

# LENGUAJES DE MARCAS

## 4.1 Introducción

Los lenguajes de marcas [10], también denominados lenguajes de marcado o lenguajes de descripción de documentos, definen la estructura, la semántica y controlan el procesamiento de un documento digital. No son lenguajes de formato similares a los lenguajes que se usan en Internet como los de descripción de páginas (archivos PostScript, archivos .pdf, etc.) ni son lenguajes de programación (Java, Perl, C++...).

En un documento existen distintos niveles de información: por un lado, los datos que conforman el contenido de un documento (caracteres de contenido), y por otro, una información superpuesta al contenido, que es lo que constituye el etiquetado, marcado o “*markup*” (caracteres de etiquetado). En general una etiqueta indica el principio de un elemento y otra el final del mismo.

Un lenguaje de marcado cumple con dos objetivos esenciales a la hora de diseñar y procesar un documento digital:

- Especifica las operaciones tipográficas y las funciones que debe ejecutar el programa navegador/visualizador sobre dichos elementos. Las operaciones tipográficas son instrucciones de formato que se aplican a cada uno de los elementos de un documento digital como, por ejemplo, imprimir un título en negrita y a un determinado tamaño.
- Separa un texto en los elementos de los que se compone, como por ejemplo un párrafo, un capítulo, un encabezamiento, etc.

Los lenguajes de marcas surgieron dada la gran variedad de formatos existentes a la hora de archivar documentos y de los problemas que se planteaban cuando se intercambiaban archivos con las consiguientes pérdidas de formato y otras características. Se acordó, pues, que las marcas deberían constar de unos caracteres ASCII especialmente reservados para ello y que deberían seguirse unas reglas sintácticas especiales.

Así nació un lenguaje especial normalizado internacional para la formulación de este tipo de reglas llamado *Standard Generalized Markup Language* o SGML [11]. SGML no es pues, un lenguaje en sí mismo, sino un metalenguaje y de él se deriva el lenguaje HTML [12] o lenguaje de marcas de hipertexto, y otros muchos lenguajes.

Existen 3 utilizaciones básicas de los lenguajes de marcas:

- Los que sirven principalmente para describir el contenido (por ejemplo, las bases de datos).
- Los que sirven para definir el formato (por ejemplo, los procesadores de textos).
- Los que realizan las dos funciones indistintamente (por ejemplo, el lenguaje HTML).

En el siguiente apartado se estudian los diferentes tipos de lenguaje de marcas que existen, su estructura, tipo de página que es capaz de crear y ejemplos de navegadores capaces de interpretarlo.

## 4.2 Tipos de lenguajes de marcas

En este apartado se analizan los lenguajes HTML, XML [13], XHTML [14] y también los usados en entornos móviles como WML [15], cHTML [16], XHTML Basic y XHTML MP [17], que son variaciones de XHTML.

### 4.2.1 HTML

El HTML es uno de los primeros lenguajes de marcas que surgieron. Este lenguaje servirá como base para explicar la estructura general de los lenguajes de marcado.

#### 4.2.1.1 Estructura básica de un documento HTML

HTML (*Hipertext Markup Language*) o lenguaje de marcas de hipertexto es el lenguaje más usado en la *World Wide Web*.

La mayoría de los efectos se especifican de la misma forma: rodeando el texto que se quiere marcar entre dos etiquetas o directivas (*tags*, en inglés), que definen el efecto o unidad lógica que se desea. Las etiquetas están formadas por determinados códigos metidos entre los signos “<” y “>”, y con la barra “/” cuando se trata de la segunda etiqueta de un efecto (la de cierre). Ciertas directivas sólo se ponen una vez, en el lugar del texto donde queramos que aparezca su efecto. La estructura es:

```
<TAG> . . . </TAG>
```

A veces es necesario ofrecer datos adicionales en una directiva. Para ello se incluyen parámetros en la directiva inicial (la de apertura). Un tag le dice al navegador qué hacer, un atributo le dice cómo hacerlo. Los tags pueden tener uno ó más atributos:

```
<TAG ATRIBUTO1="..." ATRIBUTO2="..." >...</TAG>
```

Los tags pueden contener otros tags. La forma correcta de escribirlas es:

```
<A><B></B></A>
```

Un documento HTML comienza con la etiqueta <html>, y termina con </html>. Dentro del documento (todo lo que queda entre las etiquetas de principio y fin de html), hay dos zonas bien diferenciadas:

- El encabezamiento, delimitado por <head> y </head>, que sirve para definir diversos valores válidos en todo el documento. Lleva la directiva <title>, que permite especificar el título del documento.
- El cuerpo, delimitado por <body> y </body>, donde reside la información del documento.

Casi toda la información que contiene un documento HTML se distribuye en el cuerpo. El conjunto de etiquetas existentes, sin entrar en más detalle, está representado en la tabla siguiente:

<b>Texto</b>	p, h1, h2, h3, h4, h5, h6, strong, em, abbr, a cronym , address , bdo, blockquote, cite, q, code, ins, del, dfn ,kbd ,pre ,samp ,var , br
<b>Links</b>	a, base
<b>Imágenes y objetos</b>	Img, area, map, object, param
<b>Listas</b>	ul, ol, li, dl, dt, dd
<b>Tablas</b>	table, tr, td, th, tbody, thead, tfoot, col, colgroup, caption
<b>Formularios</b>	form, input, textarea, select, option, optgroup, button, label, fieldset, legend
<b>Scripting</b>	script, noscript
<b>Presentación</b>	b, i, tt, sub, sup, big, small, hr

**Tabla 4.1** Etiquetas HTML.

En el siguiente apartado se explica la historia del estándar HTML.

#### 4.2.1.2 Versiones de HTML

No hay especificación oficial del HTML 1.0 porque ya existían múltiples estándares informales del HTML cuando se decidió crear un estándar oficial. Los trabajos para crear un sucesor del HTML, posteriormente llamado “HTML+”, comenzaron a finales de 1993. El HTML+ se diseñó originalmente para ser un superconjunto del HTML que permitiera evolucionar gradualmente desde el formato HTML anterior. A la primera especificación formal de HTML+ se le dio, por lo tanto, el número de versión 2.0 para distinguirla de esos “estándares no oficiales” previos. Los trabajos sobre HTML+ continuaron, pero nunca se convirtió en un estándar.

En 1994 se funda el Consorcio *World Wide Web* (W3C), asociación internacional formada por organizaciones miembro del consorcio, personal y el público en general, que trabajan conjuntamente para desarrollar estándares Web, y dentro de ellos el HTML.

El borrador del estándar HTML 3.0 fue propuesto por el recién formado *World Wide Web Consortium* en marzo de 1995. Se introdujeron muchas nuevas capacidades, tales como facilidades para crear tablas, hacer que el texto fluyese alrededor de las figuras y mostrar elementos matemáticos complejos. Aunque se diseñó para ser compatible con HTML 2.0, era demasiado complejo para ser implementado con la tecnología de la época y, cuando el borrador del estándar expiró en septiembre de 1995, se abandonó debido a la carencia de apoyos de los fabricantes de navegadores web. El HTML 3.1 nunca llegó a ser propuesto oficialmente, y el estándar siguiente fue el HTML 3.2, que abandonaba la mayoría de las nuevas características del HTML 3.0 y, a cambio, adoptaba muchos elementos desarrollados inicialmente por los navegadores web Netscape y Mosaic. La posibilidad de trabajar con fórmulas matemáticas que se había propuesto en el HTML 3.0 pasó a quedar integrada en un estándar distinto llamado MathML.

El HTML 4.0 también adoptó muchos elementos específicos desarrollados inicialmente para un navegador web concreto, pero al mismo tiempo comenzó a limpiar el HTML señalando algunos de ellos como “desaprobados” (*deprecated*).

