

2 — Estructuración Web: HTML5

2.1 Concepto y primer contacto con HTML5

HTML (*Hyper Text Markup Language*, lenguaje de marcado de hipertexto), es un lenguaje para la creación de páginas web. El lenguaje HTML no es un lenguaje de programación sino un lenguaje de marcas y este lenguaje permite describir documentos Web (página Web). Estos documentos se componen de la información o contenido (textos, imágenes, etc.) que se va a presentar, y una serie de marcas o etiquetas (*tags*). Éstas últimas contribuyen a la estructura del documento junto a sus propios atributos y establecen cómo se va a presentar esa información.

HTML5 es el nuevo estándar para el lenguaje de marcas HTML. HTML5 se encuentra aún en una fase de desarrollo/adaptación, sin embargo, la mayoría de los navegadores modernos reconocen la mayor parte de las nuevas etiquetas proporcionadas por HTML5. Los principales navegadores (Safari, Chrome, Firefox, Opera, Internet Explorer) continúan agregando nuevas características HTML5 a sus últimas versiones. Así que, HTML5 se puede considerar como una actualización de las etiquetas de HTML, eliminando etiquetas que son hoy en día ineficaces o confusas y agregando nuevas etiquetas que dan una mayor riqueza semántica a la estructuración de documentos Web. Son muchas las nuevas características que aporta HTML5 y se pueden enumerar en las siguientes:

- Se reduce la necesidad del uso de plugins (por ejemplo Flash), tratando de hacer el código HTML lo más independiente posible a cualquier compañía privada.
- Se mejora el manejo de errores (por ejemplo en formularios Web).
- Se aumenta el número de etiquetas (tags) para reemplazar secuencia de comandos por ejemplo en Javascript. Aparecen etiquetas específicas como:
 - **Canvas** para dibujar gráficos.
 - **Video y audio** para insertar y reproducir archivos multimedia.
 - Soporte de **almacenamiento local y offline**.
 - Nuevos elementos de **contenido** específicos, como article, footer, header, nav, section.
 - Nuevos controles de **formularios**, como calendar, date, time, email, url, search ,para facilitar la validación.
- También se permite aplicar mediante JavaScript una API de **Geolocalización**, para conocer las coordenadas (latitud y longitud) del usuario según su posición.

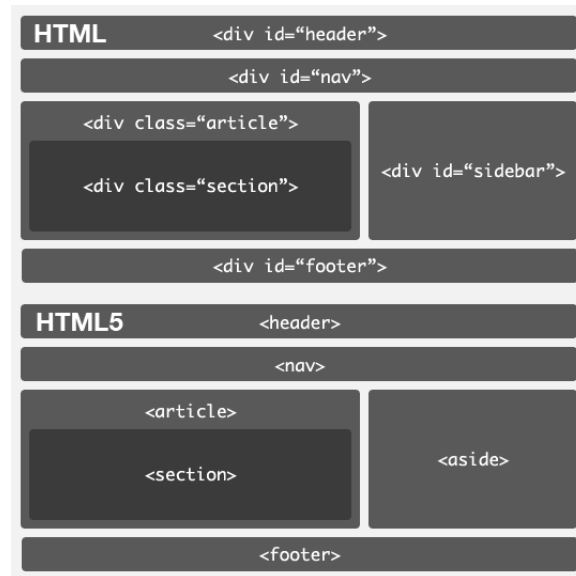


Figura 2.1: Comparación de la estructura Web de un documento HTML clásico frente a HTML5.

2.1.1 Estructura de un documento HTML5

Al igual que en otro lenguaje de marcas, por ejemplo XML, en un documento HTML se pueden encontrar las siguientes entidades:

- **Etiquetas (tags):** Se encierran entre los corchetes angulares `<...>` como inicio de etiqueta y `</...>` como cierre de etiqueta.
- **Elementos:** Los elementos definen la estructura del documento HTML e indican al navegador cómo presentar el contenido. Por lo general, los elementos están formados por una etiqueta de inicio, el contenido (información u otros elementos), y una etiqueta de cierre. Existen elementos que no poseen contenidos, denominados elementos vacíos. Este tipo de elemento no posee etiqueta de cierre aunque su etiqueta de inicio acaba con `/>`.
- **Atributos:** Poseen la estructura **nombreAtributo = valorAtributo** y se encuentran dentro de la etiqueta de inicio de un elemento. Indican las propiedades asociadas a cada elemento.

La estructura de un documento en HTML5 es ligeramente diferente a la estructura de las versiones anteriores de HTML. Se han incorporado nuevas etiquetas para delimitar con mayor semántica las diferentes partes del documento a las ya existentes *head* y *body*. En la Figura 2.1 se muestra una comparativa de las diferentes partes que componen un documento Web HTML5 frente a uno clásico. Las partes que componen la estructura de un documento HTML5 encajan perfectamente a las partes de la Web que se describieron en la Sección 1.1. A continuación se describen cada una de las partes:

- **<header>:** Define el encabezamiento del documento o de la sección (*section*), se suele incluir en el encabezado el título utilizando las etiquetas de cabeceras (`h1`, `h2`, ..., `h6`).
- **<nav>:** Define los elementos de navegación del sistema de navegación de la página Web, por ejemplo, el menú principal de navegación.
- **<section>:** Define una porción del documento con un contenido de temática similar, además se pueden anidar estas etiquetas para ir definiendo diferentes subsecciones temáticas.

- **<article>**: Define una parte de una sección, cada una de las secciones se puede subdividir en diferentes artículos.
- **<aside>**: Define el contenido que amplía al contenido principal, por ejemplo, una imagen o información extra relacionada con el contenido.
- **<footer>**: Define la información que está en el pie del documento, tal como el autor o copyright.

Una vez definidas las zonas en las que se distribuye un documento HTML5 se puede abarcar el desarrollo de la primera página en HTML5. A continuación se va a describir cada una de las partes del Código 2.1 en el cual se muestra una página HTML5 básica.

Código 2.1: Estructura de la página Web en HTML5.

```

1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="utf-8">
5     <title>Título de página</title>
6     <link rel="stylesheet" href="./T02_01/T02_01.css" type="text/
      css" media="screen" />
7   </head>
8   <body>
9     <header>
10      <h1>Título de página</h1>
11    </header>
12    <main>
13      <nav>
14        <ul>
15          <li><a href="#">Inicio</a></li>
16          <li><a href="#">Archivo</a></li>
17          <li><a href="#">Contacto</a></li>
18        </ul>
19      </nav>
20      <section id="intro">
21      </section>
22      <section>
23        <article></article>
24      </section>
25      <aside>
26        <!-- Barra Lateral -->
27      </aside>
28    </main>
29    <footer>
30      <!-- Pie de Página -->
31    </footer>
32  </body>
33 </html>

```

Línea 1: En primer lugar hay que especificar al navegador que el documento es un documento HTML puesto que los navegadores hoy en día pueden leer documentos de varios formatos. Esto se consigue con la etiqueta `<!DOCTYPE>`.

```
1 <!DOCTYPE html>
```

Líneas 2, 33: Una vez declarado el tipo de documento, se debe comenzar a estructurar el documento HTML. La estructura jerárquica tiene como nodo raíz el elemento `<html>`, es decir, todo el código HTML tiene que estar comprendida entre dichas etiquetas. Además,

en HTML5 se debe especificar el atributo **lang**, el cual define el idioma del contenido del documento HTML. En nuestro caso particular se ha definido el idioma Español [41].

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

```
33 </html>
```

Líneas 3-7: En HTML5 se ha mantenido al igual que las versiones anteriores de HTML la cabecera del documento HTML. La cabecera del documento es diferente a la cabecera del diseño Web. En esta cabecera se describe información propia del documento HTML mientras que en la cabecera del diseño está reservado para diseñar la cabecera del sitio Web. Entre las etiquetas <head> se define el título de la página Web, el juego de caracteres y se enlazan los archivos externos como pueden ser los estilos (CSS) o códigos JavaScript.

Línea 4: A partir de HTML5 se puede describir el juego de caracteres dentro de la etiqueta <meta> utilizando el atributo *charset*, en este caso se ha definido el juego de caracteres UTF-8. Dentro de la etiqueta meta se pueden describir otros atributos como pueden ser la descripción o un conjunto de palabras clave. Estos atributos se estudiarán en capítulos posteriores.

Línea 5: La etiqueta <title> permite definir el título del documento. Este texto suele aparecer en la barra del navegador.

Línea 6: Un elemento muy importante es <link> el cual permite enlazar con documentos externos como pueden ser estilos, códigos JavaScript, o elementos gráficos. En el ejemplo mostrado se está haciendo referencia al documento llamado *T02_C1.css* que es una hoja de estilos como queda especificado en el atributo *rel*. Las hojas de estilos se estudiarán a fondo en el Capítulo 3.

```
3 <head>
4   <meta charset="utf-8">
5   <title>Título de página</title>
6   <link rel="stylesheet" href="./T02_01/T02_01.css" type="
      text/css" media="screen" />
7 </head>
```

Líneas 8, 32: El código que se encuentra entre las etiquetas <body> genera la parte visible del documento. Es decir, todo lo que aparezca entre estas etiquetas serán los elementos que formarán el contenido del sitio Web.

```
8 <body>
```

```
32 </body>
```

Líneas 9-11: El elemento header es añadido en la versión HTML5 y permite definir la cabecera del diseño de nuestro sitio Web. No debe confundirse con el elemento head como se describió anteriormente. El elemento header no es exclusivamente utilizado para describir la cabecera del sitio Web. También puede utilizarse anidado en otros elementos para definir la cabecera de dicho elemento. En el ejemplo mostrado se ha incluido un elemento *h1* en el elemento header, el cual es utilizado para especificar que un texto es una cabecera. Existen hasta 6 niveles diferentes: h1, h2, ..., h6. En la siguiente sección se describirá con mayor detalle su funcionamiento.

```
9 <header>
10 <h1>Título de página</h1>
11 </header>
```

Línea 12, 28 : El elemento `<main>` es utilizado en HTML5 para especificar el contenido principal de la página Web, es decir, es donde aparece el contenido más relevante de la página Web. Este elemento contendrá los elementos de `<section>` y `<article>`.

```
12      <main>
```

```
28      </main>
```

Líneas 13-19: En HTML5 se define el elemento `<nav>` para describir el sistema de navegación del sitio Web. En el ejemplo mostrado el sistema de navegación se encuentra después del elemento `<header>` porque es un sistema de navegación del sitio Web completo. En todo caso, el elemento `<nav>` se puede definir dentro del elemento `<header>` si lo que se desea es construir un sistema de navegación para la cabecera. Es tarea de los diseñadores definir cómo se distribuirán los elementos (HTML5) y cuál será su apariencia (CSS3). En el código se muestra como el sistema de navegación está compuesto por tres opciones: Inicio, Archivo y Contacto. Estas opciones están estructuradas entre los elementos `` y `` que permiten formar listas como se describirá en la Sección 2.3 y el elemento `<a>` que permite crear hiperenlaces los cuales se describirán en la Sección 2.4.

```
13      <nav>
14          <ul>
15              <li><a href="#">Inicio</a></li>
16              <li><a href="#">Archivo</a></li>
17              <li><a href="#">Contacto</a></li>
18          </ul>
19      </nav>
```

Líneas 20, 21: La identificación de un bloque de contenidos principal en un sitio Web es referenciado utilizando el elemento `<section>`. Los bloques de secciones pueden ser anidados para construir diferentes secciones. Normalmente, se utilizan para agrupar contenidos similares.

```
20      <section id="intro">
21      </section>
```

Líneas 22-24: El elemento `<article>` define las zonas del documento donde se ubica el contenido del documento. Normalmente, dentro del elemento `<section>` se pueden encontrar varios elementos `<article>`. No se debe caer en el error de pensar que el elemento `<article>` solamente puede contener artículos de noticias puesto que su objetivo es poder contener unidades independientes de contenido (agrupadas por secciones). Por lo tanto, dentro del elemento `<article>` se puede encontrar artículos, mensajes de foros, comentarios de usuarios, entradas de blogs, anuncios, etc.

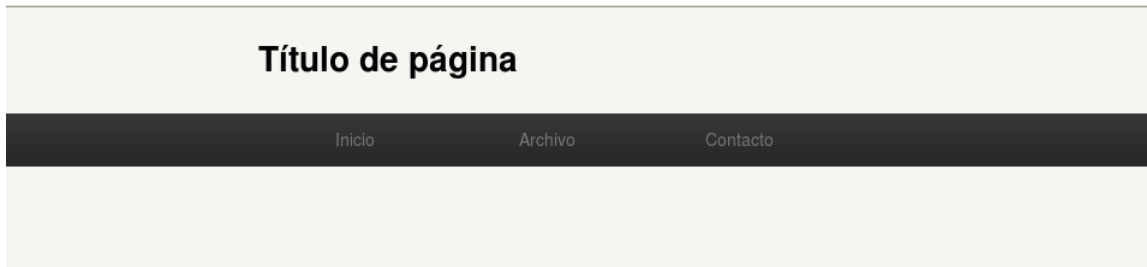
```
22      <section>
23          <article></article>
24      </section>
```

Líneas 25-27: El elemento `<aside>` es añadido en la especificación de HTML5 y es utilizado para suministrar el contenido que está relacionado con el contenido principal. Es decir, suele ser contenido adicional, anexo, que sirve para complementar el contenido principal (que aparece en los elementos `<section>` y `<article>`). El elemento `<aside>` no especifica su ubicación, eso se especifica en la fase de maquetación (CSS3). No obstante, tradicionalmente se han ubicado en un lateral del contenido principal. Por ejemplo, si se ha diseñado una Web con recetas de cocina, el contenido principal son las recetas mientras que la información adicional o complementaria puede ser opiniones de usuarios al respecto.

