

## 1.8 Generación de documentos y sitios web.

### Diseño web con HTML5

El lenguaje de marcas HTML5 es la versión actual del lenguaje HTML. Es el lenguaje que se utiliza para el diseño de páginas web y hace referencia a un estándar desarrollado por W3C.



El estándar HTML5 contiene un amplio conjunto de tecnologías que permite crear sitios web, así como aplicaciones web de envergadura. Algunas características de HTML5 agrupadas por su función son:

- **Semántica:** permite describir con mayor precisión cuál es su contenido.
- **Rendimiento e integración:** Proporciona una mayor optimización de la velocidad y un mayor uso del hardware a través de la introducción de un diseño adaptativo.
- **CSS3:** lenguaje de hojas de estilos que nos ofrece una gran variedad de opciones para realizar diseños más sofisticados
- **Multimedia:** nos proporciona un excelente soporte para utilizar contenido multimedia, como son audio y vídeo nativamente.
- **Gráficos y efectos 2D/3D:** proporciona una amplia gama de nuevas características que se ocupan de los gráficos en la web, como son canvas 2D, WebGL, SVG, etc.

Lo que se llama **web semántica** es un concepto que propone la inclusión de metadatos semánticos en la web, para que agentes informáticos, como pueden ser los navegadores o robots de los motores de búsqueda, puedan procesar su contenido, razonar con éste, combinarlo y realizar deducciones lógicas para resolver y dar una respuesta más acertada. Así pues, los **elementos que la especificación HTML5 aporta para poder estructurar un documento web con semántica estandarizada** son: <header>, <footer>, <section>, <aside>, <nav>, <article>, <hgroup>, <time>, etc.

**'Web crawler' o robot web**  
Software que inspecciona páginas web de forma metódica y automatizada. Se utiliza para poder referenciar las páginas visitadas a través de un motor de búsqueda, como Google, Yahoo, Bing, etc.

El lenguaje HTML5 también permite realizar diseños web adaptativos, en inglés *Responsive Web Design*. Así pues, dispone de técnicas para que una web sea visible de forma óptima en todo tipo de dispositivos capaces de visualizar páginas web, es decir, desde móviles hasta ordenadores o tabletas. Por eso se utilizan estructuras con imágenes fluidas y hojas de estilos CSS.

### Formato del HTML5

Los objetivos del HTML5 son la simplicidad y la permisividad. Así pues, HTML5 reanuda la sintaxis permisiva de HTML4 y hace desaparecer las reglas estrictas de sintaxis de XHTML. Sin embargo, es recomendable seguir algunas de las normas de sintaxis de XHTML:

- Usar sólo minúsculas.
- Usar las comillas dobles para indicar el valor de los atributos.
- Cerrar todas las etiquetas.
- Sangrar el código para aumentar la legibilidad.

El **formato** de una etiqueta HTML es:

```
<NombreEtiqueta atributo1="valor atributo1" ... atributoN="valor atributoN">
  Elementos afectados por la etiqueta
</NombreEtiqueta>
```

Cada etiqueta puede tener una serie de **atributos** que modifican el comportamiento de la misma, por ejemplo:

### Ejemplo de formato de una etiqueta

```
<!-- En la siguiente línea definimos un párrafo -->
<p>Esta es la primera página web</p>

<!-- Fijémonos en el cierre de la siguiente etiqueta, directamente con la / -->
<br />

<!-- En la siguiente línea estamos definiendo un link -->
<a href="https://iesagora.educarex.es/" target="_blank">Link IES</a>
```

a es el nombre de la etiqueta, href es el atributo a través del cual definimos dónde debe llevar el enlace, target permite especificar dónde se debe abrir el enlace; \_blank significa que se abrirá en una nueva ventana, Link IES es el texto que hace de *link* y que se ve afectado por la etiqueta <a>, </a> sirve para cerrar la etiqueta y marcar donde deja de surtir efecto la misma.

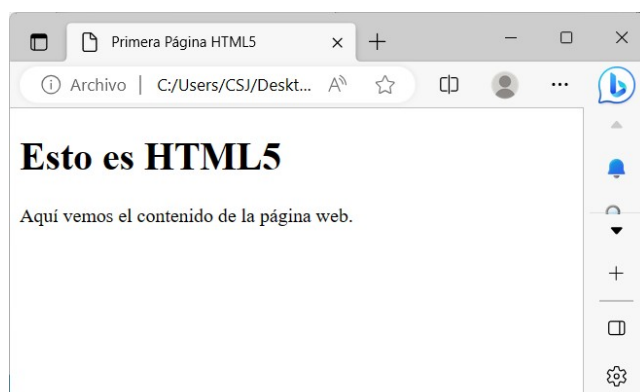
Como puede apreciarse en el ejemplo anterior, los comentarios de HTML5 se ponen entre los caracteres <!-- y -->.

## Estructura del código HTML5

Generalmente, la estructura de una página web en HTML5 suele tener un patrón similar.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <title>Primera Página HTML5</title>
</head>
<body>
  <h1>Esto es HTML5 </h1>
  <p>Aquí vemos el contenido de la página web. </p>
</body>
</html>
```

En la siguiente imagen se puede apreciar cómo se visualiza una página web en el navegador.



En el ejemplo anterior se puede apreciar que el documento html se diferencia en dos partes: *head* y *body*.

En el **head** se encuentra la información adicional de la página web, como puede ser el título que se visualiza en la pestaña del navegador a través del elemento <title>, así como también elementos <meta>, <script> y <link>, entre otros.

En el **body** está el contenido que se visualiza de la página web, el cual puede estar formado por multitud de elementos html que dan formato al documento.

Aunque en la especificación HTML5 se ha hecho un gran esfuerzo para relajar la sintaxis y que las etiquetas <html>, <head> y <body> no serían obligatorias, es del todo recomendable que estén allí.

Veamos algunos de los **apartados que se pueden definir en una página web**:

**1. El tipo de documento:** la primera línea de un documento HTML corresponde a la declaración del tipo de documento, el **doctype**, y sirve para indicar qué versión de HTML se ha utilizado en el documento.

```
<!--Declaración del doctype en HTML5-->
<!DOCTYPE html>
```

```
<!-- Declaración del doctype en HTML4.01 transicional -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<!--Declaración del doctype en XHTML 1.0 estricto-->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

DTD, Document Type Declaration, es una descripción de estructura y sintaxis de un documento, en este caso HTML. Se utiliza para validar el documento.

Así pues, en el ejemplo anterior puede apreciarse que uno de los objetivos de HTML5 es la simplicidad y sobre todo la compatibilidad en versiones anteriores, así como versiones futuras; por tanto, incluir el número de versión no tendría sentido.

**2. El documento HTML:** el elemento html indica el principio del contenido de la página HTML.

```
<!-- Sintaxis en HTML5. -->
<html lang="es">
```

```
<!-- Sintaxis en XHTML 1.0 -->
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es" >
```

**3. La codificación de caracteres:** se suele indicar qué codificación de caracteres se ha utilizado en una página web.

```
<!-- Sintaxis en HTML5 -->
<meta charset="UTF-8" />
```

```
<!-- Sintaxis en HTML 4.01 transicional -->
<meta http-equiv="Content-Type" content="text/html; charset="UTF-8" >
```

```
<!-- Sintaxis en XHTML 1.0 estricto -->
<meta http-equiv="Content-Type" content="text/html; charset="UTF-8" />
```

**4. Estilos CSS** : para indicar los estilos que se utilizarán en la página html.

```
<!-- Sintaxis en HTML5 -->
<link href="MisEstils.css" rel="stylesheet" />

<!-- Sintaxis en XHTML 1.0 estricto -->
<link href="MisEstils.css" rel="stylesheet" type="text/css" />
```

**5. Los *scripts***: en una página web se pueden incluir *scripts*.

```
<!-- Sintaxis en HTML5 -->
<script src="MeusScripts.js"> </script>

<!-- Sintaxis en XHTML 1.0 estricto -->
<script src="MeusScripts.js" type="texto/javascript"> </script>
```

## Elementos de HTML5

Todos los elementos del estándar HTML5 y sus atributos, tanto los obligatorios o requeridos como los opcionales, se pueden consultar, por ejemplo, en la web: [Search: "HTML5 lista elementos" \(mozilla.org\)](#)

En la especificación HTML5 también se determina qué elementos se han declarado obsoletos. HTML5 debe ser compatible con las versiones anteriores; así pues, los elementos obsoletos seguirán siendo interpretados correctamente por los navegadores.