La herencia del HTML se mantiene en XHTML, que se basa en XML y que se verá en un apartado posterior. A continuación se muestra un ejemplo de página HTML.

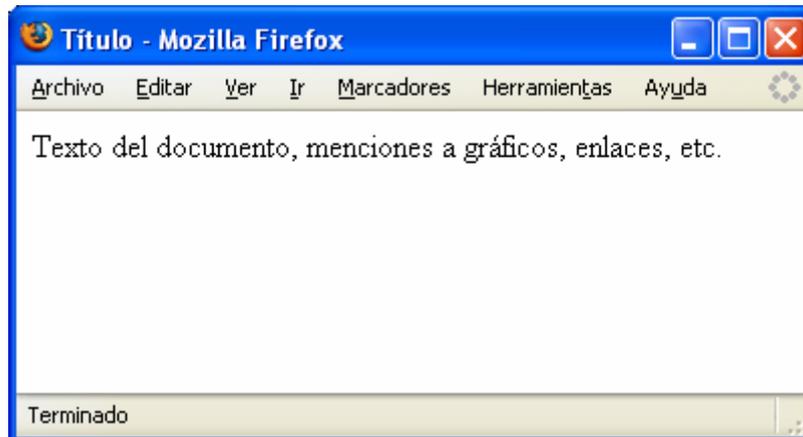
#### 4.2.1.3 Ejemplo de página HTML

El siguiente código es un ejemplo de página escrita en HTML:

```
<html>
<head>
<title>Título</title>
</head>
<body>
Texto del documento, menciones a gráficos, enlaces, etc.
</body>
</html>
```

**Figura 4.1** Ejemplo de documento HTML.

En la siguiente figura se muestra la ejecución de ese código HTML. Para ello se ha usado el navegador Firefox 1.5.



**Figura 4.2** Visualización de documento HTML.

Este ejemplo sólo contenía un título y un poco de texto plano. En el siguiente apartado se incluye un ejemplo más complejo que demuestra el potencial de este lenguaje.

#### **4.2.1.4 Ejemplo complejo de HTML**

La página del buscador Google tiene versiones en todos los lenguajes de marcas existentes, como HTML, cHTML, WML, XHTML, etc. Algunos de ellos pertenecientes al ámbito móvil.

Gracias a la versatilidad de este buscador se utilizarán sus distintas versiones para comparar los lenguajes. De especial interés resulta comprobar cómo los lenguajes pensados para dispositivos móviles adaptan el diseño de las páginas a su ámbito de funcionamiento.

La versión en HTML de Google se encuentra en la dirección <http://www.google.es>. Es una página mucho más compleja que el ejemplo anterior, incluye imágenes, cuadros de texto, botones, enlaces, y otros efectos. El código de la página es demasiado extenso y no es relevante para este proyecto, por eso no se ha incluido en esta memoria. Para consultarlo puede usarse cualquier navegador que sea capaz de interpretar este lenguaje.

En nuestro caso se ha utilizado el navegador Firefox 1.5 para visualizar la página. En la siguiente figura puede verse el resultado.

[Página principal personalizada](#) | [Acceder](#)



[La Web](#) [Imágenes](#) [Grupos](#) [Noticias](#) [Más »](#)  
 [Búsqueda avanzada](#)  
  [Preferencias](#)  
[Herramientas del idioma](#)  
 Búsqueda:  la Web  páginas en español  páginas de España

Google.es ofrecido en: [Català](#) [galego](#) [euskara](#)

[Publicidad](#) - [Soluciones Empresariales](#) - [Todo acerca de Google](#) - [Google.com in English](#)

©2007 Google

**Figura 4.3** Google en HTML.

#### 4.2.2 XML y XHTML

XML, acrónimo en inglés de *eXtensible Markup Language* (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium* (W3C).

XML se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

En XML se separa el contenido de la presentación de forma total. Para representar sus documentos existen una serie de estándares acompañantes:

- **Klink** describe un método estándar para añadir hiperenlaces a un fichero XML.
- **Xpointer** y **Xfragments** son sintaxis para señalar partes de un documento XML.
- **XSL** es un lenguaje desarrollado para la generación de hojas de estilo.
- **DOM** es un conjunto de estándares de llamadas de funciones para la manipulación de ficheros XML y HTML a partir de un lenguaje de programación.
- **XML Namespace** es una especificación que describe cómo puede conectarse un URL con los diversos tags y atributos de un documento XML.
- **XML Schemas** ofrecen soporte a los desarrolladores para una definición exacta de sus propios formatos basados en XML.

El lenguaje estandarizado XHTML 1.0 del W3C es lenguaje HTML con la estructura de datos XML. Esto significa que todos los elementos de HTML 4.0 contenidos en XHTML están definidos de tal forma que cumplen con las especificaciones y requisitos de XML. La diferencia principal entre un documento XML y un documento HTML es que el XML contiene datos que se autodefinen exclusivamente, mientras que un documento HTML contiene datos mal definidos, mezclados con elementos de formato.

La ventaja de basarse en XML es que mediante XHTML será más difícil para los creadores de navegadores apartarse de las especificaciones de W3C mediante extensiones propietarias. La definición XHTML-DTD (*Document Type Definition*) indica con exactitud cómo funcionan los diversos tags existentes. Para los programadores, esto significa más seguridad en la publicación de documentos, pues cuando un fichero XHTML funciona, lo hace en todos los clientes.

Los documentos XML y XHTML son documentos “bien formados” (del inglés *well formed*), es decir que cumplen con todas las definiciones básicas de formato y pueden, por lo tanto, ser analizados correctamente por cualquier “parser” (analizador de código) que cumpla con la norma. Por otro lado está el concepto de “validez”, que implica que no sólo el documento es bien formado sino que también su estructura se corresponde con la definida en un documento externo (expresada como DTD o como XSchema).

Para que un documento esté “bien formado” debe tener las siguientes propiedades:

- Los documentos han de seguir una estructura estrictamente jerárquica con lo que respecta a las etiquetas que delimitan sus elementos. Una etiqueta debe estar correctamente incluida en otra, es decir, las etiquetas deben estar correctamente anidadas. Los elementos con contenido deben estar correctamente cerrados.
- Se permite elementos sin contenido pero la etiqueta debe ser de la forma: `<elemento_sin_contenido/>`
- Sólo se permite un elemento raíz del que todos los demás sean parte, es decir, sólo puede tener un elemento inicial.
- Los valores atributos siempre deben estar encerrados entre comillas simples o dobles.
- Son sensibles a mayúsculas y minúsculas. Existe un conjunto de caracteres llamados espacios en blanco (espacios, tabuladores, retornos de carro, saltos de línea) que los procesadores tratarán de forma diferente.
- Es necesario asignar nombres a las estructuras, tipos de elementos, entidades, elementos particulares, etc. Los nombres deben tener alguna característica en común.
- Las construcciones como etiquetas, referencias de entidad y declaraciones se denominan marcas; son partes del documento que el procesador espera entender. El resto del documento entre marcas son los datos entendibles por las personas.

A continuación se muestra un ejemplo para entender la estructura de un documento XML:

```
<?xml version="1.0"?>
<!DOCTYPE MENSAJE SYSTEM "mensaje.dtd">
<mensaje>
  <remitente>
    <nombre>Alfredo Reino</nombre>
    <mail>alf@ibium.com</mail>
  </remitente>
  <destinatario>
    <nombre>Bill Clinton</nombre>
    <mail>president@WhiteHouse.gov</mail>
  </destinatario>
  <asunto>Hola Bill</asunto>
  <texto>
    <parrafo>¿Hola que tal? Hace <enfasis>mucho</enfasis>
    que no escribes. A ver si llamas y quedamos para tomar
    algo. </parrafo>
  </texto>
</mensaje>
```

**Figura 4.4** Ejemplo de documento XML.

Básicamente, XHTML es un documento HTML bajo la condición de estar “bien formado”. De hecho las únicas diferencias entre HTML y XHTML son:

- Etiqueta <DOCTYPE> a elegir entre una de las tres distintas DTD:

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

<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

**Figura 4.5** Formas de definir un documento XHTML.

- Todas las etiquetas deben cerrarse. Si un elemento consta de una única etiqueta, deberá contener el carácter de cierre (“/”).
- No se permite abreviar los atributos, todos deben tener un valor y este debe de estar entrecomillado.
- El código es sensible a mayúsculas y minúsculas. Los nombres y atributos de todas las etiquetas deben estar escritos en minúsculas.

