

CFGS ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN RED

MODALIDAD PRESENCIAL

PROYECTO PROPIO

SR1E

MÓDULO: LENGUAJES DE MARCAS Y SISTEMAS DE GESTIÓN DE INFORMACIÓN

Profesora: Elvira Girón

Curso: 2019/2020

IES Clara del Rey

UNIDAD 2: HTML y XHTML

CONTENIDO

2.1. Introducción al HTML.

2.1.1. Estructura de una página

2.1.2. Etiquetas de contenido

2.1.2.1. Títulos

2.1.2.2. Párrafos

2.1.2.3. Listas.

2.1.3. Elementos de formulario

2.1.3.1. Campos de texto

2.1.3.2. Botones, desplegables.

2.1.4. Otros elementos de formato y agrupamiento

2.1.4.1. Tablas

2.1.4.2. Marcos

2.1.4.3. Capas.

2.1.5. Imágenes

2.1.6. Estilos en HTML

2.1. Introducción al HTML.

HTML (HyperText Markup Language, Lenguaje de Marcado de HiperTexto) es un lenguaje utilizado para la creación de páginas web. HTML es un estándar desarrollado por W3C (World Wide Web Consortium).

Una aplicación web, es una aplicación creada usando como base el lenguaje HTML. Por lo tanto, se trata de una aplicación que se ejecuta en un navegador de Internet.

Las aplicaciones web utilizan lenguajes que apoyan la labor de HTML y que son traducibles por parte de los navegadores.

Esencialmente las aplicaciones web actuales utilizan:

HTML. Para dar significado a los contenidos de la aplicación web. Permite indicar qué textos son títulos, cuáles son párrafos normales, cuáles son celdas de una tabla, cuáles son imágenes, etc.

CSS. Lenguaje que permite dar formato y maquetación a los contenidos. Color, tamaño de letra, posición, etc.

JavaScript. Permite diseñar la interactividad de la página. Permite que las acciones del usuario se puedan capturar y que la página reaccione a ellas. Ejemplos de sus posibilidades son que cuando el usuario arrime el ratón a una imagen ésta se ilumine, o que cuando el usuario haga clic en un título se muestre el contenido relacionado si estaba oculto o que el usuario pueda aumentar el tamaño de la letra de la página haciendo clic en un icono.

2.1.1. Estructura de una página

Elemento "html"

Un documento HTML contiene marcas (etiquetas), las cuales se escriben empleando los caracteres menor que "<", mayor que ">" y barra inclinada "/".

Por ejemplo, las etiquetas de inicio (<html>) y fin (</html>) representan al elemento raíz ("html") que en todo documento HTML hay que escribir.

Elemento "head"

Dentro del elemento "html", es decir, entre <html> y </html>, se debe escribir el elemento "head" que puede contener diversa información sobre el documento

Elemento "title"

Entre <head> y </head> se pueden escribir otros elementos. Por ejemplo, el elemento "title", el cual sirve para indicar el título del documento.

Elemento "body"

En un documento HTML, después del "head", hay que escribir el elemento "body"

El elemento "body" alberga todo el contenido (párrafos, imágenes, vídeos...) del documento, los cuales se mostrarán en el navegador.

Elemento "p"

En este caso, entre <body> y </body> se incluyen dos elementos "p" delimitados por la etiqueta de inicio <p> y la de cierre </p>

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" >
    <title></title>
  </head>
  <body>

  </body>
</html>
```

ESTRUCTURA BÁSICA DE UN DOCUMENTO HTML

<!DOCTYPE html> en la primera línea. Esta es la forma de indicar al navegador donde se visualice el documento que este está escrito en la versión HTML5.

lang="es-ES" en la etiqueta de inicio <html>. Esto sirve para especificar, mediante el atributo lang, el idioma del contenido del elemento "html". En este caso español (España)

<meta charset="UTF-8"> dentro del elemento "head". Así, se indica la codificación de caracteres del documento, en esta ocasión UTF-8 (8-bit Unicode Transformation Format, Formato de Transformación Unicode de 8 bits)

Los documentos HTML están formados por elementos cuya gran mayoría deben escribirse delimitados por una etiqueta de inicio <etiqueta> y otra de fin </etiqueta>, tales como: "html", "head", "body", "title" o "p". No obstante, existen otros elementos que no necesitan ser cerrados con la etiqueta de fin, como por ejemplo: "meta".

Para HTML 4.01 Strict escribiríamos:

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

Para HTML 4.01 Transitional (recomendado dentro de HTML 4.01) escribiríamos:

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

Comentarios en HTML

En un documento HTML, los comentarios pueden escribirse entre los caracteres "<!--" y "-->".

Ejemplo:

