El gran libro de HTML5, CSS3 y Javascript

Juan Diego Gauchat

marcombo
ediciones técnicas

El gran libro de HTML5, CSS3 y Javascript

J. D. Gauchat

marcombo
ediciones técnicas

Índice

Introducción

HTML5 no es una nueva versión del antiguo lenguaje de etiquetas, ni siquiera una mejora de esta ya antigua tecnología, sino un nuevo concepto para la construcción de sitios web y aplicaciones en una era que combina dispositivos móviles, computación en la nube y trabajos en red.

Todo comenzó mucho tiempo atrás con una simple versión de HTML propuesta para crear la estructura básica de páginas web, organizar su contenido y compartir información. El lenguaje y la web misma nacieron principalmente con la intención de comunicar información por medio de texto.

El limitado objetivo de HTML motivó a varias compañías a desarrollar nuevos lenguajes y programas para agregar características a la web nunca antes implementadas. Estos desarrollos iniciales crecieron hasta convertirse en populares y poderosos accesorios. Simples juegos y bromas animadas pronto se transformaron en sofisticadas aplicaciones, ofreciendo nuevas experiencias que cambiaron el concepto de la web para siempre.

De las opciones propuestas, Java y Flash fueron las más exitosas; ambas fueron masivamente adoptadas y ampliamente consideradas como el futuro de Internet. Sin embargo, tan pronto como el número de usuarios se incrementó e Internet pasó de ser una forma de conectar amantes de los ordenadores a un campo estratégico para los negocios y la interacción social, limitaciones presentes en estas dos tecnologías probaron ser una sentencia de muerte.

El mayor inconveniente de Java y Flash puede describirse como una falta de integración. Ambos fueron concebidos desde el principio como complementos (plug-ins), algo que se inserta dentro de una estructura pero que comparte con la misma solo espacio en la pantalla. No existía comunicación e integración alguna entre aplicaciones y documentos.

La falta de integración resultó ser crítica y preparó el camino para la evolución de un lenguaje que comparte espacio en el documento con HTML y no está afectado por las limitaciones de los plug-ins. Javascript, un lenguaje interpretado incluido en navegadores, claramente era la manera de mejorar la experiencia de los usuarios y proveer funcionalidad para la web. Sin embargo, después de algunos años de intentos fallidos para promoverlo y algunos malos usos, el mercado nunca lo adoptó plenamente y pronto su popularidad declinó. Los detractores tenían buenas razones para oponerse a su adopción. En ese momento, Javascript no era capaz de reemplazar la funcionalidad de Flash o Java. A pesar de ser evidente que ambos limitaban el alcance de las aplicaciones y aislaban el contenido web, populares funciones como la reproducción de video se estaban convirtiendo en una parte esencial de la web y solo eran efectivamente ofrecidas a través de estas tecnologías.

A pesar del suceso inicial, el uso de Java comenzó a declinar. La naturaleza compleja del lenguaje, su evolución lenta y la falta de integración disminuyeron su importancia hasta el punto en el que hoy día no es más usado en aplicaciones web de importancia. Sin Java, el mercado volcó su atención a Flash. Pero el hecho de que Flash comparte las mismas características básicas que su competidor en la web lo hace también susceptible de correr el mismo destino.

Mientras esta competencia silenciosa se llevaba a cabo, el software para acceder a la web continuaba evolucionando. Junto con nuevas funciones y técnicas rápidas de acceso a la red, los navegadores también mejoraron gradualmente sus intérpretes Javascript. Más potencia trajo más oportunidades y este lenguaje estaba listo para aprovecharlas.

En cierto punto durante este proceso, se hizo evidente para algunos desarrolladores que ni Java o Flash podrían proveer las herramientas que ellos necesitaban para crear las aplicaciones demandadas por un número creciente de usuarios. Estos desarrolladores, impulsados por las mejoras otorgadas por los navegadores, comenzaron a aplicar Javascript en sus aplicaciones de un modo nunca visto. La innovación y los increíbles resultados obtenidos llamaron la atención de más programadores. Pronto lo que fue llamado la "Web 2.0" nació y la percepción de Javascript en la comunidad de programadores cambió radicalmente.

Javascript era claramente el lenguaje que permitía a los desarrolladores innovar y hacer cosas que nadie había podido hacer antes en la web. En los últimos años, programadores y diseñadores web alrededor del mundo surgieron con los más increíbles trucos para superar las limitaciones de esta tecnología y sus iniciales deficiencias en portabilidad. Gracias a estas nuevas implementaciones, Javascript, HTML y CSS se convirtieron pronto en la más perfecta combinación para la necesaria evolución de la web.

HTML5 es, de hecho, una mejora de esta combinación, el pegamento que une todo. HTML5 propone estándares para cada aspecto de la web y también un propósito claro para cada una de las tecnologías involucradas. A partir de ahora, HTML provee los elementos estructurales, CSS se encuentra concentrado en cómo volver esa estructura utilizable y atractiva a la vista, y Javascript tiene todo el poder necesario para proveer dinamismo y construir aplicaciones web completamente funcionales.

Las barreras entre sitios webs y aplicaciones finalmente han desaparecido. Las tecnologías requeridas para el proceso de integración están listas. El futuro de la web es prometedor y la evolución y combinación de estas tres tecnologías (HTML, CSS y Javascript) en una poderosa especificación está volviendo a Internet la plataforma líder de desarrollo. HTML5 indica claramente el camino.

> **IMPORTANTE:** En este momento no todos los navegadores soportan HTML5 y la mayoría de sus funciones se encuentran actualmente en estado de desarrollo. Recomendamos leer los capítulos y ejecutar los códigos con las últimas versiones de Google Chrome y Firefox. Google Chrome ya implementa muchas de las características de HTML5 y además es una buena plataforma para pruebas. Por otro lado, Firefox es uno de los mejores navegadores para desarrolladores y también provee total soporte para HTML5.

Capítulo 1
Documentos HTML5

1.1 Componentes básicos

HTML5 provee básicamente tres características: estructura, estilo y funcionalidad. Nunca fue declarado oficialmente pero, incluso cuando algunas APIs (Interface de Programación de Aplicaciones) y la especificación de CSS3 por completo no son parte del mismo, HTML5 es considerado el producto de la combinación de HTML, CSS y Javascript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y Javascript hace el resto que (como veremos más adelante) es extremadamente significativo.

Más allá de esta integración, la estructura sigue siendo parte esencial de un documento. La misma provee los elementos necesarios para ubicar contenido estático o dinámico, y es también una plataforma básica para aplicaciones. Con la variedad de dispositivos para acceder a Internet y la diversidad de interfaces disponibles para interactuar con la web, un aspecto básico como la estructura se vuelve parte vital del documento. Ahora la estructura debe proveer forma, organización y flexibilidad, y debe ser tan fuerte como los fundamentos de un edificio.

Para trabajar y crear sitios webs y aplicaciones con HTML5, necesitamos saber primero cómo esa estructura es construida. Crear fundamentos fuertes nos ayudará más adelante a aplicar el resto de los componentes para aprovechar completamente estas nuevas tecnologías.

Por lo tanto, empecemos por lo básico, paso a paso. En este primer capítulo aprenderá cómo construir una plantilla para futuros proyectos usando los nuevos elementos HTML introducidos en HTML5.

> **Hágalo usted mismo:** Cree un archivo de texto vacío utilizando su editor de textos favorito para probar cada código presentado en este capítulo. Esto lo ayudará a recordar las nuevas etiquetas HTML y acostumbrarse a ellas.
>
> **Conceptos básicos:** Un documento HTML es un archivo de texto. Si usted no posee ningún programa para desarrollo web, puede simplemente utilizar el Bloc de Notas de Windows o cualquier otro editor de textos. El archivo debe ser grabado con la extensión **`.html`** y el nombre que desee (por ejemplo, **`micodigo.html`**).

IMPORTANTE: Para acceder a información adicional y a los listados de ejemplo, visite nuestro sitio web *www.minkbooks.com*.

1.2 Estructura global

Los documentos HTML se encuentran estrictamente organizados. Cada parte del documento está diferenciada, declarada y determinada por etiquetas específicas. En esta parte del capítulo vamos a ver cómo construir la estructura global de un documento HTML y los nuevos elementos semánticos incorporados en HTML5.

<!DOCTYPE>

En primer lugar necesitamos indicar el tipo de documento que estamos creando. Esto en HTML5 es extremadamente sencillo:

```
<!DOCTYPE html>
```

Listado 1-1. *Usando el elemento <doctype>.*

IMPORTANTE: Esta línea debe ser la primera línea del archivo, sin espacios o líneas que la precedan. De esta forma, el modo estándar del navegador es activado y las incorporaciones de HTML5 son interpretadas siempre que sea posible, o ignoradas en caso contrario.

Hágalo usted mismo: Puede comenzar a copiar el código en su archivo de texto y agregar los próximos a medida que los vamos estudiando.