Título de página

- [Inicio](#)
- [Archivo](#)
- [Contacto](#)

(a)



(b)

Figura 2.2: Resultado de la primera Web en HTML5: (a) sin hoja de estilos (b) con hoja de estilos.

```

25         <aside>
26             <!-- Barra Lateral -->
27         </aside>

```

Líneas 29-31 El elemento `<footer>` permite especificar el pie de la página Web, como ya tratamos en el capítulo anterior, esta zona de la Web tiene una gran importancia hoy en día para mostrar información institucional de la empresa. Al igual que el elemento `<header>` el elemento `<footer>` puede ser utilizado en diferentes ubicaciones y anidada con otros elementos para definir los pie de dichos elementos.

```

29         <footer>
30             <!-- Pie de Página -->
31         </footer>

```

El resultado que se obtiene de interpretar el código anterior en cualquier navegador es el mostrado en la Figura 2.2(a). El resultado es muy poco vistoso, puesto que el navegador no tiene información del aspecto visual que tiene que tener cada uno de los elementos principales de HTML5. No obstante, si se elabora un fichero de estilos (el cual es relacionado utilizando el elemento `<link>`) se puede obtener el resultado mostrado en la Figura 2.2(b).

Ejercicio 2.1 Escribe el contenido del Código 2.1 y estudia cada una de las partes. Observa como se visualiza en los diferentes navegadores Web.

Ejercicio 2.2 Busca estilos CSS que se adapten a HTML5 y enlázalos a tu diseño Web. No debes entender, ni modificar los ficheros CSS, solamente debes utilizarlos.

2.1.2 Estructura completa de un documento HTML5

Una vez presentada la estructura básica de un documento HTML5, aún nos falta por presentar varios elementos de estructuración incorporados en HTML5. Estos elementos son descritos

utilizando un ejemplo de documento HTML5 con todos los elementos de estructuración de HTML5, véase el Código 2.2.

Código 2.2: Estructura de una página Web HTML5 con todos los elementos de estructuración.

```

1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8" />
5   <title>Mi primera página Web HTML5</title>
6   <link rel="stylesheet" href="styles.css" type="text/css" media="screen"
   />
7 </head>
8 <body id="wrapper">
9   <header>
10    <hgroup>
11      <h1><a href="#">Mi primera página Web HTML5</a></h1>
12      <h2>Un interesante subtítulo</h2>
13    </hgroup>
14  </header><!-- end of header -->
15  <nav><!-- top nav -->
16    <ul>
17      <li><a href="#">Principal</a></li>
18      <li><a href="#">Nosotros</a></li>
19      <li><a href="#">Productos</a></li>
20      <li><a href="#">Servicios</a></li>
21      <li><a href="#">Soporte</a></li>
22      <li><a href="#">Contacto</a></li>
23    </ul>
24  </nav><!-- end of top nav -->
25  <main><!-- #main content and sidebar area -->
26    <section id="content"><!-- #content -->
27      <article>
28        <h2><a href="#">Mi primer artículo</a></h2>
29        <figure>
30          
31          <figcaption>
32            Caption primera imagen.
33          </figcaption>
34        </figure>
35        <p>En un lugar de la Mancha, de cuyo nombre no quiero acordarme
          , no ha mucho tiempo que vivía un hidalgo de los de lanza en
          astillero, adarga antigua, rocín flaco y galgo corredor.
          Una olla de algo más vaca que carnero, salpicón las más
          noches, duelos y quebrantos los sábados, lantejas los
          viernes, algún palomino de anhadiduría los domingos, consum
          ían las tres partes de su hacienda.
36      </p>
37    </article>
38    <article>
39      <h2><a href="#">Segundo artículo</a></h2>
40      <p>...lo mismo, y los días de entresemana se honraba con su
          vellorí de lo más fino. Tenía en su casa una ama que pasaba
          de los cuarenta y una sobrina que no llegaba a los veinte, y
          un mozo de campo y plaza que así ensi llaba el rocín como
          tomaba la podadera. </p>
41    </article>
42  </section><!-- end of #content -->
43  <aside id="sidebar"><!-- sidebar -->
44    <h3>Cosas que hacer</h3>
45    <ul>

```



```

46     <li><a href="#">Jugar videojuegos</a></li>
47     <li><a href="#">Chatear con amigos</a></li>
48     <li><a href="#">Leer historias</a></li>
49     <li><a href="#">Vender cosas</a></li>
50     <li><a href="#">Comprar cosas</a></li>
51     <li><a href="#">Intercambiar cosas</a></li>
52 </ul>
53 <h3>Anuncios</h3>
54 <figure>
55     
56     <figcaption>
57         Imagen de anuncio.
58     </figcaption>
59 </figure>
60 <h3>Conecta con nosotros</h3>
61 <ul>
62     <li><a href="#">Twitter</a></li>
63     <li><a href="#">Facebook</a></li>
64 </ul>
65 </aside><!-- end of sidebar -->
66 </main><!-- end of #main content and sidebar-->
67 <footer>
68     <section id="footer-area">
69         <section id="footer-outer-block">
70             <aside class="footer-segment">
71                 <h4>Amigos</h4>
72                 <ul>
73                     <li><a href="#">one linkylink</a></li>
74                     <li><a href="#">two linkylinks</a></li>
75                     <li><a href="#">three linkylinks</a></li>
76                 </ul>
77             </aside><!-- end of #first footer segment -->
78             <aside class="footer-segment">
79                 <h4>Cosas increíbles</h4>
80                 <ul>
81                     <li><a href="#">one linkylink</a></li>
82                     <li><a href="#">two linkylinks</a></li>
83                     <li><a href="#">three linkylinks</a></li>
84                 </ul>
85             </aside><!-- end of #second footer segment -->
86             <aside class="footer-segment">
87                 <h4>Chuladas</h4>
88                 <ul>
89                     <li><a href="#">one linkylink</a></li>
90                     <li><a href="#">two linkylinks</a></li>
91                     <li><a href="#">three linkylinks</a></li>
92                 </ul>
93             </aside><!-- end of #third footer segment -->
94             <aside class="footer-segment">
95                 <h4>Blahdyblah</h4>
96                 <p>&copy; 2015 <a href="#">yoursite.com</a> Praesent
97                     libero. Sed cursus ante dapibus diam. Sed nisi.</p>
98             </aside><!-- end of #fourth footer segment -->
99         </section><!-- end of footer-outer-block -->
100     </section><!-- end of footer-area -->
101 </footer>
102 </body>
103 </html>
104 <!-- Free template created by http://freehtml5templates.com -->

```


El resultado obtenido de la estructuración del segundo código es el mostrado en la Figura 2.3(a) sin aplicar la maquetación y en la Figura 2.3(b) una vez aplicada las hojas de estilos. En el código de la Web propuesto se observan varias etiquetas nuevas que se describen a continuación:

Líneas 10-13: El elemento `hgroup` se utiliza para agrupar elementos de cabecera H (`h1`, `h2`, ... `h6`). Esto se hace así para estructurar los títulos y subtítulos en entidades de mayor envergadura. Se recuerda que el objetivo es dar valor semántico a la estructura del documento HTML para en etapas posteriores u otro software pueda hacer una manipulación adecuada. No tiene sentido utilizar el elemento `hgroup` para agrupar un solo elemento H puesto que para eso se utiliza directamente el elemento H.

```
10     <hgroup>
11         <h1><a href="#">Mi primera página Web HTML5</a></h1>
12         <h2>Un interesante subtítulo</h2>
13     </hgroup>
```

Líneas 29-34, 54-59: El nuevo elemento `<figure>` de HTML5 permite especificar la aparición de un elemento multimedia, ya sea, una imagen, fotografía o vídeo. Esto permite que se pueda situar fácilmente en el diseño el elemento multimedia en el espacio adecuado según cada diseño. Es decir, se pueden definir elementos flotantes. Además, se incorpora el elemento `<figcaption>` asociada al elemento `<figure>` que permite describir una etiqueta (caption) para el elemento multimedia. En el ejemplo presentado se ilustra la aparición de dos elementos multimedia, en este caso imágenes, con sus correspondientes etiquetas.

```
29     <figure>
30         
31         <figcaption>
32             Caption primera imagen.
33         </figcaption>
34     </figure>
```

```
54     <figure>
55         
56         <figcaption>
57             Imagen de anuncio.
58         </figcaption>
59     </figure>
```

En el código anteriormente descrito aparecen otros elementos, que aunque no son novedosos en HTML5, permiten estructurar y especificar texto, imágenes, enlaces o listas. Estos elementos, y algunos más, a lo largo de este capítulo van a ser presentados.

2.2 Texto

En esta sección se describen las etiquetas de versiones anteriores que aún se mantienen en la nueva especificación junto a las nuevos elementos presentados en la última revisión del estándar HTML. Hace años, el formato del texto formaba parte de HTML, pero desde hace ya años estas etiquetas han quedado en desuso y se han eliminado de la especificación. Es importante, reseñar que algunas etiquetas modifican el aspecto del texto pero esto es realizado por los propios navegadores y sin una especificación de formato en CSS3 esto queda a la interpretación de los navegadores.

Mi primera página Web HTML5

Un interesante subtítulo

- [Principal](#)
- [Nosotros](#)
- [Productos](#)
- [Servicios](#)
- [Soporte](#)
- [Contacto](#)

Mi primer artículo



Caption primera imagen.

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor. Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados, lantejas los viernes, algún palomino de añadidura los domingos, consumían las tres partes de su hacienda. El resto d'ella concluían sayo de velarte, calzas de velludo para las fiestas, con sus pantuflos de...

Segundo artículo

...lo mismo, y los días de entresemana se honraba con su vellorí de lo más fino. Tenía en su casa una ama que pasaba de los cuarenta y una sobrina que no llegaba a los veinte, y un mozo de campo y plaza que así ensi llaba el rocín como tomaba la podadera.

Cosas que hacer

- [Jugar videojuegos](#)
- [Chatear con amigos](#)
- [Leer historias](#)
- [Vender cosas](#)
- [Comprar cosas](#)
- [Intercambiar cosas](#)

Anuncios



(a)



(b)

Figura 2.3: Resultado de la segunda Web en HTML5: (a) sin hoja de estilos (b) con hoja de estilos.

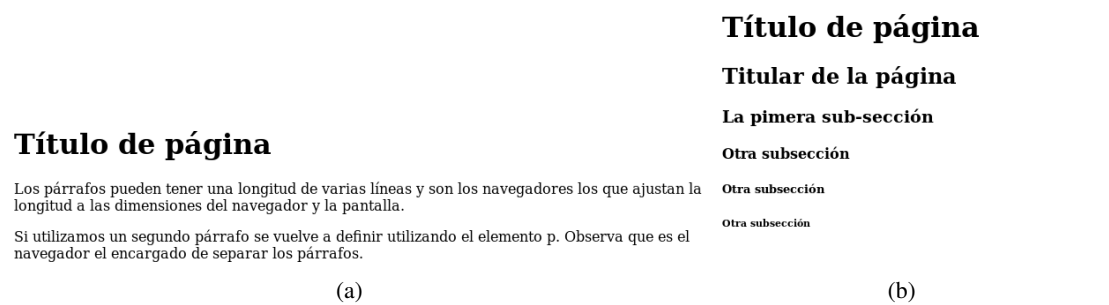


Figura 2.4: HTML5: (a) Uso del elemento p y span (b) Títulos.

2.2.1 Estructurar textos

El texto compone aproximadamente el 90 % del contenido de la Web y es por ello que debe estar bien estructurado para que sea una tarea cómoda y fácil su modificación y actualización. Esto se consigue si previamente tenemos claramente estructurado las diferentes regiones del texto, es decir, estamos definiendo un valor semántico al contenido de la Web.

Párrafos

Lo primero que se debe definir en un texto es la diferencia entre los diferentes párrafos que componen el texto. Esto se consigue utilizando el elemento <p>. En el Código 2.3 se muestra un ejemplo de utilización de los párrafos, el resultado se muestra en la Figura 2.4(a). En este caso, se han especificado dos párrafos. En el segundo párrafo aparece el elemento span el cual se utiliza para especificar líneas de un texto que quieran estar marcadas para posteriormente asignarle un valor semántico o estético. El elemento span es utilizado cuando no existe un elemento específico definido por el propio estándar de HTML.

Código 2.3: HTML: Uso del elemento p y span.

```
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="utf-8">
5     <title>Ejemplo de uso de párrafos</title>
6   </head>
7   <body>
8     <header>
9       <h1>Título de página</h1>
10    </header>
11    <main>
12      <p>Los párrafos pueden tener una longitud de varias líneas y son los
        navegadores los que ajustan la longitud a las dimensiones del
        navegador y la pantalla.</p>
13      <p>Si utilizamos un segundo párrafo se vuelve a definir utilizando el
        elemento p. <span> Observa que es el navegador el encargado de
        separar los párrafos. </span></p>
14    </main>
15    <footer>
16      <!-- Pie de Página -->
17    </footer>
18  </body>
19 </html>
```

Títulos

En las páginas HTML se suelen estructurar en diferentes secciones y cada una de estas secciones o bloques de contenido tiene un título propio. En HTML se definen hasta 6 niveles de cabeceras de títulos. Estos títulos tienen de mayor a menor importancia desde `<h1>` hasta `<h6>`. Por lo tanto, el elemento `<h1>` tiene mayor importancia que `<h2>`, y `<h2>` mayor que `<h3>` y así sucesivamente hasta `<h6>`. En el Código 2.4 se muestra un ejemplo de utilización de los títulos, el resultado se muestra en la Figura 2.4(b). En este caso, se han especificado los seis niveles de títulos. Por defecto los navegadores asignan un tamaño inferior a cada uno de los subtítulos.

