mencionar Adobe Dreamweaver.

También existen otras herramientas de edición, que podemos utilizar para la realización de diferentes páginas web de forma más profesional, como pueden ser, entre otras, Aptana, Amaya, Dreamweaver y Kompozer. Estos editores utilizan diferentes menús e iconos en los que podemos añadir:

- Algunas etiquetas (directivas) de HTML sin teclear.
- Reglas estilo CSS.
- Diferentes funciones destinadas a la creación y mantenimiento de la página web

2.7.3 Gestores de lenguajes HTML

Actualmente, en el mercado existen distintos navegadores, entre los que destacamos: Internet Explorer, Mozilla Firefox y Google Chrome.

2.7.4 Otras Opciones

Una alternativa son los editores tipo WYSIWYG (*what you see is what you get*), que puede traducirse como *lo que ves, es lo que obtienes*. De esta forma, podemos escribir diferentes documentos HTML y ver, de forma simultánea, cómo quedaría el resultado final de la página web, de la misma forma que se vería cuando la publicásemos en Internet.

2.8. Hojas de Estilo

El consorcio W3C desarrolló lo que se conoce como CSS (Cascade Style Sheets). CSS son las hojas de estilo en cascada y permiten definir estilos de cada uno de los elementos de un documento HTML o XHTML, permitiendo realizar agrupaciones de elementos por tipo, y de esta forma modificar el estilo de todos los elementos de un tipo determinado con un único cambio.

Este lenguaje presenta numerosas ventajas al utilizar separación entre contenidos y presentación. Podemos destacar las siguientes:

- Necesitamos menos código a la hora de escribir.
- Es más fácil generar código y mantenerlo.
- Los documentos son más legibles.

Sintaxis

Forma 1: CSS en línea: consiste en colocar el código CSS dentro de las etiquetas HTML usando el atributo *style*. De ese modo, los cambios de estilo CSS solo se aplican a dicho elemento y sus elementos hijos, si los hubiere.

Forma 2: CSS interno: consiste en colocar el código CSS en la cabecera del documento HTML, es decir, dentro de la etiqueta `<head>` del documento.

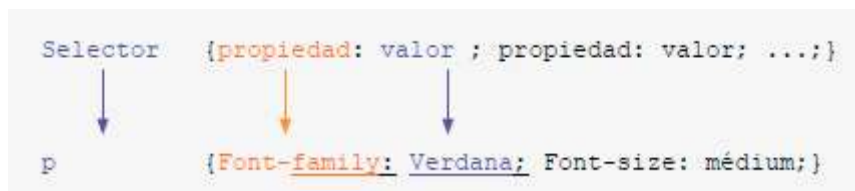
Forma 3: CSS externo: con esta opción separamos el código HTML del código CSS de manera tajante, de modo que en el archivo *.html* solo habrá código HTML y todo lo relativo al estilo de la página que estamos creando estará contenido en CSS.

Si se produce colisión, la prioridad debe ser:

1. **css en línea +prioridad**
2. **css interno**
3. **css externo -prioridad**

En las tres formas que hemos analizado, en caso de que se produzca repetición, debe ser la última regla la que se aplique. Si queremos dar más prioridad, podemos hacerlo haciendo uso de la palabra reservada: ***!important.***

Construir reglas CSS: los selectores son un elemento básico en CSS, porque nos ayudan a seleccionar un elemento o un conjunto de elementos HTML a los cuales se les dará un formato de estilo determinado.



Selectores

Vamos a ver de forma detallada una clasificación de los diferentes tipos de selectores:

- **Selector Universal:** Este selector permite seleccionar todos los elementos de una página. Este selector se indica a través de un asterisco (*). Su uso no es muy habitual
- **Selector de tipo o etiqueta:** El selector de tipo o etiqueta permite seleccionar todos los elementos de una página cuya etiqueta sea igual que la del selector
- **Selector descendiente:** El selector descendiente selecciona los elementos que se encuentran en el interior de otros elementos. La única condición es que debe estar dentro del otro elemento, independientemente del nivel de profundidad en el que se encuentre.
- **Selector de clase:** Este tipo de selector ofrece la posibilidad de seleccionar una determinada instancia de un elemento en concreto.
 - **Selector de ID:** nos ofrece la posibilidad de seleccionar una sola línea de una página completa.

