

Curso completo de HTML

por Jorge Ferrer

por Víctor García

por Rodrigo García

Curso completo de HTML

por Jorge Ferrer

por Víctor García

por Rodrigo García

Este es un curso tremendamente completo, guiado y con ejemplos y prácticas pensadas para convertirlo en una guía didáctica para aprender a hacer páginas web. No se requieren más que conocimientos básicos de informática para llevarlo a cabo y se cubren desde los aspectos más básicos de HTML hasta los más avanzados. Todo ello resaltando en todo momento las partes más importantes de este lenguaje y la mejor forma de emplearlo. Esperamos que lo disfrutes.

Copyright Jorge Ferrer, Rodrigo Garcia y Victor García. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation. Puede consultar una copia de la licencia en: <http://www.gnu.org/copyleft/fdl.html>

Para obtener la última versión de este documento, informar de fallos que hayas encontrado o colaborar a su mejora visita <http://freak.jorgeferrer.com>.

Tabla de contenidos

Introducción al lenguaje HTML	1
Introducción	1
Características del lenguaje HTML	2
Qué se necesita para crear una página web	2
Algunas razones para usar HTML	5
HTML es un lenguaje descriptivo	6
Las bases de HTML	8
Primeros pasos con HTML	10
Estructura de una página	11
Los encabezados	15
El ejemplo mejorado	18
Conclusión	20
La primera página Web	21
Párrafos en HTML	21
Saltos de línea	22
Cómo se insertan párrafos	22
Líneas horizontales para separar párrafos	26
La etiqueta <blockquote>... </blockquote>	28
La etiqueta <address>... </address>	29
La etiqueta <center>... </center>	30
Cómo insertar imágenes en una página.	31
Cómo crear enlaces hipertexto.	32
La etiqueta <a>... 	33
Páginas en otros directorios	34
Enlaces externos	34
PRÁCTICA 1: La página web de "La chistera"	36
La propuesta	36
La solución	40
Anclas	41
¿Qué es un ancla?	41
Creando anclas	42
Enlaces a anclas de otras páginas	47
Uso de enlace y ancla en la misma etiqueta	49
El atributo 'title'	50
Los atributos 'rel' y 'rev'	51
Formato de texto y Caracteres especiales	53
Formato de texto	53
Etiquetas de estilo físico	53
Etiquetas de estilo lógico	59
La etiqueta FONT.	63
Cambiando el tamaño del texto.	63
Cambiando el color del texto	68
Texto preformateado.	69
Uso de otras etiquetas dentro de <PRE>...</PRE>	72
Acentos y otros caracteres especiales	72
Entidades de caracteres para caracteres españoles	73
Entidades de caracteres para caracteres reservados	74
Otras entidades de caracteres	75
Listas y Comentarios en HTML	77
Listas en HTML	77
Etiquetas necesarias para crear listas	79

Listas no ordenadas	80
Listas ordenadas	82
Listas de glosario	85
Directorios y menús	87
Anidamiento de listas	89
Comentarios en HTML	92
Creación de comentarios en HTML	93
Práctica 2: Un sitio web completo.	94
Planificación	95
Secciones de la página	95
Diseño gráfico	99
Programación en HTML	99
Imágenes y color en el WWW	103
Usando color en el WWW.	105
Cómo cambiar la apariencia de una página.	107
Color de fondo.	107
Color del texto y Links.	108
Imágenes de fondo:	110
Imágenes en las páginas Web.	114
Insertando Imágenes.	114
Imágenes y texto.	115
Alineamiento de imágenes.	119
Texto que envuelve a las imágenes o alineamiento horizontal.	123
Como terminar el texto envolvente.	126
Márgenes alrededor de las imágenes.	128
Imágenes enlace.	129
Cómo cambiar la apariencia de las imágenes	130
Borde de las imágenes	130
Modificando el tamaño de las imágenes	133
Alternativas a las imágenes inline	137
El atributo ALT	137
Imágenes externas	140
Imágenes mapa	145
Mejoras en las imágenes	150
Formatos GIF y JPEG	150
Transparencia, Entrelazado y GIFs animados	153
Creación de tablas HTML	157
Cómo se crea una tabla	159
Borde de las tablas	164
Insertar un título para la tabla	170
Tamaño de las tablas	173
Ejercicio 7.1	174
Alineamiento de tablas	174
Alineamiento de celdas y filas	177
Ancho de tablas y celdas	180
Tamaño de las celdas	181
EJERCICIO 7-2	181
El atributo NOWRAP	185
Celdas que abarcan varias filas o columnas	187
Márgenes y separación de celdas	188
Ejercicio 7.3	190
Conclusión	191
Anidamiento	191

Práctica 3 - Tabla de Datos.	194
Paso 1.	196
Paso 2.	197
Paso 3.	197
Paso 4.	198
Paso 5.	200
Práctica 4 - Página Web con una estructura creada con tablas.	202
Paso 1 - Planificación.	203
Paso 2 - Código básico.	208
Paso 3 - Estructura de la página principal.	209
Paso 4 - Estructura del resto de páginas.	211
Paso 5 - Contenido de la página principal.	212
Paso 6 - Contenido gráfico.	215
Paso 7 - Cambiar el fondo y color del texto.	219
Paso 8 - Realización del resto de las páginas.	222
Paso 9 - Retoques, pruebas, revisión y mantenimiento.	224
HTML avanzado	225
Estándares y extensiones del lenguaje HTML	227
El estándar HTML 2.0	230
El estándar HTML 3.2	232
HTML 4.0 y XHTML	235
El futuro de HTML	237
Conclusiones	237
Extensiones a las listas	241
Listas no ordenadas	241
Listas ordenadas	245
Alineamiento	249
La etiqueta META	253
Carga automática de páginas	253
Información para los buscadores	253
Otros usos	255
Otras etiquetas del lenguaje HTML 3.2	255
Descriptor de tipo de documento	255
Otras etiquetas	256
Frames, las ventanas llegan al WWW	257
El atributo TARGET	261
La etiqueta BASE	265
Creación de páginas con frames	269
Creación del documento de definición de frames	270
División en columnas	271
División en filas	276
Combinando COLS y ROWS	276
La etiqueta FRAME	278
Página completa con frames	279
Atributos de la etiqueta FRAME	282
Compatibilidad: NOFRAME	284
Anidamiento de framesets	287
Bordes de los frames	292
Cambiando el color del borde	292
Como quitar el borde	295
Grosor del borde	297
Valores especiales del atributo TARGET	299
Ejercicio con TARGET	301

Práctica 5: Web de Los Alpes con frames	302
Documento de definición de frames	303
El frame Banner	306
Frame principal	309
Últimos retoques	310
Frames flotantes	312
Atributos de los frames flotantes	313
Vídeos, Sonido y Multimedia	319
Sonido vídeo y otros ficheros externos	319
Ficheros de sonido	322
Ficheros de vídeo	324
Otros tipos de archivos	326
Multimedia Inline	328
Vídeo inline	328
El atributo LOWSRC	329
Sonidos Inline	331
Marquesinas animadas	333
Texto que parpadea	337
Controles multimedia	337
Plug-ins y objetos incrustados	339
Como usar los plug-ins	342
Alternativas para navegadores sin soporte para PLUG-INS	342
Sonido inline con la etiqueta EMBED	344
Applets de Java y Controles ActiveX	347
Java en las Páginas Web.	347
Inserción de Applets Java.	350
Especificando la dirección del applet.	352
Otros atributos de la etiqueta APPLET.	353
La etiqueta PARAM.	356
Alternativas a los applets Java.	357
El nuevo estándar: la etiqueta OBJECT.	361
Inserción de un applet Java.	362
ActiveX, la tecnología del futuro.	363
Inserción de un control ActiveX.	366
Formularios.	371
Formularios en HTML.	371
Controles de Formularios.	379
Cajas de texto.	380
Cajas de texto para claves.	383
Botones de elección.	385
Cajas de selección.	388
Imágenes.	390
Botón de borrado	395
Botón genérico	399
Campos ocultos	402
Los formularios vistos desde Lynx	403
Innovaciones de Microsoft	404
OTROS CONTROLES	408
Cajas de texto multilínea: TEXTAREA	408
Cuadros de selección	412
PRÁCTICA 6-FORMULARIO PARA "Los Alpes"	418
COMO USAR LOS DATOS DE UN FORMULARIO	427
Formularios sin CGIs	427

ENVÍO DE FICHEROS USANDO FORMULARIOS 429

Lista de figuras

1.1. Aunque al principio el texto predominaba en el WWW, en la actualidad las imágenes son mayoría como podemos observar en una de las principales webs del proyecto KDE.	1
1.2. Un editor de texto simple, como el block de notas de Windows, es todo lo necesario para crear una página Web	3
1.3. Algunos editores WYSIWYG permiten, además, modificar el código HTML directamente.	4
1.4. Los navegadores en modo texto como Lynx también pueden mostrar páginas Web gracias al carácter descriptivo de HTML.	6
1.5. Uso básico de una etiqueta HTML sin ningún tipo de parámetros. Ejemplo usando la etiqueta <hr> para crear una línea horizontal.	9
1.6. Las etiquetas tienen atributos para cambiar aspectos del efecto que producen. En este ejemplo vemos como se puede cambiar el grosor de una línea con el atributo SIZE.	9
2.1. Estructura de bloques de un documento HTML.	11
2.2. El título de la página es mostrado en la parte superior de la ventana del navegador.	12
2.3. Cuando los navegantes incluyan nuestra página a su menú de favoritos en su menú aparecerá el título que le hayamos dado a nuestra página con la etiqueta <TITLE>.	13
2.4. En esta figura observamos la zona de la ventana en la que los navegadores mostrarán el texto que pongamos en el cuerpo de la página.	14
2.5. Distintos tipos de encabezados vistos con Microsoft Internet Explorer.	16
2.6. Distintos tipos de encabezados vistos con Netscape Navigator	18
2.7. Ejemplo de página Web con dos tipos de encabezados y dos párrafos.	19
2.1. Uso básico de una etiqueta HTML sin ningún tipo de parámetros. Ejemplo usando la etiqueta <hr> para crear una línea horizontal.	21
2.2. Las etiquetas tienen atributos para cambiar aspectos del efecto que producen. En este ejemplo vemos como se puede cambiar el grosor de una línea con el atributo 'size'	22
2.3. Ejemplo de un párrafo con un salto de línea y la división del mismo texto en dos párrafos.	24
2.4. Con el atributo 'align' se pueden crear párrafos alineados a la derecha, centrados o alineados a la izquierda.	25
2.5. Con los nuevos atributos de la etiqueta <hr> podemos controlar el ancho, grosor alineamiento y sombra de la línea horizontal.	27
2.6. La etiqueta <blockquote> es usada para insertar bloques de texto que citan a otro autor. Los navegadores suelen introducir un sangrado en este texto.	29
2.7. En esta página se observa como al final de la página se han incluido datos de los autores y de la página. Para ello se ha usado la etiqueta <address>.	30
2.8. Con la etiqueta <center> es posible centrar cualquier elemento de nuestra página como por ejemplo el verso de Lorca de la figura 2.6.	31
2.9. Al insertar una imagen esta se entremezcla con el texto. Cuando la imagen es pequeña se crean interesantes efectos como este.	31
2.10. Al pulsar sobre un enlace hipertexto saltaremos a otra página de Internet. Los navegadores resaltan estos enlaces para que sea fácil identificarlos.	33
2.11. Página que agrupa todos los enlaces de un sitio web	35
2.12. La práctica 1 propone realizar un página a partir de lo mostrado en esta figura.	37
2.13. Estructura de los enlaces normales. Al pulsar sobre el texto activo no podemos elegir el lugar de la página al que saltamos, iremos siempre al comienzo.	42
2.14. En la sección superior puede verse ver una serie de enlaces (las letras mayúsculas) a anclas. En la sección superior se muestra el lugar al que se ha saltado al pulsar sobre la letra S. Podemos deducir que antes de la palabra Salud hay un ancla.	43
2.15. Estructura de un enlace a un ancla de la misma página. Cuando pulsamos sobre el enlace el navegador nos transporta hasta el lugar de la página donde está el ancla.	46
2.16. Estructura de un enlace a un ancla de otra página. Cuando pulsamos sobre el enlace vamos hasta otra página, pero no al comienzo sino al lugar concreto donde está el ancla.	48

2.17. Con el atributo 'title' podemos incluir información adicional sobre el Link. Explorer mostrará esta información con un bocadillo.	50
3.1. Con el tipo de letra Times New Roman no todas las letras ocupan lo mismo.	53
3.2. Con el tipo de letra Courier todas las letras ocupan lo mismo (monoespaciadas)	55
3.3. Los navegadores gráficos como Explorer pueden mostrar perfectamente todos los tipos de estilos lógicos como vemos en esta figura.	56
3.4. El navegador Lynx sólo puede mostrar texto y usa los medios a su alcance para formatear el texto que debería ser negrita, cursiva...	58
3.5. En esta figura observamos los estilos lógicos con Explorer. La gran mayoría de ellos son mostrados como letra en negrita o cursiva.	61
3.6. Usando estilos lógicos no se nota tanto la diferencia entre Explorer y Lynx aunque los medios de este sean más limitados.	62
3.7. Ejemplo de uso de la etiqueta FONT y su atributo SIZE para cambiar el tamaño del texto.	64
3.7. Ejemplo de uso de la etiqueta FONT y su atributo SIZE para cambiar el tamaño del texto.	65
3.8. Los tamaños del texto que hallamos modificado usando el método relativo se calcularán a partir del al tamaño por defecto del texto normal, pero sin pasar nunca de 7.	67
3.9. Gracias a la etiqueta FONT y a su atributo COLOR nuestras páginas tomarán un colorido totalmente diferente.	69
3.10. Como se puede ver se pueden crear firmas muy originales usando únicamente caracteres y sin necesidad de imágenes.	70
3.11. Si no usamos la etiqueta de texto preformateado la figura construida con caracteres se convertirá en algo ilegible.	70
3.12. Una entidad de caracter con nombre tiene tres partes: '&' + 'nombre' + ';'	72
3.13. Una entidad de caracter numérica tiene cuatro partes: '&' + '#' + 'número' + ';'	73
3.1. Entidades de caracteres numéricas y nominales para caracteres acentuados, 'u' con diéresis y eñe.	73
3.2. Entidades de caracter numéricas y nominales para caracteres reservados del lenguaje HTML.	74
3.3. Entidades de caracteres numéricas y nominales para algunos caracteres especiales de uso común.	75
4.1. En esta figura podemos apreciar la presentación de la segunda práctica del curso que realizaremos al final del capítulo.	77
4.2. Como aperitivo mostramos una lista no numerada en la que ya se pueden apreciar las características fundamentales que tienen en HTML.	79
4.3. Los elementos de las listas no numeradas van precedidos de una viñeta. En Explorer esta viñeta es habitualmente un punto grande.	81
4.4. Lista no ordenada en Lynx. Cada uno de los elementos de lista comienza con un asterisco simulando a los símbolos gráfico que usan Netscape o Explorer.	81
4.5. En las listas ordenadas cada uno de los elementos de lista aparecen numerados y sangrados hacia la derecha.	82
4.6. Cuando insertamos un nuevo elemento de lista en una lista ordenada los elementos siguientes son automáticamente renumerados.	83
4.7. En el caso de las listas numeradas la diferencia entre la visualización con Lynx y con Explorer es menor.	84
4.8. Las listas de glosario constan de dos partes la palabra de definición y la propia de definición que es mostrada como un nuevo párrafo con sangrado.	85
4.9. Listas de glosario en Lynx. De nuevo las diferencias respecto a la visualización con Explorer son escasas.	86
4.10. En Explorer las listas de menú (<MENU>) y de directorio (<DIR>) son mostradas de igual manera que las listas no numeradas ().	88
4.11. En Lynx las listas de menú y de directorio son mostradas simplemente insertando cada elemento de lista en una nueva línea.	88
4.12. Gracias al anidamiento podemos crear listas muy complejas en HTML. El segundo nivel de lista no numerada usa una viñeta distinta.	90
4.13. Comparando esta figura con la 4.8 podemos ver las ligeras diferencias que existen en los distintos tipos de listas entre Netscape y Explorer	91