Si se desea consultar todos elementos HTML estándar, los no estándar y los que están obsoletos y los que no, se puede acceder, por ejemplo, a: [developer.mozilla.org/en-US/docs/Web/HTML/Element](#).

Puedes consultar los elementos obsoletos en la siguiente dirección  
<https://www.w3.org/TR/2014/REC-html5-20141028/obsolete.html>

## Encabezamientos

Los encabezamientos, *headings*, son muy importantes en HTML5. Se definen con las etiquetas <h1> hasta <h6>, siendo <h1> la más importante, y <h6> definiría el encabezamiento menos importante.

Automáticamente, los navegadores añadirían un espacio (margen) antes y después de cada encabezamiento.

Es mejor utilizar los encabezados para escribir títulos en lugar de utilizar, por ejemplo, las etiquetas <bold> o <big>, ya que los motores de búsqueda usan los encabezados para indexar la estructura y el contenido del sitio web.

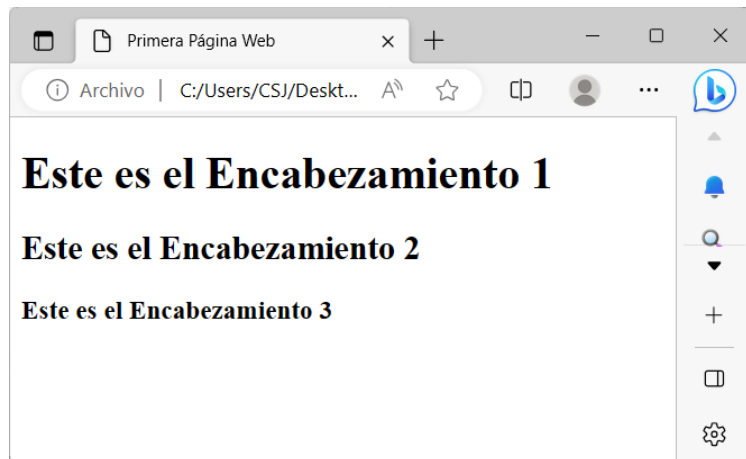
**Ejemplo** de visualización de diferentes encabezados

```
<!doctype html>

<html lang="es">
<head>
  <meta charset="UTF-8" />
  <title>Primera Página Web</title>
</head>
<body>
```

```
<h1>Este es el Encabezamiento 1</h1>
<h2>Este es el Encabezamiento 2</h2>
<h3>Este es el Encabezamiento 3</h3>
</body>
</html>
```

Visualización de diferentes encabezados en un navegador



## Párrafos

Los párrafos se definen con el elemento `<p>`. Realmente no podemos estar seguros de cómo se visualizará un párrafo, ya que existen diferentes dimensiones de pantalla y los párrafos se verán utilizando más o menos líneas.

Con HTML no se puede cambiar la salida añadiendo espacios en blanco o saltos de línea, puesto que el navegador abolirá estos caracteres extras para mostrar la página. De esta forma, cualquier número de espacios en blanco o de saltos de línea contará como un solo espacio en blanco.

Si se desea que se tengan en cuenta los espacios en blanco y/o saltos de línea, podemos utilizar el elemento `<pre>`.

## Ejemplo de visualización de párrafos

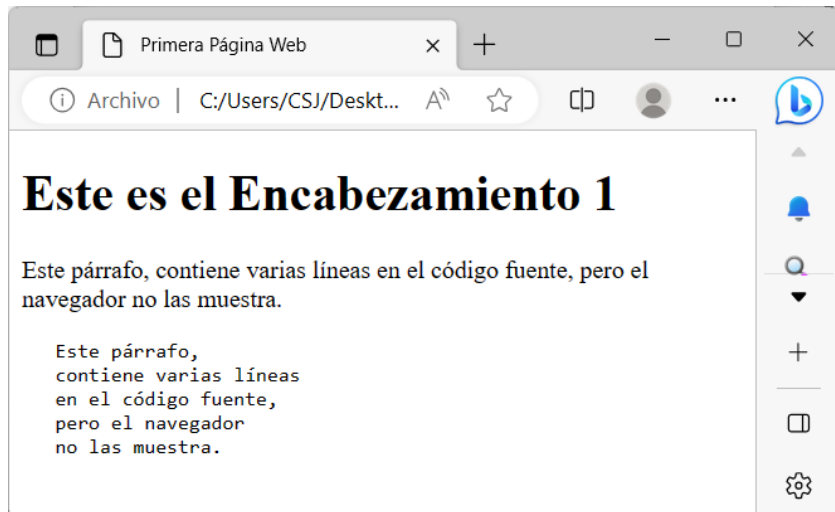
```
<!doctype html>

<html lang="es">
<head>
  <meta charset="UTF-8" />
  <title>Primera Página Web</title>
</head>
<body>
  <h1>Este es el Encabezamiento 1</h1>
  <p>
    Este párrafo,
    contiene varias líneas
    en el código fuente,
    pero el navegador
    no las muestra.
  </p>

  <pre>
    Este párrafo,
    contiene varias líneas
    en el código fuente,
    pero el navegador
```

```
    sí las muestra.  
</pre>  
</body>  
</html>
```

El resultado del ejemplo se muestra así:



## Colores en HTML5

Existen varias formas de especificar los diferentes colores que se utilicen para mostrar el contenido del sitio web:

- con el nombre del color en inglés
- con su código RGB
- con su código hexadecimal

Puedes encontrar una lista de colores en la siguiente dirección:  
[es.wikipedia.org/wiki/Colores\\_web](https://es.wikipedia.org/wiki/Colores_web)

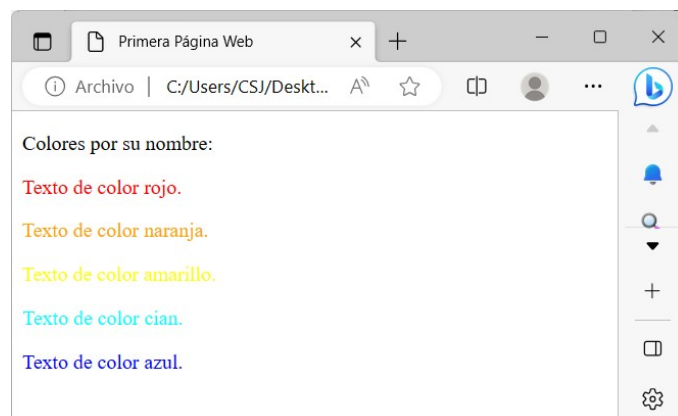
En HTML5, los colores se suelen representar a través de estilos en CSS; así, una forma de aplicar un color a un elemento es a través de su atributo style.

Veamos con algo más de detalle cada una de las opciones anteriores:

**1. Con el nombre del color con inglés:** es una manera sencilla de especificar un color, aunque este método limita la cantidad de colores representables.

### Ejemplo de visualización de colores

```
<p>Colores por su nombre:</p>  
<p style="color:red">Texto de color rojo.</p>  
<p style="color:orange">Texto de color naranja.</p>  
<p style="color:yellow">Texto de color amarillo.</p>  
<p style="color:cyan">Texto de color cian.</p>  
<p style="color:blue">Texto de color azul.</p>
```



**2. Con su código RGB:** la paleta de colores RGB proviene de la combinación de tres colores primarios: rojo, verde y azul. Por tanto, para especificar el código de un color en RGB lo haremos a través de tres números referentes a la cantidad de rojo, verde y azul, respectivamente. De esta forma podemos especificar cualquier combinación de colores.