Tema 4: Aplicación de los lenguajes de marcas a la sindicalización de contenidos

RSS (*Really Simple Syndication* o *Sindicación Realmente Simple*)

Tecnología que tiene como función principal, compartir o distribuir la información que se encuentre en la web.

RSS hace referencia a una técnica que podemos utilizar para conseguir extender aquellos contenidos que se encuentran en un sitio web. Es un lenguaje derivado de XML, mediante el cual podemos construir los diferentes archivos que almacenan el contenido a difundir.

Es Atom, también deriva de XML, aunque añade una serie de ventajas, entre las que podemos destacar la resolución de algunas ambigüedades que ofrecía RSS y la extensión de su alcance.

Son distintos, ya que el primero (RSS) se refiere a la redifusión web, mientras que el segundo (ATOM) hace referencia a un archivo determinado que posee la información que se desea difundir.

Entre sus principales ventajas destacamos:

- Aquellos usuarios que utilicen este programa no necesitan hacer una comprobación para saber si la información está actualizada en los lugares en los que se encuentran los contenidos de interés.
- Sigue un formato para los datos de tipo texto plano. Es bastante rápido a la hora de transmitir los datos, por lo que es muy recomendable para dispositivos móviles.
- Ofrece la posibilidad de filtrar información de diferentes sitios sin perder tiempo utilizando la agrupación.

4.1 Ámbitos de aplicación

La sindicación o redifusión web tienes varios ámbitos de aplicación. Desde diarios digitales hasta redes sociales, pasando por aplicaciones de trabajo o programas de descargas.

Con la RSS podemos compartir el contenido de una página web en tiempo real con otros usuarios. De esta forma, podemos ofrecer contenidos propios para que sean mostrados en otras páginas web de forma integrada.

4.2 Estructura de los canales de contenidos

Los documentos RSS son documentos XML, por tanto, han de cumplir las normas sintácticas establecidas por la W3C para todos los archivos XML. Además, deben tener una declaración donde se informe sobre la versión que se está utilizando.

Podemos desgranar los archivos RSS en 3 partes:

Etiqueta	Función
<?xml...?>	Información general sobre la versión XML a utilizar y el tipo de codificación de caracteres.
<rss...>	Información sobre la versión RSS que se va a utilizar.
<channel.>	Contenido del documento.

4.3 Elementos principales de un RSS

4.3.1 <CHANNEL>

Su función principal es describir la fuente RSS mediante tres elementos obligatorios.

- **<title>**: hace referencia al nombre del canal.
- **<link>**: define el hipervínculo al canal.
- **<description>**: se encarga de describir el contenido correspondiente al canal.

Este elemento **<channel>** tiene posibilidad de contener un elemento o varios **<item>**

Elementos opcionales en **<channel>**

- **<category>**: Permite definir una o más categorías en las que se clasifica el canal.
- **<cloud>**: Proporciona información sobre los cambios en el canal de forma inmediata.
- **<copyright>**: Indica información sobre los derechos de autor del contenido publicado.
- **<docs>**: Especifica la dirección donde se encuentra la documentación sobre el formato RSS utilizado.
- **<generator>**: Muestra el software o herramienta utilizada para generar el canal.
- **<image>**: Contiene una imagen representativa del canal, utilizada por los agregadores de RSS.
- **<language>**: Indica el idioma en el que está escrito el contenido del canal.
- **<lastBuildDate>**: Define la fecha y hora en la que el contenido del canal fue modificado por última vez.
- **<link>**: Contiene la URL principal del canal.
- **<pubDate>**: Indica la fecha de la última publicación realizada en el canal.
- **<rating>**: Especifica una valoración PICS (Platform for Internet Content Selection) del canal.
- **<skipDays>** / **<skipHours>**: Define los días u horas en los que los agregadores deben omitir la actualización del canal.
- **<textInput>**: Proporciona un campo de entrada de texto dentro del canal.
- **<ttl>** (*Time to Live*): Indica el tiempo, en minutos, que el canal puede permanecer en caché antes de ser actualizado.
- **<webMaster>**: Contiene la dirección de correo electrónico del webmaster encargado del canal.