```
<!--La siguiente tabla muestra el horario de clase-->
```

Los comentarios no se mostrarán en el navegador donde se visualice el documento HTML.

En un navegador se mostrarán solamente los párrafos escritos en el "body"

No obstante, al ver el código fuente del documento en un navegador, sí se mostrarán los comentarios.

2.1.2. Etiquetas de Contenido

2.1.2.1. Títulos en HTML

h1 marcará títulos de primer nivel, h2 de segundo nivel,... y así hasta h6 (sexto nivel).

Ej:

`<h1>Sistema Solar</h1>`

`<h2> Planetas</h2>`

`<h3>Mercurio</h3>`

2.1.2.2. Párrafos en HTML

`<p> y </p>.`

`
` Salto de línea

`<p>Texto uno
`

`Texto dos
`

`Texto tres.</p>`

`<pre>` párrafo con sus espacios y saltos de línea `</pre>` muestra espacios y saltos de línea

`<hr>` línea horizontal

Ej.

`<p>Primer párrafo</p>`

`<hr>`

`<p>Segundo párrafo</p>`

Elementos que se emplean para señalar texto interior a un párrafo.

`` texto importante``

`` texto enfatizado``

`<mark>` texto resaltado`</mark>`

`` subíndice

`` superíndice

`<small></small>` letra pequeña

`<s></s>`efecto tachado

`` negrita uso desaconsejado en v.5

`<i></i>` cursiva uso desaconsejado en v.5

Consultar el siguiente enlace para escribir caracteres especiales <https://dev.w3.org/html5/html-author/charref>

2.1.2.3. Listas

Los elementos “ul”, “ol”, “li”, “dl”, “dt” y “dd”, permiten representar listas ordenadas (ordered lists), listas desordenadas (unordered lists) y listas de descripción de términos (description lists).

```
<body>
  <p>Lista desordenada de colores:</p>
    <ul>
      <li>Rojo</li>
      <li>Amarillo</li>
      <li>Verde</li>
      <li>Blanco</li>
      <li>Naranja</li>
    </ul>
</body>
```


Si cambiamos la etiqueta por , la lista aparecerá ordenada con números del uno en adelante y si queremos que el inicio sea distinto a 1 lo indicaremos con el atributo value="n"

```
<body>
  <p>Lista desordenada de colores:</p>
  <ol>
    <li value="5">mayo</li>
    <li>junio</li>
    <li>julio</li>
    <li>agosto</li>
    <li>septiembre</li>
  </ol>
</body>
```

Si lo que vamos a hacer es un listado de definiciones, podemos usar las etiquetas <dl>, <dt> y <dd>. Vamos a explicarlas por partes:

La etiqueta <dl> viene de los términos ingleses "Definiton list" y nos indica que dentro de ella, entre ella y su cierre, va a ir una definición.

La etiqueta <dt> viene de los términos "Definition term" y dentro de ella irá el término que vamos a definir. Para entendernos mejor, dentro de <dt> iría el título de la definición.

La etiqueta <dd> viene de los términos "Definition description" y nos dice que dentro de ésta irá la definición.

Si escribimos varios listados de definición, éstas se separarán automáticamente entre ellas para facilitar su diferenciación.