### Ejemplo de colores definidos con su código RGB

Las siglas RGB significan  
Red, Green, Blue.

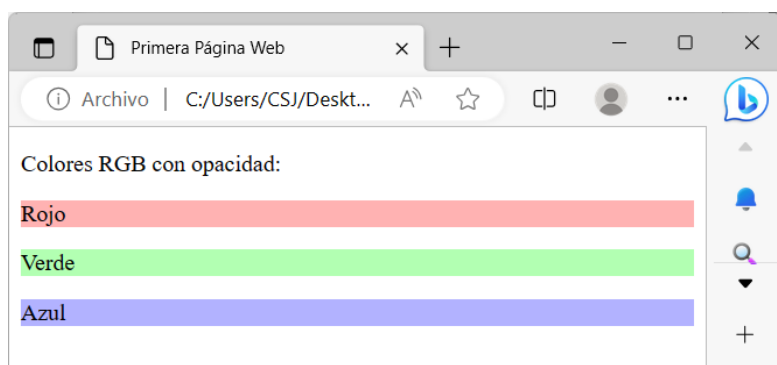
```
<p>Colores con su código RGB:</p>
<p style="color:rgb(255,0,0)">Texto de color rojo.</p>
<p style="color:rgb(255,165,0)">Texto de color naranja.</p>
<p style="color:rgb(255,255,0)">Texto de color amarillo.</p>
<p style="color:rgb(0,255,255)">Texto de color cian.</p>
<p style="color:rgb(0,0,255)">Texto de color azul.</p>
```

**3. Con su código RGBA:** es una extensión de los valores de color RGB añadiendo un **valor *alpha*** que especifica la opacidad del objeto.

Donde el parámetro alfa es un número entre **0.0 (completamente transparente)** y **1.0 (totalmente opaco)**.

### Ejemplo de colores RGBA

```
<p>Colores RGB con opacidad:</p>
<p style="background-color: rgba(255, 0, 0, 0.3);">Rojo</p>
<p style="background-color: rgba(0, 255, 0, 0.3);">Verde</p>
<p style="background-color: rgba(0, 0, 255, 0.3);">Azul</p>
```



**4. Con su código hexadecimal:** el código RGB también se puede especificar en hexadecimal, en lugar de especificar tres números decimales. Este código se expresa con una tira de caracteres que comienza con el carácter # seguido de seis caracteres hexadecimales, dos para cada color primario.

### Ejemplo de colores definidos en hexadecimal

```
<p> Colores con su código HEX:</p>
<p style="color:#FF0000">Texto de color rojo.</p>
<p style="color:#FFA500">Texto de color naranja.</p>
<p style="color:#FFFF00">Texto de color amarillo.</p>
<p style="color:#00FFFF">Texto de color cian.</p>
<p style="color:#0000FF">Texto de color azul.</p>
```

## Estilos

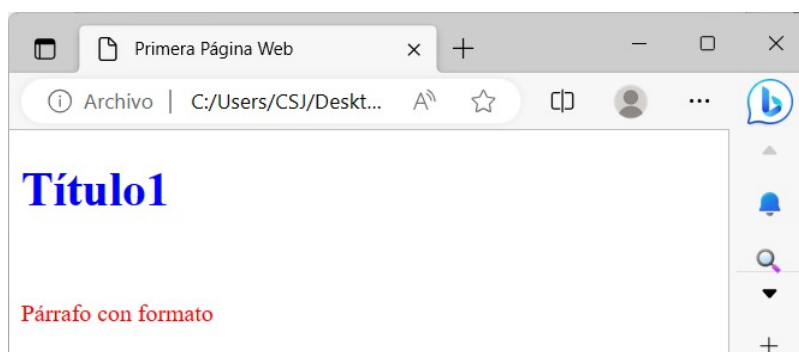
Para dar un formato de estilo a un elemento HTML se utilizará el **atributo style**. Para aplicar estilos a través de este atributo se realizará de la siguiente manera:

```
style="propiedad1:valor1; ... propiedadN:valorN;"
```

Donde *propiedad* hace referencia a una propiedad CSS y *valor* hace referencia a un valor CSS.

### Ejemplo de atributo "style"

```
<h1 style="fuente-family:Verdana; color:blue;">Título1</h1>
<p style="fuente-family:arial; color:red; fuente-size:200%;">
Párrafo con formato
</p>
```



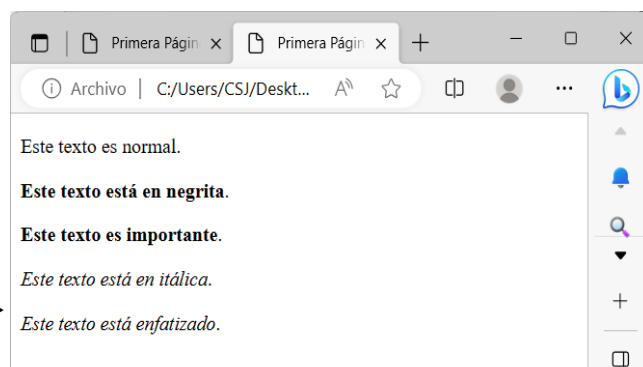
## Formato del texto

Aparte de dar formato al texto a través del atributo style, el lenguaje HTML también proporciona otros elementos para hacerlo, como los que se pueden ver en la siguiente tabla:

Elemento	Descripción
<b>	Define texto en negrita
<em>	Define texto enfatizado
<i>	Define texto en itálico, cursiva
<small>	Define texto más pequeño
<strong>	Define texto importante
<sub>	Define texto que debe aparecer en el subíndice
<sup>	Define texto que debe aparecer en el superíndice
<ins>	Define texto insertado, subrayado
<del>	Define texto tachado

### Formato negrita e itálica

```
<p>Este texto es normal.</p>
<p><b>Este texto está en negrita</b>.</p>
<p><strong>Este texto es importante</strong>.</p>
<p><i>Este texto está en itálica</i>.</p>
<p><em>Este texto está enfatizado</em>.</p>
```



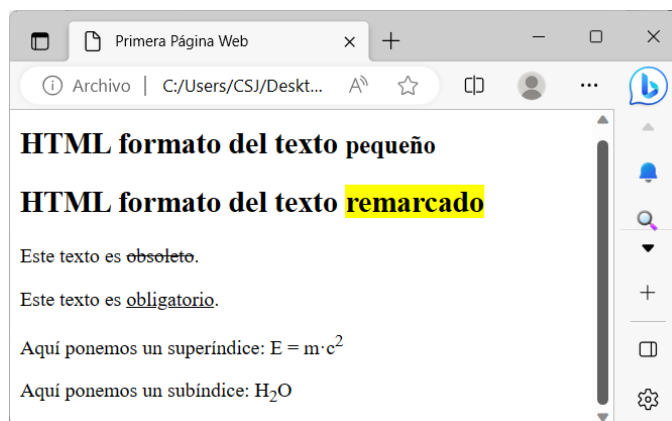


En el ejemplo anterior la visualización del texto a través de los elementos `<b>` y `<strong>` es prácticamente igual, pero si se utiliza el elemento `<strong>` se añade significado semántico al texto y se está indicando que éste es importante.

Puede apreciar lo mismo entre los elementos `<i>` y `<em>`. De esta forma, si se utiliza el elemento `<em>` también se está añadiendo significado semántico al texto.

## Otros formatos

```
<h2>HTML formato del texto
<small>pequeño</small></h2>
<h2>HTML formato del texto
<mark>remarcado</mark></h2>
<p>Este texto es <del>obsoleto</del>.</p>
<p>Este texto es <ins>obligatorio</ins>.</p>
<p> Aquí ponemos un superíndice: E =
m·c<sup>2</sup></p>
<p> Aquí ponemos un subíndice: H<sub>2</sub>O</p>
```



## Imágenes

En HTML, las imágenes se definen a través del elemento `<img>`.