4.14. Al mostrar la página los navegadores actúan como si los comentarios no existieran. Como vemos Explorer permite comentarios multilínea y con etiquetas dentro.	93
4.15. Pantalla de presentación de la segunda página web de nuestro website. Es importante mantener un parecido entre todas las páginas que lo componen.	96
4.16. En la sección 'Los Alpes - 20 años a su servicio' podemos apreciar el uso de la etiqueta BLOCKQUOTE y de una lista no ordenada.	97
4.17. Logo de los alpes.	99
4.18. Letrero de 'Los Alpes' que, junto con el logo, aparecerá en todas las páginas del sitio web.	99
5.1. Aprovechando las capacidades del HTML para incluir colores e imágenes en un Web podemos conseguir resultados tan atractivos como el que observamos en el web de Disney.	103
tabla 5.1. Colores usados más frecuentemente	105
5.2. El lenguaje HTML nos permite cambiar el color de fondo de nuestras páginas. Podemos indicar el color por su nombre en inglés o con su número RGB.	108
5.3. También podemos cambiar el color del texto, de los enlaces, de los enlaces visitados y de los enlaces activos a nuestro gusto, igual que hacíamos con el fondo.	109
5.4. Para crear un mayor impacto visual podemos usar como fondo una imagen, esta imagen se replicará (es puesta en mosaico) por toda la página, luego los bordes deben coincidir.	111
5.5. En esta ocasión la imagen de fondo es un logo sobre un fondo de color sólido (gris). Esta es una manera sencilla para que no se noten los bordes cuando el navegador la pone en mosaico.	111
5.6. Esta es la imagen usada como fondo en la página de la figura 5.4. Como vemos, es de pequeño tamaño, pero al coincidir sus bordes simula una imagen que ocupa toda la pantalla.	112
5.7. También podemos poner fotos como fondo de la página, pero hay que tener cuidado porque los bordes no encajan y además la página queda demasiado sobrecargada.	113
5.8. Podemos apreciar la diferencia entre insertar una imagen dentro del encabezado o insertarla fuera, en cuyo caso queda en la línea superior.	115
5.9. Como veíamos en el fascículo 2 (fig. 2.9) cuando la imagen es pequeña puede entremezclarse con el texto sin problema.	117
5.10. Cuando la imagen es grande, al insertarla en una línea de texto ésta queda muy separada de la línea anterior en lugar de envolver la imagen.	117
5.11. Con las técnicas de alineamiento del lenguaje HTML que estudiamos en este capítulo podemos crear interesantes efectos como texto envolviendo a una imagen.	119
5.12. Ejemplos de align=top. En el primer caso la parte superior de la imagen coincide con la del texto y en el segundo con la parte superior de la imagen azul.	120
5.13. Ejemplos de align=middle. En el primer caso el centro de la imagen coincide con la base del texto, en el segundo intenta centrarse con la imagen azul.	121
5.14. Ejemplos de align=bottom. En ambos casos la parte inferior de la imagen se alinea con la base del texto, aunque la imagen azul baje por debajo.	122
5.15. Alineando una imagen a la izquierda con align=left conseguiremos que el texto envuelva a la imagen.	124
5.16. Usando align=right conseguimos igualmente que el texto envuelva a la imagen pero ahora ésta se situará a la derecha de la página.	125
5.17. Usando el atributo clear de la etiqueta br podemos terminar el texto envolvente y seguir escribiendo debajo de la imagen.	126
5.18. Usando los atributos hspace y vspace podemos controlar la separación entre la imagen y el texto que la envuelve.	128
5.19. Combinando las etiquetas img y a adecuadamente podemos crear imágenes enlace. Los navegadores indican esto mostrando un borde azul alrededor de estas imágenes.	130
5.20. Las imágenes enlace tienen un borde azul por defecto, pero podemos quitarlo o variar su grosor usando el atributo BORDER de la etiqueta IMG.	131
5.21. Las imágenes enlace se usan a menudo para crear iconos de navegación como los que aquí vemos. En estos casos es conveniente quitar el borde usando BORDER=0.	132
5.22. Las imágenes normales también pueden tener un borde de grosor variable, aunque en este caso el color no es azul, sino negro.	132

5.23. Usando los atributos WIDTH y HEIGHT podemos variar el tamaño de las imágenes. Si sólo usamos uno la otra dimensión se ajusta para mantener las proporciones.	134
5.24. Si incluimos junto con la imagen la información de sus dimensiones el navegador podrá reservar espacio para ellas con sólo leer el código.	136
5.25. Con el atributo ALT podemos especificar un texto alternativo a las imágenes. En esta figura vemos como lo muestran Explorer y Netscape.	138
5.26. Algunos navegadores como Explorer muestran el texto de ALT como un bocadillo que aparece cuando ponemos el cursor encima de la imagen.	139
5.2. Extensiones y breve descripción de los principales formatos de imágenes usados en el WWW. ..	141
5.27. Ejemplos de Thumbnails. En el primer caso se muestra un trozo de la imagen y en el segundo la imagen completa reducida, el efecto es similar en ambos casos.	142
5.28. Imagen completa que será mostrada cuando el visitante seleccione los enlaces que acompañan a los thumbnails de la figura 5.27	143
5.29. En la página principal de POST-IT vemos un excelente ejemplo de la utilidad de las imágenes mapa.	145
5.30. En esta figura podemos ver los valores que puede tomar el atributo COORDS en función de si la zona es rectangular, circular o poligonal.	148
5.31. Imagen mapa que puede sustituir el menu en forma de lista de la página de la agencia de viajes Los Alpes.	150
5.32. Mientras que el tamaño de la imagen JPEG es mucho menor la pérdida de calidad apenas es apreciable.	151
5.33. En ocasiones, especialmente si usamos una imagen como fondo, se consigue un efecto mucho mejor haciendo que nuestras imágenes sean transparentes.	153
5.34. Aspecto final de la página de Los Alpes después de aplicar los conocimientos sobre imágenes que hemos adquirido en este capítulo.	155
7.1. En el web de GNOME Hispano encontramos multitud de elementos, gráficos y texto, con una disposición muy cuidada. ¿Cómo lo hacen? Con tablas.	157
7.2. Ya hemos creado nuestra primera tabla, aunque es sencilla ya podemos observar varias cosas curiosas en ella, por ejemplo que por defecto no se dibujan los bordes.	161
7.3. En esta segunda tabla ya podemos ver como son mostrados los encabezados de tabla.	163
7.4. Tabla con bordes. En esta tabla podemos observar lo que ocurre cuando una fila tiene más celdas que otra o cuando insertamos una celda sin nada dentro.	165
7.5. Tabla con celdas vacías. Poniendo o las celdas que antes no existían ahora si son dibujadas, pero sin nada dentro.	169
7.6. Con la etiqueta CAPTION podemos poner un título a la tabla. Este título puede situarse encima (ALIGN="top") o debajo (ALIGN="bottom") de la tabla.	172
7.7. Con el atributo ALIGN podemos controlar el alineamiento de la tabla. Si elegimos ALIGN="left" el texto rodeará a la tabla al igual que ocurría con las imágenes.	174
7.8. Usando alineamiento a la derecha (ALIGN="right") conseguimos un efecto muy similar al alineamiento a la izquierda.	175
7.9. Con ALIGN="right" la tabla se sitúa en el centro de la ventana del navegador y el texto ya no la envuelve.	176
7.10. Distintas posibilidades de los alineamientos vertical y horizontal de las celdas de una tabla.	177
7.11. El punto azul de esta figura nos permite observar el alineamiento por defecto de los encabezados de tabla (TH) y de las celdas normales (TD).	178
7.12. Usando los atributos ALIGN y VALIGN en la etiqueta TR cambiamos el alineamiento por defecto de las celdas de esa fila, pero aún podemos modificar ese alineamiento usando los mismos atributos en cada celda.	180
7.13. Este será el objetivo del ejercicio 7.2. Una pista: debemos usar los atributos WIDTH, HEIGHT, ALIGN y VALIGN.	182
7.14. Usando el atributo NOWRAP evitamos que el texto sea partido en varias líneas, sin embargo no es recomendable usarlo en líneas muy largas como esta.	185

7.15. Los atributos COLSPAN y ROWSPAN nos permiten crear celdas que se extiendan varias columnas o varias filas.	187
7.16. El atributo CELLSPACING controla el espaciado entre los bordes de cada una de las celdas de la tabla.	189
7.17. Con el atributo CELLPADDING podemos aumentar los márgenes entre el contenido de las celdas y su borde.	189
7.18. Ejercicio 7.3. Aunque pueda parecer que hay una sola imagen, en realidad hay cuatro cuidadosamente colocadas.	190
6.20. Las etiquetas anidadas son aquellas que se encuentran entre la instrucción de inicio y la instrucción de fin de otra etiqueta. De esta forma conseguimos que los efectos de ambas se sumen.	192
6.21. En este ejemplo vemos como anidando una tabla dentro de otra podemos crear interesantes efectos en las páginas Web.	193
6.22. Nuestro objetivo en esta práctica 3 será realizar esta tabla. Para su realización necesitaremos usar gran parte de los atributos que hemos aprendido.	194
6.23. Una vez que sabemos como queremos que quede la tabla es muy útil desglosarla en columnas y filas para no liarnos al escribir el código.	195
6.24. Ya tenemos hecha una parte de la tabla, algunas de las celdas se expanden 2 filas, 2 columnas o ambas, aunque en algún caso todavía no es apreciable.	197
6.25. Ya tenemos la tabla prácticamente terminada. Sólo quedan algunos detalles para darla por concluida.	199
6.26. Los asistente (wizard) para la creación de tablas como este, incluido con el editor de HTML HomeSite, son de gran ayuda en la creación de tablas complicadas.	201
6.27. Estructura del sitio Web de Los Alpes. La página principal tendrá enlaces a las otras tres que además estarán enlazadas entre sí.	203
6.28. Boceto de la disposición de los elementos en la página principal y las tablas que vamos a necesitar crear para conseguir esa visualización.	204
6.29. Boceto de la disposición de los elementos del resto de páginas. Aunque conserva un parecido con la página principal hay algunos cambios.	206
6.30. Usando las opciones de cortar y pegar nos ahorraremos el proceso reescribir el código de las prácticas anteriores.	208
6.31. La coletilla aporta un toque de profesionalidad a nuestra página además de ofrecer información interesante a los visitantes.	210
6.32. Página principal de la agencia de viajes Los Alpes. Aunque no podemos ver los bordes de las tablas podemos intuir por donde irían.	212
6.33. Visualización de medida.htm	216
6.34. Resultado final de la página mes.htm	219
6.35. Resultado final de la página superOFE.htm	222
7.1. Usando adecuadamente las etiquetas del lenguaje HTML y con algunos conocimientos de diseño se pueden crear webs tan atractivos como este.	225
7.2. En el CERN (http://www.cern.ch), un instituto de física con poca relación con Internet, se engendró la idea del World Wide Web.	227
7.3. El World Wide Web Consortium, o W3C, fue creado para regular el World Wide Web. Esta regulación incluye una estandarización del lenguaje HTML.	230
7.4. Tanto Microsoft como Netscape crearon sitios web especialmente dedicados a programadores de Internet en el que informan de las tecnologías propietarias soportadas por sus navegadores.	232
7.5. En el servidor web del W3C podemos ver la especificación del nuevo estándar HTML 4.0. La dirección concreta es: http://www.w3c/TR/PR-html40/	235
7.6. En esta figura vemos un excelente uso de los frames o marcos, que dividen la pantalla en cuatro partes para facilitar la navegación.	239
7.7. En HTML 3.2 existen 3 tipos de viñetas para las listas no ordenadas. Con el atributo TYPE podemos elegir entre ellos.	242
7.8. Comparando esta figura con la anterior vemos la similitud entre las distintas viñetas de los elementos de las listas usadas por Netscape Navigator e Internet Explorer.	243