<html>

Luego de declarar el tipo de documento, debemos comenzar a construir la estructura HTML. Como siempre, la estructura tipo árbol de este lenguaje tiene su raíz en el elemento **<html>**. Este elemento envolverá al resto del código:

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

</html>
```

Listado 1-2. *Usando el elemento <html>.*

El atributo **lang** en la etiqueta de apertura **<html>** es el único atributo que necesitamos especificar en HTML5. Este atributo define el idioma humano del contenido del documento que estamos creando, en este caso **es** por español.

> **Conceptos básicos:** HTML usa un lenguaje de etiquetas para construir páginas web. Estas etiquetas HTML son palabras clave y atributos rodeados de los signos mayor y menor (por ejemplo, **<html lang="es">**). En este caso, **html** es la palabra clave y **lang** es el atributo con el valor **es**. La mayoría de las etiquetas HTML se utilizan en pares, una etiqueta de apertura y una de cierre, y el contenido se declara entre ellas. En nuestro ejemplo, **<html lang="es">** indica el comienzo del código HTML y **</html>** indica el final. Compare las etiquetas de apertura y cierre y verá que la de cierre se distingue por una barra invertida antes de la palabra clave (por ejemplo, **</html>**). El resto de nuestro código será insertado entre estas dos etiquetas: **<html> ... </html>**.
>
> **IMPORTANTE:** HTML5 es extremadamente flexible en cuanto a la estructura y a los elementos utilizados para construirla. El elemento **<html>** puede ser incluido sin ningún atributo o incluso ignorado completamente. Con el propósito de preservar compatibilidad (y por algunas razones extras que no vale la pena mencionar aquí) le recomendamos que siga algunas reglas básicas. En este libro vamos a enseñarle cómo construir documentos HTML de acuerdo a lo que nosotros consideramos prácticas recomendadas.

Para encontrar otros lenguajes para el atributo **lang** puede visitar el siguiente enlace: *www.w3schools.com/tags/ref_language_codes.asp.*

<head>

Continuemos construyendo nuestra plantilla. El código HTML insertado entre las etiquetas **<html>** tiene que ser dividido entre dos secciones principales. Al igual que en versiones previas de HTML, la primera sección es la cabecera y la segunda el cuerpo. El siguiente paso, por lo tanto, será crear estas dos secciones en el código usando los elementos **<head>** y **<body>** ya conocidos**.**

El elemento **<head>** va primero, por supuesto, y al igual que el resto de los elementos estructurales tiene una etiqueta de apertura y una de cierre:

```
<!DOCTYPE html>
<html lang="es">
<head>

</head>

</html>
```

Listado 1-3. *Usando el elemento <head>.*

La etiqueta no cambió desde versiones anteriores y su propósito sigue siendo exactamente el mismo. Dentro de las etiquetas **<head>** definiremos el título de nuestra página web, declararemos el set de caracteres correspondiente, proveeremos información general acerca del documento e incorporaremos los archivos externos con estilos, códigos Javascript o incluso imágenes necesarias para generar la página en la pantalla.

Excepto por el título y algunos íconos, el resto de la información incorporada en el documento entre estas etiquetas es invisible para el usuario.

<body>

La siguiente gran sección que es parte principal de la organización de un documento HTML es el cuerpo. El cuerpo representa la parte visible de todo documento y es especificado entre etiquetas **<body>**. Estas etiquetas tampoco han cambiado en relación con versiones previas de HTML:

```
<!DOCTYPE html>
<html lang="es">
<head>

</head>
<body>

</body>
</html>
```

Listado 1-4. *Usando el elemento <body>.*

Conceptos básicos: Hasta el momento tenemos un código simple pero con una estructura compleja. Esto es porque el código HTML no está formado por un conjunto de instrucciones secuenciales. HTML es un lenguaje de etiquetas, un listado de elementos que usualmente se utilizan en pares y que pueden ser anidados (totalmente contenidos uno dentro del otro). En la primera línea del código del Listado 1-4 tenemos una etiqueta simple con la definición del tipo de documento e inmediatamente después la etiqueta de apertura **<html lang="es">**. Esta etiqueta y la de cierre **</html>** al final del listado están indicando el comienzo del código HTML y su final. Entre las etiquetas **<html>** insertamos otras etiquetas especificando dos importantes partes de la estructura básica: **<head>** para la cabecera y **<body>** para el cuerpo del documento. Estas dos etiquetas también se utilizan en pares. Más adelante en este capítulo veremos que más etiquetas son insertadas entre estas últimas conformando una estructura de árbol con **<html>** como su raíz.

<meta>

Es momento de construir la cabecera del documento. Algunos cambios e innovaciones fueron incorporados dentro de la cabecera, y uno de ellos es la etiqueta que define el juego de caracteres a utilizar para mostrar el documento. Ésta es una etiqueta **<meta>** que especifica cómo el texto será presentado en pantalla:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset=”iso-8859-1”>

</head>
<body>

</body>
</html>
```

Listado 1-5. *Usando el elemento <meta>.*

La innovación de este elemento en HTML5, como en la mayoría de los casos, es solo simplificación. La nueva etiqueta **<meta>** para la definición del tipo de caracteres es más corta y simple. Por supuesto, podemos cambiar el tipo **iso-8859-1** por el necesario para nuestros documentos y agregar otras etiquetas **<meta>** como **description** o **keywords** para definir otros aspectos de la página web, como es mostrado en el siguiente ejemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name=”description” content=”Ejemplo de HTML5”>
  <meta name=”keywords” content=”HTML5, CSS3, Javascript”>

</head>
<body>

</body>
</html>
```

Listado 1-6. *Agregando más elementos <meta>.*

> **Conceptos básicos:** Hay varios tipos de etiqueta **<meta>** que pueden ser incluidas para declarar información general sobre el documento, pero esta información no es mostrada en la ventana del navegador, es solo importante para motores de búsqueda y dispositivos que necesitan hacer una vista previa del documento u obtener un sumario de la información que contiene. Como comentamos

anteriormente, aparte del título y algunos íconos, la mayoría de la información insertada entre las etiquetas **<head>** no es visible para los usuarios. En el código del Listado 1-6, el atributo **name** dentro de la etiqueta **<meta>** especifica su tipo y **content** declara su valor, pero ninguno de estos valores es mostrado en pantalla. Para aprender más sobre la etiqueta **<meta>**, visite nuestro sitio web y siga los enlaces proporcionados para este capítulo.

En HTML5 no es necesario cerrar etiquetas simples con una barra al final, pero recomendamos utilizarlas por razones de compatibilidad. El anterior código se podría escribir de la siguiente manera:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1" />
  <meta name="description" content="Ejemplo de HTML5" />
  <meta name="keywords" content="HTML5, CSS3, JavaScript" />

</head>
<body>

</body>
</html>
```

Listado 1-7. *Cierre de etiquetas simples.*

<title>

La etiqueta **<title>**, como siempre, simplemente especifica el título del documento, y no hay nada nuevo para comentar:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Ejemplo de HTML5">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Este texto es el título del documento</title>

</head>
<body>

</body>
</html>
```

Listado 1-8. *Usando la etiqueta <title>.*

Conceptos básicos: El texto entre las etiquetas **<title>** es el título del documento que estamos creando. Normalmente este texto es mostrado en la barra superior de la ventana del navegador.

<link>

Otro importante elemento que va dentro de la cabecera del documento es **<link>**. Este elemento es usado para incorporar estilos, códigos Javascript, imágenes o iconos desde archivos externos. Uno de los usos más comunes para **<link>** es la incorporación de archivos con estilos CSS:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Ejemplo de HTML5">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Este texto es el título del documento</title>
  <link rel="stylesheet" href="misestilos.css">
</head>
<body>

</body>
</html>
```

Listado 1-9. *Usando el elemento <link>.*

En HTML5 ya no se necesita especificar qué tipo de estilos estamos insertando, por lo que el atributo **type** fue eliminado. Solo necesitamos dos atributos para incorporar nuestro archivo de estilos: **rel** y **href**. El atributo **rel** significa "relación" y es acerca de la relación entre el documento y el archivo que estamos incorporando por medio de **href**. En este caso, el atributo **rel** tiene el valor **stylesheet** que le dice al navegador que el archivo **misestilos.css** es un archivo CSS con estilos requeridos para presentar la página en pantalla (en el próximo capítulo estudiaremos cómo utilizar estilos CSS).

Conceptos básicos: Un archivo de estilos es un grupo de reglas de formato que ayudarán a cambiar la apariencia de nuestra página web (por ejemplo, el tamaño y color del texto). Sin estas reglas, el texto y cualquier otro elemento HTML sería mostrado en pantalla utilizando los estilos estándar provistos por el navegador. Los estilos son reglas simples que normalmente requieren solo unas pocas líneas de código y pueden ser declarados en el mismo documento. Como veremos más adelante, no es estrictamente necesario obtener esta información de archivos externos pero es una práctica recomendada. Cargar las reglas CSS desde un documento externo (otro archivo) nos permitirá organizar el documento

principal, incrementar la velocidad de carga y aprovechar las nuevas características de HTML5.

Con esta última inserción podemos considerar finalizado nuestro trabajo en la cabecera. Ahora es tiempo de trabajar en el cuerpo, donde la magia ocurre.

1.3 Estructura del cuerpo

La estructura del cuerpo (el código entre las etiquetas **<body>**) generará la parte visible del documento. Este es el código que producirá nuestra página web.

HTML siempre ofreció diferentes formas de construir y organizar la información dentro del cuerpo de un documento. Uno de los primeros elementos provistos para este propósito fue **<table>**. Las tablas permitían a los diseñadores acomodar datos, texto, imágenes y herramientas dentro de filas y columnas de celdas, incluso sin que hayan sido concebidas para este propósito.