### Ejemplo de imágenes

```

```

**src:** Donde especificamos en el atributo la URL o la ruta de la imagen.

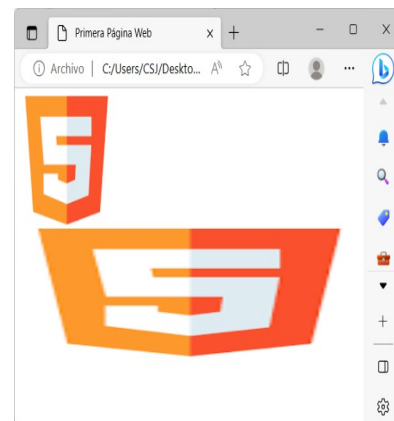
En el **atributo alt** especificamos un texto alternativo por si la imagen no puede ser mostrada. En el **atributo style** especificamos el estilo de la imagen; por ejemplo, las dimensiones de la imagen a mostrar.

El tamaño de la imagen también puede especificarse a través de los **atributos width y height**. Sin embargo, es **recomendable hacerlo a través del atributo style**, para prevenir posibles hojas de estilos que puedan cambiar el tamaño de la imagen.

```
<head>
  <meta charset="utf-8"/>
  <title>Primera Página Web</title>
  <style>
    img {width:100%;}
  </style>
</head>
<body>

</body>
```



Vemos, por tanto, que **un estilo enCSS podría modificar la visualización de la imagen**.

A menudo, para que el diseño del **sitio web sea adaptativo**, es necesario mostrar una imagen u otra más pequeña en función del tamaño de la pantalla o de la resolución del dispositivo, así como en función del tamaño del navegador desde el que accedamos al sitio web.

El navegador Internet Explorer no soporta los elementos `<picture>` ni los atributos `srcset` ni `sizes`.

Por este motivo, HTML5 incluye el nuevo elemento **`<picture>`** y nuevos **atributos, `srcset` y `sizes`, para el elemento `<img>`**.

Sin embargo, a la hora de visualizar una imagen hay que tener claros los conceptos de resolución de pantalla y de la imagen:

En la unidad “Elementos multimedia en la web: creación e integración” trabajará como incluir imágenes en diseños adaptativos.

La **resolución de pantalla** se expresa en píxeles por pulgada (PPP) y relaciona el número total de píxeles de un monitor con su tamaño en pulgadas (inches), en la dirección horizontal y la vertical. En inglés, Pixels Per Inch, PPI.

La **resolución de una imagen** indica la cantidad de detalle que puede observarse en la misma. Se expresa con dos números enteros, referentes a la cantidad de columnas de píxeles (número de píxeles de ancho) ya la cantidad de filas de píxeles (número de píxeles de alto).

Para una correcta visualización y para reducir la cantidad de información descargada es importante intentar adaptar al máximo los píxeles en los que se visualizará la imagen con los píxeles que realmente tiene la imagen.

Evitaremos mostrar una **imagen grande**, es decir, con mucha resolución, en un espacio pequeño de la pantalla.

## Enlaces

Los enlaces son los elementos que permiten al usuario navegar de una página web a otra.

El elemento que permite definir un enlace en HTML es **`<a>`**.

Los atributos que se pueden especificar para este elemento aparecen en la siguiente tabla.

Atributo	Descripción
<b>href</b>	Permite definir el destino del vínculo.
<b>target</b>	Permite definir dónde abrir el documento enlazado.
<b>id</b>	Permite definir una marca dentro de la página.
<b>style</b>	Permite definir el estilo del <i>link</i> .

## Ejemplo de enlaces

```
<a href="http://www.google.es" target="_blank">Enlace a Google</a>
<br />
<a href="pagina.html" style="color: darkblue; text-decoration:none"> Enlace a una página de
nuestro sitio web </a>
<br />
<a href="https://es.wikipedia.org/wiki/C%C3%Alceres"></a>
```



**Consideraciones** sobre los ejemplos anteriores:

- El atributo **href** es **obligatorio**.
- Si el destino del enlace es una **página externa** a su sitio web, es necesario introducir la **URL entera**.
- Si el destino es una **página de nuestro sitio web** sólo tiene que poner la **ruta de esta página**, y puede obviar su dominio y el protocolo.
- Si desea que el enlace **permita la descarga de un archivo o recurso**, en el destino del enlace habrá la ruta de este archivo descargable. Una imagen puede ser un enlace.
- El atributo **target** puede tomar por valor el nombre del elemento al que desea que se abra el enlace. El valor predeterminado **\_blank** significa que se abre en una pestaña nueva.
- Los **enlaces, por defecto, son de color azul y aparecen subrayados**, pero puede modificar su estilo mediante el atributo **style**.

## Listas

Las listas son una de las construcciones más usadas a la hora de elaborar textos, no sólo en HTML. No obstante, en HTML están también presentes y pueden ser de 3 tipos:

- **Listas Numeradas:** que son aquellas que expresan un orden entre los diferentes elementos de la lista. Este orden podrá ser numérico, alfabético etc..
- **Listas no numeradas:** que simplemente muestra los elementos de la lista uno tras otro.
- **Listas de definición:** que muestran diversos términos junto con su definición.

**Ejemplo de listas numeradas:**

```
<!-- Listas numeradas -->
<h3>Lista Numerada - Ordered List (ol)</h3>
<ol>
  <li>Primer elemento</li>
  <li>Segundo elemento</li>
  <li>Tercer Elemento</li>
  <li>Cuarto elemento</li>
</ol>
```

**ol** sería la etiqueta padre y cada uno de los elemento de la lista iría en una etiqueta **li**.

El atributo **start** nos permite indicar el número por el cual queremos que empiece la lista, ya que por defecto las listas ordenadas empiezan por el número 1.

```
<ol start="numero"> ... </ol>
```

De igual manera podemos indicar el tipo de lista ordenada. Entre los tipos de listas que podemos representar tenemos:

- 1 - Listas decimales
- a - Listas alfabéticas en minúsculas
- A - Listas alfabéticas en mayúsculas
- i - Listas de números romanos en minúsculas
- I - Listas de números romanos en mayúsculas

Los valores indicados al principio son los que le podemos asignar al atributo **type**

```
<ol type="tipo"> ... </ol>
```

El atributo **reversed** para las listas ordenadas. El atributo **reversed** nos permite invertir el orden de la lista. Es decir, realiza la numeración de la lista de forma inversa. Para utilizarlo simplemente indicamos el atributo **reversed** sobre el elemento **OL**.

```
<ol reversed> ... </ol>
```

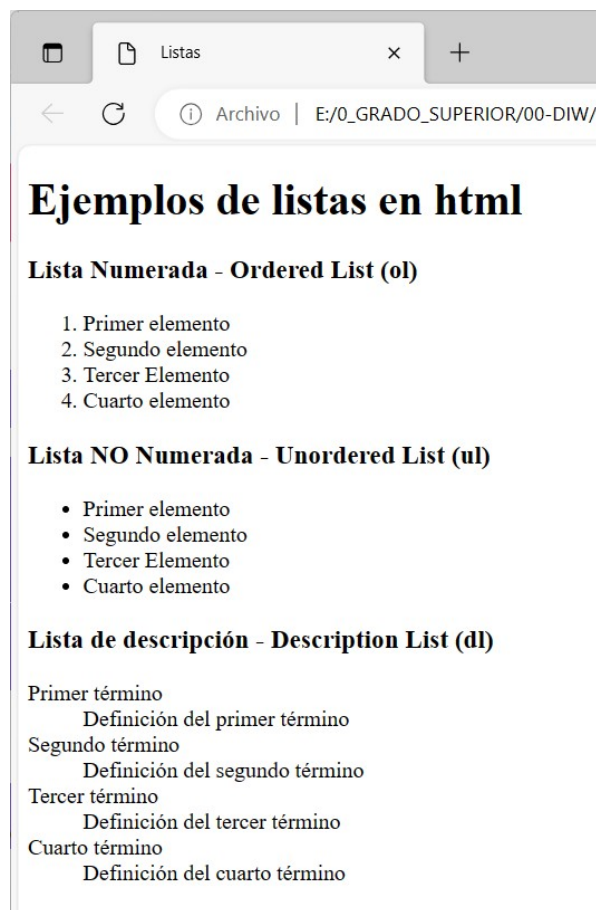
### Ejemplo de listas no numeradas:

```
<!-- Listas no numeradas-->
<h3>Lista NO Numerada - Unordered List (ul)</h3>
<ul>
  <li>Primer elemento</li>
  <li>Segundo elemento</li>
  <li>Tercer Elemento</li>
  <li>Cuarto elemento</li>
</ul>
```