### 4.2.3 Lenguajes de marcas en entornos móviles

Debido a las restricciones de los dispositivos móviles no se suele usar HTML o XHTML de manera completa, sino que se definen nuevos tipos de documentos que vienen a ser una simplificación de los anteriores:

- Para WAP 1.x se crea WML (*Wireless Markup Language*).
- Para WAP 2.0 se crean XHTML Basic y XHTML-MP (*Mobile Profile*).
- Para i-Mode se crea cHTML (*Compact HTML*).

La generación del estándar WAP 2.0, desarrollada por el WAP Forum, se basa en las versiones reducidas de XHTML. De esta forma, se agrupan los dos mundos de Internet: el WAP móvil y la Web de acceso tradicional. Como XHTML es compatible con WAP, es el próximo desarrollo lógico para las operadoras de red y los proveedores de servicios. El modelo Nokia 7360, lanzado en el año 2006, fue de los primeros en incorporar un navegador capaz de interpretarlo. Actualmente desde el Instituto Internacional de Ingeniería [18], se está desarrollando un microbrowser XHTML escrito en J2ME.

#### 4.2.3.1 WML

*Wireless Markup Language* (WML) se utiliza para construir las páginas que aparecen en las pantallas de los teléfonos móviles y los asistentes personales digitales (PDA) dotados de tecnología WAP. La visualización de la página dependerá del dispositivo que se use y de la forma en que este interprete el código, ya que varían entre sí.

##### 4.2.3.1.1 Características principales de WML

Los documentos escritos en WML tienen las siguientes características:

- Soporte para imágenes y texto, con posibilidad de texto con formato.
- Tarjetas agrupadas en barajas. Una página WML es como una página HTML en la que hay una serie de cartas, al conjunto de estas cartas se le suele llamar baraja.
- Posibilidad de navegar entre cartas y barajas de la misma forma que se navega entre páginas Web.
- Manejo de variables y formularios para el intercambio de información entre el teléfono celular y el servidor.
- WML es un lenguaje de marcas perteneciente al estándar XML 1.0, esto conlleva que WML debe cumplir con la sintaxis de XML 1.0.
- Todos los elementos de WML son sensibles a mayúsculas/minúsculas, esto incluye las etiquetas, los atributos, los identificadores, las variables...
- El conjunto de caracteres definido por defecto es el ISO/IEC-10646, que es el mismo que el Unicode 2.0 WAP soporta los siguientes subconjuntos de Unicode:
  - UTF-8
  - ISO-8859-1 o ISO Latin-1
  - UCS-2

Se definen en la etiqueta `<encoding>`. Un ejemplo del uso de esta etiqueta sería:

```
<?xml version="1.0" encoding="UTF-8"?>
```

### 4.2.3.1.2 Etiquetas

Todas las etiquetas en WML se escriben en minúsculas. Hay dos tipos de etiquetas, las que contienen elementos, para lo cual hay una etiqueta de inicio, <etiqueta>, y otra de fin, </etiqueta> y las que no, que solo tienen la de fin, </etiqueta>. Los atributos de las etiquetas han de ir siempre en la etiqueta de inicio. Los comentarios, al igual que en HTML, tienen el siguiente formato: <!--Comentario-->.

### 4.2.3.1.3 Un ejemplo sencillo: Hola mundo

Para comprender cómo funciona una página WAP vamos a hacer un sencillo ejemplo con el que mostraremos el mensaje "Hola mundo". Vamos a explicar todas las etiquetas que hacen falta:

- <?xml version="1.0"?>. Indica que es un documento XML de versión 1.0 por lo tanto cumple todas las restricciones y reglas de los documentos XML.
- <!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml\_1.1.xml">. Indica el tipo de documento XML y donde localizar la especificación del tipo de documento.
- <wml>. Indica que comienza una página WML.
- <card id="t1" title="Tarjeta 1">. Indica que comienza una tarjeta que tiene como identificador "t1" y cuyo título es "Tarjeta 1". Más adelante se explicará qué es una tarjeta.
- <p>. Indica que comienza un párrafo de texto. A diferencia de HTML esta etiqueta es obligatoria si queremos escribir texto, además debe cerrarse con la correspondiente </p>
- Hola Mundo !. Este es el texto que aparecerá en nuestro navegador.
- </p>. Con esta etiqueta indicamos que el párrafo ha terminado y no vamos a escribir más texto dentro de este párrafo.
- </card>. Fin de la tarjeta.
- </wml>. Fin de la página WML.

El código completo es el siguiente:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="t1" title="Tarjeta 1">
  - <p>Hola mundo !</p>
</card>
</wml>
```

**Figura 4.6** Ejemplo de documento WML.

El resultado de ese código ejecutado por el microbrowser jBrowser, que está especializado en WML, se muestra en la siguiente figura:



**Figura 4.7** Hola mundo en WML.

#### 4.2.3.1.4 Barajas y cartas

En el ejemplo anterior se ha visto que existe el concepto de baraja y el de carta, su finalidad se explica en este apartado.

A las páginas WML se les suele llamar barajas porque están compuestas por cartas. Una carta es la unidad de información que un navegador WAP puede mostrar. El navegador nos permite pasar de una carta a otra dentro de la baraja para así poder acceder a todas las cartas.

Una baraja de cartas se marca con las etiquetas `<wml>` y `</wml>` dentro de estas marcas irán todas las cartas de la baraja. Puede contener las etiquetas `<head>`, `<template>` y es obligatorio que al menos tenga una etiqueta `<card>`.

Una carta es la unidad de información que se muestra en un navegador WAP, una carta puede contener texto, campos de datos, enlaces...

Las etiquetas de las cartas son `<card>` y `</card>`. Algunos de los atributos son "title", que nos permite indicar el título de la carta, "id", que nos proporciona una manera de identificar la carta. El atributo "id" es común para todas las etiquetas WML y nos permite identificar un elemento dentro de un documento WML. La estructura a base de cartas se muestra en el siguiente ejemplo:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="t1" title="Tarjeta 1">
  - <p>Hola mundo !</p>
</card>
<card id="t2" title="Otra tarjeta">
  - <p>Bienvenido</p>
</card>
</wml>
```

**Figura 4.8** Cartas en WML.

#### 4.2.3.1.5 Texto

El texto debe ir entre las etiquetas `<p>` y `</p>`, ya que así está definido en el DTD de WML. Podemos incluir saltos de línea con la etiqueta `<br/>`.