En los primeros días de la web, las tablas fueron una revolución, un gran paso hacia adelante con respecto a la visualización de los documentos y la experiencia ofrecida a los usuarios. Más adelante, gradualmente, otros elementos reemplazaron su función, permitiendo lograr lo mismo con menos código, facilitando de este modo la creación, permitiendo portabilidad y ayudando al mantenimiento de los sitios web.

El elemento ** ** comenzó a dominar la escena. Con el surgimiento de webs más interactivas y la integración de HTML, CSS y Javascript, el uso de ** ** se volvió una práctica común. Pero este elemento, así como **<table>**, no provee demasiada información acerca de las parte del cuerpo que está representando. Desde imágenes a menús, textos, enlaces, códigos, formularios, cualquier cosa puede ir entre las etiquetas de apertura y cierre de un elemento ** **. En otras palabras, la palabra clave **div** solo especifica una división en el cuerpo, como la celda de una tabla, pero no ofrece indicio alguno sobre qué clase de división es, cuál es su propósito o qué contiene.

Para los usuarios estas claves o indicios no son importantes, pero para los navegadores la correcta interpretación de qué hay dentro del documento que se está procesando puede ser crucial en muchos casos. Luego de la revolución de los dispositivos móviles y el surgimiento de diferentes formas en que la gente accede a la web, la identificación de cada parte del documento es una tarea que se ha vuelto más relevante que nunca.

Considerando todo lo expuesto, HTML5 incorpora nuevos elementos que ayudan a identificar cada sección del documento y organizar el cuerpo del mismo. En HTML5 las secciones más importantes son diferenciadas y la estructura principal ya no depende más de los elementos ** ** o **<table>**.

Cómo usamos estos nuevos elementos depende de nosotros, pero las palabras clave otorgadas a cada uno de ellos nos dan ayudan a entender sus funciones. Normalmente una página o aplicación web está dividida entre varias áreas visuales para mejorar la experiencia del usuario y facilitar la interactividad. Las palabras claves que representan

cada nuevo elemento de HTML5 están íntimamente relacionadas con estas áreas, como veremos pronto.

Organización

La Figura 1-1 representa un diseño común encontrado en la mayoría de los sitios webs estos días. A pesar del hecho de que cada diseñador crea sus propios diseños, en general podremos identificar las siguientes secciones en cada sitio web estudiado:

Figura 1-1. *Representación visual de un clásico diseño web.*

En la parte superior, descripto como **Cabecera**, se encuentra el espacio donde usualmente se ubica el logo, título, subtítulos y una corta descripción del sitio web o la página.

Inmediatamente debajo, podemos ver la **Barra de Navegación** en la cual casi todos los desarrolladores ofrecen un menú o lista de enlaces con el propósito de facilitar la navegación a través del sitio. Los usuarios son guiados desde esta barra hacia las diferentes páginas o documentos, normalmente pertenecientes al mismo sitio web.

El contenido más relevante de una página web se encuentra, en casi todo diseño, ubicado en su centro. Esta sección presenta información y enlaces valiosos. La mayoría de las veces es dividida en varias filas y columnas. En el ejemplo de la Figura 1-1 se utilizaron

solo dos columnas: **Información Principal** y **Barra Lateral**, pero esta sección es extremadamente flexible y normalmente diseñadores la adaptan acorde a sus necesidades insertando más columnas, dividiendo cada columna entre bloques más pequeños o generando diferentes distribuciones y combinaciones. El contenido presentado en esta parte del diseño es usualmente de alta prioridad. En el diseño de ejemplo, **Información Principal** podría contener una lista de artículos, descripción de productos, entradas de un blog o cualquier otra información importante, y la **Barra Lateral** podría mostrar una lista de enlaces apuntando hacia cada uno se esos ítems. En un blog, por ejemplo, esta última columna ofrecerá una lista de enlaces apuntando a cada entrada del blog, información acerca del autor, etc…

En la base de un diseño web clásico siempre nos encontramos con una barra más que aquí llamamos **Institucional**. La nombramos de esta manera porque esta es el área en donde normalmente se muestra información acerca del sitio web, el autor o la empresa, además de algunos enlaces con respecto a reglas, términos y condiciones y toda información adicional que el desarrollador considere importante compartir. La barra **Institucional** es un complemento de la **Cabecera** y es parte de lo que se considera estos días la estructura esencial de una página web, como podemos apreciar en el siguiente ejemplo:

Figura 1-2. *Representación visual de un clásico diseño para blogs.*

La Figura 1-2 es una representación de un blog normal. En este ejemplo se puede claramente identificar cada parte del diseño considerado anteriormente.

1. **Cabecera**
2. **Barra de Navegación**
3. Sección de **Información Principal**
4. **Barra Lateral**
5. El pie o la barra **Institucional**

Esta simple representación de un blog nos puede ayudar a entender que cada sección definida en un sitio web tiene un propósito. A veces este propósito no es claro pero en esencia se encuentra siempre allí, ayudándonos a reconocer cualquiera de las secciones descriptas anteriormente en todo diseño.

HTML5 considera esta estructura básica y provee nuevos elementos para diferenciar y declarar cada una de sus partes. A partir de ahora podemos decir al navegador para qué es cada sección:

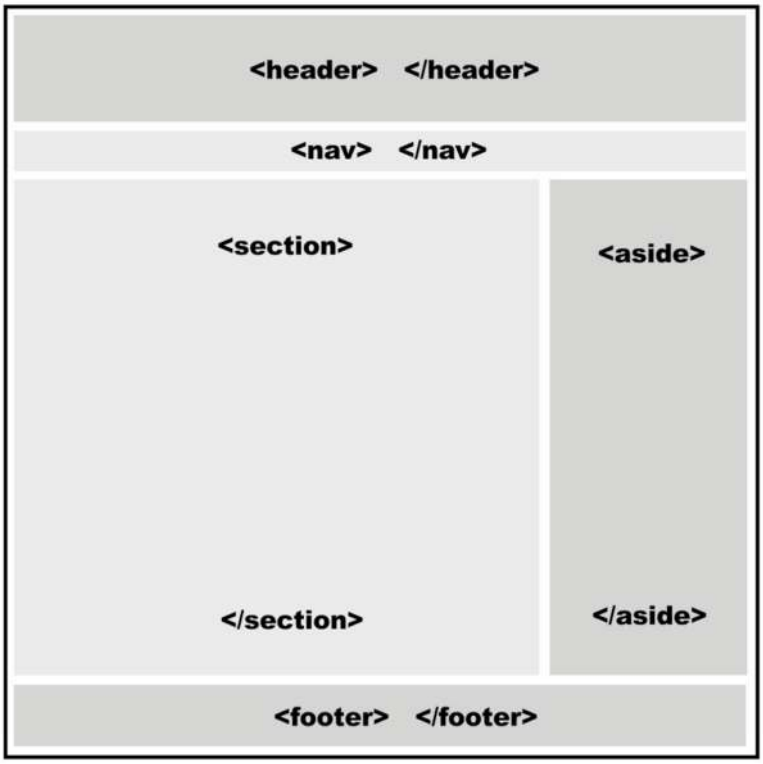

Figura 1-3. *Representación visual de un diseño utilizando elementos HTML5.*

La Figura 1-3 muestra el típico diseño presentado anteriormente, pero esta vez con los correspondientes elementos HTML5 para cada sección (incluyendo etiquetas de apertura y cierre).

<header>

Uno de los nuevos elementos incorporados en HTML5 es **<header>**. El elemento **<header>** no debe ser confundido con **<head>** usado antes para construir la cabecera del documento. Del mismo modo que **<head>**, la intención de **<header>** es proveer información introductoria (títulos, subtítulos, logos), pero difiere con respecto a **<head>** en su alcance. Mientras que el elemento **<head>** tiene el propósito de proveer información acerca de todo el documento, **<header>** es usado solo para el cuerpo o secciones específicas dentro del cuerpo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Ejemplo de HTML5">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Este texto es el título del documento</title>
  <link rel="stylesheet" href="misestilos.css">
</head>
<body>
  <header>
    <h1>Este es el título principal del sitio web</h1>
  </header>