7.9. Usando el atributo TYPE en una etiqueta LI podemos cambiar la viñeta de un único elemento de lista sin afectar a los demás.	244
7.1. Los cinco tipos distintos que puede tomar el atributo TYPE para listas ordenadas	245
7.10. Con el atributo TYPE podemos elegir entre 5 tipos distintos de numeración en las listas de HTML. En esta figura vemos los 4 nuevos.	246
7.11. Si usamos el atributo START podemos variar el número (o letra en estos casos) por los que se empiezan a numerar los elementos de la lista.	247
7.12. Con el atributo VALUE podemos indicar el valor de la numeración de cada uno de los elementos de las listas numeradas.	248
7.13. Con la nueva etiqueta DIV podemos cambiar el alineamiento de todo tipo de elementos HTML (imágenes, listas, párrafos, etc.) de una sola vez.	250
7.14. Los buscadores permiten a los navegantes encontrar la información que buscan. Usando la etiqueta META podemos proporcionarles datos para que incluyan nuestras páginas en sus bases de datos.	253
8.1. Los frames nos permiten dividir la ventana del navegador en varias secciones diferentes. En este caso la ventana a sido dividida en 4 frames.	257
8.2. En el Web www.chess.com podemos ver un excelente ejemplo del uso de frames para mantener visibles las barras de navegación.	259
8.3. Usando el atributo TARGET podremos lograr que pulsando el enlace de un frame el resultado se muestre en otro.	261
8.4. Este debe ser el resultado final tras llevar a cabo todos los pasos en el ejemplo de uso del atributo TARGET para crear varias ventanas.	264
8.5. Variando la dirección de base con la etiqueta BASE variará el significado de las direcciones relativas.	267
8.6. En este Web pulsando sobre los enlaces del frame izquierdo nos movemos por los diferentes temas del frame derecho.	268
8.7. Los Webs con frames constarán de varios documentos HTML. En este ejemplo al haber tres frames serán necesarios 4 documentos HTML.	270
8.8. La estructura de las páginas con frames es diferente a las que no tienen. La principal diferencia es la sustitución de BODY por FRAMESET.	270
8.9. Como vemos la ventana del navegador ha quedado dividida en dos columnas con un ancho del 20% y del 80% del ancho total de la ventana.	272
8.10. Las dos primeras columnas ocupan un ancho absoluto de 100 y 300 pixeles. La tercera al haber usado el asterisco ocupará el resto, que en este ejemplo son 372 pixeles.	273
8.11. En esta ocasión vemos un ejemplo del uso de FRAMESET combinando los distintos modos de indicar el tamaño y el ancho resultante de cada uno para una ventana con 800 pixeles de ancho.	275
8.12. Es posible usar los atributos COLS y ROWS simultáneamente para dividir la ventana en varias filas y columnas.	277
8.13. Ejemplo completo de una página con frames formado por un frame izquierdo pequeño que puede servir de índice y otro derecho con el contenido principal.	281
8.14. Usando los diferentes atributos de la etiqueta FRAME podemos tener un mayor control sobre las diferentes características de los frames.	283
8.15. El navegador NCSA Mosaic no soporta frames, con lo que los usuarios de dicho navegador no verán nada si no usamos la etiqueta NOFRAME.	285
8.16. En este caso sí hemos usado NOFRAMES ofreciendo al navegante un enlace para ver una versión sin frames de las páginas.	286
8.17. Para conseguir estructuras de frames más complejas puede anidarse la etiqueta FRAMESET. En esta figura se indica cómo hacerlo.	287
8.18. Nuestra primera página con frames. La ventana está dividida en 3 apartados (los frames) el izquierdo es el índice, el superior el título y el central el que tiene el contenido.	290
8.19. En general los navegadores dibujan líneas con aspecto 3-D para separar los frames.	292
170. Con el atributo BORDERCOLOR podemos elegir el color que queramos para los bordes de los frames.	293

171. Usando FRAMEBORDER="0" ocultamos los bordes de los frames con lo que se consigue una gran mejora en la apariencia.	295
8.22. Aún usando el atributo FRAMEBORDER si hacemos la ventana del navegador pequeña aparecerán barras de desplazamiento en cada uno de los frames.	297
8.23. Con el atributo BORDER podemos cambiar el grosor de los bordes. En este caso el borde horizontal tiene un grosor de 20 pixeles.	298
8.24. Representación de los cuatro valores especiales del atributo TARGET: _self, _parent, _top, _blank.	300
175. Página Web de Los Alpes creada en la práctica 4 de este curso. Ahora la modificaremos para crear una versión con frames.	302
176. El frame superior será de 163 pixeles, mientras que el inferior ocupará el resto del espacio de la ventana del navegador.	304
177. El contenido de banner.htm, el logo y encabezado de Los Alpes así como los botones de navegación, serán mostrados en el frame superior.	306
178. Todos los botones tienen como destino el frame principal. Por tanto es conveniente usar la etiqueta BASE con el atributo TARGET=principal.	308
179. Página sin cabecera, lista para ser insertada en un frame.	309
180. Estructura de frames mostrando la página main.htm en el frame central.	311
181. Los frames flotantes nos permiten insertar un documento HTML completo en una página Web. .	313
182. Los navegadores que no soporten frames flotantes mostrarán el código alternativo incluido entre las instrucciones <IFRAME> y </IFRAME>	314
183. Usando FRAMBORDER=0 podemos eliminar el borde del frame. Sin embargo si el contenido no cabe en frame la barra de desplazamiento sí se mostrará.	316
184. Con los atributos ALIGN, MARGINWIDTH y SCROLL podremos controlar las diversas características de los frames flotantes.	317
10.2. Netscape Navigator permite configurar en este cuadro de diálogo las aplicaciones externas asociadas a cada navegador.	320
10.3. RealAudio es un formato de sonido específicamente creado para la red el software necesario para su reproducción puede ser obtenido a través de su Web.	322
tabla 10.1.	324
10.4. La organización MPEG es la creadora de este estándar que permite la codificación de audio y vídeo de gran calidad en archivos de pequeño tamaño.	324
tabla 10.2.	325
10.5. Al pulsar sobre un enlace a un archvo que no puede mostrar él mismo, Internet Explorer muestra este cuadro de diálogo preguntando al usuario que quiere hacer.	326
10.6. Cuadro de diálogo para archivos externos mostrado por Netscape Navigator. Si elegimos open la aplicación externa asociada será ejecutada.	327
10.7. Usando el atributo LOWSRC se pueden crear animaciones sencillas pero impactantes como es el caso del web de Spectacle.	330
10.8. La marquesina animada muestra un texto que va moviéndose de derecha a izquierda en la ventana del navegador. En esta figura vemos 7 momentos consecutivos.	333
10.9. La etiqueta MARQUEE está acompañada de numerosos atributos que nos permiten controlar la apariencia de la marquesina.	336
10.10. Internet Explorer 4.0 incluye una serie de nuevas tecnologías entre las que destacan los nuevos controles multimedia.	338
10.11. Macromedia distribuye a través de su Web el plug-in necesario para visualizar en un Web animaciones creadas por su herramienta Director.	340
10.12. Si el navegador tiene algún problema con el plug-in muestra un icono y un espacio en blanco que puede estropear la apariencia de nuestra página.	342
10.13. Usando los atributos WIDTH y HEIGHT de la etiqueta EMBED podemos lograr que los controles de sonido aparezcan. Internet Explorer usa los controles de Windows.	344
10.14. El plug-in usado por Netscape navigator ha creado sus propios controles que tienen la capacidad añadida de poder controlar el volumen.	345

11.1. En la página Web de Sun Microsystems, creadora del lenguaje Java, podemos encontrar gran cantidad de documentación y ejemplos de sus posibilidades.	347
11.2. Gamela es una excelente fuente de recursos Java conocida desde hace tiempo. En la actualidad su dirección está en proceso de cambio a: www.developer.com.	348
11.3. Este es el aspecto que muestra nuestro primer applet insertado en una página Web. En un navegador con soporte Java podemos pulsar sobre los botones para obtener diferentes sonidos.	351
11.4. Usando applets Java podemos insertar juegos interactivos como en este caso. Usando el atributo CODEBASE no es necesario que el applet esté en nuestro servidor.	353
11.5. Usando los atributos ALIGN, VSPACE y HSPACE podemos insertar un applet rodeado de texto.	354
11.6. El applet Animator.class es uno de los más conocidos y usados en las páginas Web por la facilidad que ofrece para crear animaciones.	357
11.7. El navegador Opera no soporta la tecnología Java, por lo que mostrará el código alternativo, en este caso uno de los GIFs de la animación.	358
11.8. Resumen de todos los atributos de la etiqueta APPLET y de su subetiqueta PARAM.	359
11.9. La etiqueta OBJECT ha sido creada por los expertos del Web Consortium para englobar la inclusión de todo tipo de objetos en las páginas Web.	361
11.10. ForecastX ofrece un servicio meteorológico en tiempo real gracias a la tecnología ActiveX. ..	364
11.11. Usando este control ActiveX podemos crear efectos variados con texto, con lo que pueden sustituir el uso de imágenes con la consiguiente disminución tiempo de carga de nuestras páginas.	366
11.12. Usando el mismo control ActiveX que en la figura 11.25 pero con un código algo más complejo podemos crear este efecto tan atractivo.	368
12.1. En la página principal de Tower Communications vemos un formulario que es usado para permitir a los visitantes introducir una cadena de texto que deseen buscar.	371
12.2. En otra sección de la página de Tower Communications vemos un formulario más complicado formado por numerosas cajas de texto y otros elementos.	373
12.3. Nuestro primer formulario está compuesto por una caja de texto. Para introducirla hemos usado la etiqueta INPUT con TYPE="text".	375
12.4. Pulsando sobre la caja de texto nos aparecerá un cursor parpadeante que nos indica que podemos empezar a escribir texto.	376
12.5. Usando el código <INPUT TYPE="submit"> podemos insertar un botón de envío de manera que al pulsar sobre él se envían los datos del formulario.	378
12.6. Con el atributo VALUE podemos cambiar el texto por defecto del botón de envío por uno de nuestra elección. El ancho del botón se adaptará a la longitud del texto.	378
12.7. El atributo SIZE nos permite variar la longitud de la caja de texto. Debemos tener en cuenta que en diferentes navegadores un mismo valor no se corresponde a una misma longitud.	380
12.8. El atributo ALIGN de la etiqueta INPUT nos permite alinear los controles. En este caso vemos una caja de texto alineada a la izquierda (ALIGN="left").	382
12.9. Las cajas de texto para claves muestran asteriscos en lugar del texto que escribimos para aumentar la confidencialidad.	384
12.10. Los botones de elección permiten elegir entre un conjunto de opciones, de las cuales podemos elegir sólo una.	385
12.11. En un mismo formulario podemos incluir varios grupos de botones de elección. Los visitantes podrán seleccionar una opción en cada uno.	386
12.12. Las cajas de selección son independientes unas de otras, por los que es posible tener varias seleccionadas.	388
12.13. Este es el aspecto que tiene el código que hemos escrito hasta ahora. En el formulario de la clave sólo se podrán insertar 4 caracteres.	391
12.14. Este es el resultado final del formulario del ejercicio 12.1 en el que destaca el uso de una imagen personalizada como botón de envío.	393
12.15. El botón de borrado (<INPUT TYPE="reset" >) permite al usuario borrar los datos introducidos y recuperar los iniciales.	396
12.16. Los botones genéricos tienen multitud de usos, entre los más usados nos encontramos con el botón enlace y el botón para cerrar una ventana, que, como todos, requieren hacer uso de un lenguaje de script.	

400

12.17. Los navegadores en modo texto como Lynx también son capaces de mostrar formularios, que aunque no sean gráficos tienen la misma funcionalidad.	403
12.18. En esta figura podemos apreciar cajas de texto y de clave, botones de elección y cajas de selección tal y como son mostradas por Lynx.	403
12.19. El atributo TITLE es una extensión de Microsoft que permite dar una descripción a los controles que será mostrada en un bocadillo.	405
12.20. Los atributos DISABLED y READONLY permiten deshabilitar un control de manera que sólo sirva para mostrar datos.	407
12.21. Las áreas de texto (<TEXTAREA> </TEXTAREA>) permiten al usuario introducir texto de varias líneas como si de un sencillo editor de texto se tratara.	409
12.22. Lynx muestra las áreas de texto como un conjunto de líneas (tantas como hemos indicado con el atributo COLS) sobre las que el usuario podrá escribir.	410
12.24. Los cuadros de selección permiten mostrar varias opciones para que el usuario pueda elegir. En esta ocasión vemos su forma de persiana desplegable.	413
12.25. Usando el atributo SIZE con un valor mayor que uno conseguimos que el cuadro de selección se muestre como una lista.	414
12.26. Usando el atributo MULTIPLE el usuario podrá seleccionar varias de las opciones que se ofrecen en la lista.	416
12.27. El navegador Lynx también es capaz de mostrar cuadros de selección, por lo que podremos usarlo con libertad.	417
12.28. Este es el aspecto inicial de la plantilla del sitio Web de Los Alpes.	419
12.29. El primer paso será modificar la plantilla, cambiando el título y la fecha y escribiendo un texto introductorio.	419
12.30. Usando los conocimientos adquiridos en este capítulo utilizamos las etiquetas INPUT y SELECT para crear los controles del formulario.	421
12.31. Usando una tabla de nueve filas y tres columnas controlamos la disposición de todos los elementos de la tabla para conseguir un diseño profesional.	423
12.32. Este es el aspecto final del formulario una vez introducido en la estructura con frames del sitio Web creado en la práctica 5.	425
12.33. Como método alternativo a los CGIs pueden mandarse los datos del formulario por correo electrónico. En ese caso el navegador pedirá al usuario confirmación de la acción.	428
12.33. Como método alternativo a los CGIs pueden mandarse los datos del formulario por correo electrónico. En ese caso el navegador pedirá al usuario confirmación de la acción.	430

Introducción al lenguaje HTML

Jorge Ferrer

Victor García

Versión 1.0

Copyright Jorge Ferrer y Victor García. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation. Puede consultar una copia de la licencia en: <http://www.gnu.org/copyleft/fdl.html>

En esta primera entrega de este Curso completo de HTML haremos un primer estudio del lenguaje HTML y analizaremos las herramientas que son necesarias o al menos recomendables para realizar una página web. Además aprenderemos qué es y qué no es el lenguaje HTML; cual es la estructura de una página web; como dar un título a nuestra página y como crear encabezados y texto en ella.