Código 2.4: HTML5: Títulos.

```
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="utf-8">
5     <title>Ejemplo de uso de párrafos</title>
6   </head>
7   <body>
8     <header>
9       <h1>Título de página</h1>
10    </header>
11    <main>
12      <h2>Titular de la página</h2>
13      <h3>La primera sub-sección</h3>
14      <h4>Otra subsección</h4>
15      <h5>Otra subsección</h5>
16      <h6>Otra subsección</h6>
17    </main>
18    <footer>
19      <!-- Pie de Página -->
20    </footer>
21  </body>
22 </html>
```

2.2.2 Marcado de textos

Una vez que se ha estructurado un documento y el texto principal en secciones y párrafos el siguiente paso es asignarle un valor semántico al texto que aparece en los párrafos. Son muchos los elementos encargados de asignar valor semántico al texto y cada vez aparecen más. Además, en caso de que no exista un elemento específico se debe utilizar el elemento `` que se presentó en la sección anterior.

En esta sección se presentan algunos de los elementos principales de marcado. No obstante, se debe tener en cuenta que en cualquier momento pueden ir apareciendo nuevos elementos o irse eliminando de la especificación de HTML. En el Código 2.5 se presentan algunos de los principales elementos de marcado. El resultado es mostrado en la imagen de la Figura ??.

Código 2.5: HTML5: Marcado de textos.

```
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="utf-8">
5     <title>Ejemplo de uso de marcado de texto</title>
6   </head>
7   <body>
8     <header>
9       <h1>Título de página</h1>
```

```

10  </header>
11  <main>
12    <p>El lenguaje HTML5 permite especificar fragmentos de texto según
      algunos segmentos de texto según su importancia como <em>
      importantes</em> o como <strong>los más importantes</strong>. Ade
      más, si este <mark>texto es relevante en este contexto</mark>
      también se dispone de una etiqueta propia. Finalmente, se puede
      utilizar un elemento antiguo <b> para marcar el texto</b>. </p>
13    <p>El uso de la etiqueta i <del datetime="20100112" cite="http://www.
      w3c.org"> es ampliamente utilizada para remarcar texto importante
      </del> <ins> ha quedado en desuso en el nuevo estándar</ins>.</p>
14    <blockquote cite="http://stackoverflow.com">
15      <p>Type HTML in the textarea above, and it will magically appear
      in the frame below.</p>
16    </blockquote>
17    <p>Como dijo Edsger W. Dijkstra: <q url="http://en.wikipedia.org/wiki
      /Edsger_W._Dijkstra"> La Ciencia de la Computación no tiene que
      ver con las computadoras más que la Astronomía con los telescopios
      . </q>
18    </p>
19    <p>El lenguaje <abbr title="HyperText_Markup_Language">HTML</abbr> es
      estandarizado por el <abbr title="World_Wide_Web_Consortium">W3C
      </abbr>.</p>
20  </main>
21  <footer>
22    <small>
23      Derechos Reservados &copy; <time datetime="2013-10-12">2013 Mi
      empresita S.L </time>
24      <address> C\Torrelaguna, 23, Madrid</address>
25    </small>
26  </footer>
27 </body>
28 </html>

```

Lo primero que se debe decidir es el nivel de relevancia que tiene un fragmento de texto. De este modo se encuentran actualmente hasta 4 elementos que permiten definir este nivel. Observe la línea 12 del Código 2.5 donde se muestra el uso de estos 4 elementos:

- ****: Especifica énfasis en un texto (sustituye al elemento <i> que refleja texto en cursiva, el cual se reflejará en las hojas de estilos). Por defecto, los navegadores muestran los elementos en cursiva.
- ****: Especifica importancia en un texto. Por defecto, los navegadores muestran los elementos en negrita.
- **<mark>**: Es para resaltar texto relevante a las circunstancias. Por ejemplo, cuando se realiza una búsqueda por palabras y esas palabras se encuentran en un determinado texto.
- ****: Especifica importancia cuando no se adapta el texto a ninguna de las anteriores circunstancias. Antiguamente era utilizado para poner el texto en negrita. Esta idea está ya en desuso.

HTML también permite especificar cuando se han realizado modificaciones en el contenido de una página. Es decir, se puede especificar el texto que ha sido eliminado o insertado en un documento previamente elaborado. Esto se consigue utilizando los elementos <ins> y para información insertada y eliminada respectivamente. Además, se suele incluir los atributos *cite* y *datetime* los cuales permiten especificar una dirección Web donde se describe el motivo por el que se realizó la modificación y la fecha en la que se llevó a cabo. Por defecto, el texto eliminado

se muestra tachado mientras que el nuevo texto aparece por defecto subrayado. Observe la línea 13 del Código 2.5 donde se refleja el uso de estas etiquetas.

Al crear documentos Web es importante poder hacer referencia a los autores de la información o desde donde se ha extraído. Entre las líneas 13 y 18 del Código 2.5 se muestra el uso de las referencias. En la actualidad se disponen de varios elementos para realizar el marcado de las referencias:

- **cite:** Permite marcar el origen de una cita, a partir de HTML5 es utilizado exclusivamente para mencionar títulos de obras. Antiguamente, se utilizaba también para referenciar a personas. Hoy en día, existen muchas Webs que siguen referenciando a personas y los validadores automáticos son incapaces de detectar dicha situación como un error puesto que no pueden distinguir el nombre de una obra o de un autor.
- **blockquote:** Se utiliza para citas largas, en bloques de textos (párrafos, listas, etc.). Suele tener el atributo *cite* (No confundir con el elemento anteriormente descrito) que permite especificar la URL de origen de la información.
- **q:** Se utiliza para citas cortas, dentro normalmente de párrafos o listas. Es decir, no es un bloque en sí, sino parte de un bloque de contenido.

Otro interesante elemento por su accesibilidad es `<abbr>` el cual permite describir abreviaturas y acrónimos. En versiones anteriores estaba aceptado el elemento `<acronym>` pero este quedó en desuso debido a que con el elemento `<abbr>` se abarcan tanto abreviaturas como acrónimos. En la línea 19 del Código 2.5 se muestra el uso del elemento `<abbr>` para la definición de los acrónimos HTML y W3C. En la mayoría de los navegadores se muestra por defecto un borde inferior punteado que al posicionar el puntero del ratón sobre la palabra subrayada muestra el valor del atributo *title*.

Finalmente, en HTML5 se incorporan tres elementos muy interesantes que son `<small>`, `<address>` y `<time>`. El objetivo del elemento `<small>` es suministrar un espacio para texto pequeño, de menor tamaño. Normalmente utilizado para describir información institucional en el pie de página. El elemento `<address>` se incorpora para especificar direcciones de contacto, estas pueden ser tanto físicas como virtuales, es decir, direcciones postales, URL o correos electrónicos. Finalmente, el elemento `<time>` permite definir el momento en el cual se actualizó el texto. Esto permite saber si un texto es obsoleto o es novedoso. Algo muy útil para automatizar sistemas. En las líneas 23 y 24 del Código 2.5 se muestra la utilización de estas etiquetas.

Ejercicio 2.3 Estructura y marca utilizando los elementos necesarios para conseguir los resultados de las imágenes de la Figura 2.5.

Ejercicio 2.4 Estructurar y marcar utilizando todas las etiquetas anteriores un texto cualquiera sobre el tema que más te interese (Deporte, cine, música, ...).

2.2.3 Espacios en blanco y saltos de líneas

A los usuarios normalmente acostumbrados a sistemas de edición WYSIWYG (*What You See is What You Get*, lo que ves es lo que obtienes) les sorprende que en la edición de documentos HTML cuando aparecen varios espacios en blancos solamente son tomadas como un solo espacio o que los saltos de líneas no son reflejados en el documento. Esto es algo muy común y extendido en varios formatos, por ejemplo LaTeX.

Para definir los saltos de líneas se define el elemento `
`. Al ser un elemento vacío la misma de apertura es la de cierre. El resultado obtenido es equivalente al salto de línea en un procesador de texto. Por otro lado, los espacios en blancos son incluidos haciendo uso de un carácter especial el cual debe escribirse con el texto ** **. Es importante que el primer

GNU/Linux

GNU/Linux es uno de los términos empleados para referirse a la combinación del núcleo o kernel libre similar a Unix denominado Linux con el sistema GNU. Su desarrollo es uno de los ejemplos más prominentes de software libre; todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la [GPL](#) y otra serie de licencias libres.

A pesar de que *Linux* es, en sentido estricto, el núcleo del sistema operativo, parte fundamental de la interacción entre el hardware y el usuario (o los programas de aplicación) se maneja usualmente con las herramientas del proyecto GNU y con entornos de escritorio basados en **GNOME**, que también forma parte del proyecto GNU aunque tuvo un origen independiente. Sin embargo, una parte significativa de la comunidad, así como muchos medios generales y especializados, prefieren utilizar el término Linux para referirse a la unión de ambos proyectos.

A las variantes de esta unión de programas y tecnologías, a las que se les adicionan diversos programas de aplicación de propósitos específicos o generales se las denomina distribuciones. Su objetivo consiste en ofrecer ediciones que cumplan con las necesidades de un determinado grupo de usuarios. Algunas de ellas son especialmente conocidas por su uso en servidores y supercomputadoras donde tiene la cuota más importante del mercado. Según un informe de *IDC*, *GNU/Linux es utilizado por el 78% de los principales 500 servidores del mundo*, otro informe le da una *cuota de mercado de 89% en los 500 mayores supercomputadores*. Con menor cuota de mercado el sistema GNU/Linux también es usado en el segmento de las computadoras de escritorio, portátiles, computadoras de bolsillo, teléfonos móviles, sistemas embebidos, videoconsolas y otros dispositivos.

Historia de Linux

El nombre GNU, *GNU's Not Unix* (GNU no es Unix), viene de las herramientas básicas de sistema operativo creadas por el proyecto GNU, iniciado por **Richard Stallman** en 1983 y mantenido por la **FSF**. El nombre Linux viene del núcleo Linux, inicialmente escrito por **Linus Torvalds** en 1991.

La contribución de GNU es la razón por la que existe controversia a la hora de utilizar Linux o GNU/Linux para referirse al sistema operativo formado por las herramientas de GNU y el núcleo Linux en su conjunto.

(a)

Edsger Wybe Dijkstra

Edsger Wybe Dijkstra- (Róterdam, Países Bajos, 11 de mayo de 1930 - Nuenen, Países Bajos, 6 de agosto de 2002) fue un científico de la computación de los Países Bajos.

Poco después de su muerte en el 2002, recibió la distinción ACM PODC Influential Paper Award en computación distribuida por su trabajo en la auto-estabilización en programas computacionales. Este premio fue renombrado a Premio Dijkstra el siguiente año en su honor. Algunas de sus citas más relevantes fueron:

"Es prácticamente imposible enseñar programación correctamente a estudiantes que han estado expuestos al lenguaje BASIC con anterioridad. Como potenciales programadores, tienen la mente mutilada sin esperanza alguna de regeneración."

"Si la depuración es el proceso de eliminar errores, entonces la programación debe ser el proceso de introducirlos."

(b)

Figura 2.5: Ejercicios de HTML5: Marcado de texto.

Título de página

Este primer párrafo no contiene saltos de línea ni otro tipo de espaciado.

Este segundo párrafo sí que contiene saltos
de
línea
y otro tipo de espaciado.

Derechos Reservados © 2013 Mi empresita S.L
C\Torrelaguna, 23, Madrid

Figura 2.6: HTML5: Espacios en blanco y saltos de líneas.

Finalmente, puede surgir la necesidad de mostrar texto tal y cual lo escribimos no por cuestiones estéticas sino porque debe ir de ese modo. Así que existe un elemento especial (<pre>) que permite mostrar la información tal y como aparece entre dicho elemento. En el Código 2.7 se muestra el uso de este elemento para un famoso poema.

Código 2.7: HTML5: Elemento <pre>.

```
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="utf-8">
5     <title>Ejemplo de uso de espacios blancos y saltos de líneas</title>
6   </head>
7   <body>
8     <header>
9       <h1>Título de página</h1>
10    </header>
11    <main>
12      <pre>
13 En la torre
14 amarilla,
15 dobla una campana.
16
17 Sobre el viento
18 amarillo,
19 se abren las campanadas.
20
21 En la torre
22 amarilla,
23 cesa la campana.
24
25 El viento con el polvo,
26 hace proras de plata.
27    </pre>
28  </main>
29    <footer>
30      <small>
31        Derechos Reservados &copy; <time datetime="2013-10-12">2013 Mi
          empresita S.L </time>
32                                <address> C\Torrelaguna, 23, Madrid</address>
33      </small>
34    </footer>
35  </body>
36</html>
```

Ejercicio 2.6 Utilizando el elemento <pre> elabora un documento HTML que obtenga el resultado similar al de la Figura 2.7. ■

2.2.4 Codificación de caracteres

Algunos caracteres no se pueden escribir directamente puesto que serían interpretados de manera diferente a nuestro objetivo, por ejemplo, en la sección anterior se presentaba como carácter especial el espacio en blanco. Pero imagina que se quieren mostrar los tags (< >) o caracteres que no se encuentran en español pero no se ha definido el juego de caracteres adecuadamente. Muchos de estos problemas se han minimizado en HTML5 con la elección del juego de caracteres y con la definición del lenguaje pero aún así es importante conocer la existencia de este juego de caracteres. Los textos que sustituyen un carácter especial se denominan entidades HTML y en la tabla 2.1 se muestran algunas de las entidades más representativas.