</body>
</html>
```

Listado 1-10. *Usando el elemento <header>.*

En el Listado 1-10, definimos el título de la página web utilizando el elemento **<header>**. Recuerde que esta cabecera no es la misma que la utilizada previamente para definir el título del documento. La inserción del elemento **<header>** representa el comienzo del cuerpo y por lo tanto de la parte visible del documento. De ahora en más será posible ver los resultados de nuestro código en la ventana del navegador.

> **Hágalo usted mismo:** Si siguió las instrucciones desde el comienzo de este capítulo ya debería contar con un archivo de texto creado con todos los códigos estudiados hasta el momento y listo para ser probado. Si no es así, todo lo que debe hacer es copiar el código en el Listado 1-10 dentro de un archivo de texto vacío utilizando cualquier editor de texto (como el Bloc de Notas de Windows, por ejemplo) y grabar el archivo con el nombre de su agrado y la extensión

`.html`. Para ver el código en funcionamiento, abra el archivo en un navegador compatible con HTML5 (puede hacerlo con un doble clic sobre el archivo en su explorador de archivos).

Conceptos básicos: Entre las etiquetas **<header>** en el Listado 1-10 hay un elemento que probablemente no conoce. El elemento **<h1>** es un viejo elemento HTML usado para definir títulos. El número indica la importancia del título. El elemento **<h1>** es el más importante y **<h6>** el de menor importancia, por lo tanto **<h1>** será utilizado para mostrar el título principal y los demás para subtítulos o subtítulos internos. Más adelante veremos cómo estos elementos trabajan en HTML5.

<nav>

Siguiendo con nuestro ejemplo, la siguiente sección es la **Barra de Navegación**. Esta barra es generada en HTML5 con el elemento **<nav>**:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Ejemplo de HTML5">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Este texto es el título del documento</title>
  <link rel="stylesheet" href="misestilos.css">
</head>
<body>
  <header>
    <h1>Este es el título principal del sitio web</h1>
  </header>
  <nav>
    <ul>
      <li>principal</li>
      <li>fotos</li>
      <li>videos</li>
      <li>contacto</li>
    </ul>
  </nav>

</body>
</html>
```

Listado 1-11. *Usando el elemento <nav>.*

Como se puede apreciar en el Listado 1-11, el elemento **<nav>** se encuentra dentro de las etiquetas **<body>** pero es ubicado después de la etiqueta de cierre de la cabecera

(**</header>**), no dentro de las etiquetas **<header>**. Esto es porque **<nav>** no es parte de la cabecera sino una nueva sección.

Anteriormente dijimos que la estructura y el orden que elegimos para colocar los elementos HTML5 dependen de nosotros. Esto significa que HTML5 es versátil y solo nos otorga los parámetros y elementos básicos con los que trabajar, pero cómo usarlos será exclusivamente decisión nuestra. Un ejemplo de esta versatilidad es que el elemento **<nav>** podría ser insertado dentro del elemento **<header>** o en cualquier otra parte del cuerpo. Sin embargo, siempre se debe considerar que estas etiquetas fueron creadas para brindar información a los navegadores y ayudar a cada nuevo programa y dispositivo en el mercado a identificar las partes más relevantes del documento. Para conservar nuestro código portable y comprensible, recomendamos como buena práctica seguir lo que marcan los estándares y mantener todo tan claro como sea posible. El elemento **<nav>** fue creado para ofrecer ayuda para la navegación, como en menús principales o grandes bloques de enlaces, y debería ser utilizado de esa manera.

> **Conceptos básicos:** En el ejemplo del Listado 1-11 generamos las opciones del menú para nuestra página web. Entre las etiquetas **<nav>** hay dos elementos que son utilizados para crear una lista. El propósito del elemento **<ul>** es definir la lista. Anidado entre las etiquetas **<ul>** encontramos varias etiquetas **<li>** con diferentes textos representando las opciones del menú. Las etiquetas **<li>**, como probablemente ya se ha dado cuenta, son usadas para definir cada ítem de la lista. El propósito de este libro no es enseñarle conceptos básicos sobre HTML, si necesita más información acerca de elementos regulares de este lenguaje visite nuestro sitio web y siga los enlaces correspondientes a este capítulo.

<section>

Siguiendo nuestro diseño estándar nos encontramos con las columnas que en la Figura 1-1 llamamos **Información Principal** y **Barra Lateral**. Como explicamos anteriormente, la columna **Información Principal** contiene la información más relevante del documento y puede ser encontrada en diferentes formas (por ejemplo, dividida en varios bloques o columnas). Debido a que el propósito de estas columnas es más general, el elemento en HTML5 que especifica estas secciones se llama simplemente **<section>**:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Ejemplo de HTML5">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Este texto es el título del documento</title>
  <link rel="stylesheet" href="misestilos.css">
</head>
```

```
<body>
  <header>
    <h1>Este es el título principal del sitio web</h1>
  </header>
  <nav>
    <ul>
      <li>principal</li>
      <li>fotos</li>
      <li>videos</li>
      <li>contacto</li>
    </ul>
  </nav>
  <section>

  </section>

</body>
</html>
```

Listado 1-12. *Usando el elemento <section>.*

Al igual que la **Barra de Navegación**, la columna **Información Principal** es una sección aparte. Por este motivo, la sección para **Información Principal** va debajo de la etiqueta de cierre **</nav>**.

> **Hágalo usted mismo:** Compare el último código en el Listado 1-12 con el diseño de la Figura 1-3 para comprender cómo las etiquetas son ubicadas en el código y qué sección cada una de ellas genera en la representación visual de la página web.
>
> **IMPORTANTE:** Las etiquetas que representan cada sección del documento están localizadas en el código en forma de lista, unas sobre otras, pero en el sitio web algunas de estas secciones se ubicarán lado a lado (las columnas **Información Principal** y **Barra Lateral** son un claro ejemplo). En HTML5, la responsabilidad por la representación de los elementos en la pantalla fue delegada a CSS. El diseño será logrado asignando estilos CSS a cada elemento HTML. Estudiaremos CSS en el próximo capítulo.

<aside>

En un típico diseño web (Figura 1-1) la columna llamada **Barra Lateral** se ubica al lado de la columna **Información Principal**. Esta es una columna o sección que normalmente contiene datos relacionados con la información principal pero que no son relevantes o igual de importantes.

En el diseño de un blog, por ejemplo, la **Barra Lateral** contendrá una lista de enlaces. En el ejemplo de la Figura 1-2, los enlaces apuntan a cada una de las entradas del blog y ofrecen información adicional sobre el autor (número 4). La información dentro de esta

barra está relacionada con la información principal pero no es relevante por sí misma. Siguiendo el mismo ejemplo podemos decir que las entradas del blog son relevantes pero los enlaces y las pequeñas reseñas sobre esas entradas son solo una ayuda para la navegación pero no lo que al lector realmente le interesa.

En HTML5 podemos diferenciar esta clase secundaria de información utilizando el elemento **<aside>**:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Ejemplo de HTML5">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Este texto es el título del documento</title>
  <link rel="stylesheet" href="misestilos.css">
</head>
<body>
  <header>
    <h1>Este es el título principal del sitio web</h1>
  </header>
  <nav>
    <ul>
      <li>principal</li>
      <li>fotos</li>
      <li>videos</li>
      <li>contacto</li>
    </ul>
  </nav>
  <section>

   </section>
   <aside>
     <blockquote>Mensaje número uno</blockquote>
     <blockquote>Mensaje número dos</blockquote>
   </aside>

</body>
</html>
```

Listado 1-13. *Usando el elemento <aside>.*

El elemento **<aside>** podría estar ubicado del lado derecho o izquierdo de nuestra página de ejemplo, la etiqueta no tiene una posición predefinida. El elemento **<aside>** solo describe la información que contiene, no el lugar dentro de la estructura. Este elemento puede estar ubicado en cualquier parte del diseño y ser usado siempre y cuando su contenido no sea considerado como el contenido principal del documento. Por ejemplo, podemos usar **<aside>** dentro del elemento **<section>** o incluso insertado entre la información relevante, como en el caso de una cita.

<footer>

Para finalizar la construcción de la plantilla o estructura elemental de nuestro documento HTML5, solo necesitamos un elemento más. Ya contamos con la cabecera del cuerpo, secciones con ayuda para la navegación, información importante y hasta una barra lateral con datos adicionales, por lo tanto lo único que nos queda por hacer es cerrar nuestro diseño para otorgarle un final al cuerpo del documento. HTML5 provee un elemento específico para este propósito llamado **<footer>**:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Ejemplo de HTML5">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Este texto es el título del documento</title>
  <link rel="stylesheet" href="misestilos.css">
</head>
<body>
  <header>
    <h1>Este es el título principal del sitio web</h1>
  </header>
  <nav>
    <ul>
      <li>principal</li>
      <li>fotos</li>
      <li>videos</li>
      <li>contacto</li>
    </ul>
  </nav>
  <section>

  </section>
  <aside>
    <blockquote>Mensaje número uno</blockquote>
    <blockquote>Mensaje número dos</blockquote>
  </aside>
  <footer>
    Derechos Reservados &copy; 2010-2011
  </footer>