La etiqueta `<p>` puede llevar los siguientes atributos:

- align="" Puede contener los valores left, right y center. Indica la alineación del texto. Por defecto es left
- mode="" Puede ser warp o nowrap. warp significa que el texto puede ir en varias líneas y nowrap quiere decir que el texto no puede ser roto en varias líneas. Por defecto es warp

También hay que reseñar que aunque indiquemos el conjunto de caracteres que vamos a emplear, existen algunos que no pueden escribirse directamente. Entre ellos están los siguientes:

&#34;	"	Comillas
&#38;	&	Ampersand
&#39;	'	Apóstrofe
&#60;	<	Menor que
&#62;	>	Mayor que
&#160;		Espacio fijo

**Tabla 4.2** Caracteres en WML.

Un código que enseñar el uso de texto es el siguiente:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="t1" title="Tarjeta Texto">
  - <p>Esto es un <br/>ejemplo<br/> de &#34;texto&#34;.</p>
</card>
</wml>
```

**Figura 4.9** Código WML para representar texto.

La ejecución de dicho código da como resultado la imagen de la figura:



**Figura 4.10** Ejemplo de “texto” en WML.

#### 4.2.3.1.6 Formato de texto

Aunque los navegadores WAP estén muy limitados en lo referente al apartado visual se pueden hacer algunos efectos en el texto. Estas son las etiquetas para dar formato al texto:

Etiqueta apertura	Etiqueta cierre	Significado
<em>	</em>	Enfatizado
<strong>	</strong>	Fuerte
<i>	</i>	Cursiva
<b>	</b>	Negrita
<u>	</u>	Subrayado
<big>	</big>	Grande
<small>	</small>	Pequeño

**Tabla 4.3** Etiquetas de formato en WML.

Un código de ejemplo de texto con formato es el siguiente:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="t1" title="Formato de texto">
  - <p>Esto es un <i>texto</i> con algo de <b>formato</b> para
  - ver como <u>funciona</u>.</p>
</card>
</wml>
```

**Figura 4.11** Código WML para dar formato al texto.

La ejecución de dicho código da el siguiente resultado:



**Figura 4.12** Texto con formato en WML.

#### 4.2.3.1.7 Tablas

En WML también se pueden hacer tablas, aunque éstas están mucho más limitadas que en HTML. En WML no se puede poner tablas dentro de tablas y se debe especificar el número de columnas en la etiqueta <table>. Los atributos de la etiqueta <table> son:

- columns=numero. Con este atributo ponemos el número de columnas que tendrá nuestra tabla. Es obligatorio indicarlo.
- title="" Título de la tabla, puede ser representado o no por el navegador.
- align="" Con este atributo se especifica la alineación de las columnas dentro de la tabla. C significa centrado, L alineado a la izquierda y R alineado a la derecha. Se escribe una letra por cada columna, de manera que CCR, significaría que la primera y segunda columnas van centradas y la tercera alineada a la derecha.

Por cada columna pondremos una etiqueta `<tr>...</tr>` y por cada celda una etiqueta `<td>...</td>` tal y como se hace en HTML. Salvo que en WML las etiquetas `<tr>` y `<td>` solo pueden llevar los atributos genéricos de todas las etiquetas.

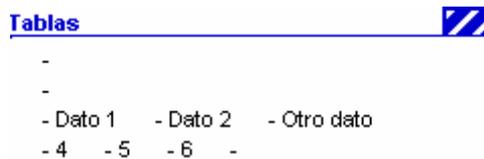
Un código de ejemplo para crear tablas es el siguiente:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="t1" title="Tablas">
  - <p>
  - <table columns="3">
<tr>
  - <td>Dato 1</td>
  - <td>Dato 2</td>
  - <td>Otro dato</td>
</tr>
<tr>
  - <td>4</td>
  - <td>5</td>
  - <td>6</td>
</tr>
</table>

  - </p>
</card>
</wml>
```

**Figura 4.13** Código WML para crear tablas.

La ejecución de dicho código da como resultado la siguiente imagen:



**Figura 4.14** Tablas en WML.

#### 4.2.3.1.8 Eventos y tareas

Los eventos y las tareas proporcionan un mecanismo para realizar acciones sobre las tarjetas, permitiéndonos navegar entre tarjetas y construir pequeños interfaces para dar más funcionalidad a las páginas.

El más importante de todos es el elemento `<do>`, que permite asignar una tarea sobre una acción. La etiqueta `<do>` puede contener una de las siguientes tareas: `<go>`, `<prev>`, `<noop>`, `<refresh>` que indican la acción que realiza la etiqueta.

Los atributos más importantes que contiene la etiqueta `<do>` son:

- `type=""`. indica sobre qué botón del navegador se aplica la acción, las más comunes son "accept", "prev" y "help".
- `label=""`. Texto que aparece asociado a la acción.
- `name=""`. Nombre de la acción, es imprescindible si vamos a asignar más de una acción a un mismo tipo.

El contenido de la etiqueta `<do>` es la tarea que se realizará al seleccionar esa acción, y estas tareas pueden ser:

- `<go href="" />` Permite ir a la dirección indicada en el atributo "href".
- `<prev/>` Permite ir a la tarjeta anterior, en la historia del navegador.
- `<noop/>` Es una acción que no realiza nada.
- `<refresh>...</refresh>` Refresca el contenido de la tarjeta actual, volviéndola a pedir al servidor.

Un código de ejemplo para crear eventos es el siguiente:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>

<card id="primera" title="Eventos">
- <do type="accept" label="dos" name="dos">
-   - <go href="#segunda"/>
- </do>
-
- <do type="accept" label="tres" name="tres">
-   - <go href="#tercera"/>
- </do>   -
-
- <do type="accept" label="nada" name="tres">
-   - <noop/>
- </do>
-
- <p>Primera tarjeta</p>
</card>

<card id="segunda">
<p>Segunda tarjeta</p>
</card>

<card id="tercera">
<p>Tercera tarjeta</p>
</card>
</wml>
```

**Figura 4.15** Código WML para crear eventos.

La ejecución de dicho código da el siguiente resultado:



Figura 4.16 Eventos en WML.

#### 4.2.3.1.9 Enlaces (*Links*)

Al igual que en HTML podemos seleccionar parte de un texto o una imagen y que este sea un enlace a otra página o que realice una tarea.

Para realizar esto existen dos etiquetas `<anchor>...</anchor>` y `<a>...</a>`. `<anchor>` es más completa y `<a>` es más sencilla de usar. Así que empezaremos por la etiqueta `<a>` para pasar luego a `</anchor>`.

La etiqueta `<a>...</a>` nos permite seleccionar un texto o imagen y hacer que sea un enlace a otra página, de una manera muy similar a la etiqueta `<a>...</a>` de HTML. Contiene el atributo `href` en el cual ponemos la URL a donde nos llevará el enlace. Por ejemplo:

```
<a href="#tarjeta2">Ir a tarjeta 2</a>
```

Con la etiqueta `<anchor>...</anchor>` podemos además de indicar que se realice una acción al activar el enlace. Las acciones que podemos incluir son:

- `<go href="" />` Permite ir a la dirección indicada en el atributo `href`.
- `<prev/>` Permite ir a la tarjeta anterior, en la historia del navegador.
- `<refresh>...</refresh>` Refresca el contenido de la tarjeta actual, volviéndola a pedir al servidor.

Por ejemplo:

```
<anchor>
  Ir a la tarjeta anterior.
<prev/>
</anchor>
```

Otro ejemplo:

```
<anchor>
  Ir a la tarjeta 2
<go href="#tarjeta2"/>
</anchor>
```

#### 4.2.3.1.10 Imágenes

El uso de imágenes en los navegadores WAP está limitado, pero aún así es posible poner imágenes en dichas páginas.

La etiqueta para poner imágenes en las páginas WAP es `<img/>`, con ella podemos poner una imagen e indicar un texto alternativo por si nuestro navegador WAP no es capaz de representarla.

La etiqueta `<img/>` tiene los siguientes atributos:

- `alt=""`. Texto alternativo que se visualiza si nuestro navegador no es capaz de visualizar la imagen.
- `src=""`. URL de la imagen. Normalmente está en formato `.wbmp` pero también soporta otros.
- `vspace="numero"`. Espacio vertical en blanco entre la imagen y el resto de la página.
- `hspace="numero"`. Espacio horizontal en blanco entre la imagen y el resto de la página.
- `align=""`. Puede ser `top`, `middle` o `bottom` indica la alineación de la imagen con respecto al texto.
  - `top`. Alineado a la parte superior.
  - `middle`. Alineado al centro.
  - `bottom`. Alineado con la parte inferior del texto.
- `height="numero"`. Altura de la imagen.
- `weight="numero"`. Anchura de la imagen.