Entidad	Carácter	Entidad	Carácter	Entidad	Carácter
<	<	>	>	&	&
"	“	 	espacio en blanco	'	'
ñ	ñ	Ñ	Ñ	á	á
é	é	í	í	ó	ó
ú	ú	Á	Á	É	É
Í	Í	Ó	Ó	Ú	Ú

Cuadro 2.1: Entidades HTML

Sintáxis de la etiqueta abbr

La sintaxis de la etiqueta `<abbr>` se muestra a continuación:

```
<abbr title="Texto a mostrar">Texto en abreviatura</abbr>
```

Figura 2.8: Ejercicio de HTML5: Caracteres especiales.

Ejercicio 2.7 Reescriba el Código 2.7 utilizando entidades HTML en lugar del juego de caracteres.

Ejercicio 2.8 Escriba el código HTML para conseguir el resultado mostrado en la Figura 2.8.

Ejercicio 2.9 Busca información y describe la utilidad de los siguientes elementos: `<q>`, `<dfn>`, `<var>`, `<s>`, `<code>`, `<samp>`, `<kbd>`, `<ruby>`, `<rt>`, `<rp>`, `<bdi>`, `<bdo>`, `<wbr>`.

Ejercicio 2.10 Sobre un documento HTML aplica adecuadamente cada uno de los elementos anteriores.

2.3 Listas

El ser humano está acostumbrado a clasificar y catalogar los elementos del mundo. Es por ello, que normalmente se construyen listas que permiten estructurar los elementos del discurso de un modo cómodo para el receptor del mensaje. Además, en los sitios Web también son utilizados para especificar las diferentes opciones de los sistemas de navegación. En HTML existen tres tipos de listas diferentes: Listas no ordenadas, listas enumeradas y listas de definición. En las siguientes secciones se describen cómo se hace uso de este tipo de listas en HTML.

2.3.1 Listas no ordenadas

Las listas no ordenadas son las más sencillas y las que más se utilizan tanto en HTML como en la redacción de documentos. Una lista no ordenada es un conjunto de elementos relacionados entre sí pero para los que no se indica un orden o secuencia determinados. Para formar listas no ordenadas se hace uso del elemento `` para indicar que existe una lista no ordenada. Posteriormente, el elemento `` es el que define cada uno de los elementos de la lista. En el Código 2.8 se muestra la creación de una lista no ordenada que puede ser utilizada como sistema de navegación.

Código 2.8: HTML5: Listas no ordenadas.

```
12 <ul>
13   <li>Inicio</li>
14   <li>Productos</li>
15   <li>Servicios</li>
16   <li>Contacto</li>
17 </ul>
```

2.3.2 Listas ordenadas

En las listas ordenadas los elementos relacionados muestran un orden determinado y configurable. Una de las listas ordenadas más conocidas es la tabla de contenidos o índice de un libro. Para definir listas ordenadas existe el elemento ``, así que para especificar la lista ordenada se hace uso del elemento `` y para cada uno de los elementos de esta lista se sigue utilizando el elemento ``. En el Código 2.9 se muestra el uso de las listas ordenadas.

Código 2.9: HTML5: Listas ordenadas.

```
18 <ol>
19   <li>Percepción visual</li>
20   <li>Estructuración Web: HTML5</li>
21   <li>Maquetación Web: CSS3</li>
22 </ol>
```

2.3.3 Listas de descripción

Las listas de definición o descripción son ampliamente utilizado en la redacción de documentos para construir glosarios. No obstante, en los sitios Web está poco extendido su utilización. Para la especificación de las listas de descripción se hace uso de tres nuevos elementos `<dl>`, `<dt>` y `<dd>`. El elemento `<dl>` es el encargado de definir la creación de la lista de definición y los elementos `<dt>` y `<dd>` especifican el término y la descripción de cada elemento de la lista respectivamente. En el Código 2.10 se muestra el uso de las listas de definición.

Código 2.10: HTML5: Listas de descripción.

```
23 <dl>
24   <dt>ac</dt>
25   <dd>Imprime estadísticas acerca del tiempo que han estado
        conectado los usuarios.</dd>
26   <dt>adduser</dt>
27   <dd>Ver useradd</dd>
28   <dt>alias</dt>
29   <dt>apt-get</dt>
30   <dt>arp</dt>
31   <dt>arping</dt>
32   <dt>arptables</dt>
33   <dd>Firewall similar en funciones a iptables pero para control
        de tráfico de protocolo arp.</dd>
34 </dl>
```

Ejercicio 2.11 Escribe el documento HTML para obtener como resultado el mostrado en la Figura 2.9. ■

Menú

- Inicio
- **Productos**
 - Recientes
 - **Más vendidos**
 - Mejor valorados
- Servicios
 1. **Consultoría**
 2. Diseño Web
 3. Desarrollo de Apps
 4. Otros
- Contacto

Email
nombre@direccion.com

Teléfono
900 900 900

Dirección
C\ Torrelaguna

Figura 2.9: Ejercicio de HTML5: Listas.

2.4 Hiperenlaces e imágenes

En esta sección se describirán los elementos de estructuración para crear hiperenlaces e insertar imágenes en nuestros documentos HTML.

2.4.1 Hiperenlaces

Los hiperenlaces o enlaces se utilizan para establecer relaciones entre dos recursos. Una de las principales características de la Web es la posibilidad de relacionar los recursos entre sí. Dentro de los recursos que se pueden relacionar no solamente son documentos HTML sino que se pueden enlazar cualquier tipo de documento. Por tanto, un enlace está compuesto por dos extremos (recursos) y un sentido. Cada uno de estos extremos se llaman anclas (*anchors*) y pueden referenciarse a través de su propio nombre.

El elemento `<a>` es el encargado de crear los enlaces. Este elemento tiene dos atributos muy importantes, *href* que permite definir la ruta del recurso enlazado y *name* que especifica el nombre del enlace para que se pueda acceder desde otros enlaces.

Código 2.11: HTML5: Enlaces.

```
12 <a href="http://www.google.com">Página principal de Google</a><br/>
13 <a href="http://www.ejemplo.com/imagen.jpg">Imagen</a><br/>
14 <a href="http://www.ejemplo.com/fichero.pdf">[PDF]</a><br/>
15 <a href="http://www.ejemplo.com/fichero.doc">[DOC]</a><br/>
```

En el Código 2.13 se muestran varios ejemplos de creación de hiperenlaces haciendo uso del atributo href. En la línea 12 se muestra un enlace a la página Web de Google, entre las líneas 13 y 15 se muestran enlaces a diferentes archivos (imágenes, PDF y Microsoft Word).

Código 2.12: HTML5: Enlaces.

```
17 <a href="http://www.ejemplo.com/pagina1.html">Enlace página 1</a>
18 <a href="http://www.ejemplo.com/pagina1.html#segunda_seccion">Enlace a la
    sección 2 de la página 1</a>
19
```

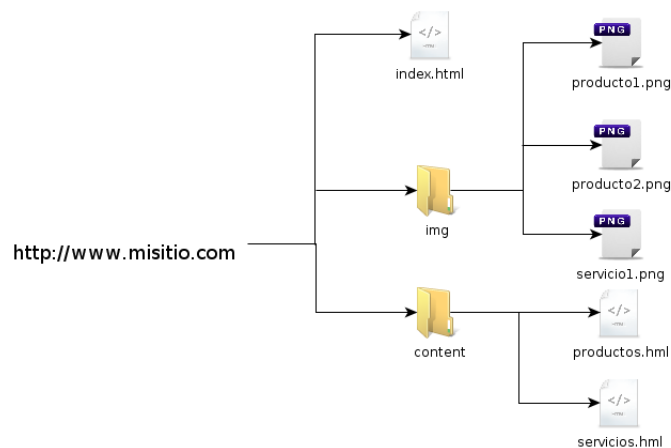


Figura 2.10: Ejercicio de HTML5: Enlaces.

```

20 <a name="inicio"></a>
21 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id
    ligula eu felis adipiscing ultrices. Duis gravida leo ut lectus.
    Praesent condimentum mattis ligula.</p>
22 <a href="#inicio">Volver al inicio de la página</a>
23 <p>Pellentesque malesuada. In in lacus. Phasellus erat erat, lacinia a,
    convallis eu, nonummy et, odio. Aenean urna elit, ultrices id,
    placerat varius, facilisis eget, dolor.</p>
24 <a href="#inicio">Volver al inicio de la página</a>

```

Además de enlazar a recursos el atributo href permite abrir el gestor de correos predeterminado del usuario completándole cierta información como el correo o el asunto. En lugar de utilizar el protocolo http:// se debe utilizar *mailto*:

```

1 <a href="mailto:nombre@direccion.com?subject=Preguntas/Comentarios">
    Contacto </a>

```

En el código anterior se muestra como se especifica un correo electrónico *nombre@direccion.com* y como asunto *Preguntas/Comentarios*. Al pulsar sobre la palabra contacto se lanzará la aplicación que gestiona los correos en el sistema operativo y usuario operativo en el momento. Esta técnica está quedando totalmente en desuso por varias razones:

1. **Mala usabilidad.** Al estar navegando por una Web y tener que cambiar a otro tipo de herramienta, que quizás no tengamos configurada puesto que estemos navegando desde un ordenador que no sea el nuestro o simplemente como usuarios no hagamos uso de ese tipo de herramientas.
2. **Peligro de spam.** Al dejar fácilmente la dirección de contacto, pueden robots programados tomar esa cuenta de correo y enviar correos no deseados.

Así que, aunque hoy en día el estándar lo permite, es muchísimo más recomendable construir un formulario Web y hacer que el usuario nos escriba sus inquietudes desde ese formulario. Si se dispone de conocimientos de lado de servidor se podrá incluso hacer un tratamiento de la información con mayor precisión que si solamente se hace uso del lado del cliente.

Ejercicio 2.12 Debe elaborar la estructura de ficheros y los enlaces entre los ficheros mostrados en la Figura 2.10. Debe cumplir como mínimo los siguientes requisitos:

- Todas las páginas deben estar estructuradas en HTML5.
- Todas las páginas deben permitir volver al inicio.
- Las páginas de productos y servicios deben enlazar las imágenes de sus respectivos productos o servicios (no deben mostrarse las imágenes, solo enlazarse).
- Las páginas deben tener enlaces en la misma página hacia secciones de la misma y hacia otras páginas externas.

Ejercicio 2.13 Busca la función del atributo `target` y otro más para el elemento `a`. Construye una Web en la que se compruebe cada una de las funcionalidades implementadas.

2.4.2 Imágenes

Las imágenes son uno de los primeros elementos multimedia que aparecieron en las páginas Web. Hoy en día, en todas las páginas Web aparecen imágenes ya sean imágenes de contenido o imágenes de adorno.

Las imágenes de contenido son las que complementan o proporcionan información. Por otro lado, las imágenes de adorno son las que cumplen funciones en el diseño del interfaz gráfico para proporcionar algunas de las características vistas en el Capítulo 1. Las imágenes de adorno se deben incluir en las hojas de estilos mientras que las imágenes de contenido se incluyen directamente en el documento HTML a través del elemento vacío ``. El elemento `` tiene varios atributos que son ampliamente utilizados:

- **src:** Especifica la URL de la imagen que se muestra.
- **name:** Nombre del elemento imagen.
- **height:** Indica la altura con la que se debe mostrar la imagen.
- **width:** Indica la anchura con la que se debe mostrar la imagen.
- **alt:** Especifica un texto descriptivo de la imagen en caso de que no se pueda realizar la carga de la misma.
- **longdesc:** Especifica una URL donde se puede encontrar una descripción más detallada de la imagen.

El atributo `src` tiene un funcionamiento similar al atributo `href` de los enlaces ya que establece la URL de la imagen que se va a mostrar. El atributo `alt` es utilizado para describir la imagen en caso de que no se pueda cargar, esta descripción debe ser breve (menos de 1024 caracteres). La recomendación hoy en día es utilizar formatos gráficos soportados por todos los navegadores como son GIF, JPG/JPEG y PNG. En el Capítulo 4 se estudiará detenidamente los tipos de formatos.

Los atributos `width` y `height` se utilizan para indicar las dimensiones a mostrar de las imágenes. Lo lógico sería que esta especificación estuviera en la maquetación y no en la estructuración y contenido del documento pero al existir una gran cantidad de imágenes estoy provocaría que las especificaciones CSS se sobrecargaran haciéndolas complicadas de mantener y provocando una pérdida de usabilidad en las páginas Web. Es por ello, que esta especificación es la excepción a la norma en el desarrollo de Webs con HTML/CSS. Los valores de los atributos `width` y `height` se expresan mediante un número entero. Si no se indica lo contrario los navegadores toman dicho valor como la medida en píxeles. Aunque, también es posible especificar las dimensiones

utilizando porcentajes. En este caso, el porcentaje hace referencia a la altura o anchura del elemento en el que está contenida la imagen.

Las novedades aparecidas en HTML5 respecto a la estructuración de las imágenes, elementos `<figcaption>` y `<figure>` fueron presentadas en el capítulo anterior.

Código 2.13: HTML5: Imágenes.

```

12 <figure>
13   
14 </figure>
15 <figure>
16   
17   <figcaption>TS-0212K</figcaption>
18 </figure>
19 <figure>
20   
21   <figcaption>TS-0232K</figcaption>
22 </figure>
23 <figure>
24   
25   <figcaption>TS-1500P</figcaption>
26 </figure>

```

En el código anterior se muestra la aparición de cuatro imágenes diferentes, la primera corresponde al logotipo de la empresa y suele ir situado en la cabecera de la estructura Web. Las siguientes imágenes son relativas a microondas en las que los tamaños de las imágenes son especificados con valores absolutos de píxeles o con porcentajes. Es recomendable ajustar los valores según porcentajes puesto que los valores de píxeles absolutos pueden visualizarse incorrectamente según los tamaños de las pantallas de visualización. No es lo mismo un equipo sobremesa que un dispositivo móvil o tablet.