</body>
</html>
```

Listado 1-14. *Usando el elemento <footer>.*

En el típico diseño de una página web (Figura 1-1) la sección llamada **Institucional** será definida por etiquetas **<footer>**. Esto es debido a que la barra representa el final (o pie)

del documento y esta parte de la página web es normalmente usada para compartir información general sobre el autor o la organización detrás del proyecto.

Generalmente, el elemento **<footer>** representará el final del cuerpo de nuestro documento y tendrá el propósito descripto anteriormente. Sin embargo, **<footer>** puede ser usado múltiples veces dentro del cuerpo para representar también el final de diferentes secciones (del mismo modo que la etiqueta **<header>**). Estudiaremos esta última característica más adelante.

1.4 Dentro del cuerpo

El cuerpo de nuestro documento está listo. La estructura básica de nuestro sitio web fue finalizada, pero aún tenemos que trabajar en el contenido. Los elementos HTML5 estudiados hasta el momento nos ayudan a identificar cada sección del diseño y asignar un propósito intrínseco a cada una de ellas, pero lo que es realmente importante para nuestro sitio web se encuentra en el interior de estas secciones.

La mayoría de los elementos ya estudiados fueron creados para construir una estructura para el documento HTML que pueda ser identificada y reconocida por los navegadores y nuevos dispositivos. Aprendimos acerca de la etiqueta **<body>** usada para declarar el cuerpo o parte visible del documento, la etiqueta **<header>** con la que agrupamos información importante para el cuerpo, la etiqueta **<nav>** que provee ayuda para la navegación del sitio web, la etiqueta **<section>** necesaria para contener la información más relevante, y también **<aside>** y **<footer>** para ofrecer información adicional de cada sección y del documento mismo. Pero ninguno de estos elementos declara algo acerca del contenido. Todos tienen un específico propósito estructural.

Más profundo nos introducimos dentro del documento más cerca nos encontramos de la definición del contenido. Esta información estará compuesta por diferentes elementos visuales como títulos, textos, imágenes, videos y aplicaciones interactivas, entre otros. Necesitamos poder diferenciar estos elementos y establecer una relación entre ellos dentro de la estructura.

<article>

El diseño considerado anteriormente (Figura 1-1) es el más común y representa una estructura esencial para los sitios web estos días, pero es además ejemplo de cómo el contenido clave es mostrado en pantalla. Del mismo modo que los blogs están divididos en entradas, sitios web normalmente presentan información relevante dividida en partes que comparten similares características. El elemento **<article>** nos permite identificar cada una de estas partes:

```
<!DOCTYPE html>
<html lang="es">
<head>
```

```
    <meta charset="iso-8859-1">
    <meta name="description" content="Ejemplo de HTML5">
    <meta name="keywords" content="HTML5, CSS3, JavaScript">
    <title>Este texto es el título del documento</title>
    <link rel="stylesheet" href="misestilos.css">
  </head>
  <body>
    <header>
      <h1>Este es el título principal del sitio web</h1>
    </header>
    <nav>
      <ul>
        <li>principal</li>
        <li>fotos</li>
        <li>videos</li>
        <li>contacto</li>
      </ul>
    </nav>
    <section>
      <article>
        Este es el texto de mi primer mensaje
      </article>
      <article>
        Este es el texto de mi segundo mensaje
      </article>
    </section>
    <aside>
      <blockquote>Mensaje número uno</blockquote>
      <blockquote>Mensaje número dos</blockquote>
    </aside>
    <footer>
      Derechos Reservados &copy; 2010-2011
    </footer>
  </body>
  </html>
```

Listado 1-15. *Usando el elemento <article>.*

Como puede observarse en el código del Listado 1-15, las etiquetas **<article>** se encuentran ubicadas dentro del elemento **<section>**. Las etiquetas **<article>** en nuestro ejemplo pertenecen a esta sección, son sus hijos, del mismo modo que cada elemento dentro de las etiquetas **<body>** es hijo del cuerpo. Y al igual que cada elemento hijo del cuerpo, las etiquetas **<article>** son ubicadas una sobre otra, como es mostrado en la Figura 1-4.

> **Conceptos básicos:** Como dijimos anteriormente, la estructura de un documento HTML puede ser descripta como un árbol, con el elemento **<html>** como su raíz. Otra forma de describir la relación entre elementos es nombrarlos como padres, hijos y hermanos, de acuerdo a la posición que ocupan dentro de esa misma estructura. Por ejemplo, en un típico documento HTML el elemento **<body>** es

hijo del elemento **<html>** y hermano del elemento **<head>**. Ambos, **<body>** y **<head>**, tienen al elemento **<html>** como su padre.

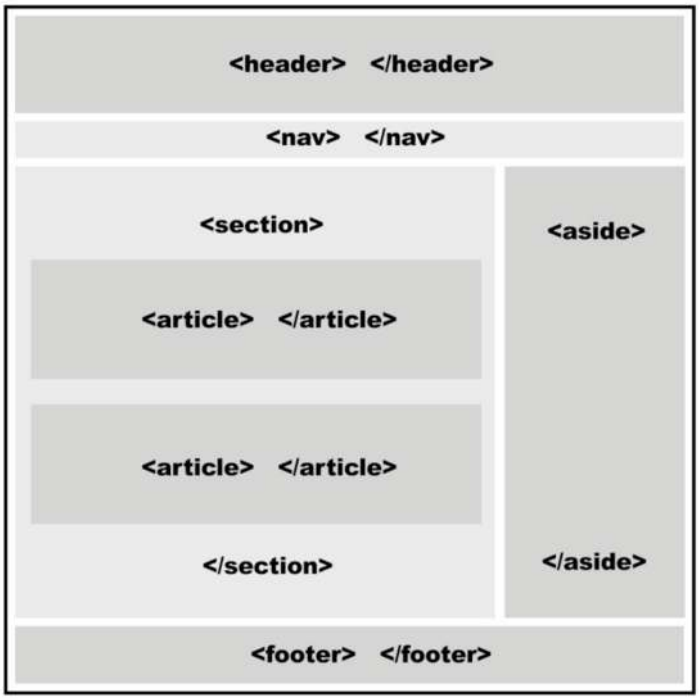

Figura 1-4. *Representación visual de las etiquetas <article> que fueron incluidas para contener información relevante de la página web.*

El elemento **<article>** no está limitado por su nombre (no se limita, por ejemplo, a artículos de noticias). Este elemento fue creado con la intención de contener unidades independientes de contenido, por lo que puede incluir mensajes de foros, artículos de una revista digital, entradas de blog, comentarios de usuarios, etc… Lo que hace es agrupar porciones de información que están relacionadas entre sí independientemente de su naturaleza.

Como una parte independiente del documento, el contenido de cada elemento **<article>** tendrá su propia estructura. Para definir esta estructura, podemos aprovechar la versatilidad de los elementos **<header>** y **<footer>** estudiados anteriormente. Estos elementos son portables y pueden ser usados no solo para definir los límites del cuerpo sino también en cualquier sección de nuestro documento:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
```

```
    <meta name="description" content="Ejemplo de HTML5">
    <meta name="keywords" content="HTML5, CSS3, JavaScript">
    <title>Este texto es el título del documento</title>
    <link rel="stylesheet" href="misestilos.css">
  </head>
  <body>
    <header>
      <h1>Este es el título principal del sitio web</h1>
    </header>
    <nav>
      <ul>
        <li>principal</li>
        <li>fotos</li>
        <li>videos</li>
        <li>contacto</li>
      </ul>
    </nav>
    <section>
      <article>
        <header>
          <h1>Título del mensaje uno</h1>
        </header>
        Este es el texto de mi primer mensaje
        <footer>
          <p>comentarios (0)</p>
        </footer>
      </article>
      <article>
        <header>
          <h1>Titulo del mensaje dos</h1>
        </header>
        Este es el texto de mi segundo mensaje
        <footer>
          <p>comentarios (0)</p>
        </footer>
      </article>
    </section>
    <aside>
      <blockquote>Mensaje número uno</blockquote>
      <blockquote>Mensaje número dos</blockquote>
    </aside>
    <footer>
      Derechos Reservados &copy; 2010-2011
    </footer>
  </body>
  </html>
```

Listado 1-16. *Construyendo la estructura de <article>.*

Los dos mensajes insertados en el código del Listado 1-16 fueron construidos con el elemento **<article>** y tienen una estructura específica. En la parte superior de esta estructura incluimos las etiquetas **<header>** conteniendo el título definido con el elemento

<h1>, debajo se encuentra el contenido mismo del mensaje y sobre el final, luego del texto, vienen las etiquetas **<footer>** especificando la cantidad de comentarios recibidos.

<hgroup>

Dentro de cada elemento **<header>**, en la parte superior del cuerpo o al comienzo de cada **<article>**, incorporamos elementos **<h1>** para declarar un título. Básicamente, las etiquetas **<h1>** son todo lo que necesitamos para crear una línea de cabecera para cada parte del documento, pero es normal que necesitemos también agregar subtítulos o más información que especifique de qué se trata la página web o una sección en particular. De hecho, el elemento **<header>** fue creado para contener también otros elementos como tablas de contenido, formularios de búsqueda o textos cortos y logos.