**ul** sería la etiqueta padre y cada uno de los elemento de la lista iría en una etiqueta **li**.

En el caso de las listas desordenadas, no podemos indicarle el tipo de lista mediante **html**. En este caso tenemos que recurrir a **CSS** para poder indicar el tipo de lista utilizando el atributo **list-style-type** con **circle**, **square**, **upper-roman** y **lower-alfa**... Consultar este [enlace](#).

```
<style type="text/css">
ul {
  list-style-type: circle;
}
</style>
```



## Ejemplo de listas de definición:

```
<!-- Listas de definición-->
<h3> Lista de descripción - Description List (dl)</h3>
<dl>
  <dt>Primer término</dt>
  <dd>Definición del primer término</dd>
  <dt>Segundo término</dt>
  <dd>Definición del segundo término</dd>
  <dt>Tercer término</dt>
  <dd>Definición del tercer término</dd>
  <dt>Cuarto término</dt>
  <dd>Definición del cuarto término</dd>
</dl>
```

dl sería la etiqueta padre y cada término se define mostrando consecutivamente las etiquetas **dt**, que se corresponde con el término que vamos a definir, y **dd** que es la definición del término anterior.

## Tablas

Si se desea crear una tabla en HTML5 se utilizarán los elementos siguientes:

Elemento	Descripción
<b>&lt;table&gt;</b>	Permite definir una tabla.
<b>&lt;tr&gt;</b>	Permite definir una fila de la tabla.
<b>&lt;td&gt;</b>	Permite definir una columna de una fila.
<b>&lt;th&gt;</b>	Permite definir las cabeceras de la tabla.
<b>&lt;caption&gt;</b>	Permite definir el título de una tabla.

No siempre es necesario utilizar todos los elementos para definir una tabla, pero verá cómo los elementos **<table>**, **<tr>** y **<td>** son indispensables para visualizarla correctamente.

## Ejemplo de tabla:

```
<table id="tabla1" border="1" style="width:100%;">
  <tr>
    <td>Concepto</td>
    <td>Precio</td>
    <td>Descripción</td>
  </tr>
  <tr>
    <td>Objeto1</td>
    <td>10€</td>
    <td>Descripción del objeto 1</td>
  </tr>
  <tr>
    <td>Objeto2</td>
    <td>20€</td>
    <td>Descripción del objeto 2</td>
  </tr>
</table>
```

Concepto	Precio	Descripción
Objeto1	10€	Descripción del objeto 1
Objeto2	20€	Descripción del objeto 2

Aunque en el ejemplo anterior hemos utilizado el atributo **border** para especificar que se vea el borde de la tabla, todos los aspectos del formato de visualización de una tabla suelen especificarse en lenguaje **CSS** a través, por ejemplo, de su atributo **style**.

Para **combinar celdas** de una tabla se utilizan los atributos **colspan** y **rowspan** y elementos `<td>` o `<th>`.

### Ejemplo de tabla con filas combinadas (rowspan)

```
<table id="tabla2" style="border:1px solid black">
  <caption> Notas Finales</caption>
  <tr>
    <th>Alumno</th>
    <th>Módulo</th>
    <th>Nota</th>
  </tr>
  <tr>
    <td rowspan="3" style="vertical-align: text-top;">Nombre Alumno1</td>
    <td>M1 Sistemas Operativos</td>
    <td>9</td>
  </tr>
  <tr>
    <td>M2 Bases de Datos</td>
    <td>8</td>
  </tr>
  <tr>
    <td>M3 Programación</td>
    <td>7</td>
  </tr>
  <tr>
    <td rowspan="3" style="vertical-align: text-top;">Nombre Alumno2</td>
    <td>M1 Sistemas Operativos</td>
    <td>7</td>
  </tr>
  <tr>
    <td>M2 Bases de Datos</td>
    <td>6</td>
  </tr>
  <tr>
    <td>M3 Programación</td>
    <td>5</td>
  </tr>
</table>
```

Alumno	Módulo	Nota
Nombre Alumno1	M1 Sistemas Operativos	9
	M2 Bases de Datos	8
	M3 Programación	7
Nombre Alumno2	M1 Sistemas Operativos	7
	M2 Bases de Datos	6
	M3 Programación	5

Como ve en el ejemplo anterior, **en las cabeceras, en la fila 1 y en la fila 5 se definen tres columnas**. En cambio, **en las filas 2, 3, 6 y 7 sólo se definen dos columnas**, puesto que la primera columna está ya ocupada.

### Ejemplo de tabla con celdas combinadas (colspan)

```
<table id="tabla3" border="1">
  <caption> Notas Finales</caption>
  <tr>
    <td rowspan="2" style="vertical-align: bottom;">Alumno</td>
    <td colspan="3">Módulo 1 Sistemas</td>
  </tr>
  <tr>
    <td>EAC</td>
    <td>PAF</td>
    <td>Final</td>
  </tr>
  <tr>
    <td>Nombre Alumno1</td>
    <td>8</td>
    <td>8</td>
    <td>8</td>
  </tr>
  <tr>
    <td>Nombre Alumno2</td>
    <td>7</td>
    <td>7</td>
    <td>7</td>
  </tr>
  <tr>
    <td>Nombre Alumno3</td>
    <td>6</td>
    <td>6</td>
    <td>6</td>
  </tr>
```

Alumno	Módulo 1 Sistemas		
	EAC	PAF	Final
Nombre Alumno1	8	8	8
Nombre Alumno2	7	7	7
Nombre Alumno3	6	6	6

```
<td>8</td>
<td>8</td>
<td>8</td>
</tr>

<tr>
<td>Nombre Alumno2</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>

<tr>
<td>Nombre Alumno3</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>

</table>
```

Como se ve en el ejemplo anterior, la celda “alumno” ocupa dos filas y la celda “Módulo1 Sistemas” ocupa tres columnas.

## Logotipo o icono

Si desea ver el logotipo del sitio web o el icono en la barra de direcciones o en la pestaña del navegador deberá crear el icono .ico , *renombrarlo* con el nombre **favicon.ico** y **ponerlo en la raíz de nuestro sitio web**.

El motivo de que el archivo se llame favicon.ico y se ponga en la raíz del sitio es porque, por defecto, los navegadores como Firefox, Safari u Opera buscan en esta ubicación.

Para asegurarnos de que el navegador encuentra nuestro *favicon* es recomendable insertar la siguiente línea de código en la cabecera de nuestro documento HTML (entre las etiquetas <head> y </head>), indicando la ruta del *favicon*, que en este caso es la raíz del sitio:

```
<link rel="Shortcut Icon" href="favicon.ico" type="image/x-icon" />
```

**Para crear nuestro icono, primero es necesario realizar una imagen cuadrada, por ejemplo 32×32 píxeles, en formato JPG, GIF o PNG.**

Si se desea que el icono tenga un **fondo transparente**, entonces esa imagen inicial debe ser en formato **PNG y con el fondo transparente**.

Para convertir esta imagen a .ico se puede hacer a través de la web [www.favicon.cc](http://www.favicon.cc)

## Formularios con HTML5

Los formularios se utilizan para que el usuario pueda introducir datos para después ser enviados y procesados por el servidor.

El elemento HTML que permite crear formularios es `<form>`, y en su contenido se definen los diferentes campos que forman parte del mismo.