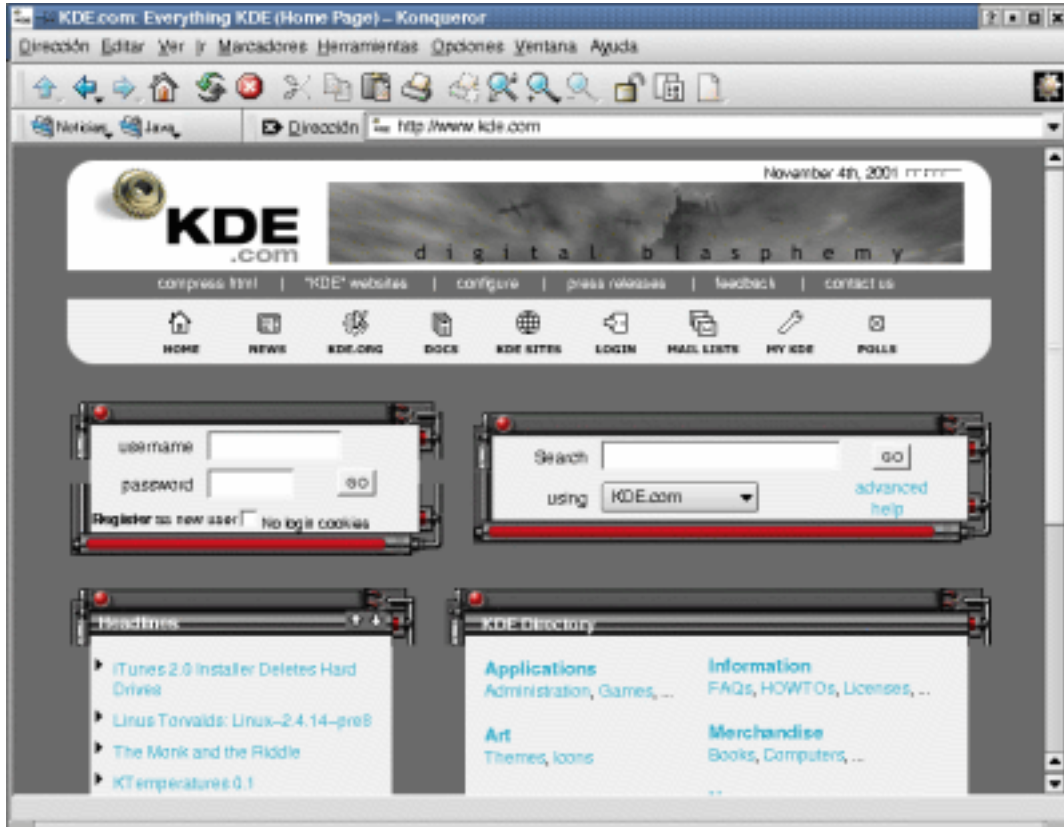
Introducción

El World Wide Web (WWW) es un sistema que contiene una cantidad de información casi infinita. Pero esta información debe estar ordenada de alguna forma de manera que sea posible encontrar lo que se busca. La unidad básica donde está almacenada esta información son las páginas Web. Estas páginas se caracterizan por contener texto, imágenes, animaciones... e incluso sonido y video.

Una de las características más importantes de las páginas Web es que son hipertexto. Esto quiere decir que las páginas no son elementos aislados, sino que están unidas a otras mediante los *links* o enlaces hipertexto. Gracias a estos enlaces el navegante de internet puede pulsar sobre un texto (texto al que llamaremos de ahora en adelante *activo*) de una página para *navegar* hasta otra página. Será cuestión del programador de la página inicial decidir que palabras o frases serán *activas* y a donde nos conducirá pulsar sobre ellas.

En algunos ambientes se discute que el World Wide Web ya no es un sistema hipertexto sino hipermedia. Los que defienden el cambio se apoyan en que aunque en sus orígenes el WWW constaba únicamente de texto en la actualidad es un sistema principalmente gráfico y se puede hacer que las zonas *activas* sean, no sólo texto, sino imágenes, videos, botones,... en definitiva cualquier elemento de una página. Aún así, el término original no ha sido reemplazado todavía y por tanto será el empleado a lo largo de esta obra.

Figura 1.1. Aunque al principio el texto predominaba en el WWW, en la actualidad las imágenes son mayoría como podemos observar en una de las principales webs del proyecto KDE.



Características del lenguaje HTML

Pero empecemos ya con lo que nos interesa. ¿Cómo se hace una página Web? Cuando los diseñadores del WWW se hicieron esta pregunta decidieron que se debían cumplir, entre otras, las siguientes características:

- El Web tenía que ser distribuido: La información repartida en páginas no muy grandes enlazadas entre sí.
- El Web tenía que ser hipertexto y debía ser fácil navegar por él.
- Tenía que ser compatible con todo tipo de ordenadores (PCs, Macintosh, estaciones de trabajo...) y con todo tipo de sistemas operativos (Windows, MS-DOS, UNIX, MAC-OS,...).
- Debía ser dinámico: el proceso de cambiar y actualizar la información debía ser ágil y rápido.

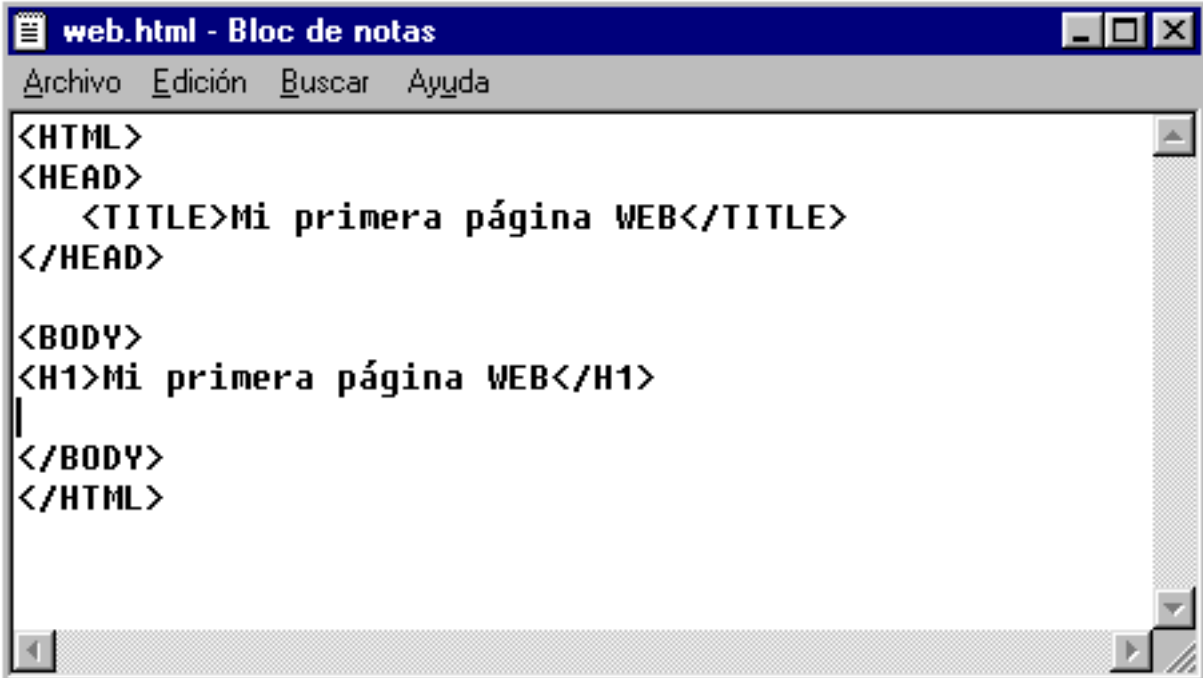
Estas características son las que marcaron el diseño de todos los elementos del WWW incluida la programación de páginas Web. Como respuesta a todos estos requisitos se creó el lenguaje HTML (HiperText Markup Language), cuyas siglas significan "lenguaje hipertexto de marcas".

Este lenguaje será el encargado de convertir un inocente archivo de texto inicial en una página web con diferentes tipos y tamaños de letra, con imágenes impactantes, animaciones sorprendentes, formularios interactivos, etc.

Qué se necesita para crear una página web

Una de las características de este lenguaje más importantes para el programador es que no es necesario ningún programa especial para crear una página Web. Gracias a ello se ha conseguido que se puedan crear páginas con cualquier ordenador y sistema operativo. El código HTML, como hemos adelantado en el párrafo anterior, no es más que texto y por tanto lo único necesario para escribirlo es un editor de texto como el que acompañan a todos los sistemas operativos: edit™ en MS-DOS, block de notas en Windows, vi™ en UNIX, etc. Por supuesto estos no son los únicos editores de texto que pueden ser usados, sino cualquier otro. También se puede usar procesadores de texto, que son editores con capacidades añadidas, como pueden ser Microsoft Word™ o WordPerfect™ pero hay que tener cuidado porque en ocasiones hacen traducciones automáticas del código HTML que no siempre son deseadas. En estos dos últimos casos, también hay que tener en cuenta que deberemos guardar el archivo en modo texto.

Figura 1.2. Un editor de texto simple, como el block de notas de Windows, es todo lo necesario para crear una página Web



```
<HTML>
<HEAD>
  <TITLE>Mi primera página WEB</TITLE>
</HEAD>
<BODY>
<H1>Mi primera página WEB</H1>
|
</BODY>
</HTML>
```

Una vez hemos escrito el código deberemos guardar el archivo (con formato de texto) con la extensión .html (o .htm en MS-DOS, Windows 3.1 o cualquier otro sistema que sólo acepte tres letras en la extensión. Los siguientes son nombres válidos de archivos que contengan código HTML: index.html, index.htm, principal.html, PRINCIPAL.htm, etc...

NOTA: Si el editor o procesador de textos usado para crear la página obliga a usar la extensión .txt al guardar el archivo en modo texto, deberemos guardarlo con esta extensión, por ejemplo como index.txt, y posteriormente

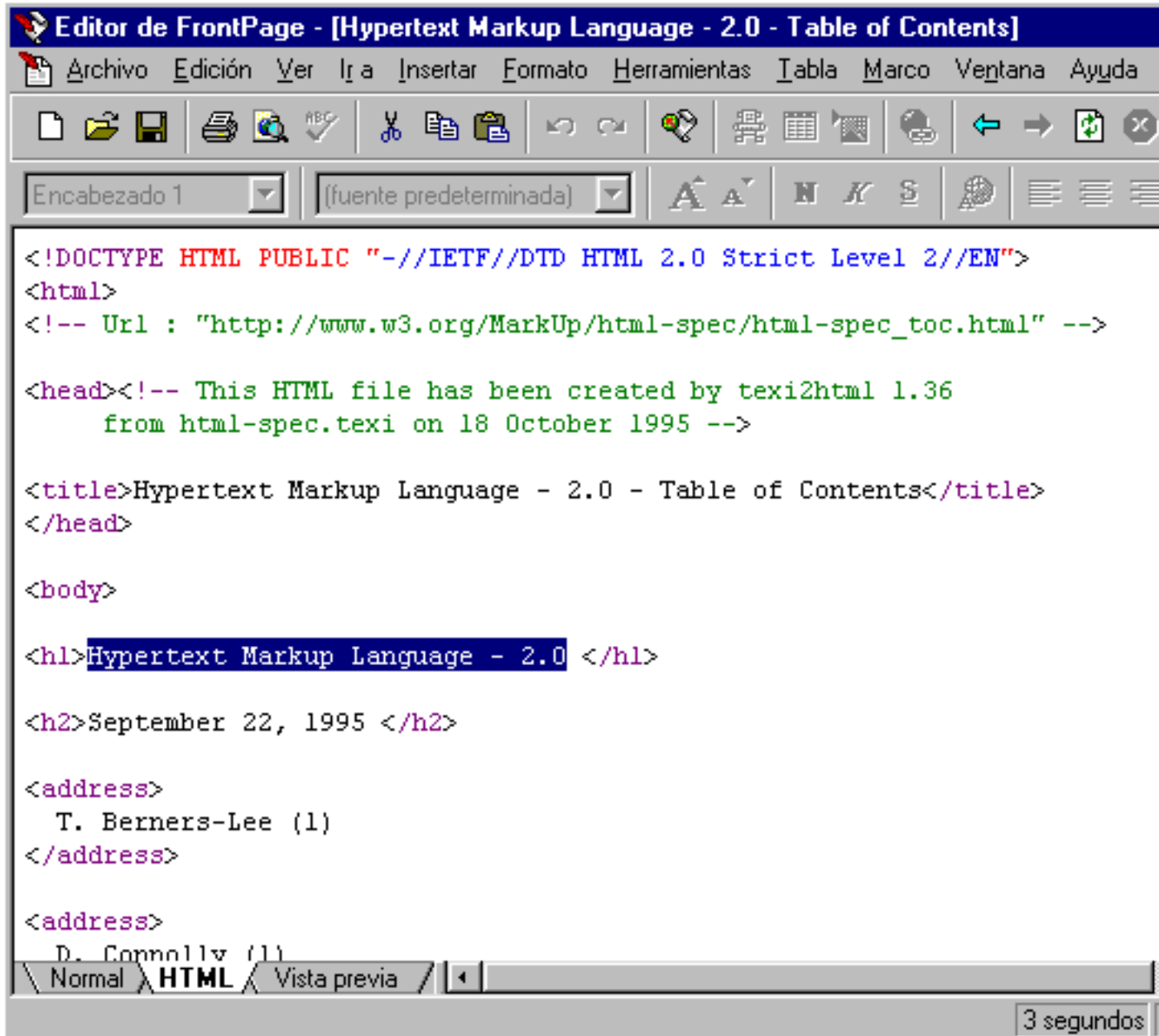
cambiar el nombre del archivo desde fuera del programa a `index.html` o `index.htm`. Para ello usaremos el comando **rename** en MS-DOS; en Windows 3.1 lo haremos mediante el administrador de archivos y en Windows 95 con el explorador. En otros entornos, como Linux, es más raro que se de esta situación.

ATENCIÓN: El World Wide Web es un sistema que diferencia las mayúsculas de las minúsculas. Es un error común llamar a un archivo `index.html` y luego referirse a él como `Index.html`. Aunque en nuestro ordenador puede funcionar al publicarlo seguramente no lo hará. Por esta razón es una norma general usar siempre minúsculas para los nombres de los archivos html.