Para construir este tipo de cabeceras, podemos aprovechar el resto de las etiquetas H, como **<h1>, <h2>**, **<h3>**, **<h4>**, **<h5>** y **<h6>**, pero siempre considerando que por propósitos de procesamiento interno, y para evitar generar múltiples secciones durante la interpretación del documento por parte del navegador, estas etiquetas deben ser agrupadas juntas. Por esta razón, HTML5 provee el elemento **<hgroup>**:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Ejemplo de HTML5">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Este texto es el título del documento</title>
  <link rel="stylesheet" href="misestilos.css">
</head>
<body>
  <header>
    <h1>Este es el título principal del sitio web</h1>
  </header>
  <nav>
    <ul>
      <li>principal</li>
      <li>fotos</li>
      <li>videos</li>
      <li>contacto</li>
    </ul>
  </nav>
  <section>
    <article>
      <header>
        <hgroup>
          <h1>Título del mensaje uno</h1>
          <h2>Subtítulo del mensaje uno</h2>
        </hgroup>
        <p>publicado 10-12-2011</p>
```

```
      </header>
      Este es el texto de mi primer mensaje
      <footer>
        <p>comentarios (0)</p>
      </footer>
    </article>
    <article>
      <header>
        <hgroup>
          <h1>Título del mensaje dos</h1>
          <h2>Subtítulo del mensaje dos</h2>
        </hgroup>
        <p>publicado 15-12-2011</p>
      </header>
      Este es el texto de mi segundo mensaje
      <footer>
        <p>comentarios (0)</p>
      </footer>
    </article>
  </section>
  <aside>
    <blockquote>Mensaje número uno</blockquote>
    <blockquote>Mensaje número dos</blockquote>
  </aside>
  <footer>
    Derechos Reservados &copy; 2010-2011
  </footer>
</body>
</html>
```

Listado 1-17. *Usando el elemento <hgroup>.*

Las etiquetas H deben conservar su jerarquía, lo que significa que debemos primero declarar la etiqueta **<h1>**, luego usar **<h2>** para subtítulos y así sucesivamente. Sin embargo, a diferencia de anteriores versiones de HTML, HTML5 nos deja reusar las etiquetas H y construir esta jerarquía una y otra vez en cada sección del documento. En el ejemplo del Listado 1-17, agregamos un subtítulo y datos adicionales a cada mensaje. Los títulos y subtítulos fueron agrupados juntos utilizando **<hgroup>**, recreando de este modo la jerarquía **<h1>** y **<h2>** en cada elemento **<article>**.

> **IMPORTANTE:** El elemento **<hgroup>** es necesario cuando tenemos un título y subtítulo o más etiquetas H juntas en la misma cabecera. Este elemento puede contener solo etiquetas H y esta fue la razón por la que en nuestro ejemplo dejamos los datos adicionales afuera. Si solo dispone de una etiqueta **<h1>** o la etiqueta **<h1>** junto con datos adicionales, no tiene que agrupar estos elementos juntos. Por ejemplo, en la cabecera del cuerpo (**<header>**) no usamos este elemento porque solo tenemos una etiqueta H en su interior. Siempre recuerde que **<hgroup>** fue creado solo con la intención de agrupar etiquetas H, exactamente como su nombre lo indica.

Navegadores y programas que ejecutan y presentan en la pantalla sitios webs leen el código HTML y crean su propia estructura interna para interpretar y procesar cada elemento. Esta estructura interna está dividida en secciones que no tienen nada que ver con las divisiones en el diseño o el elemento **<section>**. Estas son secciones conceptuales generadas durante la interpretación del código. El elemento **<header>** no crea una de estas secciones por sí mismo, lo que significa que los elementos dentro de **<header>** representarán diferentes niveles e internamente pueden generar diferentes secciones. El elemento **<hgroup>** fue creado con el propósito de agrupar las etiquetas H y evitar interpretaciones incorrectas por parte de los navegadores.

> **Conceptos básicos:** lo que llamamos "información adicional" dentro de la cabecera en nuestra descripción previa es conocido como Metadata. Metadata es un conjunto de datos que describen y proveen información acerca de otro grupo de datos. En nuestro ejemplo, Metadata es la fecha en la cual cada mensaje fue publicado.

<figure> y <figcaption>

La etiqueta **<figure>** fue creada para ayudarnos a ser aún más específicos a la hora de declarar el contenido del documento. Antes de que este elemento sea introducido, no podíamos identificar el contenido que era parte de la información pero a la vez independiente, como ilustraciones, fotos, videos, etc… Normalmente estos elementos son parte del contenido relevante pero pueden ser extraídos o movidos a otra parte sin afectar o interrumpir el flujo del documento. Cuando nos encontramos con esta clase de información, las etiquetas **<figure>** pueden ser usadas para identificarla:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Ejemplo de HTML5">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Este texto es el título del documento</title>
  <link rel="stylesheet" href="misestilos.css">
</head>
<body>
  <header>
    <h1>Este es el título principal del sitio web</h1>
  </header>
  <nav>
    <ul>
      <li>principal</li>
      <li>fotos</li>
      <li>videos</li>
      <li>contacto</li>
    </ul>
```

```
    </nav>
    <section>
      <article>
        <header>
          <hgroup>
            <h1>Título del mensaje uno</h1>
            <h2>Subtítulo del mensaje uno</h2>
          </hgroup>
          <p>publicado 10-12-2011</p>
        </header>
        Este es el texto de mi primer mensaje
        <figure>
          <img src=”http://minkbooks.com/content/myimage.jpg”>
          <figcaption>
             Esta es la imagen del primer mensaje
           </figcaption>
        </figure>
        <footer>
          <p>comentarios (0)</p>
        </footer>
      </article>
      <article>
        <header>
          <hgroup>
            <h1>Título del mensaje dos</h1>
            <h2>Subtítulo del mensaje dos</h2>
          </hgroup>
          <p>publicado 15-12-2011</p>
        </header>
        Este es el texto de mi segundo mensaje
        <footer>
          <p>comentarios (0)</p>
        </footer>
      </article>
    </section>
    <aside>
      <blockquote>Mensaje número uno</blockquote>
      <blockquote>Mensaje número dos</blockquote>
    </aside>
    <footer>
      Derechos Reservados &copy; 2010-2011
    </footer>
  </body>
  </html>
```

Listado 1-18. *Usando los elementos <figure> y <figcaption>.*

En el Listado 1-18, en el primer mensaje, luego del texto insertamos una imagen (**<img src="http://minkbooks.com/content/myimage.jpg">**). Esta es una práctica común, a menudo el texto es enriquecido con imágenes o videos. Las etiquetas **<figure>**

nos permiten envolver estos complementos visuales y diferenciarlos así de la información más relevante.

También en el Listado 1-18 se puede observar un elemento extra dentro de **<figure>**. Normalmente, unidades de información como imágenes o videos son descriptas con un corto texto debajo. HTML5 provee un elemento para ubicar e identificar estos títulos descriptivos. Las etiquetas **<figcaption>** encierran el texto relacionado con **<figure>** y establecen una relación entre ambos elementos y su contenido.

1.5 Nuevos y viejos elementos

HTML5 fue desarrollado con la intención de simplificar, especificar y organizar el código. Para lograr este propósito, nuevas etiquetas y atributos fueron agregados y HTML fue completamente integrado a CSS y Javascript. Estas incorporaciones y mejoras de versiones previas están relacionadas no solo con nuevos elementos sino también con cómo usamos los ya existentes.

<mark>

La etiqueta **<mark>** fue agregada para resaltar parte de un texto que originalmente no era considerado importante pero ahora es relevante acorde con las acciones del usuario. El ejemplo que más se ajusta a este caso es un resultado de búsqueda. El elemento **<mark>** resaltará la parte del texto que concuerda con el texto buscado:

```
<span>Mi <mark>coche</mark> es rojo</span>
```

Listado 1-19. *Uso del elemento <mark> para resaltar la palabra "coche".*

Si un usuario realiza una búsqueda de la palabra "coche", por ejemplo, los resultados podrían ser mostrados con el código del Listado 1-19. La frase del ejemplo representa los resultados de la búsqueda y las etiquetas **<mark>** en el medio encierran lo que era el texto buscado (la palabra "coche"). En algunos navegadores, esta palabra será resaltada con un fondo amarillo por defecto, pero siempre podemos sobrescribir estos estilos con los nuestros utilizando CSS, como veremos en próximos capítulos.

En el pasado, normalmente obteníamos similares resultados usando el elemento **<b>**. El agregado de **<mark>** tiene el objetivo de cambiar el significado y otorgar un nuevo propósito para éstos y otros elementos relacionados:

- **** es para indicar énfasis (reemplazando la etiqueta **<i>** que utilizábamos anteriormente).
- **** es para indicar importancia.

- **<mark>** es para resaltar texto que es relevante de acuerdo con las circunstancias.
- **<b>** debería ser usado solo cuando no hay otro elemento más apropiado para la situación.

<small>

La nueva especificidad de HTML es también evidente en elementos como **<small>**. Previamente este elemento era utilizado con la intención de presentar cualquier texto con letra pequeña. La palabra clave referenciaba el tamaño del texto, independientemente de su significado. En HTML5, el nuevo propósito de **<small>** es presentar la llamada letra pequeña, como impresiones legales, descargos, etc…

```
<small>Derechos Reservados &copy; 2011 MinkBooks</small>
```

Listado 1-20. *Inclusión de información legal con el elemento <small>.*

<cite>

Otro elemento que ha cambiado su naturaleza para volverse más específico es **<cite>**. Ahora las etiquetas **<cite>** encierran el título de un trabajo, como un libro, una película, una canción, etc…

```
<span>Amo la película <cite>Tentaciones</cite></span>
```

Listado 1-21. *Citando una película con el elemento <cite>.*

<address>

El elemento **<address>** es un viejo elemento convertido en un elemento estructural. No necesitamos usarlo previamente para construir nuestra plantilla, sin embargo podría ubicarse perfectamente en algunas situaciones en las que debemos presentar información de contacto relacionada con el contenido del elemento **<article>** o el cuerpo completo.