```
<body>
  <p>Glosario de siglas:</p>
  <hr>
  <dl>
    <dt>CSS</dt>
    <dd>Cascading Style Sheets</dd>
    <dt>HTML</dt>
    <dd>HyperText Markup Language</dd>
    <dt>W3C</dt>
    <dd>World Wide Web Consortium</dd>
  </dl>
</body>
```

2.1.3. Elementos de formulario

Los formularios html nos ayudan a aumentar la interactividad de nuestra Web y nos permiten recibir información de los usuarios de nuestro sitio. Los formularios html están compuestos por campos de texto y botones.

Una vez el usuario ha rellenado el formulario e introducido los valores en los campos, éstos son enviados para poder procesarlos. El envío de estos datos puede hacerse a un correo electrónico o a un programa que procese toda la información y nos ayude a hacer un seguimiento.

La etiqueta de los formularios es `<form>` y su cierre `</form>`. Todo lo que vaya dentro de estas etiquetas, serán partes del formulario.

El atributo **“action”** indica el tipo de acción que va a realizar el formulario. Anteriormente indicamos que la información podía enviarse a un correo electrónico o a un programa que la gestione. Es mediante esta etiqueta que se gestionan los formularios.

Si queremos que el formulario se envíe a un correo, la acción quedaría escrita de la siguiente manera:

<form action=mailto:direcciondelcorreo@correo.com...></form>.

Este tipo de envíos se utilizarían para casos de formularios de contacto, de sugerencias, etc.

Si lo que queremos es que la información sea enviada a un programa que la gestione, debemos indicarle en la acción, la url del archivo donde se encuentra ubicado el programa que la gestionará. Lo escribiríamos de la siguiente manera:

<form action="dirección completa del archivo que la gestionará" ...> </form>.

Este tipo de envíos se utilizarían para casos de formularios de encuestas, cuestionarios, etc.

Mediante el atributo "**method**" le indicamos al formulario la forma en la que el formulario será enviado. Existen dos valores posibles: get y post.

El valor get es el valor por defecto. Si no concretamos el method, la información se enviará a través de este medio.

"get" indica que los datos enviados se adjuntarán en la barra de direcciones del cliente, al final de la url correspondiente y después de un signo de interrogación de cierre. Si se envía más de un dato, éstos irán separados por el símbolo &.

Un ejemplo de un formulario enviado por el method="get" sería el siguiente:

http://www.....?nombre1=valor1&nombre2=valor2

El valor post indica que el método de envío no se hará a través de la url, sino formando parte del cuerpo de la petición.

Mediante el atributo "**enctype**" indicaremos la forma en la que viajará la información que se mande a través del formulario HTML.

La forma puede ser de varios tipos, aunque el más común es que la información se envíe como texto plano (enctype="text/plain").

Ejemplo de código de un formulario estándar enviado a un correo electrónico:

```
<form action="mailto:direcciondelcorreo@correo.com" method="post" enctype="text/plain"> </form>
```

2.1.3.1. Campos de texto

Los campos de entrada de los formularios se definen mediante la etiqueta `<input>` y sus diferentes valores. Estos valores nos permiten utilizar varios tipos de cajas y de formas.

`<input type= "text">`

Podemos completar el estilo de esta caja gracias a los siguientes atributos:

size: mediante el atributo `size` podemos definir el tamaño de la caja. Es decir, la apariencia de la misma. Si el texto que vamos a escribir no cabe en la caja, éste se desplazará pero sin que la caja varíe su tamaño. El texto irá desapareciendo por la izquierda.

maxlength: Este atributo nos sirve para definir la cantidad máxima de letras que se pueden escribir en la caja. No se podrán escribir en la caja más caracteres que los indicados por el `maxlength`.

value: Define si queremos que en la ventana haya un texto ya preescrito. Este texto puede ayudar al usuario a saber cuál es el tipo de dato que se piden en ese campo.

name: para ponerle un nombre al campo. Esto resulta muy útil a la hora de gestionar la información que nos llegue.

Ejemplo:

```
<input type="text" size="15" maxlength="30" value="Nombre" name="nombre">
```

Tipo de caja “text” con un tamaño de 15 y un límite de caracteres de 30. Además, tiene un texto preescrito en él Nombre.

Un **<textarea>** es como una caja de texto sencilla, tipo “text”, pero en la que dejamos al usuario espacio de sobra para poder escribir un texto largo. Son útiles para que el usuario contacte con nosotros, plantee sus dudas o sugerencias, etc.

Su etiqueta es <textarea> y su correspondiente cierre </textarea>. Todo lo que vaya dentro de estas etiquetas formará parte del texto.

Sus atributos básicos son “rows” y “cols” que servirán para definir el tamaño del textarea. Además, no debemos olvidar el atributo “name”, que servirá para definir el nombre de este campo.

Un textarea llamado “comentarios” y con unas dimensiones de 10 filas y de 40 columnas, tendría el código siguiente:

```
<textarea name="comentarios" rows="10" cols="40">Escribe aquí tus comentarios</textarea>
```

En muchos formularios se piden datos personales que el usuario no quiere que se lean a su alrededor: contraseñas, etc. Para ello debemos crear una caja que no muestre los datos que se escriben en ella y que los codifique. Esto es más sencillo de lo que parece.

El siguiente código es de un campo con datos codificados:

```
<input type="password" name="contraseña">
```

2.1.3.2. Botones, desplegables

Las listas de opciones son de gran ayuda para los formularios en los que queremos saber la opinión de un aspecto concreto en nuestra Web, dentro de diferentes opciones planteadas previamente por nosotros. Las listas de selección u opciones se escriben gracias a la etiqueta **<select>** y a su cierre.

```
<select name="transporte">
<option>Coche</option>
<option>Avión</option>
<option>Tren</option>
</select>
```

Pero podemos personalizar esta lista de opciones mediante una serie de atributos. Por ejemplo, podemos hacer que se vea más de una opción a la vez en la lista. Para ello utilizaremos “size”, indicando el número de opciones que queremos que se muestren.

```
<select name="transporte" size="2">
```

También podemos definir si queremos que se pueda elegir más de una opción a la vez mediante las teclas ctrl y clic. Para ello utilizaremos el atributo multiple. Mira primero cómo quedaría nuestro código:

```
<select name="transporte" size="2" multiple>
```

Otra cosa que podemos predefinir es que la opción que aparezca seleccionada al principio no sea la primera que está en la lista. Esto se realiza gracias al atributo selected y lo tenemos que poner en la opción que queramos que aparezca como seleccionada:

```
<option selected>Tren</option>
```

El atributo value es muy útil para gestionar los datos, sobre todo si le llega a un programa. Con este atributo lo que se le hace es atribuir un número o una letra a cada opción.

El atributo se escribe dentro de cada opción y se define el valor del mismo:

```
<select name="transporte">  
<option value="1">Coche</option>  
<option value="2">Avión</option>  
<option value="3">Tren</option>  
</select>
```

Listas de botones

Son otro tipo de listas de opciones o de selección. Son listas con botones de tipo radio. Estos botones se escriben mediante la etiqueta `type="radio"`.

```
<input type="radio" name="transporte" value="1">Coche  
<br>  
<input type="radio" name="transporte" value="2">Avión  
<br>  
<input type="radio" name="transporte" value="3">Tren
```

Por defecto no saldrá ninguna opción activada. Pero si queremos activar alguna de ellas podemos hacerlo mediante el atributo `checked`.

```
<input type="radio" name="transporte" value="2" checked>Avión
```

Listas de cajas: checkbox

Las cajas se escriben mediante `type="checkbox"`

Las listas de cajas permiten seleccionar una o varias opciones

```
<input type="checkbox" name="transporte" value="1">Coche  
<br>  
<input type="checkbox" name="transporte" value="2" >Avión  
<br>  
<input type="checkbox" name="transporte" value="3">Tren
```


Botones submit y reset

Para enviar la información, el formulario necesita de un botón que le indique que el formulario ha concluido y que pueden enviar la información

Los botones de envío se crean con la etiqueta “**input**” de tipo “**submit**”. Además, hay que definir el texto que queremos que esté escrito en el botón. Esto lo conseguimos con el atributo “value”. Así, el código de un botón sencillo de envío de formulario sería el siguiente:

```
<input type="submit" value="Enviar información">
```

Es posible que el usuario una vez avanzado el formulario, quiera reiniciarlo desde el principio y volver a escribir la información. Para ello es muy útil el botón para borrar la información.

El botón se escribe casi igual que el anterior, cambiando el type, que en el caso de estos botones es “**reset**” y cambiando el mensaje interno del botón, o sea, el “value”.

El ejemplo sencillo de este botón tendría el siguiente código:

```
<input type="reset" value="Borrar información">
```

2.1.4. Otros elementos de formato y agrupamiento

2.1.4.1. Tablas

Una tabla en html viene marcada por las etiquetas `<table>` `</table>`. Entre esas dos etiquetas definiremos la tabla, las celdas que queremos, las columnas y las características de cada uno de estos parámetros.

A una tabla también lo podemos definir el fondo de la misma. Esto lo podemos conseguir con el parámetro `"bgcolor"`, que nos pondrá un color de fondo, o `"background"` para poner una imagen de fondo. Recuerda que si la imagen es más pequeña que la tabla, ésta se repetirá tanto a lo ancho como a lo largo.

Otro aspecto que podemos definir de la tabla es el borde. Esto lo haremos con el parámetro `"border"`. Como en todas los parámetros que ya hemos visto escribiremos: `border= "x"` siendo la x un número. Ese número indicará el grosor del borde. Si no ponemos borde o lo escribimos `"0"`, la tabla no mostrará borde ninguno. Por supuesto, también podemos darle color al borde, escribiendo la etiqueta `"bordercolor"` e indicando el color que queramos para nuestro borde.

El parámetro `"width"` indicará la anchura de la tabla. Esta anchura la podemos poner en píxeles (`width= "300"`) o con porcentaje (`width= "100%"`).

Dos parámetros más son `cellspacing` (que define el espacio entre las celdas de la tabla) y `cellpadding` (que le marca a la tabla el espacio que debe dejar alrededor del texto dentro de una celda).

Como ejemplo, escribiremos el hipotético supuesto de querer una tabla que sea ancha al 100%, con un borde azul de un píxel de grosor y con un `cellpadding` de 10 y con un `cellspacing` de 10. El código quedaría de la siguiente forma.

```
<table width="100%" border="1" bordercolor="#0000FF" cellspacing="10" cellpadding="10"></table>
```

Las filas se escriben gracias a las etiquetas `<tr>` con su correspondiente cierre `</tr>`. El contenido de las columnas que están dentro de la fila lo podemos alinear tanto horizontal como verticalmente.

Para alinearlo verticalmente utilizaremos el atributo “valign” para poder alinearlo arriba de la celda (“top”), en el centro (“middle”) o debajo (“bottom”).

Para alinearlo horizontalmente utilizaremos el atributo “align”. Con este atributo podremos alinear el contenido de las celdas en el centro (“center”), a la izquierda (“left”), a la derecha (“right”) o justificado (“justify”).

Por supuesto a las filas también les podemos definir el color de fondo (“bgcolor”) y el color del borde (“bordercolor”).

Las celdas que van dentro de cada fila tenemos que escribirlas con la etiqueta <td> y su correspondiente cierre </td>.

Al igual que en las filas, en las celdas podemos definir la alineación del contenido que está dentro con los atributos “valign” y “align”.

Las celdas poseen unos atributos que nos ayudan a poder agrupar tantas celdas o tantas columnas como indiquemos en él. Para agrupar celdas en horizontal utilizaríamos el atributo “colspan” y para agrupar celdas en vertical el atributo “rowspan”.

Por ejemplo, para agrupar en una celda 2 columnas tenemos que escribir: <td colspan=“2”></td>. Y para agrupar dos filas, la indicación sería la siguiente: <td rowspan=“2”></td>.

Las celdas escritas con la etiqueta <th> y su correspondiente cierre, admiten los mismos atributos que las etiquetas <td> y funcionan de la misma forma, salvo que el contenido que esté dentro de una etiqueta <th> está considerado como el encabezado de la tabla, por lo que se creará en negrita y centrado sin que nosotros se lo indiquemos.

nowrap es un *atributo* que le podemos colocar a la etiqueta de la celda y que obligará al navegador a no romper esa línea, o sea, a no hacer ningún salto de línea. Con este atributo en la celda, el navegador no respeta el ancho predefinido de la tabla, si es que lo hubiera, y respeta el ancho de la frase, ya que no puede partirla.

Por tanto, si la frase es más larga que el ancho definido de la tabla, ésta se estirará todo lo necesario para albergar la frase sin saltos de línea.

Etiqueta “caption”

Esta etiqueta sirve para poder ponerle un título o encabezado a la tabla. Se puede poner el encabezado arriba o abajo, dónde se prefiera, mediante la etiqueta “align”: “align=top” para ponerlo arriba y “align=bottom” para ponerlo abajo. En el siguiente ejemplo aparecerá abajo.