Últimamente han aparecido nuevas alternativas que facilitan la programación de páginas Web. Son los editores HTML. Podemos dividir estos editores en dos grupos:

- **Asistentes:** ayudan a crear el código HTML e incluyen plantillas de código *prefabricadas*, por ejemplo HotDog™ (Win), HomeSite™ (Win), HTML Editor™ (Mac), Quanta™ (Linux, KDE) o Bluefish™, (Linux, GNOME).
- **Conversores:** son programas con otra función que la de la programación Web pero que permiten convertir a HTML. Son ejemplos de conversores Microsoft Word™, Quark XPress™ y PageMaker™.
- **Editores WYSIWYG (What You See Is What You Get, lo que ves es lo que obtienes):** estos editores permiten crear páginas web sin escribir código HTML como si se tratase de un programa de dibujo por ordenador. Algunos ejemplos de este tipo de editores son Macromedia Dreamweaver™, HotMetal™ o Microsoft Frontpage™.

Figura 1.3. Algunos editores WYSIWYG permiten, además, modificar el código HTML directamente.



Algunas razones para usar HTML

Todos estos editores HTML tipos pueden ser de gran ayuda y tienen sus ventajas e inconvenientes, pero la experiencia demuestra que conocer el lenguaje HTML ofrece bastantes ventajas:

- Seremos capaces de aprovechar todas las características de este lenguaje, incluso las más nuevas.

- Aunque se use inicialmente un editor WYSWYG tendremos los conocimientos suficientes para modificar posteriormente el código que éste ha creado. Así se pueden corregir errores o incluir etiquetas no soportadas por el editor.
- No dependeremos de la disponibilidad de una herramienta concreta para poder crear páginas web. Con un simple editor de textos será suficiente.

El principal problema de usar editores conversores y WYSIWYG es la necesidad de trabajar con dos archivos-fuentes por separado, por un lado el archivo del editor y por otro el archivo del código HTML una vez generado, lo que nos complica la vida a la hora de realizar cambios en nuestro Web. Sin embargo su utilidad es innegable y su combinación con un buen conocimiento del lenguaje HTML nos convertirá en unos grandes programadores de páginas Web capaces de realizar creaciones impactantes visualmente y de gran calidad técnica.

HTML es un lenguaje descriptivo

Probablemente el lector ha usado alguna vez un procesador de texto (Abiword™, Microsoft Word™ o KWord™) o un programa de descripción de páginas (QuarkXPress™). Con este tipo de aplicaciones el usuario tiene un control total sobre los elementos del documento: se le puede decir al programa "pon este texto con este tamaño", "sitúalo a 1 cm. del borde", "usa este sangrado para los párrafos", etc... Con HTML el programador no tiene este tipo de control sobre los elementos que incluirá en su página. El objetivo de este lenguaje será simplemente describir cómo es una página de manera que examinando esa descripción el navegador del usuario final sea capaz de mostrarlo de la mejor manera posible. Con HTML podremos indicarle al navegador que este es un titular, aquí comienza un párrafo, estos son elementos de una lista, etc. posteriormente el navegador decidirá como mostrar esos elementos.

Figura 1.4. Los navegadores en modo texto como Lynx también pueden mostrar páginas Web gracias al carácter descriptivo de HTML.

```

IENAC - Ingenieros en Acci3n

IENAC
Ingenieros en Acci3n

"Acci3n Solidaria UPM: Ingenieros en Acci3n" naci3 en
Surgi3 como forma de plasmar algunas actividades de in
diversos grupos de estudiantes, dentro de la U.P.M., v
realizando. Se buscaba crear una Asociaci3n que fuera
desde la cual se llevaran a cabo iniciativas de ayuda
social.

[USEMAP]
[Breve historia] [Actualidad] [Sugerencias] [Links] [S
discapacidad] [C3mo colaborar]

Actualmente las principales actividades de la asoci
siguientes:
(pulsa sobre el t3tulo para acceder a informaci3n ma
cada actividad):

-- press space for next page --
Arrow keys: Up and Down to move. Right to follow a link
IH)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [d
```

¿Por qué funciona así? En un principio esta característica del HTML puede resultar molesta para el creador de la página, que no puede saber como será vista su página más que de una forma aproximada. Sin embargo es fundamental para conseguir que el World Wide Web pueda ser navegado con cualquier tipo de ordenador (dicho con otras palabras, que el WWW sea multiplataforma). Debemos tener en cuenta que no todos los sistemas operativos son gráficos, que incluso usando un mismo sistema operativo existen diferentes resoluciones de pantalla, ... Por esta razón el lenguaje necesario para crear páginas debe ser descriptivo y como consecuencia tras crear una página el programador debe ver como es mostrada con varios navegadores distintos como Netscape Navigator™ o Microsoft Internet Explorer™ incluyendo algún navegador de texto como Lynx™.

Pero una consecuencia más importante aún de esta característica, es que ha permitido que aparezcan navegadores vocales, que leen las páginas en vez de mostrarlas. Gracias a ello personas ciegas también han podido disfrutar del WWW. Es más, esta misma tecnología esta siendo usada para los nuevos navegadores embarcados en coches que leen la página al conductor para que no se distraiga.

NOTA: Una vez creada una página es recomendable probar nuestra página, no sólo con varios navegadores, sino también con varias resoluciones de pantalla distintas como por ejemplo: 640x480, 800x600 y 1024x768. Si existe la posibilidad de probarlo además con un navegador vocal, mucho mejor aún.

Un buen ejemplo de este carácter descriptivo es la definición de titulares, también llamados encabezados, en HTML. Vamos a adelantarnos un poco en esta introducción para mostrar este ejemplo, en él se muestra como poner el titular de una página:

```
<h1>Bienvenido a la página de La Super Editorial</h1>
```

Con este código estamos describiendo el texto *Bienvenido a la página de La Super Editorial* como el texto de máxima importancia en nuestra página, es el titular. En los navegadores más usados como el Netscape Navigator™ o el Microsoft Internet Explorer™ este titular será mostrado como un texto en negrita y de tamaño grande (el tamaño más grande posible). Sin embargo repetimos una vez más que existen navegadores como pueden ser Lynx™ o Emacs-W3™ que son bastante usados por la comunidad académica y universitaria que sólo pueden mostrar texto y no pueden variar el tamaño de la letra. Por esta razón la etiqueta `<h1>` no dice que el texto que encierra debe ser de tal tamaño o si debe ser negrita o no. Esta etiqueta sólo le dice que este texto es el más importante de la página, el titular. Sabiendo esto el navegador que funcione en modo texto puede resaltar dicho titular con los medios de que dispone, como por ejemplo poniéndolo en mayúsculas o usando distintos sangrados.

También podemos encontrar ventajas de la descripción mirando al futuro. Imaginemos que la informática evoluciona hasta tal punto que los monitores de dentro de unos años pueden mostrar objetos en tres dimensiones. Sería muy interesante que el titular de nuestra página fuese en 3-D de manera que resaltase lo máximo posible. Pues bien, si lo hemos definido como cabecera principal (y no como letra de tamaño 20 puntos y en negrita, por ejemplo) el navegador sabe que queremos que ese texto sea el más importante y puede actuar de manera inteligente poniéndolo automáticamente en tres dimensiones, ¡sin que nosotros hayamos cambiado una sola línea de nuestro código! Parece interesante ¿no?

Una vez a quedado claro cual es el espíritu de HTML, hay que decir que últimamente se han introducido métodos para conseguir un mayor control sobre las páginas. Estos métodos permiten controlar aspectos como el tamaño de la letra la disposición (aproximada) de imágenes y texto de manera que se facilita la labor del programador. Pero hay que tener siempre en mente que siempre habrá ligeras diferencias entre como verán las páginas unos usuarios u otros y no debemos pretender poder controlar la presentación y diseño de nuestra página hasta el mínimo detalle.

Las bases de HTML

Una vez repasadas unas nociones básicas de HTML y de la programación de páginas Web en general entraremos de lleno en la programación con este lenguaje. Pronto nos daremos cuenta de la sencillez de este lenguaje lo que le convierte en un lenguaje muy fácil de aprender y que nos permitirá crear páginas con mayor facilidad aún.

Las órdenes de este lenguaje estarán formadas por unos comandos llamados etiquetas que pueden tener o bien la siguiente estructura:

```
<NOMBRE_ETIQUETA>
```

O bien esta otra:

```
<NOMBRE_ETIQUETA> TEXTO </NOMBRE_ETIQUETA>
```

Como vemos, la primera estructura está formada por una única instrucción y la segunda por dos: una que marca el inicio de la etiqueta y otra que marca el final, con texto entre ambas.

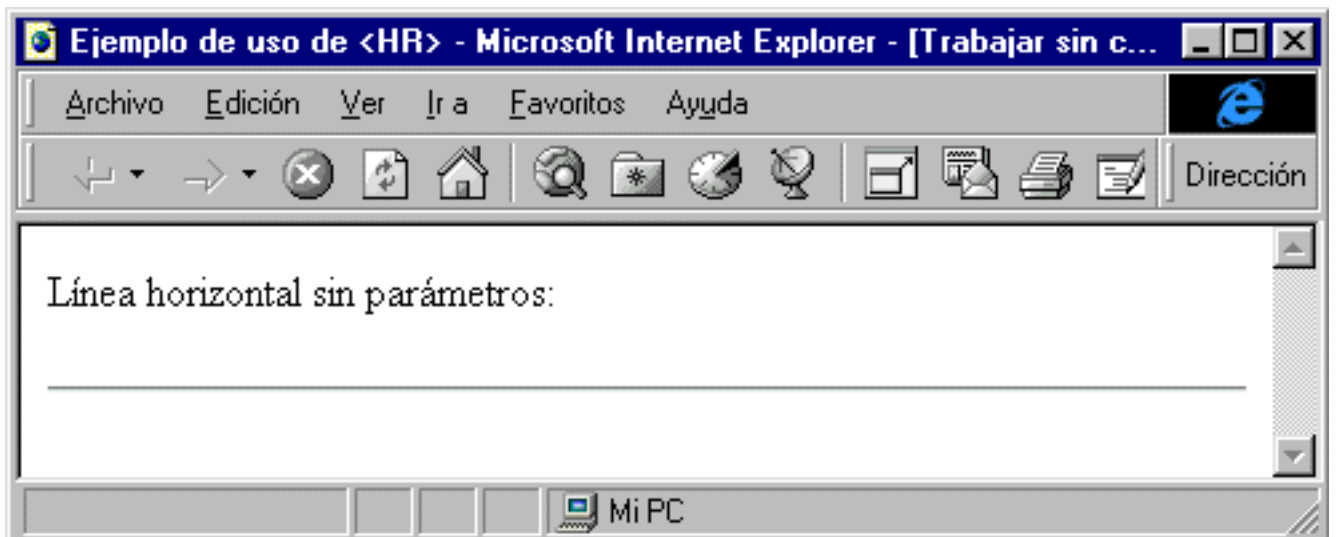
Por ejemplo, para indicarle al navegador que queremos que pinte una línea horizontal debemos escribir:

```
<hr>
```

Así de sencillo. Como vemos, el nombre de la etiqueta va delimitado por los símbolos *menor que* (<) y *mayor que* (>), todas las instrucciones de HTML deben ir encerradas entre estos dos símbolos.

Pero las etiquetas no se limitan a indicar ordenes tan sencillas, estas órdenes tienen en ocasiones parámetros. Por ejemplo la etiqueta <hr> tal y como lo hemos hecho anteriormente daría lugar a la línea que vemos en la figura 1.5, es decir le dice al navegador que dibuje una línea horizontal en la pantalla. Esta línea tiene un grosor predeterminado y un ancho variable en función del tamaño de la ventana del navegador.

Figura 1.5. Uso básico de una etiqueta HTML sin ningún tipo de parámetros. Ejemplo usando la etiqueta <hr> para crear una línea horizontal.



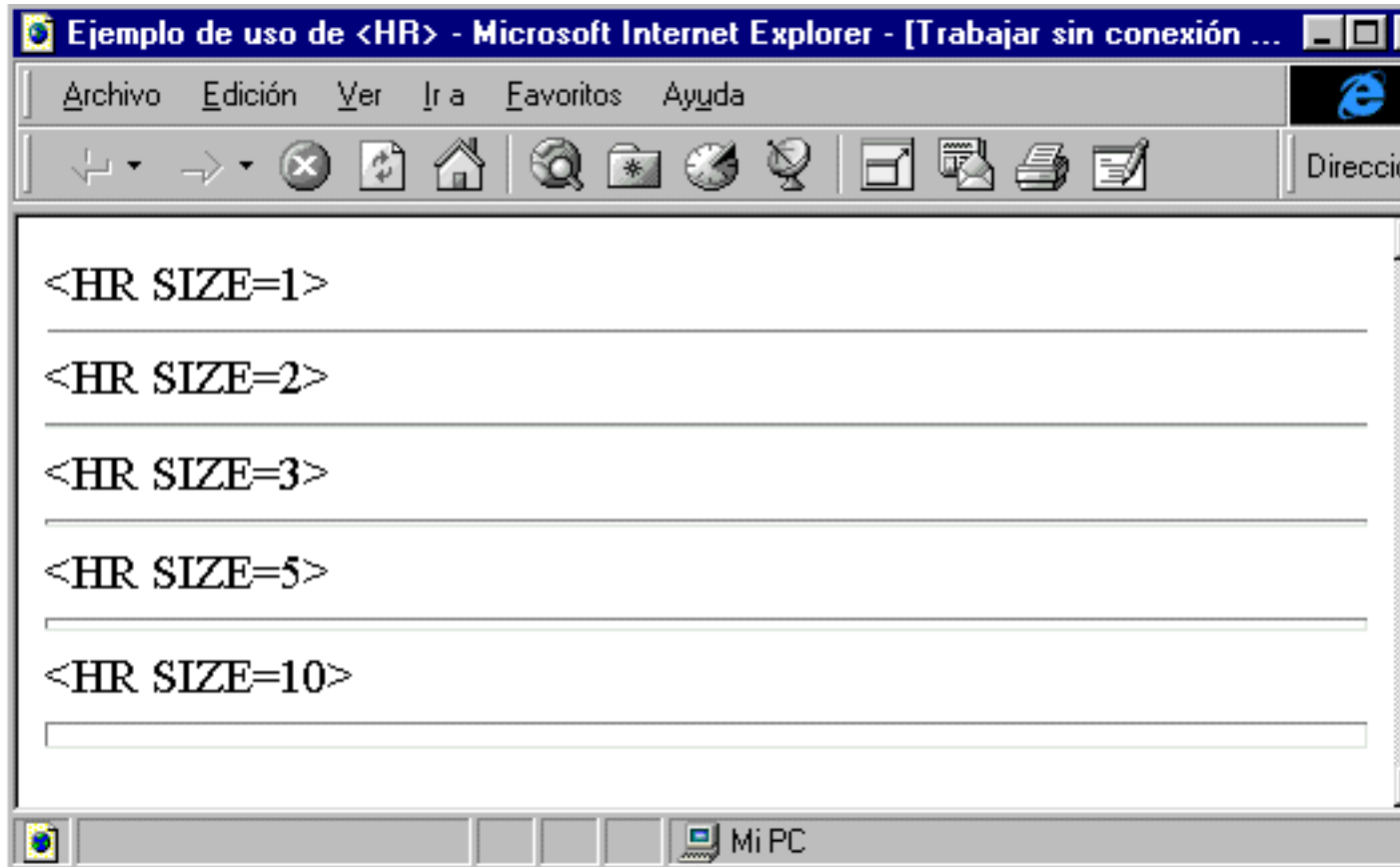
Sin embargo hay muchas formas de pintar una línea y sería deseable poder elegir detalles tales como la anchura y el grosor que va a tener dicha línea. Para especificar este tipo de detalles se crearon los atributos de las etiquetas. Este nuevo elemento se introduce en una etiqueta de la siguiente manera:

```
<etiqueta atributo="valor">
```

Es decir, en primer lugar ponemos el nombre de la etiqueta, después el nombre del atributo seguido por un signo igual y posteriormente el valor que queremos darle a ese atributo encerrado entre comillas una etiqueta puede tener tantos atributos como se deseen y en ocasiones son necesarios para que la etiqueta tenga algún significado.