4.3.2 <ITEM>

Cada elemento **<item>** puede definir un determinado artículo en el canal RSS. En este caso, también debe contar con tres elementos fundamentales:

- **<title>**: hace referencia al título del artículo.
- **<link>**: define el hipervínculo al artículo.
- **<description>**: se encarga de describir el canal.

Los elementos opcionales más importantes de **<item>**:

- **<author>**: Especifica el correo electrónico del autor del artículo.
- **<category>**: Define una o varias categorías a las que pertenece el artículo.
- **<comments>**: Proporciona un enlace a los comentarios relacionados con el artículo.
- **<enclosure>**: Permite incluir archivos multimedia, como imágenes, audios o videos.
- **<guid>** (*Globally Unique Identifier*): Asigna un identificador único al artículo para evitar duplicados.
- **<pubDate>**: Indica la fecha y hora de la última publicación del artículo.
- **<source>**: Especifica la fuente original del artículo a través de un enlace.

4.4 Tecnologías de creación de canales de contenidos

Para crear un canal de contenido lo único que hace falta es un editor de XML y página web. Para la creación de páginas de sindicación podemos utilizar las herramientas de **Joomla** o **DRupal**. Programas que permiten crear una estructura de soporte para la creación y administración de contenidos.

Presentan la ventaja de ser de código abierto. Y con implementación mayoritariamente en *php*, utilizan también servidores web Apache y gestores de bases de datos MySQL.

4.5 Validación y publicación

Validación del archivo RSS

Después de haber creado un archivo RSS, debemos comprobar que el archivo cumple con todas las normas y requisitos, tanto XML como RSS. Para realizar dicho proceso usamos la validación.

Existen múltiples webs desde las cuales podemos realizar el proceso de validación de un archivo RSS. W3C posee su propio validador de RSS y lo ofrece gratuitamente a la comunidad.

Dos opciones existentes para validar un archivo RSS:

- **Validate by URI:** en este caso indicaríamos la ubicación del archivo RSS por medio de una dirección URI. El archivo RSS debe estar accesible en la red.
- **Validate by Direct Input:** validación por código directo. En este caso colocaríamos el código RSS directamente en el cuadro de texto disponible.

Publicación del archivo RSS

4.6 Directorios de canales de contenidos

Existen una serie de canales de contenidos de donde podemos extraer la información que deseemos. Estos canales están catalogados por temática. Por tanto, podemos suscribirnos al canal deseado. De esta forma, se actualizará el contenido integrado en el portal web.

Los directorios de canales de contenidos nos ofrecen las siguientes ventajas:

- Permiten el acceso a los ficheros RSS.
- Ayudan a la localización de los archivos RSS a cualquier usuario.
- Clasifican y catalogan por temáticas (u otros criterios) los ficheros RSS.

4.7 Agregación

Es necesario que el usuario disponga de un agregador. Podemos diferenciar entre una gran variedad de lectores RSS, que vamos a clasificar en tres categorías distintas:

- **Agregadores de escritorio:** aquellos que se pueden instalar en el ordenador de un usuario.
- **Agregadores en línea:** no es necesario que el usuario los instale. Solo hay que darse de alta en el sitio para utilizarlo.
- **Agregadores como *plugins*:** existen algunos navegadores como Firefox, Netscape, Opera y algunos más, que los incluyen entre sus programas para ofrecer un servicio al usuario.

Cuando el usuario ya ha seleccionado el agregador adecuado, tiene que elegir, a continuación, qué feeds o archivos RSS va a necesitar para poder llevar a cabo la correspondiente sindicación de contenidos.