```
<caption align="bottom">Encabezado de la tabla.</caption>
```

Ejemplo:

```
<table width="100%" border="1" cellpadding="0" cellspacing="0" bordercolor="#000000">
<tr>
<th>Encabezado 1</th>
<th>Encabezado 2</th>
<th>Encabezado 3</th>
</tr>
<tr>
<td rowspan="2" valign="middle" align="left">Este texto está alineado al centro
verticalmente y a la izquierda horizontalmente</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td colspan="2">&nbsp;</td>
</tr>
</table>
```

2.1.4.2. Marcos

Un Frame HTML (en castellano “marco”), es una ventana independiente dentro de la propia ventana del navegador. Cada frame tiene sus propios bordes y también sus barras de desplazamiento. Mediante un frame conseguimos dividir la ventana del navegador en varias subventanas independientes entre ellas. Cada una de estas subventanas posee un documento html propio.

La habitual etiqueta html <body> es sustituida por <frameset> que indica que esa ventana va a dividirse en diferentes marcos.

Dentro de la etiqueta frameset debemos indicar el número de marcos que va a haber (ya sea en columnas “cols” o en filas “rows”) y el tamaño de cada uno de ellos.

```
<frameset cols="150,50%">
```

Este tamaño puede asignarse por porcentaje, por números absolutos o sobre el espacio sobrante. Esta forma de tamaño se indica mediante un asterisco y le estamos indicando al marco que coja el espacio sobrante que dejan los demás marcos.

Podemos también colocar un frameset dentro de otro. Es decir, podemos dividir un marco de frameset en otros submarcos. Esto se hace colocando la etiqueta dentro de otra de ellas.

Otra parte destacable de la estructura es <frame>. Como vemos, debemos escribir tantos <frame> como números de éstos hayamos creado en <frameset>. En nuestro caso eran dos, así que hemos escrito dos <frame>.

<frame> tiene dos atributos básicos y obligatorios. El primero es el atributo name, en el que debemos indicar el nombre de cada uno de los marcos. El segundo es la fuente o la dirección del documento html que ocupará dicho <frame>.

```
<frame name="indice" src="frame-ejemplo-2.html">
```

```
<frame name="principal" src="frame-ejemplo.html">
```

La etiqueta `<noframes>` es para el supuesto de que el navegador no acepte frames. En caso de no aceptarlos, saldrá el texto que hayamos puesto para esos supuestos.

```
<noframes>

<p>Si tu navegador no acepta frames, estarás viendo este mensaje. ¡Lo sentimos!

</p>

</noframes>
```

2.1.4.3. Capas

Con la utilización de las divisiones, las cuales son llamadas capas por muchos desarrolladores, podremos crear distintas secciones dentro del contenido de la página, dependiendo del significado que tenga dicho contenido. Dichas secciones además podrán ser situadas en cualquier punto de la página y se le podrán asociar hojas de estilos.

Para poder definir una división en HTML, es necesario utilizar la etiqueta `<div></div>`. Además, en la definición de la etiqueta, para poder asignarle estilos será recomendable utilizar el atributo `id` para asociarle un nombre identificativo mediante el cual definiremos los estilos para la sección. También existe la posibilidad de utilizar el atributo `class`, que permite definir una clase con unos estilos que puedan ser comunes a varias secciones.

Una vez que hemos definido la división podremos establecer en su interior todo el contenido que queramos mostrar en ella. Además, esta etiqueta permite utilizar un atributo para modificar sus propiedades, que es el siguiente:

`align`

Este atributo no está soportado en HTML5, pero con él se puede establecer la alineación del contenido que se defina dentro de la sección. Admite los valores `left`, `right`, `center` y `justify`.

Todas las demás características que se quieran modificar en cada sección hará necesario la utilización de estilos.

Además de la etiqueta `div`, HTML5 incorpora nuevas etiquetas para definir la estructura de la página Web y que también ayudan a identificar de una manera más fácil el contenido que se define en su interior. Se recomienda su uso para no utilizar en todas las secciones la etiqueta `div` y para aportar un mayor significado semántico a los contenidos que los motores de búsqueda vayan a indexar.

Dichas etiquetas son las siguientes:

`<header>`

`<nav>`

`<article>`

`<section>`

`<aside>`

`<footer>`

`<address>`

Mediante la definición de varias divisiones dentro de la página Web podemos crear la estructura en la que posteriormente añadiremos el contenido que queremos mostrar a los usuarios de la Web. Generalmente, a la hora de desarrollar una página Web podemos definir las siguientes secciones:

Cabecera

Menú

Contenido

Pie

Cabecera

A la hora de definir la estructura de una página Web debemos empezar por la parte superior de la página, es decir, por la cabecera, e ir bajando hasta la parte inferior, hasta el pie. Así que, para comenzar a definir la estructura, empezaremos estableciendo la cabecera, y para ello tendremos que utilizar la etiqueta <header>. En su interior podremos colocar un logotipo identificativo para la página Web, y también un título, entre otras cosas.