En nuestro ejemplo de la línea horizontal existe un atributo llamado SIZE, que significa *tamaño* en inglés, que permite controlar el grosor de la línea que será dibujada en pantalla. Veamos un ejemplo, en la figura 1.6 podemos ver la línea horizontal con diferentes grosores.

Figura 1.6. Las etiquetas tienen atributos para cambiar aspectos del efecto que producen. En este ejemplo vemos como se puede cambiar el grosor de una línea con el atributo SIZE.



El valor que toma el atributo `size` determina el número de puntos de pantalla o píxeles que debe ocupar, en grosor, la línea. En posteriores capítulos estudiaremos en detalle la etiqueta `<hr>` y todos sus atributos.

En la figura 1.6 vemos que el valor numérico que le damos al atributo `size` no va entrecomillado, por ejemplo en `<hr size=5>`. En HTML no es necesario entrecomillar los valores de los atributos cuando están formados únicamente por números [0-9], letras [a-z, A-Z], guiones y puntos. En todo caso es *muy recomendable* incluir las comillas siempre. De esta forma no tenemos que preocuparnos de cuando ponerlas y cuando no y a la vez cumplimos el nuevo estándar XHTML. Para los curiosos adelantaremos que XHTML es idéntico a HTML pero con algunas restricciones adicionales del mundo de XML.

Antes de empezar con el código HTML básico conviene hacer saber al lector que aunque en todos los ejemplos usados hasta ahora las etiquetas usadas estaban escritas en mayúsculas en realidad es indiferente el uso de estas o de minúsculas. Sin embargo las mayúsculas son de uso común para que sea más fácil distinguirlas del texto y por tanto seguiremos usando este convenio a lo largo del curso.

En los ejemplos también usaremos distinto sangrado (alineación de los párrafos) para los diferentes elementos del código para mostrar los niveles de anidamiento. Este sangrado no es necesario y el lector puede optar por ignorarlo si decide copiar los ejemplos para verlos en su navegador, sin embargo facilita la lectura del código y no afecta al modo en que los objetos de nuestra página serán mostrados.

Primeros pasos con HTML

Conocidas ya las principales características de HTML estamos listos para aprender sus principales etiquetas y para crear nuestra primera página.

Estructura de una página

Todo documento HTML está formado por los bloques que podemos apreciar en la figura 2.1. Pasemos a explicar cada uno de ellos.

Figura 2.1. Estructura de bloques de un documento HTML.



En primer lugar deberemos especificar que el archivo de texto que estamos escribiendo es un documento HTML, para ello usamos las instrucciones de inicio y fin de la etiqueta `<html>` al principio y al final de la página respectivamente:

```
<html>
Código de la página
</html>
```

El código de la página esta formado a su vez por dos grandes bloques, la cabecera y el cuerpo.

La cabecera de la página está delimitada por las instrucciones de inicio y fin de la etiqueta `head`. Estas instrucciones deben estar dentro de la etiqueta HTML de la siguiente manera:

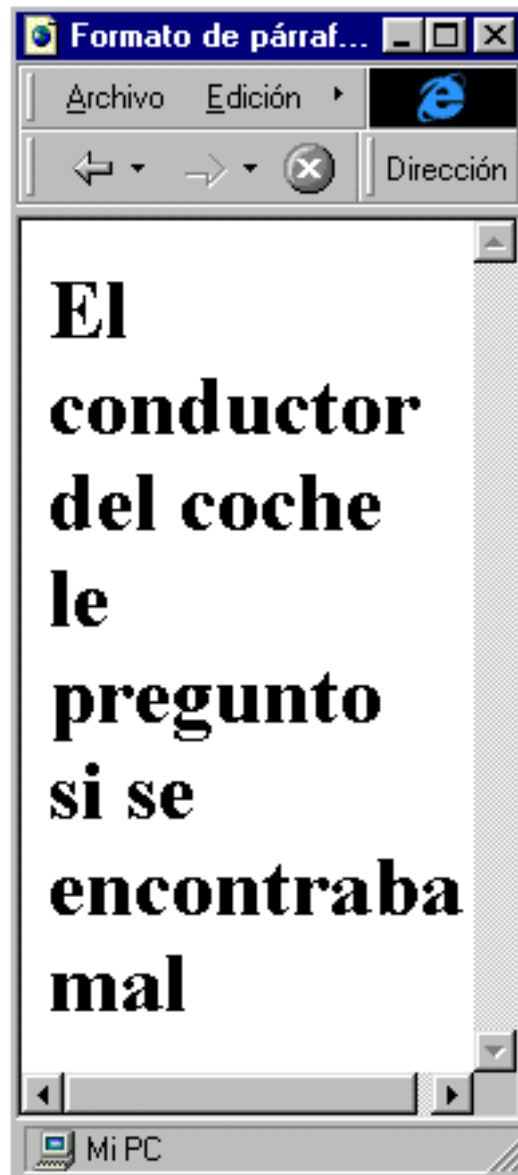
```
<html>
  <head>
    Elementos de la cabecera
  </head>
  ... Resto de código de la página ...
</html>
```

En la cabecera de la página se introduce toda aquella información que afectará a toda la página. En un principio esta información se limitará al título. Este título se indicará con la etiqueta `title` usando la siguiente sintaxis:

```
<html>
<head>
  <title>Mi primera página WEB</title>
</head>
</html>
```

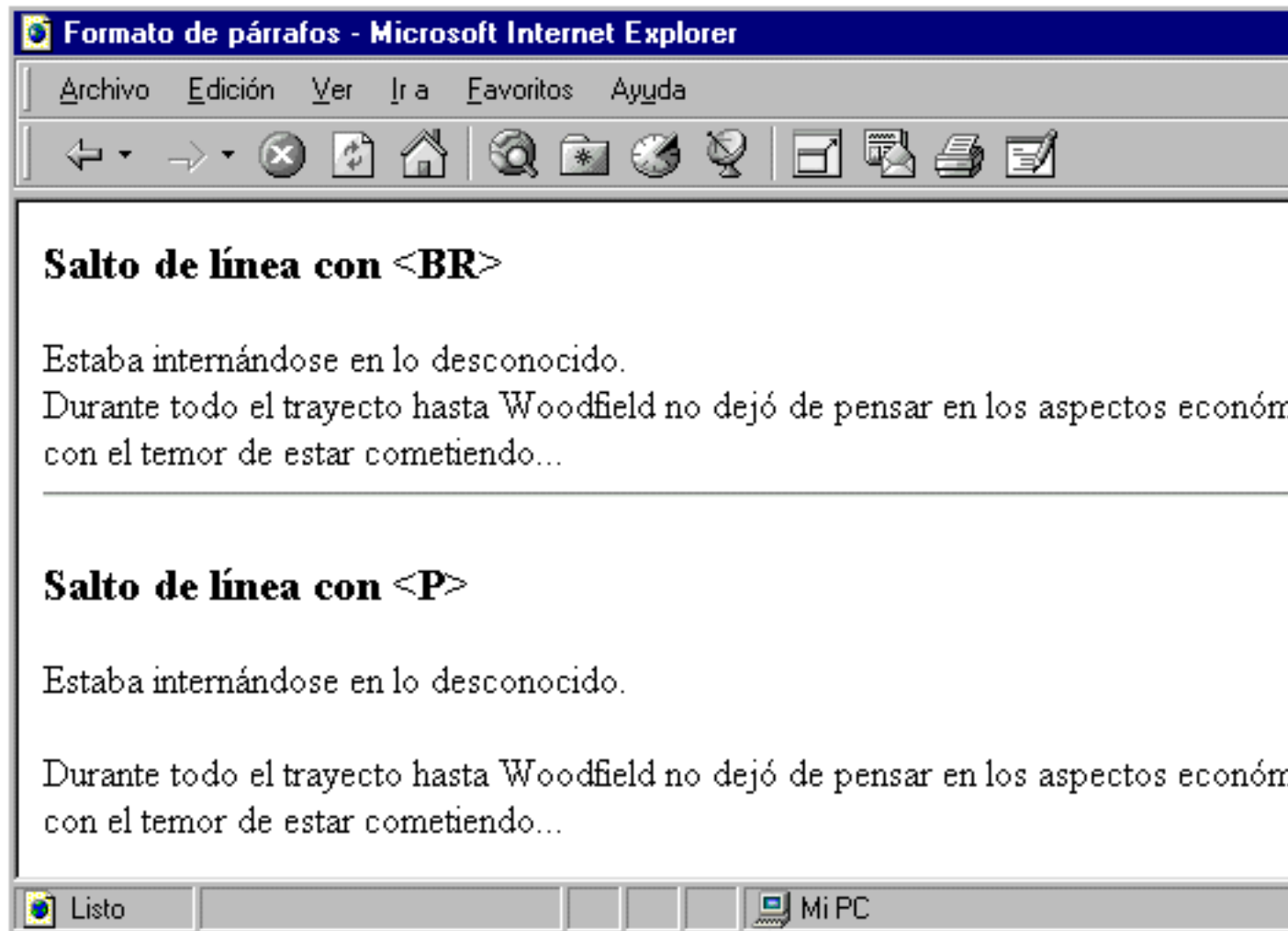
El resultado de ver esta página con nuestro navegador es el que se observa en la figura 2.2. Como vemos, el área de la pantalla donde suele estar la página está todavía vacía, pero si nos fijamos en la cabecera de la ventana del navegador vemos que el título a pasado de ser Microsoft Internet Explorer™ a "Mi primera página WEB - Microsoft Internet Explorer™". Es decir el título que le demos a nuestra página con la etiqueta <TITLE> pasará a ser el título de la ventana del navegador.

Figura 2.2. El título de la página es mostrado en la parte superior de la ventana del navegador.



Además cuando un usuario vea nuestra página y decida incluirla en su lista de webs preferidos con la opción agregar a favoritos del menú Favoritos del navegador (en este caso el explorer) añadirá el nombre que le hemos dado a nuestra página en dicho menú, tal y como podemos apreciar en la figura 2.3. Aunque el nombre con el que se almacena un Favorito puede ser cambiado por el usuario, si hemos puesto un buen título, lo suficientemente descriptivo a cada una de nuestras páginas le evitaremos esa molestia.

Figura 2.3. Cuando los navegantes incluyan nuestra página a su menú de favoritos en su menú aparecerá el título que le hayamos dado a nuestra página con la etiqueta <TITLE>



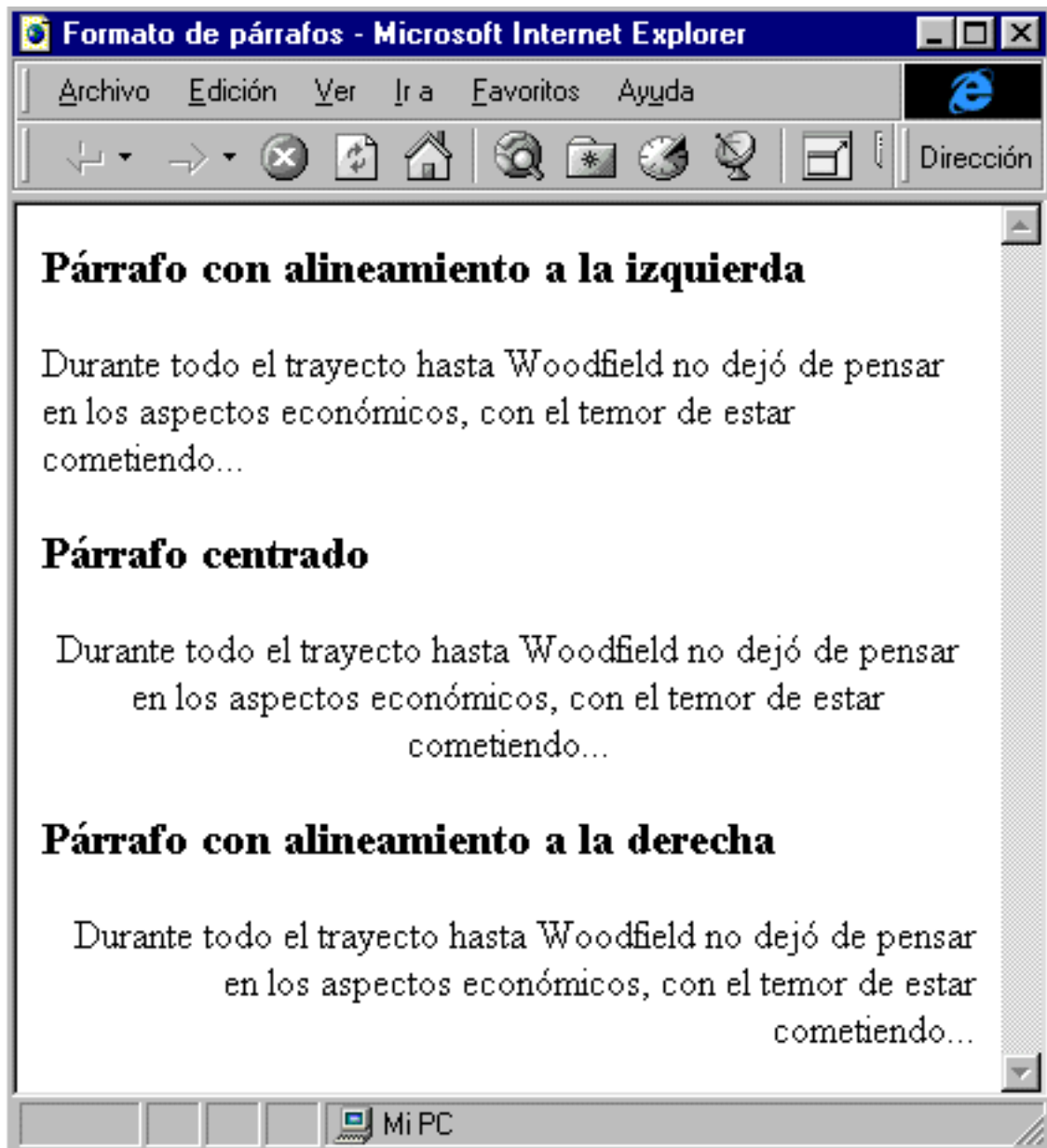
Es importante resaltar que nada de lo que se ponga en la cabecera de la página será mostrado en la zona del navegador dedicada a mostrar el contenido, es decir, en la zona que vemos en blanco en la figura 2.2.