### Ejemplo de formulario

```
<form id="ValidaUsuario" action="mailto:nombre@correo.com" method="post" autocomplete="on">
```

```
  Nombre usuario: <br />
  <input type="texto" name="nombreUsuario" /><br />
  Contraseña: <br />
  <input type="password" name="contraseña" />
  <br />
  <br />
  <input type="reset" value="Borrar"/>
  <input type="submit" value="Aceptar"/>
```

```
</form>
```

Veamos, en el ejemplo, los **atributos más importantes que tiene el elemento `<form>`**:

- **id**: indica el identificador del formulario.
- **action**: indica la URL o el nombre de la página que se encarga de procesar los datos del formulario.
- **method**: indica el método de envío del formulario, que puede ser get o post. Si es get, la información del formulario se enviará a través de la URL al que se envía el formulario. Si es post, se envía encapsulada en el paquete HTTP.
- **autocomplete**: indica si el navegador completará automáticamente los valores de los campos del formulario en base a valores que el usuario haya entrado con anterioridad. El valor de este atributo puede ser o no off.
- **novalidate**: si este atributo está presente se desactiva la validación nativa que HTML5 ofrece en los formularios.
- **name**: permite definir un nombre en el formulario.
- **enctype**: permite definir el tipo de datos que se enviarán.
- Los campos de un formulario se pueden definir a través de distintos elementos HTML5:
  - el elemento `<input>`
  - el elemento `<textarea>`
  - el elemento `<select>`
  - el elemento `<button>`
  - el elemento `<datalist>`

### El elemento `<input>`

Mediante el **elemento `<input>`**, y a través de su atributo **type**, podemos insertar diferentes tipos de campo.

Veamos los diferentes tipos en la siguiente tabla

Valor	Descripción
-------	-------------



<b>text</b>	Permite insertar un texto.
<b>password</b>	Permite insertar una contraseña, escondiendo los caracteres.
<b>hidden</b>	Permite insertar un campo oculto para el usuario.
<b>submit</b>	Permite insertar un botón para enviar los datos al servidor.
<b>image</b>	Permite insertar una imagen como botón para enviar los datos.
<b>button</b>	Permite insertar un botón.
<b>radio</b>	Permite insertar un botón <i>radio</i> .
<b>checkbox</b>	Permite insertar un <i>checkbox</i> .
<b>color</b>	Permite seleccionar un color en el sistema de selección de colores del sistema operativo.
<b>date</b>	Permite insertar una fecha. Los navegadores propondrán una interfaz de selección de la fecha.
<b>datetime</b>	Permite insertar una fecha y hora con la diferencia horaria.
<b>datetime-local</b>	Permite insertar una fecha y hora sin la diferencia horaria.
<b>time</b>	Permite insertar sólo la hora.
<b>month</b>	Permite insertar el mes de una fecha.
<b>week</b>	Permite insertar la semana.
<b>email</b>	Permite insertar un correo electrónico. Los dispositivos móviles adaptarán el teclado.
<b>number</b>	Permite insertar un número. Los dispositivos móviles adaptarán el teclado.
<b>range</b>	Mostrará una barra de selección con un cursor con el que podremos elegir un valor. No soportado en IE.
<b>search</b>	Permite insertar un campo de introducción de datos específico para las búsquedas.
<b>tel</b>	Permite insertar un número de teléfono. Los dispositivos móviles adaptarán el teclado.
<b>url</b>	Permite insertar una URL. Los dispositivos móviles adaptarán el teclado.

Veamos que **a través del elemento input se pueden crear distintos tipos de campo**. Y, de hecho, existen varios campos, como por ejemplo ***email***, ***number***, ***url***, ***etc.***, que aparentemente son iguales, pero su utilización ayudará a la validación correcta de los datos introducidos en el formulario y que en dispositivos móviles su teclado se adapte para una mejor utilización.

Aparte del atributo type, también podemos definir **otros atributos para el elemento <input>**. Veamos una recopilación en la siguiente tabla:

Atributo	Descripción
<b>id</b>	Permite definir un identificador.
<b>name</b>	Permite definir un nombre para identificar el campo.
<b>value</b>	Permite definir qué valor debe tomar el campo del formulario.
<b>readonly</b>	Permite definir que este campo es sólo de lectura.
<b>disabled</b>	Permite desactivar el campo.
<b>selected</b>	Permite definir que este campo esté seleccionado.
<b>checked</b>	Permite definir que un botón de <i>radio</i> o <i>checkbox</i> esté seleccionado.
<b>size</b>	Permite definir el tamaño del campo.
<b>maxlength</b>	Permite definir el tamaño máximo del valor del campo.
<b>autocomplete</b>	Permite activar la función autocompletar.
<b>autofocus</b>	Permite definir que estos campos son los que tienen el foco por defecto.
<b>minymax</b>	Permite definir un valor mínimo y máximo.
<b>placeholder</b>	Permite definir una descripción o ejemplo para el valor del campo.
<b>required</b>	Permite definir que este campo es obligatorio y se utiliza para la validación nativa de HTML5 del formulario.

No todos los atributos son aplicables a todos los tipos de campo; por ejemplo, no tiene sentido aplicar el atributo placeholder para un campo de fecha o aplicar los atributos min y max un campo de contraseña.

## Ejemplo de un formulario con distintos campos de tipo <input>

```
<h3>Datos del usuario</h3>
<hr />
<form method="post" action="mailto:usuario@correo.com" autocomplete >
  <input type="text" name="nombre" size="50" maxlength="10" placeholder="Escribe el nombre"
  required autofocus/><br />
  <input type="text" name="apellido" size="50" maxlength="20" placeholder="Escribe el apellido"
  required /><br />
  <input type="email" name="email" size="50" placeholder="Escribe el correo electrónico" /><br />
  <input type="password" name="contraseña" size="50" placeholder="Escribe la contraseña"
  disabled/><br />
  Fecha de nacimiento: <br />
  <input type="date" name="dataNace"/>
  <hr />
  Género: <br />
  <input type="radio" name="gender" value="masculio" checked />Hombre <br />
  <input type="radio" name="gender" value="femenio" />Mujer <br />
  <hr />
  Aficiones: <br />
  <input type="checkbox" name="aficio1" value="deporte" />Deporte <br />
  <input type="checkbox" name="aficio2" value="cine" />Cine <br />
  <input type="checkbox" name="aficio3" value="leer" />Lectura <br />
  <input type="checkbox" name="aficio4" value="otros" />Otros <br />
  Selecciona tu color favorito:
  <input type="color" name="colorPreferido" /> <br />
  <hr />
  Foto:
  <input type="file" name="foto" />
  <hr />
  <input type="button" value="Haga clic!" onclick="alert('HOLA!!!');" />
  <input type="reset" value="Borrar" />
  <input type="submit" value="Enviar" />
  <hr />
</form>
```

### Algunas consideraciones sobre el ejemplo serían:

- Para que los **botones de radio** se discriminen entre sí **deben tener el mismo nombre**, al contrario que los **botones checkbox**, que cada uno debe tener un **nombre distinto**.
- **Cada campo tiene un nombre**, que se utilizará para que el servidor que recibe los datos pueda acceder a cada uno de los datos recibidos. También se puede especificar un identificador a través del **atributo id**, y el servidor también podría acceder a él a través del atributo. Pero fijémonos en que los botones de **radio** necesitan especificar el **atributo name** para su funcionamiento.
- Hay **diferencia de los atributos maxlength y size**: el primero hace referencia al tamaño del valor que se le dará al campo y el segundo, al tamaño en el que se visualiza el campo de texto.

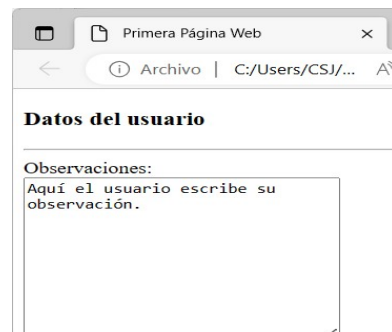
- También en algunos campos se han utilizado los **atributos disabled, placeholder y required**.

### El elemento <textarea>

Este elemento permite definir un campo de texto de varias líneas.