```
<header>

  

  <h1>Bienvenidos al portal</h1>

</header>
```


Menú

Una vez que ya hemos definido la cabecera, vamos a crear un menú de navegación justo debajo, a través del cual los usuarios puedan acceder a cualquiera de los apartados del sitio Web. Para ello tendremos que utilizar la etiqueta `<nav>`, y en su interior tendremos que colocar todos los enlaces a las distintas páginas del sitio Web. Al establecer los estilos para esta sección, es conveniente que los elementos del menú aparezcan de forma horizontal y siempre en el mismo sitio, para que el usuario no se pierda.

```
<nav>

  <a href="inicio.html">Inicio</a>

  <a href="noticias.html">Noticias</a>

  <a href="empresa.html">Nuestra empresa</a>

  <a href="contacto.html">Contactar</a>

</nav>
```

Contenido

Lo siguiente que tendremos que hacer es definir la estructura para el contenido principal de la página Web. Para ello utilizaremos varias etiquetas, ya que por ejemplo podemos crear un determinado artículo que tenga una cabecera y un pie, además de la información del propio artículo. Por lo tanto, para comenzar a definir el artículo utilizaremos la etiqueta `<article>`.

En su interior lo primero que haremos será crear una cabecera con el título del artículo, por lo que tendremos que utilizar la etiqueta <header>. Debajo tocará establecer la información del artículo, para lo que podemos utilizar las etiquetas <p>. Para acabar la redacción del artículo podemos añadirle un pie mediante la etiqueta <footer>, en la que definiremos la información respecto al autor del artículo.

```
<article>

  <header>

    <h1>Título del artículo</h1>

  </header>

  <p>Contenido del artículo</p>

  <footer>

    <p>Artículo escrito por...</p>

  </footer>

</article>
```

Pie

Por último, solamente nos quedará definir el pie para la página Web, donde se puede colocar cierta información respecto al autor de la página Web, cualquier información de contacto, logotipos de la página con un tamaño menor que en la cabecera, información sobre los derechos de autor de la Web, etc. Para ello tendremos que utilizar la etiqueta <footer>.

```
<footer>

  <p>Todos los derechos reservados</p>

  <p>Página Web desarrollada por...</p>

  

</footer>
```

Utilización de las capas

Gracias a las divisiones que podemos hacer en las páginas Web, conseguimos dotar a las secciones de contenido formato específico mediante las hojas de estilos. Además, una de las ventajas respecto a los frames es que permiten a los buscadores que indexen la página Web de manera correcta. Incluso permiten mejorar la accesibilidad de la Web, puesto que los usuarios que necesiten una herramienta externa podrán conocer en qué parte de la página se encuentran.

Por lo tanto, se puede afirmar que cuando se quiere definir la estructura de la página Web es muy importante utilizar las distintas etiquetas para realizar divisiones, ya que actualmente es el estándar que se sigue.

2.1.5. IMÁGENES

El tag básico para colocar una imagen es “img”. Este tag, a diferencia de la gran mayoría de los tags de html, no necesita un cierre. Atributos que lo pueden acompañar:

El atributo “src” es imprescindible para poder colocar una imagen. Este atributo es el que indica dónde se encuentra alojada la imagen que queremos mostrar. Se escribe así: ``, siendo “x” la dirección o la url dónde se encuentra situada la foto.

La foto podemos alinearla en la página como queramos mediante “align”, utilizando los atributos “left” para alinearla a la izquierda, “right” para alinearla a la derecha, “top” para alinearla arriba, “bottom” para alinearla abajo y “middle” para alinearla al centro.

También podemos poner una descripción de la imagen dentro de la misma para que la gente puede leerla al mantener el ratón encima de ella. Esta descripción podemos escribirla mediante el atributo “alt” y lo escribiríamos de la siguiente manera: `alt= “x”`, siendo “x” la descripción que le gente leerá al pasar el ratón por encima.

Debemos aclarar que no con todos los navegadores podemos ver el “alt” de manera visual. Hay algunos que no lo representan.

Atributos opcionales pero muy recomendables, son el height y el width. El height marca la altura de la imagen, mientras que el width marca la anchura. Son recomendables porque así ayudaremos al navegador a representar la imagen.

Se le puede adjudicar un borde a la fotografía utilizando el atributo “border”

Ejemplo:

```

```

Para hacer una imagen responsive adaptada al tamaño de su contenedor, tan solo tienes que añadir el siguiente atributo a la etiqueta img: style=”max-width:100%;width:auto;height:auto;”. Con esto conseguirás que la imagen se adapte a los diferentes cambios en el tamaño de pantalla.

2.1.6. Estilos en HTML

Para cambiar el estilo en el que por defecto se visualizan los elementos de un documento HTML en un navegador web, existen varios métodos:

- Estilo en línea (Inline): utilizando el atributo style.
 <p style="color:red"> </p> ó style="color:codhex"
- Estilo interno (Internal): usando un elemento "style"dentro de "head"
 <style>
 p {color:red;} ó {color:#codhex;} para modificar el color del texto de un elemento
 </style>

En “head”

Y en “body” el texto del elemento “p” aparecerá en el color indicado

Si se desea cambiar el color de fondo se utilizará:

```
<style>
body {
background-color: #codhex
}
</style>
```

Con la misma estructura se pueden definir colores de fondo para otros elementos como encabezados (h1, h2...), párrafos, etc

```
<head>
<style>
body {
    background-color: #93B874;
}
h1 {
    background-color: #00b33c;
}
p {
    background-color: #FFFFFF;
}
</style>
</head>
<body>
<h1>Encabezado con fondo verde</h1>
<p>Párrafo con fondo blanco</p>
</body>
```

Consultar código de color en

- <https://htmlcolorcodes.com/es/>
- <https://www.w3schools.com/colors/>

- Estilo externo (External): empleando un archivo CSS (Cascading Style Sheets) externo al documento HTML.

ANEXO

Convenciones

- Los nombres de los elementos HTML y sus atributos se deben escribir en minúsculas
- Los valores de los atributos en HTML deben ir entre comillas dobles
- Indentar el código con dos espacios, así ordenado mejora su legibilidad
- No introducir espacios antes o después del signo "igual"
- Usar UTF-8 como encoding.
- No cerrar elementos autocontenidos, por ejemplo usa `
` en lugar de `
`
- Evita el uso de estilos en línea
- Evita el uso de entidades HTML siempre que sea posible (salvo por ejemplo para `<` y `&`)
- Especifica el atributo lang en el elemento html:

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

- Especifica siempre el atributo for cuando añadas un elemento label
- Internet Explorer soporta el uso de una etiqueta de compatibilidad meta indicando cómo tratar el código, usar siempre que se pueda:

```
<meta http-equiv="X-UA-Compatible" content="IE=Edge">
```

GLOSARIO

HTTP	Hypertext Transfer Protocol
URL	Uniform Resource Locator El esquema del URL indica el protocolo de red usado para recuperar la información del recurso identificado. Entre los esquemas URL, se encuentran http, https, mailto y ftp
HTTPS	Hypertext Transfer Protocol Secure
FTP	File Transfer Protocol
mailto	Esquema de Identificador Uniforme de Recursos (URI) para direcciones de correo electrónico .
URI	Uniform Resource Identifier
Servidor web HTTP	HTTP funciona bajo el paradigma "Cliente – Servidor". El cliente es el Browser que solicita páginas Web y el servidor HTTP o servidor Web las entrega.
Alojamiento WEB (hosting)	Espacio donde se aloja un sitio web para que sea «visible» en Internet. El hosting es el uso que se da de todos o parte de los recursos de un servidor web (memoria, espacio del disco duro, ancho de banda, etc.) para alojar una web y que todo el mundo pueda verla en Internet a través del dominio que tenga.
DOM	Document Object Model

BIBLIOGRAFÍA

- Lenguajes de Marcas y Sistemas de Gestión de Información. IBERGARCETA PUBLICACIONES, S.L.
ISBN: 9788415452171

REFERENCIAS

- <https://jorgesanchez.net/manuales/html/introduccion-html.html>
- <https://www.w3schools.com/tags/>
- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element>
- <http://validator.w3.org/>
- <https://www.w3.org/TR/html5/>