Ejercicio 2.14 Modifica el Ejercicio 2.12 para que aparezcan las imágenes en lugar de textos. Añade también imágenes de cabecera y pie de página. Reflexiona si las imágenes de cabecera y pie de página corresponden a imágenes de contenido o de adorno.

Ejercicio 2.15 Existe un modo de interactuar con imágenes denominada mapas de imágenes. Aún sigue en la especificación de HTML5 aunque van surgiendo alternativas que se verán en capítulos anteriores. El alumno debe completar paso a paso los ejemplos que se describen en las siguientes direcciones:

<http://www.html-5-tutorial.com/map-and-area-elements.htm>

<http://www.tutorialized.com/view/tutorial/Image-Map-Rollover/3484>

2.5 Tablas

Las tablas son elementos que permiten mostrar datos de una forma ordenada y estructurada por filas y columnas. Una mala práctica ya desechada era la maquetación con tablas puesto que previo a la aparición de los estilos y en los primeros años de estos era muy complicado poder situar los elementos en diferentes posiciones de la pantalla. Se hacía uso de las tablas como un viejo truco en programación para conseguir una malla de posiciones de la pantalla y poder ir situando cada elemento donde se deseara. Eso hoy en día no está aceptado como una práctica de diseño de interfaces Web puesto que existen mecanismos para conseguir situar los elementos en las posiciones deseadas.

2.5.1 Estructura de tablas

Las tablas por tanto permiten mostrar datos de un modo tabulado, estos datos pueden provenir de un fichero estático o de una base de datos a través de ciertas consultas. Para la elaboración de las tablas se hace uso de varios elementos que permiten ir definiendo cada una de las zonas por las que se componen las tablas. A continuación se van a describir los principales elementos que aparecen en la elaboración de las tablas:

- **table:** Es el nodo padre de la creación de las tablas. Dentro de este elemento es donde tienen significado los diferentes elementos de las tablas.
- **caption:** Este elemento es opcional al igual que cuando aparece en el elemento `<figure>`. En este caso describe una leyenda descriptiva para la tabla.
- **colgroup:** Especifica un grupo de una o más columnas en la tabla para especificarle un formato.
- **col:** Especifica las propiedades de la columna para cada columna que son hijos del elemento `colgroup`.
- **th:** Define una celda cabecera en la tabla.
- **tr:** Define una fila de una tabla.
- **td:** Define una celda de una tabla.
- **thead:** Agrupa la cabecera de la tabla.
- **tbody:** Agrupa el cuerpo de la tabla.
- **tfoot:** Agrupa el pie de la tabla.

El elemento `<table>` tiene un conjunto de atributos que permiten adaptar la tabla a las necesidades de cada conjunto de datos. Así se pueden encontrar los siguientes atributos:

- **align:** Este atributo configura la alineación del texto. Este atributo permite valores como *left*, *right* o *center*. En la actualidad, este atributo está en desuso y se recomienda utilizar hojas de estilos.
- **bgcolor:** Permite configurar el color de fondo de la tabla. Este atributo también se encuentra en desuso por su adaptación en las hojas de estilos.
- **border:** Permite especificar el tamaño del borde entre las celdas. Si se especifica el valor 0 entonces no aparece borde entre celdas y la tabla.
- **bordercolor:** Permite especificar el color del borde. Este valor también se encuentra en desuso.
- **cellpadding:** Este atributo especifica el número de píxeles que existe de espacio entre las celdas y su contenido.
- **cellspacing:** Este atributo especifica el número de píxeles que existe de espacio entre las celdas y el borde exterior de la tabla.
- **colspan:** Permite unir varias columnas en una sola.
- **rowspan:** Permite unir varias filas en una sola.
- **height/width:** Al igual que con las imágenes este atributo especifica las dimensiones del tamaño. En el caso de que no se especifique el navegador redimensiona la tabla al mejor ajuste para su contenido.

Código 2.14: HTML5: Tablas.

```
12 <table border='1' bgcolor='cyan' width='450' height='200' cellpadding=  
    '5' cellspacing='5'>  
13   <thead>  
14     <tr bgcolor='cyan'>  
15       <th width='200'>Heading 1</th>
```

Cuota de mercado de los navegadores Web Julio de 2020

Heading 1	Heading 2	Heading 3
Row 1, Col 1	Row 1, Col 2	Row 1, Col 3
Row 2, Col 1	Row 2, Col 2	Row 2, Col 3

(a)

Browser	Share
Internet Explorer	2.2%
Mozilla Firefox	27.9%
Google Chrome	16.8%
Apple Safari	7.3%
Opera	43.5%
Others & Mobile	3.3%

(b)

Characteristics with positive and negative sides

Characteristics are given in the second column, with the negative side in the left column and the positive side in the right column.

1	4	7	10
2	5	8	
3	6	9	

(c)

Negative	Characteristic	Positive
Sad	Mood	Happy
Failing	Grade	Passing
Foot 1	Foot 2	Foot 3

(d)

Figura 2.11: HTML5: Tablas.

```

16     <th>Heading 2</th>
17     <th>Heading 3</th>
18   </tr>
19 </thead>
20 <tbody>
21   <tr bgcolor='green'>
22     <td width='200'>Row 1, Col 1</td>
23     <td>Row 1, Col 2</td>
24     <td>Row 1, Col 3</td>
25   </tr>
26   <tr bgcolor='yellow'>
27     <td>Row 2, Col 1</td>
28     <td>Row 2, Col 2</td>
29     <td>Row 2, Col 3</td>
30   </tr>
31 </tbody>
32 </table>

```

En el Código 2.14 se muestra el primer ejemplo de tabla en HTML. En esta tabla se ha configurado una tabla con varios parámetros de configuración de dos filas y tres columnas. Además, se ha incorporado una fila de cabecera con los títulos de cada columna. El resultado se muestra en la Figura 2.11(a).

Código 2.15: HTML5: Tablas.

```

35 <table border='2' cellpadding='5' cellspacing='1' width='250' bgcolor='
    lightgreen' bordercolor='blue'>
36   <caption>
37     <em>Cuota de mercado de los navegadores Web Julio de 2020</em>
38   </caption>
39   <thead>
40     <tr bgcolor='blue'>
41       <th>Browser</th>

```

```

42     <th>Share</th>
43   </tr>
44 </thead>
45 <tbody>
46   <tr>
47     <td>Internet Explorer</td>
48     <td>2.2%</td>
49   </tr>
50   <tr>
51     <td>Mozilla Firefox</td> <td>27.9%</td>
52   </tr>
53   <tr>
54     <td>Google Chrome</td>
55     <td>16.8%</td>
56   </tr>
57   <tr>
58     <td>Apple Safari</td>
59     <td>7.3%</td>
60   </tr>
61   <tr>
62     <td>Opera</td>
63     <td>43.5%</td>
64   </tr>
65   <tr>
66     <td>Others & Mobile</td>
67     <td>3.3%</td>
68   </tr>
69 </tbody>
70 </table>

```

En el Código 2.15 se muestra un segundo ejemplo de tabla en HTML. En esta ocasión se muestra una comparativa de la cuota de mercado de varios navegadores. Se ha incorporado el elemento <caption> asociado a la tabla. El resultado se muestra en la Figura 2.11(b).

Código 2.16: HTML5: Tablas.

```

35 <table border='1' width='450' height='200'>
36   <tr align='center'>
37     <td width='25%'>1</td>
38     <td width='25%'>4</td>
39     <td width='25%'>7</td>
40     <td width='25%'>10</td>
41   </tr>
42   <tr align='center'>
43     <td>2</td>
44     <td>5</td>
45     <td>8</td>
46     <td rowspan='2' bgcolor='lightgray'></td>
47   </tr>
48   <tr align='center'>
49     <td>3</td>
50     <td>6</td>
51     <td>9</td>
52     <!-- Cell omitted here -->
53   </tr>
54 </table>

```

En el Código 2.16 se muestra otro ejemplo de tabla en HTML. Se hace uso del elemento rowspan para unir dos filas en una. El resultado se muestra en la Figura 2.11(c). Finalmente, en el Código 2.17 se muestra un ejemplo completo de estructuración de una tabla en HTML5 donde

se incorpora la tabla dentro de un elemento figure. Recordamos que el elemento figure no va relacionado con imágenes, sino con elementos multimedia que pueden ser flotantes en la página Web. El resultado se muestra en la Figura 2.11(d).

Código 2.17: HTML5: Tablas.

```

35
36 <figure>
37   <figcaption>
38     <strong>Characteristics with positive and negative sides</strong>
39     <p>Characteristics are given in the second column, with the negative
        side in the left column and the positive side in the right column
        .</p>
40   </figcaption>
41   <table border="1">
42     <thead>
43       <tr>
44         <th id="n"> Negative</th>
45         <th> Characteristic</th>
46         <th> Positive</th>
47       </tr>
48     </thead>
49     <tbody>
50       <tr>
51         <td headers="n_r1"> Sad</td>
52         <th id="r1"> Mood</th>
53         <td> Happy </td>
54       </tr>
55       <tr>
56         <td headers="n_r2"> Failing</td>
57         <th id="r2"> Grade</th>
58         <td> Passing</td>
59       </tr>
60     </tbody>
61     <tfoot>
62       <tr>
63         <td> Foot 1</td>
64         <td> Foot 2</td>
65         <td> Foot 3 </td>
66       </tr>
67     </tfoot>
68   </table>
69 </figure>

```

Ejercicio 2.16 Crea los documento HTML necesarios para conseguir los resultados mostrados en la Figura 2.12. ■

2.6 Formularios Web 2.0

El estándar HTML5 ha realizado un gran avance en torno a los formularios Web, ya que estos elementos suponen una de las principales fuentes de recopilación de información que tienen los desarrolladores Web. Es importante resaltar que cada navegador soporta de manera diferente las nuevas características proporcionadas por HTML5. A pesar de esto, se debe utilizar estas características puesto que poco a poco serán adaptas por todos los navegadores de una manera similar.







1	2	3	4
5	6	7	8
9	10		

(a)

1	4	7	10
2	5	8	
3	6	9	

(b)

Tabla 4

Los Alimentos					
Energéticos	Plásticos		Reguladores		
Azúcares	Grasas	Proteínas		Vitaminas	
Dulces	Carne	Huevos	Legumbres	Frutas	Verduras
					

(c)

Figura 2.12: HTML5: Ejercicios de Tablas.

¿Por qué utilizar formularios HTML5?

La característica principal de utilizar formularios HTML5 es que la información es procesada de manera simple, nativa y sin la necesidad de recurrir a JavaScript para validaciones básicas. No obstante, JavaScript ha pasado a tener un peso específico en el desarrollo de sitios Web pero para tareas algo más complejas que validaciones simples. Otra característica interesante es que estamos añadiendo un valor semántico a nuestros formularios, lo cual permite obtener mejores resultados en los motores de búsqueda, así como un mayor conocimiento de los datos para poder ser procesados en diferentes etapas (análisis de lado del servidor en lugar de lado cliente). Normalmente en el desarrollo de un sitio Web la validación de formularios se realizaba en dos niveles diferentes, lado del servidor y lado de cliente. La validación a nivel de servidor trata de comprobar o validar la información introducida por el usuario mientras que, la validación a nivel de cliente se centra en comprobar que la información introducida tiene el formato correcto para poder ser analizada posteriormente por el servidor. Hasta la llegada de HTML5 la única manera de validar los formularios desde el lado del cliente era haciendo uso de JavaScript pero con la llegada de HTML5 se puede prescindir de JavaScript para este cometido y dejarlo para la programación Web de lado de cliente. De esta manera, se puede aumentar la productividad en el desarrollo de sitios Web, además de tener un código mucho más limpio. Además, se ha incorporado un comportamiento más inteligente a los formularios Web como pueden ser un calendario interactivo para un campo fecha, autocompletar o pistas en los formularios, medidas de tiempo, selectores de color, etc.

2.6.1 Estructura básica de un formulario

Los formularios (*form*) se usan para pasar datos al servidor. Están formados principalmente por controles y pueden ser de varios tipos. Los datos que se van a enviar al servidor tienen su origen en controles de entrada (*input*), por ejemplo, controles como cuadros de textos, casillas de selección, áreas de textos, campos de contraseñas, etc. Para enviar los valores se realiza a través de un botón de tipo *submit*.

Los formularios son un elemento (**<form>**) más dentro de HTML. Aunque, al ser un medio de comunicación entre el usuario y los administradores de la Web ha ganado peso en estos años. En la última versión de HTML5 se han incrementado en funcionalidades. No obstante,

muchas de los elementos establecidos en versiones anteriores siguen manteniéndose. Así que se comenzará revisando los elementos comunes que aún persisten en el tiempo relativos a los formularios.

El elemento `<form>` tiene una serie de atributos básicos que se describen a continuación:

- **action:** El atributo *action* indica el tipo de acción que va a realizar el formulario. Normalmente se dirige a un script previamente programado que procesará los datos de lado de servidor. También se puede especificar un correo electrónico donde se enviarán los datos o lanzar el gestor de correo del equipo donde se esté visualizando la Web. En el siguiente ejemplo se muestra el código relativo para especificar la acción de enviar un correo electrónico:

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

En el caso de que se quiera redirigir a un script que realice la manipulación de los datos sería el siguiente:

```
1 <form action="./admin/adduser"> </form>
```

- **method:** Mediante este atributo le indicamos al formulario el protocolo de comunicación para enviar los datos. Existen dos valores posibles: *get* y *post*. El valor *get* es el valor por defecto, en este caso se enviarán los datos en la barra de direcciones del cliente concatenado al final de la url.

```
1 http://www.carloscaballero.es/portfolio?nombre1=valor1&
  nombre2=valor2
```

Por otro lado, el valor *post* indica que el envío no se hará a través de la url, sino que viajará formando parte del cuerpo de la petición.