#### Ejemplo de campo de texto de varias líneas

```
Observaciones: <br />
<textarea id="observaciones" rows="10" cols="30">
Aquí el usuario escribe su observación.
</textarea>
```

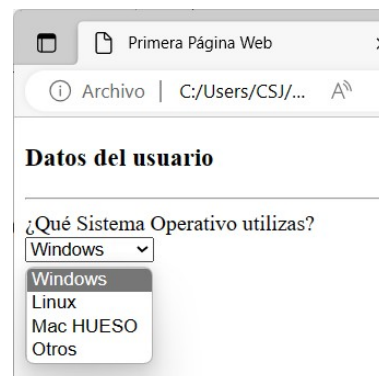


### El elemento <select>

Permite definir una **lista de opciones seleccionables**. Cada opción se especifica con el **elemento <option>**.

#### Ejemplo de utilización del elemento <select>

```
¿Qué Sistema Operativo utilizas? <br />
<select id="SO" name="SO">
  <option value="windows">Windows</option>
  <option value="linux">Linux</option>
  <option value="mac">Mac HUESO</option>
  <option value="otros">Otros</option>
</select>
```



### El elemento <button>

Con este elemento se puede **insertar un botón en el formulario**, y es muy similar a `<input type="button">`. La diferencia es que a través de éste se puede modificar no sólo el texto del botón, sino que por ejemplo se podría especificar una imagen.

#### Ejemplo de utilización del elemento <button>

```
<button type="button" onclick="alert('¡Hola a todos!')"></button>
```

Fijémonos en que a través del atributo `onclick` especifiquemos la acción a realizar al pulsar el botón. Generalmente suele ser una función en JavaScript.

#### JavaScript

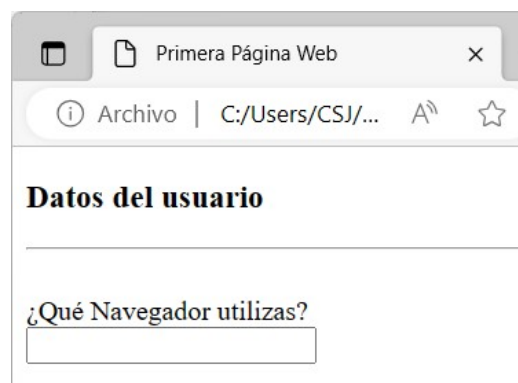
Es un lenguaje *script* que se ejecuta en la parte del cliente, es decir, en el navegador. Permite mejoras en la interfaz de usuario de aplicaciones y páginas web dinámicas.

### El elemento <datalist>

Con este elemento podemos **especificar una lista de opciones para sugerir al usuario algunas opciones predefinidas** para un elemento `<input>`. Los usuarios verán una lista desplegable con las opciones predefinidas a medida que vayan introduciendo el texto, pero pueden terminar escribiendo otro texto.

## Ejemplo de utilización del elemento &lt;datalist&gt;

```
<br>¿Qué Navegador utilizas? <br />
<input list="navegadores">
  <datalist id="navegadores">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
  </datalist>
```



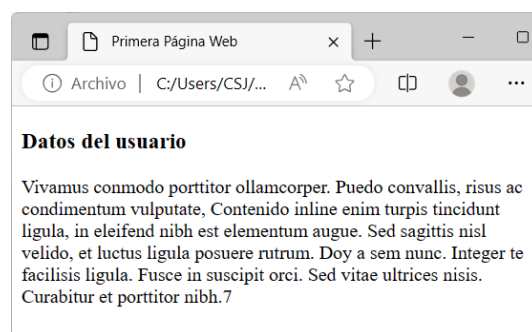
## Visualización de elementos con HTML5

Cada elemento HTML tiene un valor de vista predeterminado. En la mayoría de los elementos, este valor es *block* o *inline*. Un elemento con **visualización de block** siempre comienza a visualizarse en una nueva línea, y ocupa todo el ancho disponible. Un elemento con visualización *inline* no comienza en una nueva línea y sólo toma el ancho necesario. **Más información en:** <https://desarrolladoresweb.org/html/elementos-inline-y-block-en-html/>

La visualización por defecto de los elementos HTML tanto de tipo *block* como de tipo *inline* se puede modificar mediante propiedades CSS para conseguir un diseño adaptativo o, en inglés, *Responsive Web Design*.

## Ejemplo de elemento "block" y "inline"

```
<div>
<p>
Vivamus conmodo porttitor ollamcorper. Puedo convallis,
risus ac condimentum vulputate,
<span>Contenido inline</span>
enim turpis tincidunt ligula, in eleifend nibh est
elementum augue. Sed sagittis nisl velido,
et luctus ligula posuere rutrum. Doy a sem nunc. Integer te
facilisis ligula. Fusce in suscipit orci.
Sed vitae ultrices nisis. Curabitur et porttitor nibh.7
</p>
</div>
```



La etiqueta <div> contiene el contenido tipo block, la etiqueta <span> aloja el contenido inline.

## Estructura semántica de la página web

En contraposición a la forma de estructurar el contenido de las páginas web que utilizaba HTML4 a través de casi únicamente elemento <div>, con el nuevo estándar HTML5 llegan nuevos elementos que aportan estructura **semántica**.

Destacamos los siguientes elementos que aportan semántica a la estructura de la página (**page layout**):

Puede estructurarse una página web con tablas o marcos, pero actualmente estas técnicas se consideran obsoletas y han caído en desuso.

- Elemento **<header>**: permite insertar una zona de visualización para las cabeceras.
- Elemento **<footer>**: permite insertar una zona de visualización para los pies de página.
- Elemento **<nav>**: permite insertar una zona de visualización para la navegación con vínculos hipertexto.
- Elemento **<section>**: permite agrupar elementos que tengan la misma temática.
- Elemento **<article>**: permite insertar un contenido autónomo.

- Elemento **<aside>**: permite mostrar un contenido relacionado con el contenido al que esté vinculado.
- Elemento **<main>**: permite representar el contenido principal de la página.

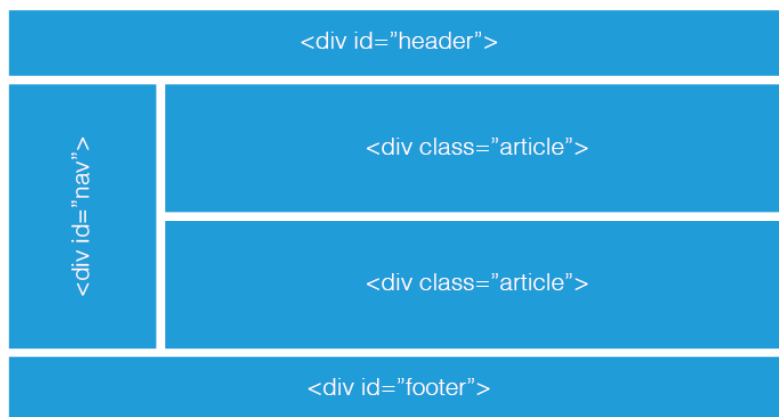
Cuando usamos alguna de estas etiquetas sucede lo siguiente:

- A nivel visual **se comportan como una etiqueta <div>**.
- **Aportan significado**, podemos conocer el qué es lo que hay dentro (no como un div). Esto ayuda a buscadores y desarrolladores tal y como se dijo antes.
- **Se utilizan para maquetar**, como los divs.
- Tienen **reglas de anidación** y éstas son todas distintas. Por ejemplo: *Sólo puede tener un main y no puede ser hijo de article, aside, footer, header o nav*

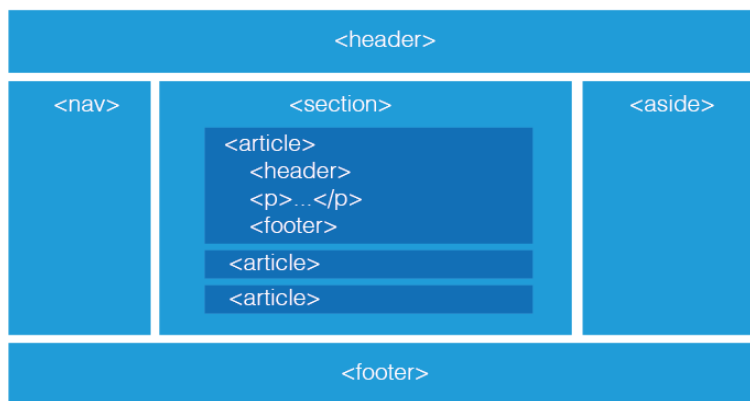
Cabe mencionar que **en HTML5 sigue siendo de mucha utilidad el uso de las cajas <div>**.