#### 4.2.3.1.11 Tipos MIMES

Para que el servidor de HTTP sepa que debe servir las páginas `*.wml` como páginas WAP debemos indicárselo, de lo contrario el servidor responderá como que el tipo de fichero es de texto. Para ello existen los tipos MIME. Con estos tipos indicamos al servidor cómo se debe comunicar con el cliente cuando le solicitan una página wml.

La configuración de tipos MIMES, si su servidor no está configurado ya, es la siguiente:

Contenido	Tipo MIME	Extensión
WML Source	text/vnd.wap.wml	wml
Compiled WML	Application/vnd.wap.wmlc	wmlc
WMLScript source	text/vnd.wap.wmlscript	wmls
Compiled WMLScript	Application/vnd.wap.wmlscriptc	wmlsc
Wireless Bitmap	image/vnd.wap.wbmp	wbmp

**Tabla 4.4** Configuración de MIMES.

#### 4.2.3.1.12 Ejemplo complejo en WML

La versión en WML del buscador Google es <http://www.google.es/wml>. El código de la página es el siguiente:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">

<wml>
  <head>
    <meta http-equiv="Content-Type" content="text/vnd.wap.wml;
charset=UTF-8"/>
    <meta http-equiv="Cache-Control" content="no-cache"/>
  </head>
  <card title="Google">
    <onevent type="onenterforward">
      <refresh>
        <setvar name="q" value=""/>
      </refresh>
    </onevent>

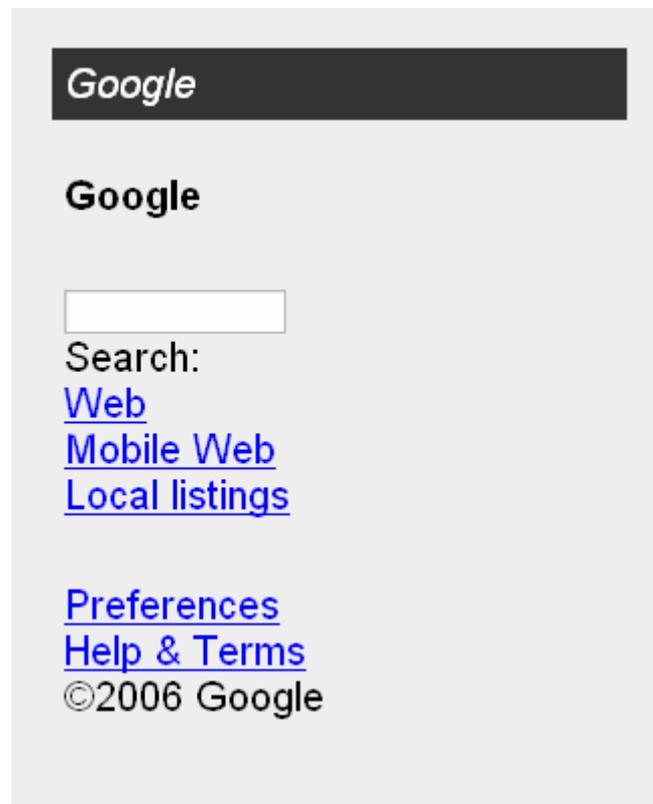
    <p>
      <b>Google</b>
    </p>
    <p>

      <input name="q" size="15" maxlength="255" emptyok="true"/>
      <br/>Search:
      <br/>
      <anchor title="search">
        <go href="/wml/search" method="get">
          <postfield name="mrestrict" value="wml"/>
          <postfield name="q" value="$(q)"/>
          <postfield name="site" value="search"/>
        </go>Web
      </anchor>
      <br/>
      <anchor title="mobile">
        <go href="/wml/search" method="get">
          <postfield name="mrestrict" value="wml"/>
          <postfield name="q" value="$(q)"/>
          <postfield name="site" value="mobile"/>
        </go>Mobile Web
      </anchor>
      <br/>
      <anchor title="local">
        <go href="/wml/search" method="get">
          <postfield name="mrestrict" value="wml"/>
          <postfield name="q" value="$(q)"/>
          <postfield name="site" value="local"/>
        </go>Local listings
      </anchor>
    </p>
    <p>
      <a href="/wml?pref=d">Preferences
      </a>
      <br/>
      <a href="/wml/help">Help & Terms
      </a>
      <br/>&#169;2006 Google
      <br/>
    </p>
  </card>
</wml>

```

Figura 4.17 Código WML de Google.

Para visualizar correctamente el código anterior se usó el navegador Opera 9.0, que mostró la siguiente imagen:



**Figura 4.18** Google en WML.

#### 4.2.3.2 XHTML Basic

XHTML Basic pretende proporcionar un tipo de documento XHTML que pueda ser compartido a través de distintos dispositivos (escritorio, TV, y teléfonos móviles), y que sea lo suficientemente rico para ser empleado en la creación de contenido simple. Fue presentada en forma de recomendación en diciembre de 2000 por la organización W3C.

Los siguientes son ejemplos de diferentes dispositivos en los que se puede usar:

- Teléfonos Móviles.
- Televisiones.
- PDAs.
- Máquinas expendedoras.
- Paginadores.
- Sistemas de Navegación de Coches.
- Máquinas de Juegos Móviles.
- Lectores de Libros Digitales.
- Relojes inteligentes.

#### 4.2.3.3 XHTML MP (*XHTML Mobile Profile*)

XHTML MP está basado en XHTML Basic pero lo amplía y mejora con nuevas características incluyendo soporte en el uso de imágenes, tablas complejas y formularios. Por ello está incluido dentro de WAP 2.0 y aparece como el futuro lenguaje oficial de esta plataforma por su origen orientado especialmente a dispositivos móviles.

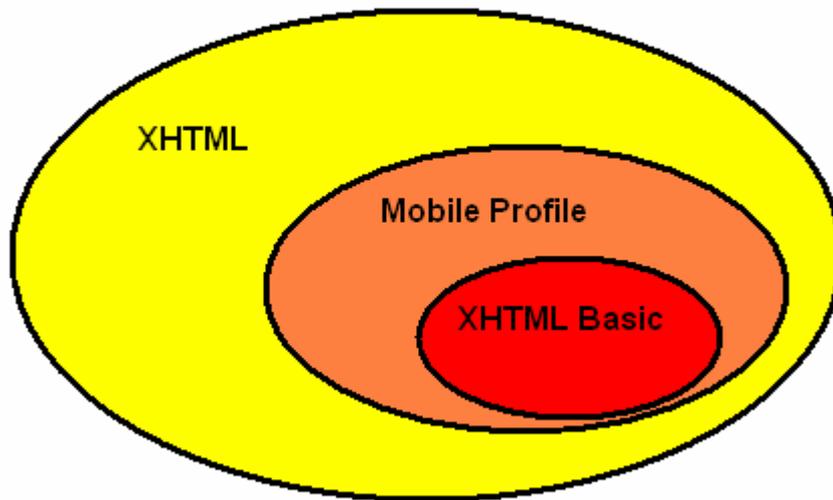
Entre las nuevas propuestas soportadas por este lenguaje resalta una muy interesante versión propia de hojas de estilo llamada WAP CSS, que se explicará al final de este capítulo.

##### 4.2.3.3.1 Características

Puesto que XHTML MP está basado en XHTML es interesante realizar una comparación entre las capacidades de ambas:

- No soporta:
  - Frames.
  - Tablas complejas.
  - Scripts.
  - Manejadores de eventos (sí a partir de la versión 1.2)
- Soporta:
  - Texto básico.
  - Vínculos.
  - Hojas de estilo.
  - Tablas sencillas:
    - table, td, th, tr, caption
  - Formularios básicos:
    - form, input, label, select, option, textarea
  - Imágenes.

En el siguiente esquema se representa el ámbito de funcionamiento de estos nuevos lenguajes surgidos del estándar XHTML. Aunque la versión Mobile Profile está basada en la Basic, amplía sus funcionalidades.



**Figura 4.19** XHTML, Mobile profile y XHTML Basic.

#### 4.2.3.3.2 Código de ejemplo

Un ejemplo básico sería el siguiente:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN"
"http://www.wapforum.org/DTD/xhtml-mobile10.dtd" >
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Subastas</title>
<link rel="stylesheet"
href="estilo.css" type="text/css"/>
</head>
<body>
<h1> Subastas </h1>
<hr/>
<p>
<a href.="personal.xhtml">Ver datos personales</a><br/>
<a href.="lista.xhtml">Ver subastas</a>
</p>
</body>
</html>
```