Esto nos induce a pensar que existe otra forma de indicar como será el contenido de la página, pues así es: el cuerpo es el siguiente gran bloque de nuestro documento HTML, éste quedará delimitado por la etiqueta `body`. En su interior introduciremos todos aquellos elementos de los que queremos que conste nuestra página como pueden ser texto, imágenes, tablas, etc. Conociendo la etiqueta `body` podemos ampliar el ejemplo anterior para que incluya texto.

```
<html>
<head>
  <title>Mi primera página WEB</title>
</head>
<body>
  Hola a todos, como han deducido por el titulo esta es la
  primera pagina web que hago, espero que les
  guste. Fdo. Jorge
</body>
</html>
```

En la figura 2.4 podemos apreciar como muestra el Internet Explorer™ este código. Como vemos todo el texto que hemos escrito entre las instrucciones de inicio y fin forman ahora el cuerpo de nuestra página.

Figura 2.4. En esta figura observamos la zona de la ventana en la que los navegadores mostrarán el texto que pongamos en el cuerpo de la página.



NOTA: Es posible que aunque no incluyamos las etiquetas `html` o `body` nuestras páginas sean visualizadas con algunos navegadores. Sin embargo no debemos confiarnos ya que lo más seguro es que con el resto de navegadores no se vean bien. Por esta razón es muy recomendable usar ambas etiquetas tal y como se ha descrito.

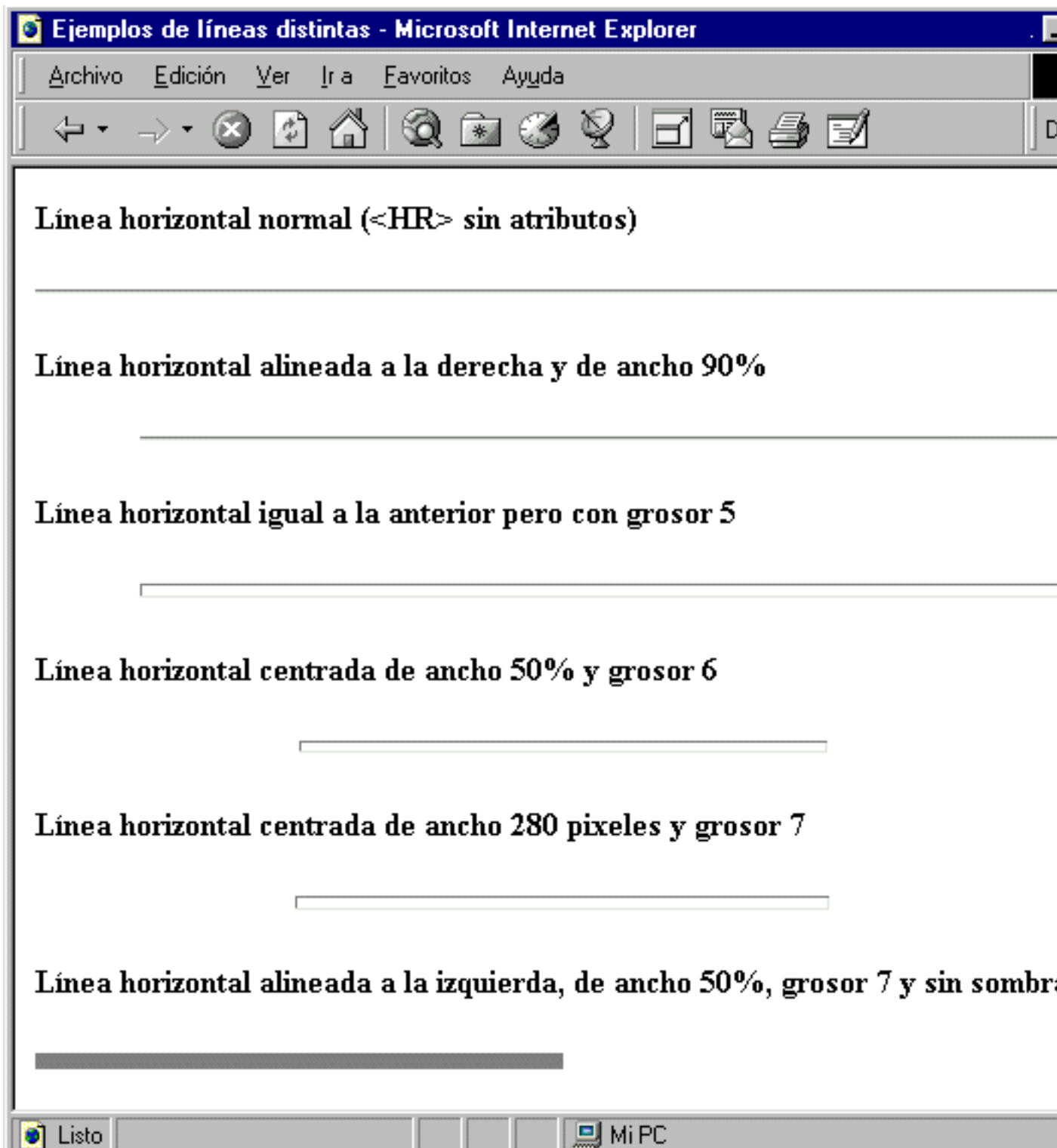
Una vez que ya sabemos insertar contenido en una página pasaremos a estudiar las diferentes etiquetas que sirven para darle formato a ese contenido.

Los encabezados

Usualmente un documento tiene, además de texto llano, una serie de encabezados o titulares. Para ello el lenguaje HTML posee una serie de etiquetas que permiten disponer de titulares de hasta 6 niveles de importancia.

Estas etiquetas son: <h1>, <h2>, <h3>, <h4>, <h5> y <h6>. La letra "h" al principio del nombre de estas etiquetas provienen de la abreviatura de la palabra inglesa *heading* que significa encabezado. Siguiendo a la hache hay un número del uno al seis que indica la importancia del titular siendo el más importante el uno y el menos importante el seis. En la figura 2.5 se puede ver un ejemplo de cómo muestra las cabeceras Internet Explorer™.

Figura 2.5. Distintos tipos de encabezados vistos con Microsoft Internet Explorer™.



El código usado en este ejemplo es el siguiente:

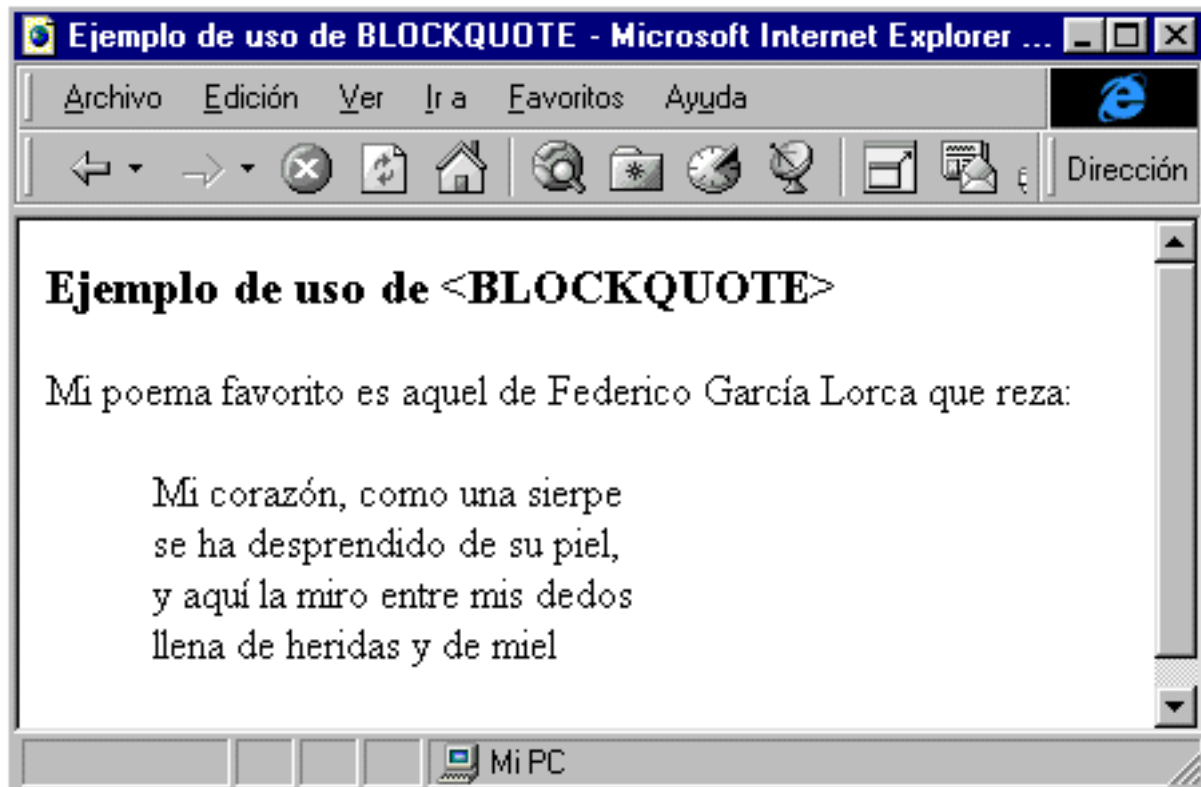
```
<h1>Encabezado 1</h1>  
<h2>Encabezado 2</h2>  
<h3>Encabezado 3</h3>  
<h4>Encabezado 4</h4>  
<h5>Encabezado 5</h5>  
<h6>Encabezado 6</h6>
```

NOTA: Los encabezados empiezan siempre en una línea nueva y el texto que les sigue comenzará en la línea siguiente. Además el navegador podrá mostrar un espacio vertical adicional entre el encabezado y el texto anterior y posterior.

En él observamos que todos los titulares están en negrita y el tamaño de la fuente usada para cada encabezado va decreciendo conforme decrece la importancia del titular. Pero hay que tener en cuenta, tal y como comentábamos en la introducción, que este es sólo un ejemplo y que otros navegadores pueden mostrar estos encabezados de manera diferente.

Por otro lado los navegadores más usados hoy en día son el Internet Explorer™ y el Netscape Navigator™ y es normal prestarles una atención especial (aunque no exclusiva). Y en la figura 2.6 podemos apreciar cómo con este último se muestra los encabezados de una manera prácticamente idéntica. Por esta razón podemos estar casi seguros de que los usuarios verán los encabezados de la forma mostrada en ambas figuras.