Puede apreciar en las figuras siguientes la estructura del contenido de una página web en HTML4 y HTML5, respectivamente.

**Estructura de los contenidos en HTML4**



**Estructura de los contenidos en HTML5**



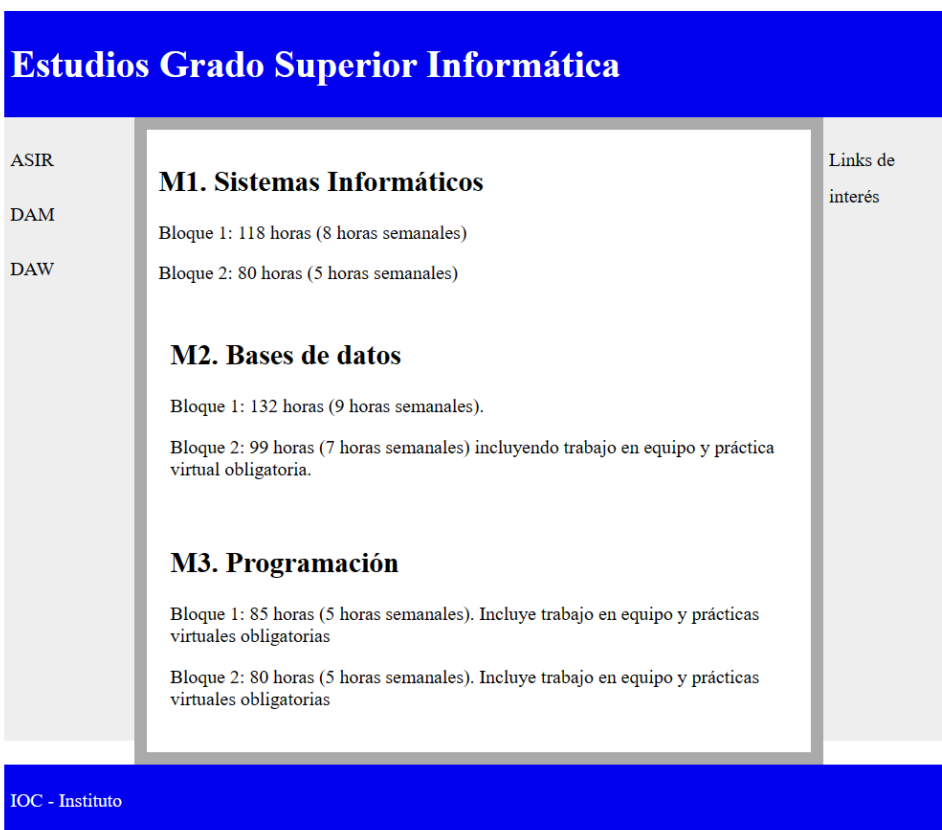
Veamos un ejemplo de página web utilizando estas etiquetas con sentido semántico.

## Estructura del contenido de una página web en HTML5

```
<!DOCTYPE html>
<html lang="es">
<head><meta charset="utf-8"/><title>Primera Página Web</title>
<style>
#capal{
    width:800px;
    margin:0 auto;
}
header {
    background-color:#0000ee;
    color:white;
    texto-align:center;
    padding:5px;
}
nav {
    line-height:30px;
    background-color:#eeeeee;
    height:515px;
    width:100px;
    float:left;
    padding:5px;
}
section {
    background-color:#aaaaaa;
    float:left;
    padding:10px;
    width:560px;
}
article {
    background-color:#ffffff;
    padding:10px;
}
aside {
    line-height:30px;
    background-color:#eeeeee;
    height:515px;
    width:100px;
    float:right;
    padding:5px;
}
footer {
    background-color:#0000ee;
    color:white;
    clear:both;
    texto-align:center;
    padding:5px;
}
</style>

</head>
<body>
<div id="capal">
    <header>
        <h1>Estudios Grado Superior Informática</h1>
    </header>

    <nav>
        <p>ASIR </p>
        <p>DAM </p>
        <p>DAW </p>
```



```
</nav>

<section>
<article>
  <h2>M1. Sistemas Informáticos</h2>
  <p> Bloque 1: 118 horas (8 horas semanales) </p>
  <p> Bloque 2: 80 horas (5 horas semanales) </p>
</article>

  <article>
    <h2>M2. Bases de datos </h2>
    <p> Bloque 1: 132 horas (9 horas semanales). </p>
    <p> Bloque 2: 99 horas (7 horas semanales) incluyendo trabajo en equipo y práctica
virtual obligatoria. </p>
  </article>

  <article>
    <h2>M3. Programación </h2>
    <p> Bloque 1: 85 horas (5 horas semanales). Incluye trabajo en equipo y prácticas
virtuales obligatorias</p>

    <p> Bloque 2: 80 horas (5 horas semanales). Incluye trabajo en equipo y prácticas
virtuales obligatorias</p>
  </article>

</section>

<aside>
  <p>Links de interés</p>
</aside>

<footer>
  <p>IOC - Instituto</p>
</footer>

</div>
</body>
</html>
```

### Visualización de una página web con estructura semántica en un navegador:

Para la visualización de cada elemento se utiliza un estilo en lenguaje CSS. Así, vemos que el tamaño, la posición y el color de cada uno de estos elementos está definido a través de un estilo CSS en el head de esta página, y por tanto, en el *body* sólo encontramos el contenido.

### Validación HTML5

Resulta de gran utilidad poder verificar los errores que pueden existir en las páginas.

La herramienta online accesible desde [validator.w3.org](http://validator.w3.org) permite validar el código HTML de diferentes maneras:

- Indicando la URL completa de la página que deseamos validar.
- Subiendo el documento HTML.
- Escribiendo el código HTML directamente.
- Una vez indicado el código HTML a validar, pulse la opción *More Options*, y allí indique el *Document*

Type, en su caso HTML5.

Por último, haga clic en el botón *Check* para que se ejecute la validación, tal y como se muestra en la siguiente figura:

W3C<sup>®</sup> Markup Validation Service  
Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI   Validate by File Upload   **Validate by Direct Input**

Validate by direct input

Enter the Markup to validate:

```
</p>>
<section>
<article>
  <h2>1. Sistemas Informáticos</h2>
  <p> Bloc 1: 115 horas (5 horas semanales) </p>
  <p> Bloc 2: 80 horas (5 horas semanales) </p>
</article>
<article>
  <h2>2. Bases de datos </h2>
  <p> Bloc 1: 102 horas (5 horas semanales) </p>
</article>
</p>>
```

More Options

☒ **Validate Full Document**  
Use Doctype:  ☐ Only if Doctype is missing

☐ **Validate HTML fragment**  
Use Doctype: ☒ HTML 4.01 ☐ XHTML 1.0

☒ List Messages Sequentially ☐ Group Error Messages by Type

☐ Show Source ☐ Clean up Markup with HTML-Tidy

☐ Show Outline ☐ Validate error pages ☐ Verbose Output

Check

En unos instantes tendrás el listado y los detalles de errores y alertas, si es que el validador detecta.

## Seguridad HTML5

Actualmente existen muchas aplicaciones web disponibles en HTML5, pero este lenguaje tiene potenciales vulnerabilidades de seguridad. Algunas de ellas proceden de los elementos del estándar en sí, y otras provienen de las implementaciones de cada navegador.

### \_\_ Más información:

[Manual HTML \(Consultar para ampliar conocimientos\)](#)