**Figura 4.20** Ejemplo de documento XHTML MP.

Para ver el código anterior se usó el navegador Firefox 1.5, que muestra la siguiente imagen:



Figura 4.21 Documento XHTML MP en Firefox 1.5.

#### 4.2.3.3.3 Ejemplo complejo de XHTML Mobile Profile

La versión del buscador Google en XHTML MP es: <http://www.google.es/xhtml>. El código es el siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN"
"http://www.wapforum.org/DTD/xhtml-mobile10.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="application/xhtml+xml;
charset=UTF-8"/>
    <style type="text/css">
      a:link {color:#0000cc}
      a:visited {color:#0000cc}
      a:hover {color:#0000cc}
      a:active {color:#0000cc}
    </style>
    <title>
      Google
    </title>
  </head>
  <body>
    <div>
    </div>
    <form action="/xhtml/search">
      <div><input type="hidden" name="mrestrict" value="xhtml"/>
      <input type="text" name="q" size="15" maxlength="2048"/>
      <br/>
      <input type="submit" name="btnG" value="Buscar"/>
      <br/>
      <input type="radio" name="site" value="search"
checked="checked"/>
      La Web<br/>
      <input type="radio" name="site" value="images"/>
      Imágenes<br/>
      <input type="radio" name="site" value="local"/>
      Directorios locales<br/>
      <input type="radio" name="site" value="mobile"/>
      Web móvil (Beta)<br/>
    </div>
  </body>
</html>
```

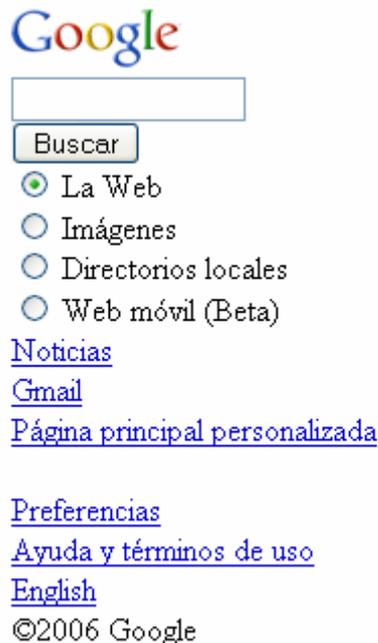
```

    </div>
  </form>
  <div><a href="/news?output=xhtml&hl=es"
  >Noticias</a>
  <br/><a href="http://mail.google.com/mail/x/"
  >Gmail</a>
  <br/><a href="/ig/mobile?output=xhtml&hl=es"
  >Página principal personalizada</a>
  <br/>
  <br/>
</div>
<div><a href="/xhtml?pref=d"
  >Preferencias</a>
  <br/><a href="/xhtml/help"
  >Ayuda y términos de uso</a>
  <br/><a href="/xhtml?hl=en"
  >English</a>
  <br/>&copy;2006 Google
  <br/>
</div>
</body>
</html>

```

**Figura 4.22** Código XHTML Mobile Profile de Google.

La ejecución de dicho código da como resultado la siguiente imagen:



**Figura 4.23** Google en XHTML Mobile Profile.

#### 4.2.3.4 CHTML

El HTML Compacto (cHTML) es el lenguaje utilizado para las aplicaciones i-Mode. El HTML Compacto está completamente basado en las recomendaciones HTML, permitiendo usar los millones de códigos HTML, herramientas software y material público e información web ya existentes.

El navegador debe ejecutarse en rangos pequeños de memoria, como por ejemplo, 150-200 Kbytes para el tratamiento sólo de datos y 150-200 Kbytes para código. El mínimo requerimiento de CPU debe estar en 1-2 MIPS. Se debe tener un buen sistema de navegación para que los usuarios puedan navegar por la información con el mínimo número de operaciones.

HTML Compacto no depende del protocolo de la red. En el caso típico el protocolo de transporte para HTML Compacto es asumido por HTTP sobre TCP/IP. De cualquier manera, la actual red de comunicación inalámbrica para teléfonos celulares es de baja velocidad. En esta área, el protocolo de transporte debe estar definido como un protocolo ligero para una mejor representación de la capa de seguridad.

#### **4.2.3.4.1 Principios de diseño.**

El HTML Compacto está diseñado para cumplir los requerimientos de las aplicaciones dirigidas a terminales móviles. Se ha diseñado basándose en los siguientes cuatro principios:

- Completamente basado en las actuales recomendaciones del W3C. HTML Compacto se ha definido como un subconjunto del HTML 2.0, HTML 3.2 y HTML 4.0. Esto conlleva que HTML Compacto hereda la flexibilidad y portabilidad del estándar HTML.
- Pequeña especificación. HTML Compacto tiene que estar implementado para pequeña memoria y baja capacidad de CPU. Frames y tablas que requieran mucha memoria están excluidas.
- Se puede visionar en una pantalla monocolor. HTML Compacto trabaja sobre pantallas de tamaño pequeño y color blanco y negro. También funciona con un bajo número de fuentes de caracteres.
- Fácil interacción con los usuarios. HTML Compacto está definido para que el usuario pueda trabajar con una combinación de cuatro botones: cursor de avance, cursor de retroceso, select, y back/stop (retorno a la página previa). Las funciones como mapa de imágenes y tablas están excluidas del HTML Compacto.

Algunas características que no se han incluido en HTML Compacto, son las siguientes:

- Imagen JPEG.
- Tablas.
- Mapa de imágenes.
- Múltiples caracteres y estilos.
- Color de fondo e imágenes.
- Frames.
- Hojas de estilo.

El HTML Compacto incluye soporte para imágenes GIF. Actualmente el navegador HTML Compacto puede visualizar las páginas como barajas y cartas. También añade:

- El atributo “*accesskey*” para asociar teclas a enlaces:  
`<a href="productos.html" accesskey="2">Productos</a>`
- Enlaces a números de teléfono:  
`<a href="tel://+555123456">Servicio técnico</a>`

Debido a la pequeña capacidad de memoria de los dispositivos en los que se suele utilizar HTML Compacto, se recomienda un límite de buffer para algunas funciones:

- La etiqueta `<input>`, empleada para introducir información en la página tiene un tamaño máximo de buffer de 512 bytes.
- La etiqueta `<select>`, que despliega un listado con diversas opciones tiene un tamaño máximo de buffer de 4096 bytes.

Una recomendación es usar la selección directa de anclas usando botones numéricos. Por ejemplo cuando existen cinco anclas contenidas en una página HTML, el tercer ancla puede ser seleccionado mediante el botón tres. La última versión de HTML, la 4.0, incluye este nuevo atributo “*accesskey*” para la asignación directa de una tecla.

#### 4.2.3.4.2 Ejemplo de página cHTML

Un ejemplo de código cHTML sería:

```
<!DOCTYPE "-//W3C//DTD Compact HTML 1.0 Draft//EN">
<html>
<head>
<META http-equiv="Content-Type" content="text/html; charset=utf-8">
<META name="cHTML" content="yes">
<META name="description" content="Documento cHTML">
<title>Menu cHTML</title>
</head>
<body bgcolor="#ffffff" text="#000000">
<center>Subastas</center>
<hr>
Menu<br>
&#59106;<A HREF="lista.htm" accesskey="1">
Ver subastas </A><BR>
&#59107;<A HREF="personal.htm" accesskey="2">
Datos personales </A><BR>
<hr>
&#63858;<A HREF="tel://555123456"> Incidencias </A><BR>
</body>
</html>
```

**Figura 4.24** Ejemplo de documento cHTML.

#### 4.2.3.4.3 Ejemplo complejo de cHTML

La versión en cHTML de Google es <http://www.google.es/imode>. El código de la página es el siguiente:

```

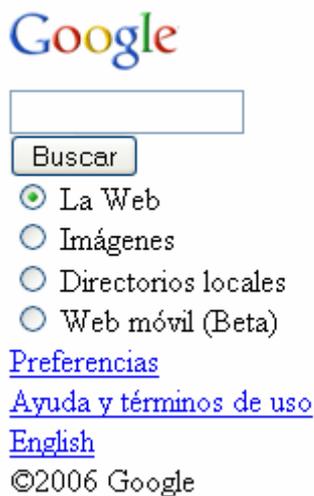
<!DOCTYPE HTML PUBLIC "-//W3C//DTD Compact HTML 1.0 Draft //EN">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-
8">
    <title>
      Google
    </title>
  </head>
  <body>
    <div>
    </div>
    <form action="/imode/search">
      <div><input type="hidden" name="mrestrict" value="chtml"><input
type="text" name="q" size="15" maxlength="2048">
      <br><input type="submit" name="btnG" value="Buscar">
      <br><input type="radio" name="site" value="search"
checked="checked">
La Web
<br><input type="radio" name="site" value="images">
Imágenes
<br><input type="radio" name="site" value="local">
Directorios locales
<br><input type="radio" name="site" value="mobile">
Web móvil (Beta)
<br>
</div>
</form>
<div><a href="/imode?pref=d">Preferencias</a>
<br><a href="/imode/help">Ayuda y términos de uso</a>
<br><a href="/imode?hl=en">English</a>
<br>&copy;2006 Google
<br>
</div>
</body>
</html>

```

**Figura 4.25** Código cHTML de Google.

La ejecución de dicho código da como resultado la siguiente imagen:



**Figura 4.26** Google en i-Mode

Con cHTML se acaba el repaso a los lenguajes de marcas. En el siguiente apartado se hablará sobre las hojas de estilo, que sirven para mejorar el aspecto de las páginas web.

## 4.3. Hojas de estilo

### 4.3.1 Introducción

Los lenguajes HTML y XML (en el que se basaba XHTML) están limitados a la hora de aplicarle forma a un documento. Esto es así porque fueron concebidos para otros usos (científicos sobretodo), distinto a los actuales, mucho más amplios. Para solucionar este problema se han desarrollado las llamadas hojas de estilo (*Style Sheets*) [21]. Una hoja de estilo es una colección de reglas que afecta a la apariencia de un documento. Estas normas se refieren al modo en que aparecerá un documento en pantalla cuando el usuario utilice un navegador gráfico, controlando por ejemplo el color, el fondo, tipo de fuente, apariencia del borde, márgenes, alineación, espacio entre caracteres, etc.

Las definiciones del formato del documento pueden colocarse en archivos separados y aplicarse a una serie de documentos. También existen muchos de tipos de hojas de estilo. Se explican a continuación.

### 4.3.2 Tipos de hojas de estilos

#### 4.3.2.1 CSS

Un primer tipo de hojas de estilo es el de “Hojas de Estilo en Cascada” (*Cascading Style Sheets*) o CSS [22].

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Permite a los desarrolladores web controlar el estilo y el formato de múltiples páginas al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne. Por ejemplo, en la siguiente regla:

```
h1 {color: red;}
```

h1 es el selector y {color: red;} es la declaración.

El selector funciona como enlace entre el documento y el estilo, especificando los elementos que se van a ver afectados por esa declaración. La declaración es la parte de la regla que establece cuál será el efecto. En el ejemplo anterior, el selector h1 indica

que todos los elementos `h1` se verán afectados por la declaración donde se establece que la propiedad `color` va a tener el valor `red` (rojo) para todos los elementos `h1` del documento o documentos que estén vinculados a esa hoja de estilos.

Las tres formas más conocidas de dar estilo a un documento son las siguientes:

- Utilizando una hoja de estilo externa que estará vinculada a un documento a través del elemento `<link>`, el cual debe ir situado en la sección `<head>`. El código sería como se indica en la siguiente figura:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN">
<html>
  <head>
    <title>Título</title>
    <link rel="stylesheet" type="text/css"
      href="http://www.w3.org/css/officeFloats.css" />
  </head>
  <body>
    .
    .
    :
    .
  </body>
</html>
```

**Figura 4.27** Hoja de estilo externa.

- Utilizando el elemento `<style>`, en el interior del documento al que se le quiere dar estilo, y que generalmente se situaría en la sección `<head>`. De esta forma los estilos serán reconocidos antes de que la página se cargue por completo. El código sería como se indica en la siguiente figura:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN">
<html>
  <head>
    <title>hoja de estilo interna</title>
    <style type="text/css">

      body {
        padding-left: 11em;
        font-family: Georgia, "Times New Roman", serif;
        color: red;
        background-color: #d8da3d;
      }

      h1 {
        font-family: Helvetica, Geneva, Arial, sans-serif;
      }

    </style>
  </head>
  <body>
    <h1>Aquí se aplicará el estilo de letra para el Título</h1>
  </body>
</html>
```

**Figura 4.28** Hoja de estilo interna.

- Utilizando estilos directamente sobre aquellos elementos que lo permiten a través del atributo `<style>` dentro de `<body>`. Pero este tipo de estilo pierde las ventajas que ofrecen las hojas de estilo al mezclarse el contenido con la presentación.

Algunas normas básicas a la hora de crear una CSS son las siguientes:

En el siguiente ejemplo, `h1{color: red;}`, el selector, `<h1>`, le dice al navegador la parte del documento que se verá afectada por esa regla. Los selectores pueden aparecer individualmente o agrupados, separándolos con comas:

```
h1, h2, h3 {
  color: red;
}
```

Equivale a:

```
h1 {color: red;}
h2 {color: red;}
h3 {color: red;}
```

La propiedad, que en este caso sería `color`, especifica qué aspecto se va a cambiar. En este ejemplo la propiedad cambiada será el color. Las propiedades que se desean modificar en una CSS para un mismo selector pueden agruparse, pero será necesario separar cada una de ellas con un punto y coma.

```
p {text-align:center;color:red}
```

Normalmente se describe una propiedad por línea, de la siguiente manera:

```
h1 {
  padding-left: 11em;
  font-family: Georgia, "Times New Roman", Times, serif;
  color: red;
  background-color: #d8da3d;
}
```

El valor, en este caso `red`, establece el valor de la propiedad. Es importante recordar que si el valor está formado por más de una palabra, hay que ponerlo entre comillas.