Este elemento debería ser incluido dentro de **<footer>,** como en el siguiente ejemplo:

```
<article>
  <header>
```

```
      <h1>Título del mensaje </h1>
    </header>
    Este es el texto del mensaje
    <footer>
      <address>
        <a href="http://www.jdgauchat.com">JD Gauchat</a>
      </address>
    </footer>
  </article>
```

Listado 1-22. *Agregando información de contacto a un <article>.*

<time>

En cada **<article>** de nuestra última plantilla (Listado 1-18), incluimos la fecha indicando cuándo el mensaje fue publicado. Para esto usamos un simple elemento **<p>** dentro de la cabecera (**<header>**) del mensaje, pero existe un elemento en HTML5 específico para este propósito. El elemento **<time>** nos permite declarar un texto comprensible para humanos y navegadores que representa fecha y hora:

```
<article>
  <header>
    <h1>Título del mensaje dos</h1>
    <time datetime="2011-10-12" pubdate>publicado 12-10-2011</time>
  </header>
  Este es el texto del mensaje
</article>
```

Listado 1-23. *Fecha y hora usando el elemento <time>.*

En el Listado 1-23, el elemento **<p>** usado en ejemplos previos fue reemplazado por el nuevo elemento **<time>** para mostrar la fecha en la que el mensaje fue publicado. El atributo **datetime** tiene el valor que representa la fecha comprensible para el navegador (timestamp). El formato de este valor deberá seguir un patrón similar al del siguiente ejemplo: **2011-10-12T12:10:45**. También incluimos el atributo **pubdate**, el cual solo es agregado para indicar que el valor del atributo **datetime** representa la fecha de publicación.

1.6 Referencia rápida

En la especificación HTML5, HTML está a cargo de la estructura del documento y provee un grupo completo de nuevos elementos para este propósito. La especificación también incluye algunos elementos con la única tarea de proveer estilos. Esta es una lista de los que consideramos más relevantes:

IMPORTANTE: Para una completa referencia de los elementos HTML incluidos en la especificación, visite nuestro sitio web y siga los enlaces correspondientes a este capítulo.

<header> Este elemento presenta información introductoria y puede ser aplicado en diferentes secciones del documento. Tiene el propósito de contener la cabecera de una sección pero también puede ser utilizado para agrupar índices, formularios de búsqueda, logos, etc…

<nav> Este elemento indica una sección de enlaces con propósitos de navegación, como menús o índices. No todos los enlaces dentro de una página web tienen que estar dentro de un elemento `<nav>`, solo aquellos que forman partes de bloques de navegación.

<section> Este elemento representa una sección general del documento. Es usualmente utilizado para construir varios bloques de contenido (por ejemplo, columnas) con el propósito de ordenar el contenido que comparte una característica específica, como capítulos o páginas de un libro, grupo de noticias, artículos, etc…

<aside> Este elemento representa contenido que está relacionado con el contenido principal pero no es parte del mismo. Ejemplos pueden ser citas, información en barras laterales, publicidad, etc…

<footer> Este elemento representa información adicional sobre su elemento padre. Por ejemplo, un elemento `<footer>` insertado al final del cuerpo proveerá información adicional sobre el cuerpo del documento, como el pie normal de una página web. Puede ser usado no solo para el cuerpo sino también para diferentes secciones dentro del cuerpo, otorgando información adicional sobre estas secciones específicas.

<article> Este elemento representa una porción independiente de información relevante (por ejemplo, cada artículo de un periódico o cada entrada de un blog). El elemento `<article>` puede ser anidado y usado para mostrar una lista dentro de otra lista de ítems relacionados, como comentarios de usuarios en entradas de blogs, por ejemplo.

<hgroup> Este elemento es usado para agrupar elementos H cuando la cabecera tiene múltiples niveles (por ejemplo, una cabecera con título y subtítulo).

<figure> Este elemento representa una porción independiente de contenido (por ejemplo, imágenes, diagramas o videos) que son referenciadas desde el contenido principal. Esta es información que puede ser removida sin afectar el fluido del resto del contenido.

<figcaption> Este elemento es utilizado para mostrar una leyenda o pequeño texto relacionado con el contenido de un elemento `<figure>`, como la descripción de una imagen.

<mark> Este elemento resalta un texto que tiene relevancia en una situación en particular o que ha sido mostrado en respuesta de la actividad del usuario.

<small> Este elemento representa contenido al margen, como letra pequeña (por ejemplo, descargos, restricciones legales, declaración de derechos, etc…).

<cite> Este elemento es usado para mostrar el título de un trabajo (libro, película, poema, etc…).

<address> Este elemento encierra información de contacto para un elemento `<article>` o el documento completo. Es recomendable que sea insertado dentro de un elemento `<footer>`.

<time> Este elemento se utiliza para mostrar fecha y hora en formatos comprensibles por los usuarios y el navegador. El valor para los usuarios es ubicado entre las etiquetas mientras que el específico para programas y navegadores es incluido como el valor del atributo `datetime`. Un segundo atributo optativo llamado `pubdate` es usado para indicar que el valor de `datetime` es la fecha de publicación.

Capítulo 2

Estilos CSS y modelos de caja

2.1 CSS y HTML

Como aclaramos anteriormente, la nueva especificación de HTML (HTML5) no describe solo los nuevos elementos HTML o el lenguaje mismo. La web demanda diseño y funcionalidad, no solo organización estructural o definición de secciones. En este nuevo paradigma, HTML se presenta junto con CSS y Javascript como un único instrumento integrado.

La función de cada tecnología ya ha sido explicada en capítulos previos, así como los nuevos elementos HTML responsables de la estructura del documento. Ahora es momento de analizar CSS, su relevancia dentro de esta unión estratégica y su influencia sobre la presentación de documentos HTML.

Oficialmente CSS nada tiene que ver con HTML5. CSS no es parte de la especificación y nunca lo fue. Este lenguaje es, de hecho, un complemento desarrollado para superar las limitaciones y reducir la complejidad de HTML. Al comienzo, atributos dentro de las etiquetas HTML proveían estilos esenciales para cada elemento, pero a medida que el lenguaje evolucionó, la escritura de códigos se volvió más compleja y HTML por sí mismo no pudo más satisfacer las demandas de diseñadores. En consecuencia, CSS pronto fue adoptado como la forma de separar la estructura de la presentación. Desde entonces, CSS ha crecido y ganado importancia, pero siempre desarrollado en paralelo, enfocado en las necesidades de los diseñadores y apartado del proceso de evolución de HTML.

La versión 3 de CSS sigue el mismo camino, pero esta vez con un mayor compromiso. La especificación de HTML5 fue desarrollada considerando CSS a cargo del diseño. Debido a esta consideración, la integración entre HTML y CSS es ahora vital para el desarrollo web y esta es la razón por la que cada vez que mencionamos HTML5 también estamos haciendo referencia a CSS3, aunque oficialmente se trate de dos tecnologías completamente separadas.

En este momento las nuevas características incorporadas en CSS3 están siendo implementadas e incluidas junto al resto de la especificación en navegadores compatibles con HTML5. En este capítulo, vamos a estudiar conceptos básicos de CSS y las nuevas técnicas de CSS3 ya disponibles para presentación y estructuración. También aprenderemos cómo utilizar los nuevos selectores y pseudo clases que hacen más fácil la selección e identificación de elementos HTML.

Conceptos básicos: CSS es un lenguaje que trabaja junto con HTML para proveer estilos visuales a los elementos del documento, como tamaño, color, fondo, bordes, etc…

IMPORTANTE: En este momento las nuevas incorporaciones de CSS3 están siendo implementadas en las últimas versiones de los navegadores más populares, pero algunas de ellas se encuentran aún en estado experimental. Por esta razón, estos nuevos estilos deberán ser precedidos por prefijos tales como `-moz-` o `-webkit-` para ser efectivamente interpretados. Analizaremos este importante asunto más adelante.

2.2 Estilos y estructura

A pesar de que cada navegador garantiza estilos por defecto para cada uno de los elementos HTML, estos estilos no necesariamente satisfacen los requerimientos de cada diseñador. Normalmente se encuentran muy distanciados de lo que queremos para nuestros sitios webs. Diseñadores y desarrolladores a menudo deben aplicar sus propios estilos para obtener la organización y el efecto visual que realmente desean.

IMPORTANTE: En esta parte del capítulo vamos a revisar estilos CSS y explicar algunas técnicas básicas para definir la estructura de un documento. Si usted ya se encuentra familiarizado con estos conceptos, siéntase libre de obviar las partes que ya conoce.

Elementos block

Con respecto a la estructura, básicamente cada navegador ordena los elementos por defecto de acuerdo a su tipo: *block* (bloque) o *inline* (en línea). Esta clasificación está asociada con la forma en que los elementos son mostrados en pantalla.

- **Elementos *block*** son posicionados uno sobre otro hacia abajo en la página.
- **Elementos *inline*** son posicionados lado a lado, uno al lado del otro en la misma línea, sin ningún salto de línea a menos que ya no haya más espacio horizontal para ubicarlos.

Casi todos los elementos estructurales en nuestros documentos serán tratados por los navegadores como elementos *block* por defecto. Esto significa que cada elemento HTML que representa una parte de la organización visual (por ejemplo, `<section>`, `<nav>`, `<header>`, `<footer>`, `<div>`) será posicionado debajo del anterior.