- **enctype:** Este atributo especifica la forma en que la información viajará a través del formulario. La forma puede ser de varios formatos, aunque el más común es el texto plano (*text/plain*).

2.6.2 Entrada de texto

Los elementos de entrada en los formularios se definen utilizando el elemento `<input>`. Este elemento dispone de un conjunto de atributos que permiten configurar los tipos de cajas y las formas de la misma. La caja vacía de un formulario se describe a continuación:

```
1 <input type="text" name="nombreCampo" />
```

En el ejemplo anterior se ha utilizado el atributo *type* el cual permite definir qué información se suministrará en la caja de texto. Algunos de los atributos que se pueden definir son los siguientes:

- **size:** Se define el tamaño de la caja. Es decir, la apariencia de la misma. En el caso de que el texto sea mayor que el tamaño se desplazará pero la caja no modificará su tamaño. El texto irá desapareciendo por la izquierda.
- **maxlength:** Especifica la cantidad máxima de caracteres que se pueden escribir en la caja.
- **value:** Permite escribir un texto predeterminado.
- **name:** Permite especificar el nombre del campo. Esto es útil para identificar fácilmente el campo a la hora de procesar la información.

```
1 <input type="text" size="20" maxlength="30" value="Saludos_Texto" name="
  nombreCampo" />
```


En el caso de que en lugar de querer que el usuario suministre un texto corto y directo se requiere tomar información textual amplia se debe utilizar el elemento **<textarea>**. Este elemento funciona de la misma manera que el input de tipo text pero con mayor tamaño en el texto. Los atributos básicos son *row* y *cols* que especifican el tamaño del elemento.

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

Finalmente, como elemento de caja de texto también se dispone de un tipo específico utilizado para la inserción de contraseñas por parte del usuario. En este caso es un tipo a utilizar en el elemento **<input>**.

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

2.6.3 Selección de opciones

Las listas de opciones son ampliamente utilizadas en los formularios puesto que permiten ofrecer al usuario de un modo cerrado las diferentes opciones. Esto permite que los errores en introducción de datos sean mínimos y que el usuario fácilmente con solo selección utilizando el ratón encuentre una opción que se adapte a sus necesidades. Existen varios métodos para mostrar varias opciones de selección en los formularios Web. A continuación se van a describir los más relevantes.

El elemento encargado de crear las listas de selección es **<select>**, dentro de este elemento se deben especificar cada una de las posibles opciones con el elemento **<option>**. A continuación se describe un ejemplo en el cual se está solicitando el medio de transporte para buscar billetes en una famosa Web.

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

Además, estos elementos tienen una serie de atributos que permiten un mayor control sobre el elemento seleccionado. Algunos de estos atributos son://

- **size:** Especifica el tamaño de la lista desplegable. Si se selecciona el valor 3 en el atributo *size* se muestran 3 opciones a los usuarios.
- **multiple:** Esta opción permite que se puedan seleccionar más de una opción simultáneamente. Es decir, haciendo uso de las teclas ctrl+shift se permiten seleccionar varias de las opciones posibles.
- **selected:** Esta opción permite especificar qué opción debe aparecer seleccionada por defecto. Esta opción puede aparecer en varias opciones siempre y cuando existan los atributos *size* y *multiple*. En caso contrario, aparecerá seleccionada la última de las opciones.
- **value:** Este valor es muy importante para automatizar la información puesto que es el valor que se transmite al seleccionar la opción para que la parte del sistema que procesa la información pueda aplicar ciertas reglas.

```
1 <select name="transporte" size="3" multiple>
2   <option value="1">Coche</option>
3   <option value="2" selected="true">Avión</option>
4   <option value="3">Tren</option>
5 </select>
```

En la Figura 2.13 se muestran varios ejemplos de aplicación de este elemento del formulario.



Figura 2.13: Formularios HTML5: Listas de selección.

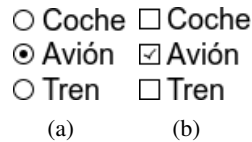


Figura 2.14: Formularios HTML5: (a) radio (b) checkbox.

Selección de opciones: Radio

Una especialización del elemento `<input>` que permite crear una lista de selección con botones radio. Estos botones permiten que se seleccione en primer lugar una sola opción de varias posibles. Esta especialización se consigue configurando el atributo `type` como del tipo `radio`. En el caso de que se quiera seleccionar una opción por defecto en lugar de utilizar el atributo `selected` se utiliza el atributo `checked`. En el siguiente código se muestra la creación de este elemento y en la Figura 2.14(a) se puede ver el resultado.

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

Selección de opciones: Checkbox

Al igual que para radio, existe una especialización denominada `checkbox` que permite crear cajas de selección múltiple. El funcionamiento de este elemento es idéntico al de radio. En el siguiente código se muestra un ejemplo de aplicación y en la Figura 2.14(b) el resultado de su aplicación.

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

2.6.4 Envío y reinicio

Es necesario pulsar un botón para enviar información, a menos que se esté haciendo uso de AJAX para la comunicación de la información, los botones pueden tener dos funcionalidades: La de envío o reinicio.

Los botones de envío son un tipo específico del elemento `<input>`, en este caso el tipo es `submit`. Además, se dispone del atributo `value` que permite especificar el texto que aparecerá en el botón. Por otro lado, si lo que se desea es que se reinicie toda la información del funcionario se puede utilizar el tipo `reset`. En el siguiente fragmento de código se muestra el uso de los dos tipos de botones. El resultado de estos dos botones es mostrado en la Figura 2.15.

```
1 <input type="submit" value="Enviar_información">
2 <input type="reset" value="Borrar_información">
```

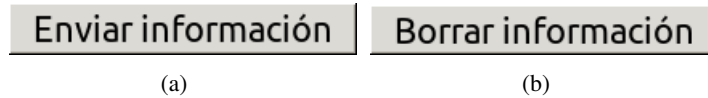


Figura 2.15: Formularios HTML5: (a) Submit (b) reset

2.6.5 Otros elementos

En esta sección se describen algunos elementos de interés a la hora de crear formularios Web. En concreto se presentará el modo de transmitir información de modo oculto en la capa de presentación (si el usuario observa el código Web de la página podrá visualizar los datos), para configurar botones con imágenes y finalmente el modo de poder enviar ficheros.

Tipo oculto: hidden

En el caso de que los desarrolladores quieran enviar información oculta al sistema que procesa la información se puede crear un elemento en el formulario oculto en la capa de presentación, es decir, no aparece en el interfaz Web del tipo *hidden*. Hay que recordar que estos datos no deben ser en ningún caso sensibles puesto que son fácilmente localizables por cualquier usuario con tan solo visualizar el código fuente.

```
1 <form action="mailto:_nombredelcorreo@correo.com_" method="post" enctype=
  "text/plain" name="mihidden">
2 <input type="hidden" name="opcion" value="si">
3 </form>
```

Tipo botón: image

En el caso que queramos sustituir el botón enviar por una imagen, algo muy para encajar bien los diseños gráficos se dispone de un tipo específico denominado *image*. En este caso sustituye el aspecto del botón de tipo *submit* pero la funcionalidad es exactamente la misma. A continuación se muestra un fragmento de código donde se aplica este tipo de elementos.

```
1 <form action="mailto:nombredelcorreo@correo.com" method="post" enctype="
  text/plain" name="image">
2 <input type="image" name="boton" src="http://html.hazunaweb.com/html/
  imagenes/boton.jpg" align="middle">
3 </form>
```

Envío de ficheros: file

Finalmente otro punto que puede ser interesante es la posibilidad de subir un fichero a través del formulario. En este caso existe un tipo específico denominado *file* pero además se debe modificar el tipo de información del formulario para conocer que no será texto plano solamente sino datos utilizando el atributo *enctype*.

```
1 <form method="post" enctype="multipart/form-data">
2 <input type="file" size=60 name="file1"><br/>
3 <input type="file" size=60 name="file2"><br/>
4 <input type="submit" value="subir">
5 </form>
```

2.6.6 Bloques temáticos y etiquetas

Un conjunto de elementos importantes para la estructuración de los formularios son la agrupación por bloques temáticos y las etiquetas de los diferentes elementos. Este conjunto

de elementos no es imprescindible para construir formularios pero si son muy recomendables puesto que ayudan a estructurar de un modo adecuado los formularios.

Bloque temático: Fieldset

El elemento `<fieldset>` permite agrupar los elementos de un formulario. Por defecto lo que se crea es un recuadro que rodea a los elementos del formulario que están comprendidos entre el elemento `<fieldset>`.

```
1 <fieldset>
2   Elemento de formulario: <input type="text" name="elemento1"><br/>
3   Otro elemento: <input type="text" name="otro">
4 </fieldset>
```

Etiquetas de grupos: Legend

El elemento `<legend>` permite asignar un nombre a un grupo creado por el elemento `<fieldset>`. Por defecto añade una nota aclaratoria del grupo temático del formulario. Un ejemplo sencillo de aplicación es el mostrado a continuación:

```
1 <form>
2   <fieldset>
3     <legend align="right">Datos personales</legend>
4     Nombre: <input type="text" name="nombre"><br/>
5     Edad: <input type="text" name="edad" size="2"><br/>
6     Dirección: <input type="text" name="direccion">
7   </fieldset>
8   <br/>
9   <fieldset>
10    <legend align="right">Datos de tu ordenador</legend>
11    Modelo de ordenador: <input type="text" name="modelo"><br>
12    Sistema que te da el problema:
13    <select>
14      <option value=cpu>CPU
15      <option value=impresora>Impresora
16    </select>
17  </fieldset>
18  <br/>
19  <fieldset>
20    <legend align="right">Descripción del problema</legend>
21    <textarea cols="55" rows="8" name="descripcion"></textarea>
22  </fieldset>
23 </form>
```

El aspecto del código anterior se muestra en la Figura 2.16.

Etiquetas de elementos: Label

El elemento `<label>` relaciona la etiqueta de cada uno de los elementos del formulario a través de un atributo denominado *for*. De esta manera la etiqueta para el campo textual input llamado edad será *Edad* tal y como muestra el siguiente ejemplo:

```
1 <label for="edad">Edad</label> <input type="text" name="edad" id="edad">
```

Ejercicio 2.17 Busca los tipos de formatos que se pueden enviar con el atributo *enctype*.

Ejercicio 2.18 Con los elementos presentados, crea un formulario de contacto para tu proyecto de Web anterior. Revisa la documentación oficial de W3C relativa al elemento input: http://www.w3schools.com/tags/tag_input.asp.

Figura 2.16: Formularios HTML5: Legend y Label

Ejercicio 2.19 Utilice la herramienta visual *pForm* para generar un formulario. Examine el código generado. ¿Es un código HTML5? ¿Le es útil para generar formularios?

2.6.7 Novedades de los formularios HTML5

Son muchas las novedades que presenta HTML5 enfocadas a suministrar mayor semántica a los formularios, y por tanto, control de la información que se transmite. En resumen, en esta sección se presentan las novedades que incorpora HTML5 entre las que destacan 16 atributos, 5 elementos y 13 tipos de atributos.

Los elementos y atributos más utilizados como *placeholder*, *autofocus*, *email*, *url*, *datalist* o *required* están soportadas por la mayoría de navegadores Web actuales. No obstante, poco a poco, todas estas novedades se irán incorporando a todos los navegadores.

Atributos

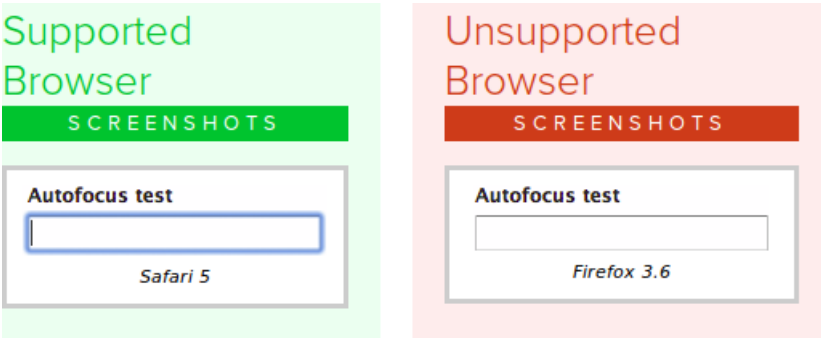
Los nuevos atributos que incorpora el estándar HTML5 al elemento `<input>` permiten un mayor control y una riqueza en el interfaz de usuario amolde a los tiempos actuales. A continuación se describen las características de estos nuevos atributos:

1. **Autofocus:** Se produce un foco en el elemento `<input>` que tenga asociado éste valor al cargar la página.

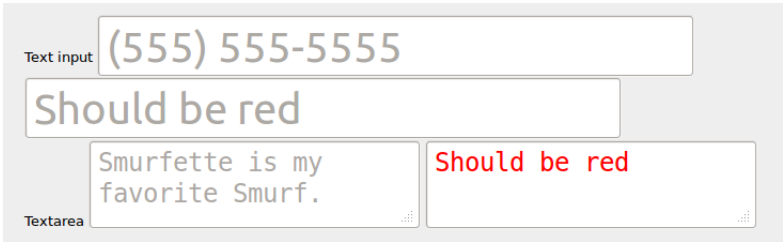
```
1 <form>
2   <input type="text">
3   <input autofocus type="text">
4 </form>
```

2. **Placeholder:** Muestra una *pista* de lo que el usuario debe ingresar en el elemento `<input>`. Antiguamente este efecto se conseguía haciendo uso del atributo *value*. No obstante, la ventaja que tiene este nuevo atributo es que cuando el usuario comienza a escribir en el elemento desaparece la pista.