```
p {font-family: "sans serif";}
```

#### 4.3.2.2 XSL

XSL (siglas de *Extensible Stylesheet Language*, expresión inglesa traducible como "lenguaje extensible de hojas de estilo") [23] es una familia de lenguajes basados en el estándar XML que permite describir cómo la información contenida en un documento XML cualquiera debe ser transformada o formateada para su presentación en un medio específico.

Esta familia está formada por tres lenguajes:

- **XSLT** (siglas de *Extensible Stylesheet Language Transformations*, lenguaje de hojas extensibles de transformación), que permite convertir documentos XML de una sintaxis a otra (por ejemplo, de un XML a otro o a un documento HTML).
- **XSL-FO** (lenguaje de hojas extensibles de formateo de objetos), que permite especificar el formato visual con el cual se quiere presentar un documento XML, es usado principalmente para generar documentos PDF.
- **XPath**, o **XML Path Language**, es una sintaxis (no basada en XML) para acceder o referirse a porciones de un documento XML.

Estas tres especificaciones son recomendaciones oficiales del W3C.

### 4.3.3 Hojas de estilo en entornos móviles

Así como existían lenguajes de marcas creados para el ámbito inalámbrico, también existen hojas de estilo propias para este entorno, que son:

- CSS Mobile Profile:
  - Estándar de W3C.
  - Subconjunto de CSS2.
- WCSS (WAP CSS):
  - Soportado por XHTML Basic y WML 2.0.
  - Definido en WAP.
  - Subconjunto de CSS2:
    - Agentes de usuario WAP CSS aceptan CSS Mobile Profile.
  - Extensiones de WAP:
    - Texto con scroll (*marquee*).
    - Formato de entrada de texto en formulario.
    - Asociar teclas del teléfono (*accesskey*).

Empezará el estudio por CSS2, del que derivarán las de hojas de estilo para móviles.

#### 4.3.3.1 CSS2

CSS2 [24] es una recomendación del W3C publicada en mayo de 1998. No está pensada para el ámbito móvil, pero este tipo de entornos usarán un subconjunto del mismo.

CSS2 se basa en CSS1 (CSS nivel 1, es la hoja de estilo usada en HTML) y, con muy pocas excepciones, todas las hojas de estilo CSS1 válidas son hojas de estilo CSS2 válidas. CSS2 brinda soporte a hojas de estilo específicas para cada medio, de modo que los autores puedan adaptar la presentación de sus documentos a los browsers visuales, a los dispositivos sonoros, a las impresoras, a los dispositivos braille, de mano, etc. Esta especificación también soporta el posicionamiento de contenidos, fuentes descargables, disposición de la página, aspectos para la internacionalización, contadores y numeradores automáticos, y algunas características relacionadas con la interfaz del usuario.

Como posterior evolución aparece CSS 2.1 [25], que es una nueva especificación de CSS que corrige algunas erratas (la más destacable es la nueva definición de posicionamiento absoluto de la altura/anchura de los elementos, más influenciado en los estilos de los atributos de HTML), modifica CSS2 en algunos aspectos en los que casi todas las implementaciones diferían del estándar, añade algunos valores de propiedades (como el color “orange”) y elimina algunas cosas que nadie había implementado ni tenía intención de implementar.

#### 4.3.3.2 CSS Mobile Profile

CSS Mobile Profile está pensada expresamente para entornos móviles. Se trata de una recomendación del W3C [26].

La versión 1.0 fue publicada en julio de 2002, subconjunto de CSS2. También existe una segunda versión, CSS Mobile Profile 2.0, sacada en diciembre de 2006, subconjunto de la especificación más reciente CSS2.1.

Las propiedades que permite CSS Mobile Profile 2.0 son:

- background
- background-attachment
- background-color
- background-image
- background-position
- background-repeat
- the border properties
- bottom
- clear
- color
- display
- float
- font
- font-family
- font-size
- font-style
- font-variant
- font-weight
- height
- left
- list-style (*shortcut*)
- list-style-image
- list-style-position
- list-style-type
- outline
- right
- the margin properties
- the padding properties
- text-align
- text-decoration
- text-indent
- text-transform
- top
- vertical-align
- visibility
- white-space
- width
- z-index

En el siguiente apartado se explica el último de los tipos de hojas de estilo para móviles que existen hasta la actualidad, el WAP CSS.

#### 4.3.3.4 WAP CSS

WAP CSS [27] especifica un subconjunto de CSS2 y de WAP. Puede usarse para dar estilo a los documentos XHTML Basic y WML 2.0, así como a cualquier otro documento XML.

Puesto que tanto CSS Mobile Profile como WAP CSS son variantes de CSS2 y siguen la semántica CSS, los agentes de usuario WAP CSS también aceptarán las hojas de estilo CSS Mobile Profile.

Las extensiones que especifica WAP son:

- Marquee. Sirve para crear efectos de animación simples. Un ejemplo de uso sería el uso de texto con scroll.
- Formato de entrada de texto en formulario.
- Accesskey. Especifica una manera opcional y adicional de activar elementos en un documento, como puede ser la asociación de teclas del teléfono con dichos elementos. Es similar a los atributos accesskey de la especificación HTML.

## 4.4 Conclusiones

En este capítulo se ha introducido el concepto de lenguajes de marcas, que se emplea para crear el contenido web. Existe tal variedad de lenguajes de marcas debido a los diferentes ámbitos de funcionamiento y capacidades de los dispositivos que van a utilizarlos.

Inicialmente se vio HTML. Este es el lenguaje más comúnmente usado en las páginas web. A pesar de no estar pensado para dispositivos móviles, algunos microbrowsers son capaces de leerlo, lo que solucionaría el problema de tener que diseñar páginas específicas para ellos. Parece ser que no existirán futuras evoluciones de HTML sino que éste derivará hacia el XHTML, que es más estructurado y fácil de estandarizar.

Después de analizar WML hay que indicar que este lenguaje no tiene unas perspectivas de futuro favorables. La razón de ello es que, por un lado es prácticamente incompatible con HTML, con la consecuencia de que las páginas deben programarse tanto en HTML como en WML para poder verse en los dos ámbitos de funcionamiento, inalámbrico y no. Por otro, WML apenas tiene perspectivas futuras de sobrevivir en el mercado móvil de banda ancha, pues con unas velocidades bajas de transferencia, representa los contenidos Internet sin gráficos y en pequeñas pantallas. Y aunque esto pudo ser suficiente para la generación de móviles pasada, no lo es para los dispositivos móviles actuales (GPRS y UMTS). De ahí que WML fuera usado en WAP 1, y en cambio WAP 2 ya no lo utilice.

El HTML Compacto, garantiza que las aplicaciones móviles pueden conectarse al servicio WWW. HTML Compacto mantiene las ventajas del HTML y solventa los problemas provocados por el tamaño de las aplicaciones para móviles. Sin embargo la baja implementación de i-Mode en nuestro país hace que existan muy pocas páginas en

español de este lenguaje. La gran mayoría de ellas son japonesas y por tanto existe una gran barrera lingüística además de tecnológica.

De las versiones reducidas de XHTML hay que decir que la versión XHTML Basic es demasiado simple, a pesar de todo es la apuesta del consorcio W3C para el desarrollo de las páginas web pensadas para móviles, tal y como se verá en el capítulo siguiente de esta memoria. Por otro lado la versión XHTML Mobile Profile es la opción elegida por los desarrolladores de WAP para la versión WAP 2.

Las hojas de estilo usadas en las páginas web convencionales no son soportadas por los terminales móviles. Esto tiene la gran desventaja de que aquellos microbrowsers que usen HTML pierden totalmente sus hojas de estilo y las muestran con un aspecto más pobre. W3C propone la utilización de CSS1 y también de CSS Mobile Profile mientras que los desarrolladores WAP han creado su propia especificación de hojas de estilo, la WAP CSS.

Tanto en lenguajes de marcas como en hojas de estilo existen divergencias entre las propuestas del W3C, que realizan las normativas, y los desarrolladores de WAP, que son los que fabrican los dispositivos y por tanto los que poseen el poder de decisión en cuanto a su introducción en el mercado.