En el Capítulo 1 creamos un documento HTML con la intención de reproducir un sitio web tradicional. El diseño incluyó barras horizontales y dos columnas en el medio. Debido a la forma en que los navegadores muestran estos elementos por defecto, el resultado en la

pantalla está muy lejos de nuestras expectativas. Tan pronto como el archivo HTML con el código del Listado 1-18, Capítulo 1, es abierto en el navegador, la posición errónea en la pantalla de las dos columnas definidas por los elementos `<section>` y `<aside>` es claramente visible. Una columna está debajo de la otra en lugar de estar a su lado, como correspondería. Cada bloque (*block*) es mostrado por defecto tan ancho como sea posible, tan alto como la información que contiene y uno sobre otro, como se muestra en la Figura 2-1.

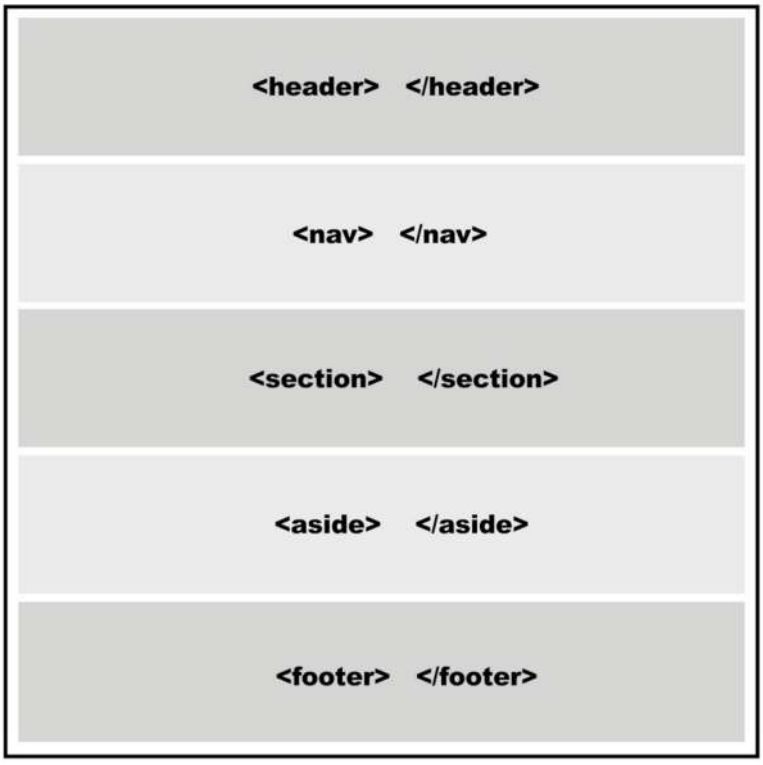

Figura 2-1. *Representación visual de una página web mostrada con estilos por defecto.*

Modelos de caja

Para aprender cómo podemos crear nuestra propia organización de los elementos en pantalla, debemos primero entender cómo los navegadores procesan el código HTML. Los navegadores consideran cada elemento HTML como una caja. Una página web es en realidad un grupo de cajas ordenadas siguiendo ciertas reglas. Estas reglas son establecidas por estilos provistos por los navegadores o por los diseñadores usando CSS.

CSS tiene un set predeterminado de propiedades destinados a sobrescribir los estilos provistos por navegadores y obtener la organización deseada. Estas propiedades no son específicas, tienen que ser combinadas para formar reglas que luego serán usadas para agrupar cajas y obtener la correcta disposición en pantalla. La combinación de estas reglas

es normalmente llamada modelo o sistema de disposición. Todas estas reglas aplicadas juntas constituyen lo que se llama un modelo de caja.

Existe solo un modelo de caja que es considerado estándar estos días, y muchos otros que aún se encuentran en estado experimental. El modelo válido y ampliamente adoptado es el llamado Modelo de Caja Tradicional, el cual ha sido usado desde la primera versión de CSS.

Aunque este modelo ha probado ser efectivo, algunos modelos experimentales intentan superar sus deficiencias, pero la falta de consenso sobre el reemplazo más adecuado aún mantiene a este viejo modelo en vigencia y la mayoría de los sitios webs programados en HTML5 lo continúan utilizando.

2.3 Conceptos básicos sobre estilos

Antes de comenzar a insertar reglas CSS en nuestro archivo de estilos y aplicar un modelo de caja, debemos revisar los conceptos básicos sobre estilos CSS que van a ser utilizados en el resto del libro.

Aplicar estilos a los elementos HTML cambia la forma en que estos son presentados en pantalla. Como explicamos anteriormente, los navegadores proveen estilos por defecto que en la mayoría de los casos no son suficientes para satisfacer las necesidades de los diseñadores. Para cambiar esto, podemos sobrescribir estos estilos con los nuestros usando diferentes técnicas.

> **Conceptos básicos:** En este libro encontrará solo una introducción breve a los estilos CSS. Solo mencionamos las técnicas y propiedades que necesita conocer para entender los temas y códigos estudiados en próximos capítulos. Si considera que no tiene la suficiente experiencia en CSS y necesita mayor información visite nuestro sitio web y siga los enlaces correspondientes a este capítulo.
>
> **Hágalo usted mismo:** Dentro de un archivo de texto vacío, copie cada código HTML estudiado en los siguientes listados y abra el archivo en su navegador para comprobar su funcionamiento. Tenga en cuenta que el archivo debe tener la extensión `.html` para ser abierto y procesado correctamente.

Estilos en línea

Una de las técnicas más simples para incorporar estilos CSS a un documento HTML es la de asignar los estilos dentro de las etiquetas por medio del atributo `style`.

El Listado 2-1 muestra un documento HTML simple que contiene el elemento `<p>` modificado por el atributo `style` con el valor `font-size: 20px`. Este estilo cambia el tamaño por defecto del texto dentro del elemento `<p>` a un nuevo tamaño de 20 pixeles.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Este es el título del documento</title>
</head>
<body>
  <p style="font-size: 20px">Mi texto</p>
</body>
</html>
```

Listado 2-1. *Estilos CSS dentro de etiquetas HTML.*

Usar la técnica demostrada anteriormente es una buena manera de probar estilos y obtener una vista rápida de sus efectos, pero no es recomendado para aplicar estilos a todo el documento. La razón es simple: cuando usamos esta técnica, debemos escribir y repetir cada estilo en cada uno de los elementos que queremos modificar, incrementando el tamaño del documento a proporciones inaceptables y haciéndolo imposible de mantener y actualizar. Solo imagine lo que ocurriría si decide que en lugar de 20 pixeles el tamaño de cada uno de los elementos **<p>** debería ser de 24 pixeles. Tendría que modificar cada estilo en cada etiqueta **<p>** en el documento completo.

Estilos embebidos

Una mejor alternativa es insertar los estilos en la cabecera del documento y luego usar referencias para afectar los elementos HTML correspondientes:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Este texto es el título del documento</title>
  <style>
    p { font-size: 20px }
  </style>
</head>
<body>
  <p>Mi texto</p>
</body>
</html>
```

Listado 2-2. *Estilos listados en la cabecera del documento.*

El elemento **<style>** (mostrado en el Listado 2-2) permite a los desarrolladores agrupar estilos CSS dentro del documento. En versiones previas de HTML era necesario

especificar qué tipo de estilos serían insertados. En HTML5 los estilos por defecto son CSS, por lo tanto no necesitamos agregar ningún atributo en la etiqueta de apertura **<style>**.

El código resaltado del Listado 2-2 tiene la misma función que la línea de código del Listado 2-1, pero en el Listado 2-2 no tuvimos que escribir el estilo dentro de cada etiqueta **<p>** porque todos los elementos **<p>** ya fueron afectados. Con este método, reducimos nuestro código y asignamos los estilos que queremos a elementos específicos utilizando referencias. Veremos más sobre referencias en este capítulo.

Archivos externos

Declarar los estilos en la cabecera del documento ahorra espacio y vuelve al código más consistente y actualizable, pero nos requiere hacer una copia de cada grupo de estilos en todos los documentos de nuestro sitio web. La solución es mover todos los estilos a un archivo externo y luego utilizar el elemento **<link>** para insertar este archivo dentro de cada documento que los necesite. Este método nos permite cambiar los estilos por completo simplemente incluyendo un archivo diferente. También nos permite modificar o adaptar nuestros documentos a cada circunstancia o dispositivo, como veremos al final del libro.

En el Capítulo 1, estudiamos la etiqueta **<link>** y cómo utilizarla para insertar archivos con estilos CSS en nuestros documentos. Utilizando la línea **<link rel="stylesheet" href="misestilos.css">** le decimos al navegador que cargue el archivo **misestilos.css** porque contiene todos los estilos necesitados para presentar el documento en pantalla. Esta práctica fue ampliamente adoptada por diseñadores que ya están trabajando con HTML5. La etiqueta **<link>** referenciando el archivo CSS será insertada en cada uno de los documentos que requieren de esos estilos:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Este texto es el título del documento</title>
  <link rel="stylesheet" href="misestilos.css">
</head>
<body>
  <p>Mi texto</p>
</body>
</html>
```

Listado 2-3. *Aplicando estilos CSS desde un archivo externo.*

> **Hágalo usted mismo:** De ahora en adelante agregaremos estilos CSS al archivo llamado **misestilos.css**. Debe crear este archivo en el mismo directorio (carpeta) donde se encuentra el archivo HTML y copiar los estilos CSS en su interior para comprobar cómo trabajan.