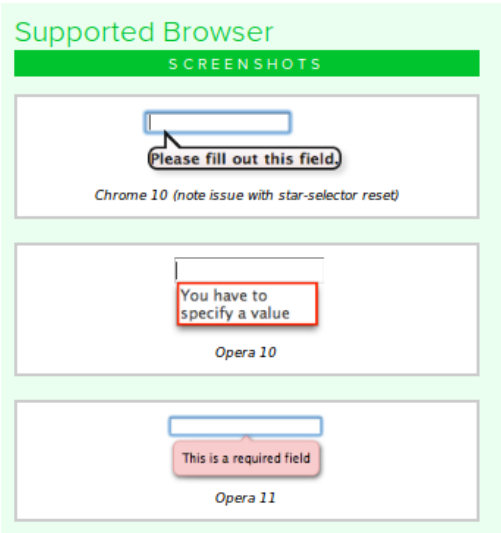
```
1 <form>
2   <input type="text" placeholder="(555)_555-5555">
3   <textarea placeholder="Smurfberries"></textarea>
4 </form>
```



(a)



(b)



(c)



interaction		Advance
interactions	108,000,000 results	Preferen
interaction design	124,000,000 results	Languag
interaction recruitment	1,990,000 results	
interaction value analysis	45,300,000 results	results. !
interaction diagram	8,630,000 results	
interactionism	483,000 results	
interaction solutions magazine	3,710,000 results	
interaction associates	8,030,000 results	
interactionist	406,000 results	
interactionist perspective	271,000 results	

(d)

Figura 2.17: Nuevos tipos de campos en los formularios HTML5: (a) Autofocus (b) placeholder (c) required (d) autocomplete.

3. **Required:** Este atributo determina si el elemento debe ser obligatorio o no por parte del usuario. Este atributo es ampliamente utilizado con otros nuevos elementos como *max/min*, *email*, *pattern* para minimizar en gran medida el uso del lenguaje JavaScript.

```
1 <form id="formulario_contacto" method="get">
2   <input type="text" id="nombre" name="nombre" required >
3   <input type="email" id="email" name="email" required>
4   <input type="submit" id="submit" value="SEND">
5 </form>
```

4. **Autocomplete:** Este atributo permite especificar si un elemento puede o no ser autocompletado por el navegador basado en las entradas previas del usuario.

```
1 <form id="formulario_contacto" method="get">
2   <label>Nombre:</label> <input type="text" autocomplete=off>
3   <label>Email:</label> <input type="email" autocomplete=on>
4   <input type="submit" id="submit" value="SEND">
5 </form>
```

5. **Max / min / step:** Permite delimitar el rango de valores numéricos para un rango: Máximo, mínimo y múltiplos de un rango.

```
1 <form>
2   <input type="number" min="0" max="100" step="5">
3 </form>
```

6. **Dirname:** Este atributo define la dirección en que se introduce el texto en un input. Por ejemplo, de derecha a izquierda se definirá con el valor *rtl* y para que la dirección sea de izquierda a derecha se utiliza el valor *ltr*.

```
1 <form action="demo_form.asp">
2   <p><label>Comentario: </label><input type=text name="comment"
3     dirname="rtl"></p>
4   <p><input type=text name="comment2" dirname="ltr"></p>
5 </form>
```

7. **Accept:** Permite que solo se pueda cargar un tipo de archivo determinado.

```
1 <form>
2   <input type="file" accept="image/*">
3 </form>
```

8. **Multiple:** Permite seleccionar múltiples archivos para ser cargados de una sola vez en el formulario, en lugar de tener que ir seleccionándolos uno a uno como sucedía anteriormente.

```
1 <form>
2   <input type="file" multiple>
3 </form>
```

9. **Form:** Permite colocar un elemento de formulario en cualquier parte de la página, y no solamente dentro de la etiqueta form y que siga siendo procesada por el formulario (antes se podía poner cualquier elemento de formulario fuera de la etiqueta form pero no era procesado). Además, permite la característica de asociar el mismo elemento de formulario a varios formularios diferentes que existan en la misma página Web.


```

1 <form id="formulario_contacto" method="get">
2   <label>Nombre:</label>
3   <input type="text" id="nombre" name="nombre">
4   <label>Email:</label>
5   <input type="email" id="email" name="email">
6   <input type="submit" id="submit" value="SEND">
7 </form>
8 <textarea id="comments" form="formulario_contacto"></textarea>

```

10. **Pattern:** Este atributo es muy interesante puesto que permite validar un elemento en base a una expresión regular. En la Web <http://html5pattern.com/> se muestran patrones utilizados para los formularios input.

```

1 <form id="formulario_contacto" method="get">
2   <label>Código postal:</label>
3   <input type="text" pattern="(0[1-9]|5[0-2])|([1-4][0-9])[0-9]{3}">
4   <label>Tarjeta de crédito </label>
5   <input type="text" pattern="[0-9]{13,16}">
6   <input type="submit" id="submit" value="SEND">
7 </form>

```

11. **Novalidate:** Evita la validación del elemento al enviar el formulario, es útil si este atributo se incorpora en el formulario dinámicamente que por alguna razón de programación no se quieran validar los campos del formulario.

```

1 <form action="demo_form.asp">
2   <p><label>Comentario: </label><input type="text" name="comment"
3     novalidate="yes"></p>
4 </form>

```

12. **Formaction, formenctype, formmethod, formnovalidate, formtarget:** Permite sobre-escribir el comportamiento, codificación, método de envío, la opción de novalidate y la ventana de destino por defecto del formulario. Esto permite que dependiendo qué botón se utilice se puede tratar el formulario en diferentes páginas usando solo un formulario, algo complicado en HTML4.

```

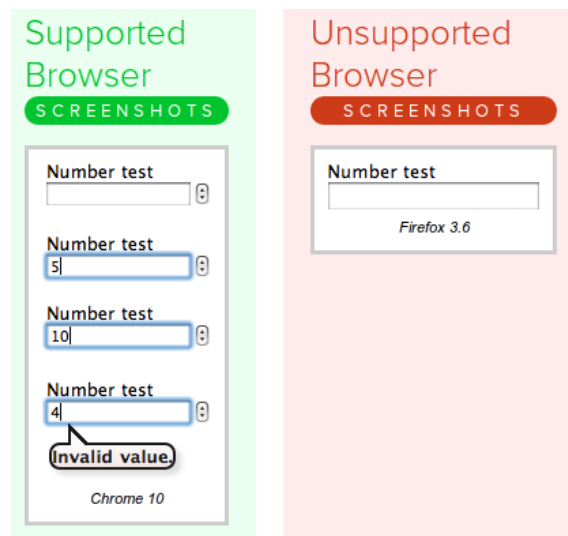
1 <form action=http://example.com method=post target="_self" enctype="
  multipart/form-data">
2   <input type="submit" value="Submit" formaction="http://wufoo.com">
3   <input type="submit" value="Submit" formmethod="get">
4   <input type="submit" value="Submit" formtarget="_target">
5   <input type="submit" value="Submit" formenctype="application/x-www-
  form-urlencoded">
6   <input type="submit" formnovalidate value="Save">
7   <input type="submit" value="Submit">
8 </form>

```

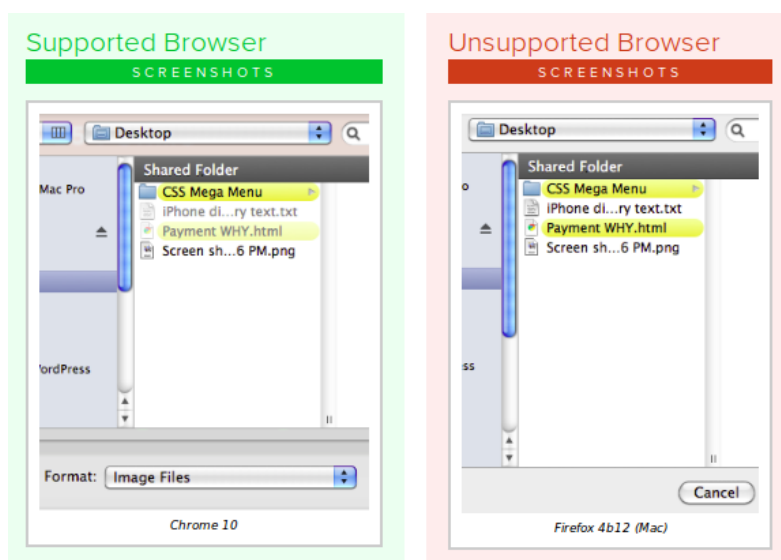
Tipos de campos

Se agregan al elemento **<input>** una gran variedad de nuevos tipos de datos. Entre ellos destacan los siguientes:

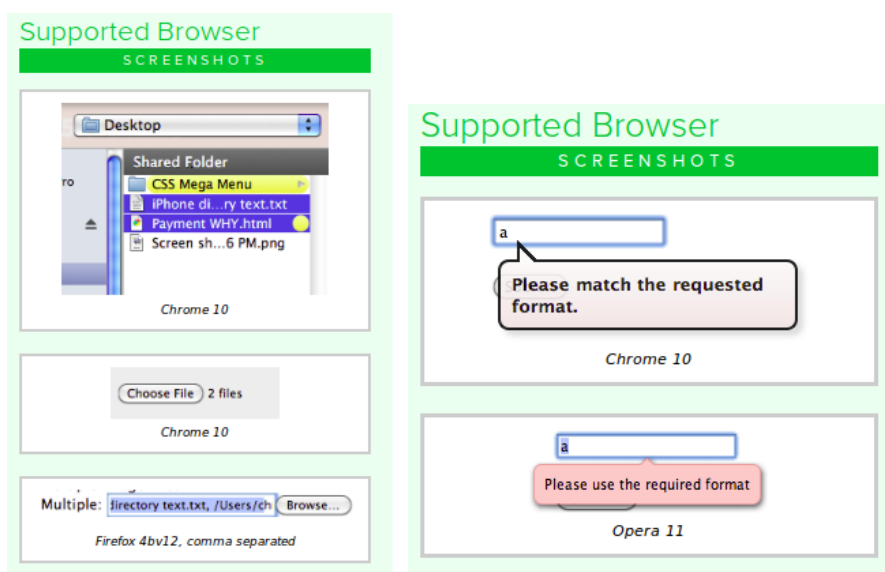
1. **Tel:** Es utilizado para definir números de teléfono. Este atributo, añade valor semántico pero no se comprueba realmente el formato del teléfono, para realizar esto se debe complementar con el atributo *pattern*. Esto es así debido a que los formatos de los teléfonos son diferentes según el país. Algunos navegadores al detectar el tipo de campo teléfono despliegan el teclado numérico en lugar del textual.



(a)



(b)



(c)

(d)

Figura 2.18: Nuevos tipos de campos en los formularios HTML5: (a) Min/Max/Step (b) accept (c) multiple (d) pattern .

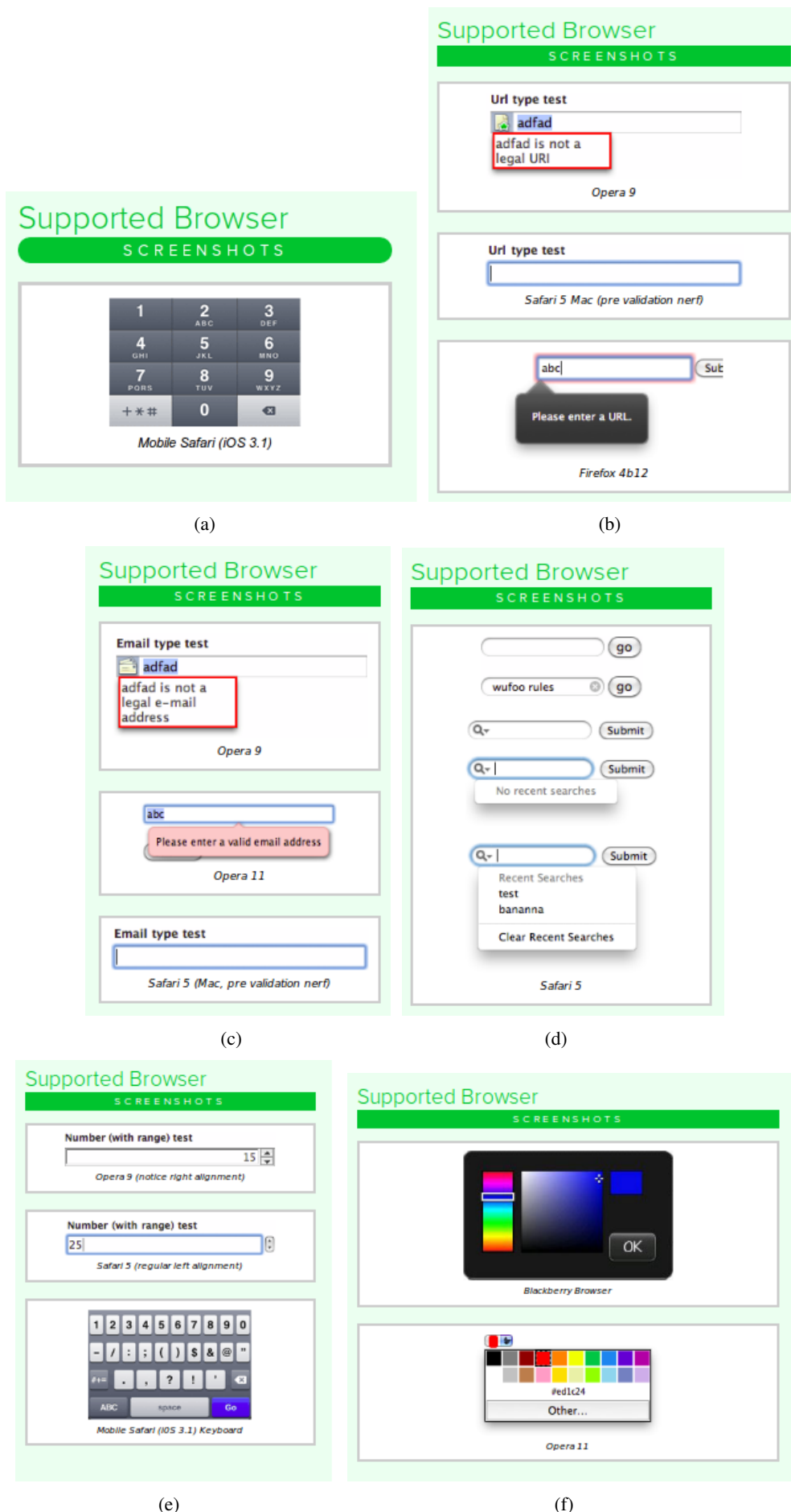


Figura 2.19: Nuevos tipos de campos en los formularios HTML5: (a) Teléfonos (b) URL (c) email (d) búsquedas (e) números (f) colores.

```

1 <form>
2   <input type="tel" name="tel">
3   <input type="submit">
4 </form>

```

2. **url:** Este tipo es utilizado para ingresar direcciones Web absolutas.

```

1 <form>
2   <input type=url name=url>
3   <input type=submit>
4 </form>

```

3. **Email:** Este tipo es utilizado para direcciones de correo electrónico.

```

1 <form>
2   <input type=email name=email>
3   <input type=submit>
4 </form>

```

4. **Search:** Al agregar este tipo se sugiere el ingreso de un determinado texto en un input.

```

1 <form>
2   <input type=search>
3   <input type=search results=5 placeholder="results_only">
4   <input type=search results=5 autosave=some_unique_value>
5   <input type=submit value=Search>
6 </form>

```

5. **Number:** Es utilizado para formato de valores numéricos.

```

1 <form>
2   <input type=number name=num>
3   <input type=submit>
4 </form>

```

6. **Color:** Es utilizado para desplegar una paleta de colores y poder seleccionar uno de ellos.

```

1 <form id="form-demo">
2   <input type=color>
3 </form>

```

7. Formatos para fechas:

- **Date:** Es utilizado para formato de fecha sin zona horaria (yyyy-mm-dd).
- **Time:**
- **Datetime:** Es utilizado para formatos de fecha y hora con zona horaria UTC (yyyy-mm-dd HH:MM).
- **Month:** Es utilizado para formato de mes y año sin zona horaria (yyyy-mm).
- **Week:** Es utilizado para formato de semana del año (yyyy-mmW).
- **Datetime-local:** Es utilizado para formato de fecha y hora sin zona horaria (yyyy-mm-dd HH:MM).

```

1 <label for="myDate">date</label><input name="myDate" type="date">
2 <label for="myDateTime">datetime</label><input name="myDateTime"
  type="datetime-local">
3 <label for="myWeek">week</label><input name="myWeek" type="week">
4 <label for="myM">month</label><input name="myM" type="month">
5 <label for="myTime">time</label><input name="myTime" type="time">

```

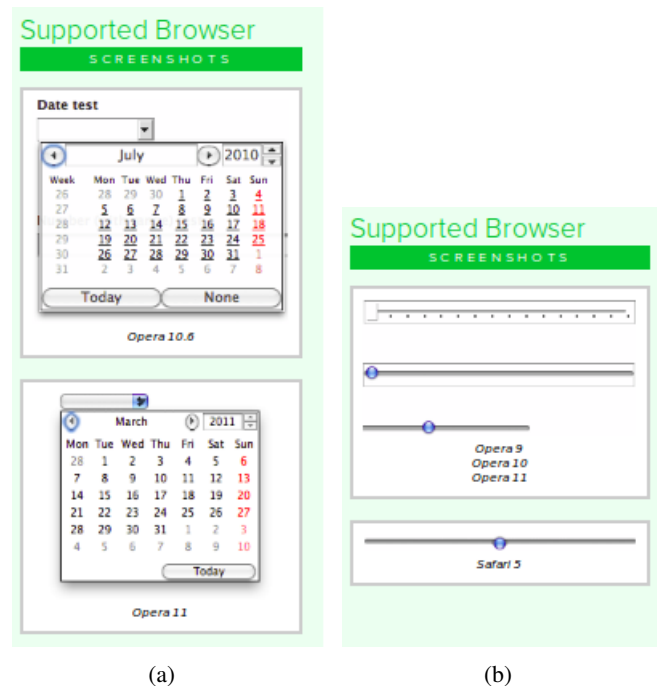


Figura 2.20: Nuevos tipos de campos en los formularios HTML5: (a) Fechas (b) rangos.

8. **Range:** Es utilizado para formato de valores numéricos en un rango, normalmente utilizado para ser controlado con una barra deslizadora. Algunos atributos interesantes del tipo rango son:

- **value:** El cual es un atributo común para el elemento input. Este valor puede ser cualquier número flotante.
- **min:** El cual es el valor mínimo del rango. Por defecto este valor es 0.
- **max:** El cual es el valor máximo del rango. Por defecto este valor es 100.
- **step:** Este es el factor de paso para la barra deslizante, el valor por defecto es 1 y solo están permitidos valores numéricos enteros.
- **list:** Es una lista de valores posibles. El elemento **<Datalist>** puede ser incorporado al tipo range, sin embargo, aún ningún navegador ha implementado esta característica.

```
1 <label for="myRange">range</label> <input name="myRange" type="range"
    min="1" max="11" value="7" step="1">
```

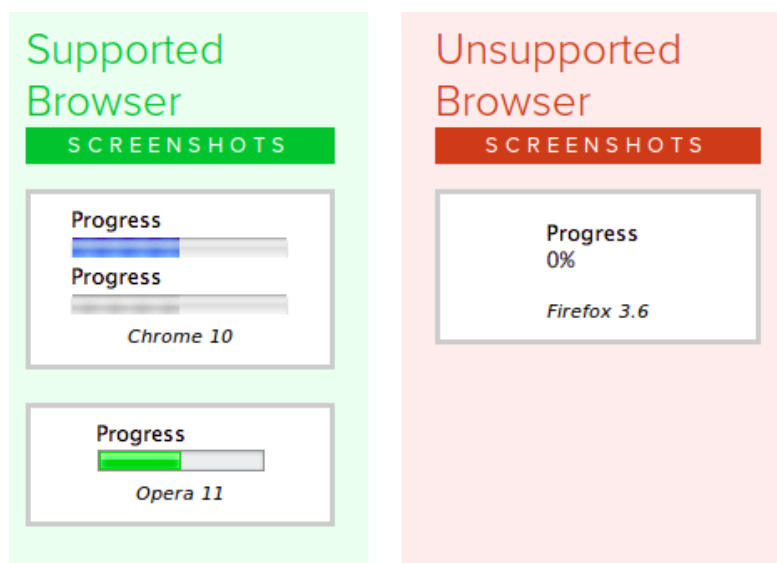
Elementos

Además de los nuevos atributos y nuevos tipos también se presentan nuevos elementos en HTML5. En principio se han diseñado 5 nuevos elementos de formularios, aunque todos no están aún estandarizados por los navegadores.

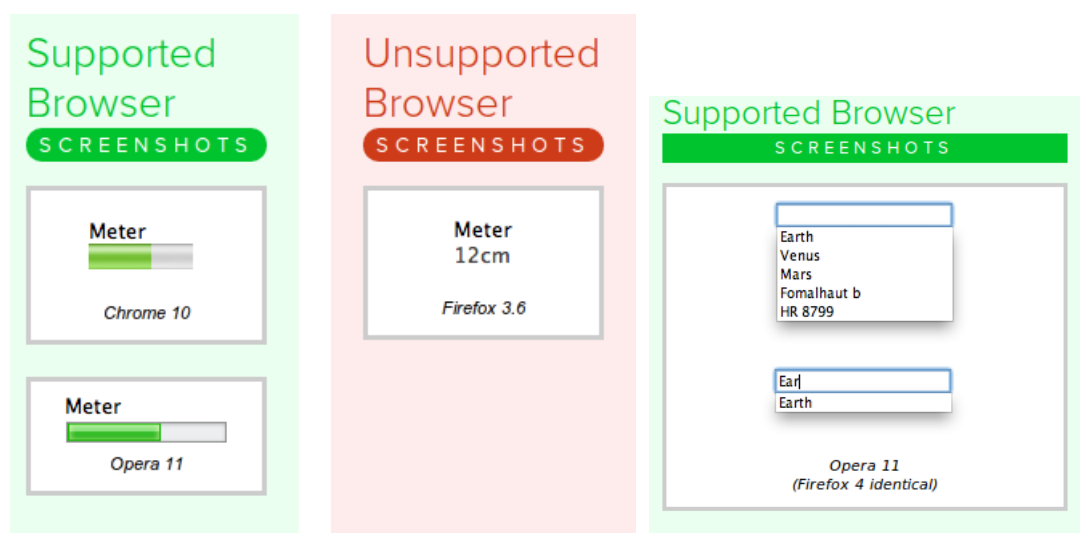
1. **Progress:** Representa el grado de progreso de una tarea o acción. Por ejemplo, representar la carga de una imagen o un fichero. Es necesario JavaScript para realizar la actualización de los datos.

```
1 <progress value="22" max="100"></progress>
```

2. **Meter:** Representa una escala para medidas tales como temperatura o distancias. Al igual que el elemento **<progress>** es necesario JavaScript para crear la animación.

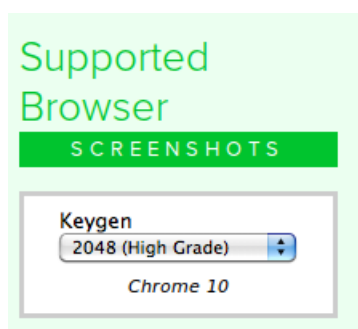


(a)



(b)

(c)



(d)



(e)

Figura 2.21: Nuevos elementos en los formularios HTML5: (a) Progress (b) meter (c) datalist (d) keygen (e) output.

```
1 <meter value="2" min="0" max="10">2 out of 10</meter><br/>
2 <meter value="0.6">60%</meter>
```

3. **Datalist:** Representa una lista de valores y se puede asociar con un elemento `<input>` de texto. De esta manera, al centrar el foco en el elemento `<input>` aparecerá una lista desplegable mostrando el contenido del elemento `datalist`.

```
1 <datalist id="lista">
2   <option value="0">
3     <option value="2">
4     <option value="4">
5     <option value="6">
6     <option value="8">
7     <option value="10">
8 </datalist>
9 <datalist id="listaurls">
10  <option value="http://www.ayudaenlaweb.com" label="Ayuda_en_la_Web">
11    <option value="http://lineadecodigo.com" label="Línea_de_Código">
12  <option value="http://www.dudasprogramacion.com" label="Dudas_de_Programación">
13 </datalist>
14 <input type="range" min="0" max="10" value="3" step="1" id="mislider" list="lista">
15 <input type="url" id="misurl" list="listaurls" size="100">
```

4. **Keygen:** Este elemento genera un par de claves de control privada que se guarda en local y pública que se envía al servidor. A continuación se muestra el formulario que genera la clave pública y se envía a la siguiente página para ser procesada a través de PHP. Se muestra como se recibe dicha clave. La clave privada es almacenada en el navegador.

```
1 <form action="formu.php" method="get">
2   User: <input type="text" name="usuario"><br/>
3   Pass: <keygen name="clave"><br/>
4   <input type="submit" value="Enviar">
5 </form>
```

```
1 <?php
2 if ($_POST['clave'])
3 echo "clave_pública_enviada:_" . $_POST['clave'];
4 ?>
```

5. **Output:** Este nuevo elemento se utiliza para realizar cálculos entre campos. Por ejemplo, la multiplicación de 2 números de distintos `inputs`.

```
1 <form onsubmit="return_false" id="form-daddy" oninput="document.
   getElementById('o').innerHTML+=parseFloat(document.
   getElementById('a').value)+parseFloat(document.getElementById('
   b').value)">
2 <input name=a type=number id=a> +
3 <input name=b type=number id=b> =
4 <input id="submitButton" type="submit" value="add">
5 <output for="form-daddy" name="o" id="o"></output>
```

Ejercicio 2.20 Actualiza el formulario de contacto de tu proyecto Web a la nueva especificación de HTML.

Ejercicio 2.21 Busca el tipo de formatos soportados por el atributo *accept*. Comprueba con un tipo de formato diferente a las imágenes.

Ejercicio 2.22 Escribe otro par de formularios en tu proyecto Web para que encajen adecuadamente. Prueba las nuevas características de autovalidación sin necesidad de JavaScript.

Ejercicio 2.23 Construye una calculadora con las operaciones suma, resta, multiplicación y división utilizando el elemento `<output>`.

Ejercicio 2.24 Observa que los nuevos elementos del navegador no se visualizan de manera adecuada en todos los navegadores. Construye tu propia tabla de los elementos que se visualizan satisfactoriamente según qué navegadores y extrae algunas conclusiones de su adaptación a los navegadores contemporáneos.

Ejercicio 2.25 Busca información sobre las bibliotecas *Modernizr* y *jQueryUI*. ¿Para qué sirven?

Ejercicio 2.26 Adapta tu formulario anterior a esta biblioteca para que te permita detectar las funcionalidades adecuadas según los navegadores. Para ello, vas a hacer uso de las bibliotecas *Modernizr* y *jQueryUI*, revisa el siguiente tutorial en el que se explica paso a paso: <http://net.tutsplus.com/tutorials/html-css-techniques/how-to-build-cross-browser-html5-forms/>. En el Capítulo relativo a la interacción con el interfaz de usuario se presentará la biblioteca jQueryUI.

Ejercicio 2.27 Adapta los formularios de tu proyecto Web para que sean compatibles para cualquier navegador Web.