Figura 2.6. Distintos tipos de encabezados vistos con Netscape Navigator

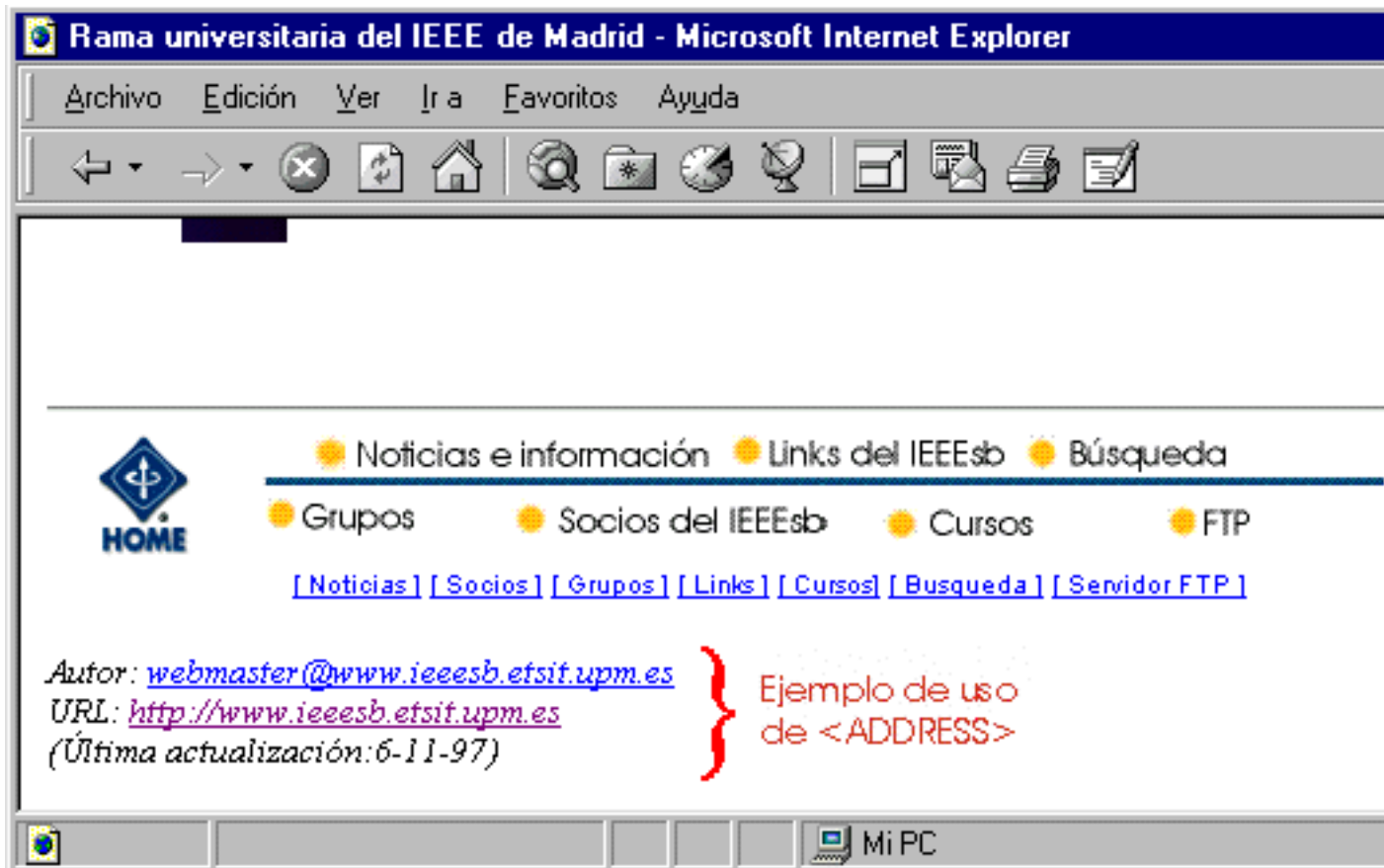


Hay que destacar que los encabezados de h1 a h4 pueden ser usados como títulos ya que los navegadores suelen usar un tamaño de letra mayor que el habitual para mostrarlos. Sin embargo los encabezados h5 y h6 suelen ser mostrados con letra más pequeña aún que el texto y por esta razón se usan como notas a pie de página o comentarios de poca importancia.

El ejemplo mejorado

Ahora que sabemos usar encabezados los utilizaremos para mejorar ligeramente nuestra página. En la figura 2.7 podemos ver cómo se muestran las modificaciones.

Figura 2.7. Ejemplo de página Web con dos tipos de encabezados y dos párrafos.



Esta página ha sido creada con el siguiente código:

```
<html>
<head>
  <title>Mi primera página WEB</title>
</head>
<body>
  <h1>Mi primera página WEB</h1>
  <h2>Bienvenida</h2>
  Hola a todos, como han deducido por el titulo esta es la
  primera pagina web que hago, espero que les guste. Fdo. Jorge
  <h2>Proposito</h2>
```

```
    En esta página iré practicando con los conocimientos que  
    adquiriera en el curso de HTML  
</body>  
</html>
```

Conclusión

Ya sabemos mucho sobre la programación de páginas. Conocemos las principales características del lenguaje HTML y qué es necesario para crear un Web. Hemos aprendido además las bases del lenguaje y hemos creado nuestra primera página de ejemplo, con texto y en cabezados, y la hemos puesto un título.

En la siguiente entrega realizaremos la primera página web completa, con párrafos, imágenes e hiperenlaces. Aprenderemos también a alinear el texto a los dos lados de la página y a combinar esta alineación con la inclusión de imágenes. Igualmente veremos cómo es posible crear enlaces a distintos puntos de una página así como a otras páginas o puntos concretos de estas.

La primera página Web

Jorge Ferrer

Victor García

Versión 1.0

Copyright Jorge Ferrer y Victor García. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation. Puede consultar una copia de la licencia en: <http://www.gnu.org/copyleft/fdl.html>

Tras un primer capítulo en el que hemos aprendido lo sencillo que es crear una página Web con el lenguaje HTML avanzamos en este segundo explicando las etiquetas más usadas a la hora de crear páginas y terminaremos elaborando la nuestra propia.

Hasta ahora hemos creado algunas páginas sencillas en HTML en las que fuimos capaces de introducir encabezados y párrafos sencillos. En este capítulo aprenderemos una serie de etiquetas con las que tendremos conocimientos suficientes para elaborar una página profesional. Los temas que serán cubiertos en este capítulo son: párrafos en HTML, la etiqueta `center`, cómo insertar imágenes y cómo crear enlaces hipertexto.

Párrafos en HTML

Ha llegado el momento de aprender a introducir un texto en una página web, pero antes es necesario comprender cómo será interpretado este texto. Cuando un navegador lee el código HTML de una página y lo interpreta para mostrar la página, todos los espaciados múltiples, tabulados o saltos de línea que se haya incluido en los textos serán interpretados como un único espacio. Esta característica suele sorprender (e incluso molestar) al principio, pero es innegable que esta forma de actuar es necesaria y concuerda con el carácter descriptivo de HTML.

La mejor manera de habituarse es probar, para ello proponemos los siguientes ejemplos de código que serán mostrados de la misma forma por el navegador aunque algunos contengan saltos de línea, otros tabulados entre las palabras, etc.

```
<h1>El conductor del coche le pregunto si se encontraba mal</h1>
```

```
<h1>
El conductor del coche le preguntó si se encontraba mal
</h1>
```

```
<h1>
El   conductor   del coche
le pregunto si se encontraba mal </h1>
```

En estos ejemplos podemos observar diversos saltos de línea y separaciones en el texto, sin embargo el navegador mostrará un solo espacio entre las palabras y producirá saltos de línea donde sea necesario en función del tamaño de la ventana. Las figuras 2.1 y 2.2 son ejemplos de cómo muestra el Explorer™ los códigos anteriores para diferentes tamaños de ventana.

Figura 2.1. Uso básico de una etiqueta HTML sin ningún tipo de parámetros. Ejemplo usando la etiqueta <hr> para crear una línea horizontal.



Conocida esta característica es momento de adentrarse en las distintas etiquetas que permiten tener un mayor control sobre el texto de nuestras páginas.

Saltos de línea

En ocasiones puede ser conveniente introducir un punto y aparte cuando estamos escribiendo un texto. En HTML para hacer esto puede usarse la etiqueta
. Cuando el navegador se encuentra con esta etiqueta provoca un salto de línea y sigue mostrando el contenido de la página. Veamos un ejemplo:

```
Estaba internándose en lo desconocido. <br>
Durante todo el trayecto hasta Woodfield no dejó de pensar en los
aspectos económicos, con el temor de estar cometiendo...
```

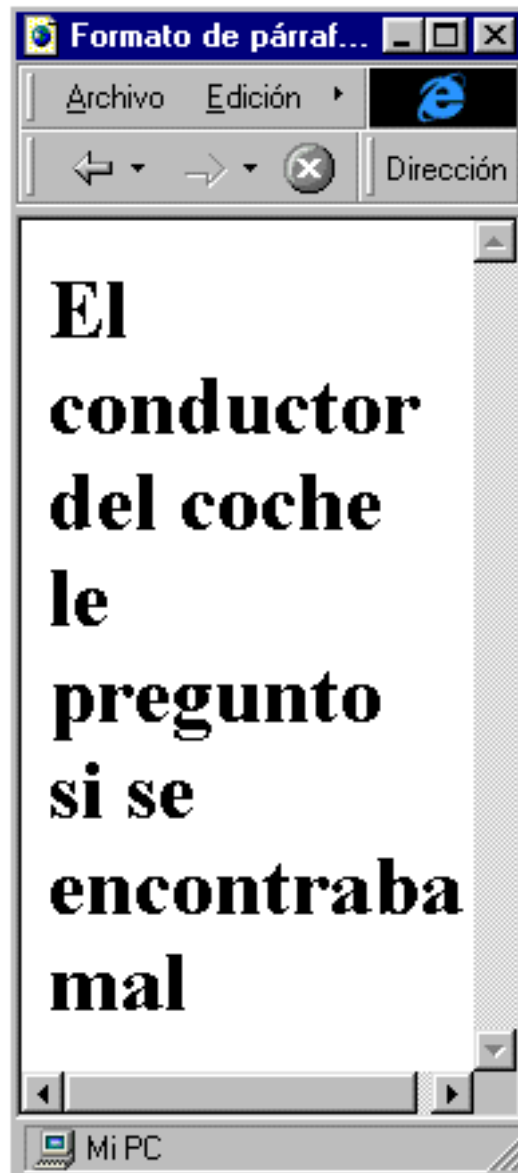
En este texto podemos observar cómo tras el primer punto era necesario un cambio de línea, por lo que hemos usado
.

Cómo se insertan párrafos

Un párrafo es un conjunto de frases sobre un mismo asunto. En HTML para demarcar un párrafo se usa la etiqueta p, situándose la instrucción de inicio <p> al comienzo del párrafo y la instrucción de fin </p> tras la última frase. Entre ellas pueden insertarse tantos saltos de línea como se deseen así como muchos otros elementos HTML. A continuación se muestra un ejemplo:

```
<p>
Estaba internándose en lo desconocido.
</p>
<p>
Durante todo el trayecto hasta
Woodfield no dejó de pensar en los aspectos económicos, con el
```

Figura 2.2. Las etiquetas tienen atributos para cambiar aspectos del efecto que producen. En este ejemplo vemos como se puede cambiar el grosor de una línea con el atributo 'size'



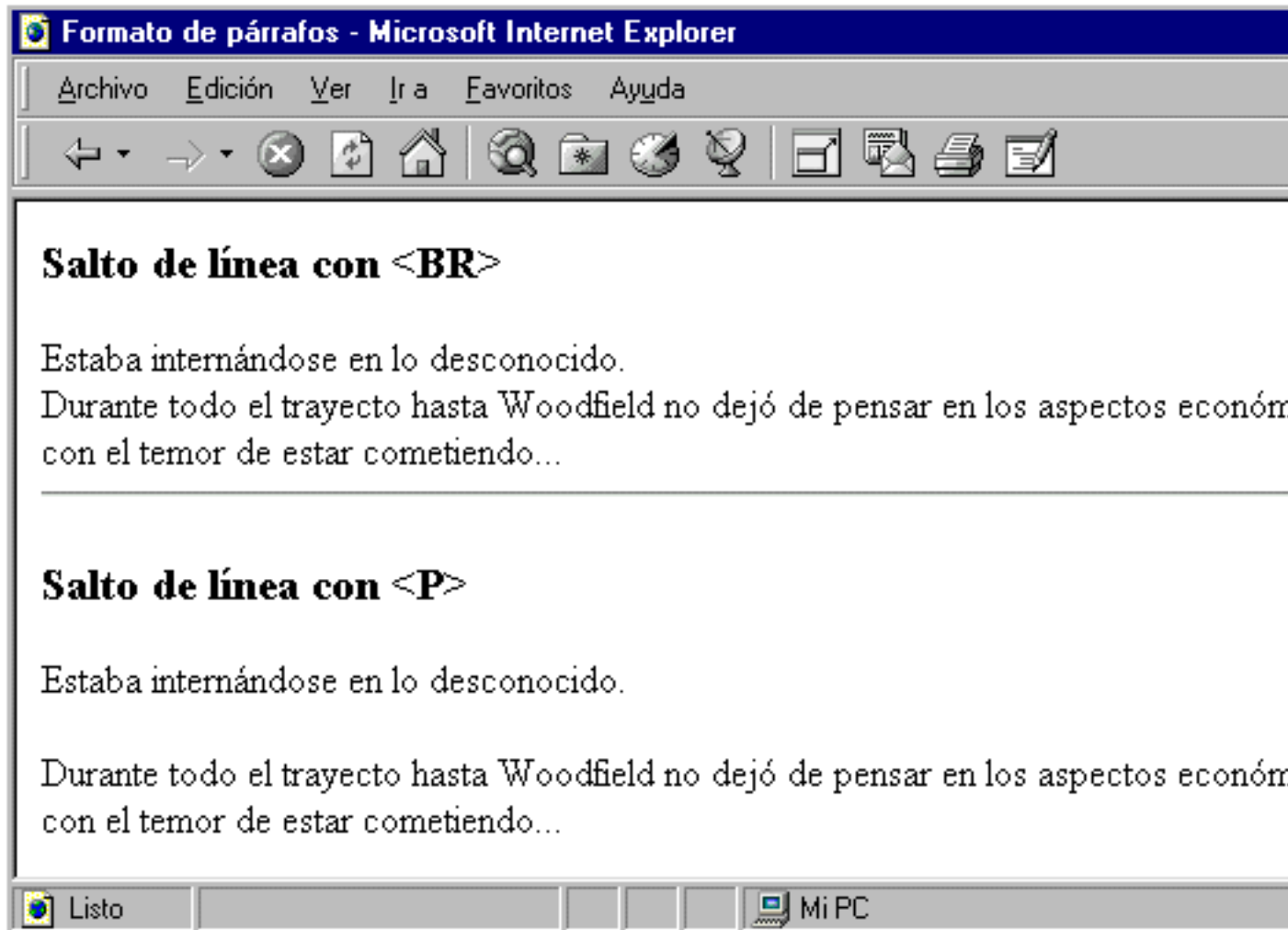
```
temor de estar cometiendo...  
</p>
```

Otra opción hubiera sido crear un único párrafo e insertar un salto de línea tras la primera frase:

```
<p>  
Estaba internándose en lo desconocido.  
<br>  
Durante todo el trayecto hasta  
Woodfield no dejó de pensar en los aspectos económicos, con el  
temor de estar cometiendo...  
</p>
```

En la figura 2.3 se puede observar la diferencia entre los dos ejemplos en Explorer™. Puede observarse que en general los navegadores dejan un espacio vertical antes y después de cada párrafo, algo que no ocurre con los saltos de línea.

Figura 2.3. Ejemplo de un párrafo con un salto de línea y la división del mismo texto en dos párrafos.



Uso incorrecto de la etiqueta p

La etiqueta `p` es usada en muchas páginas de forma incorrecta. El error consiste en emplearla simplemente para introducir un espacio (vertical) entre líneas. La forma adecuada es introducir las instrucciones de inicio y fin para marcar el comienzo y el fin de un párrafo.

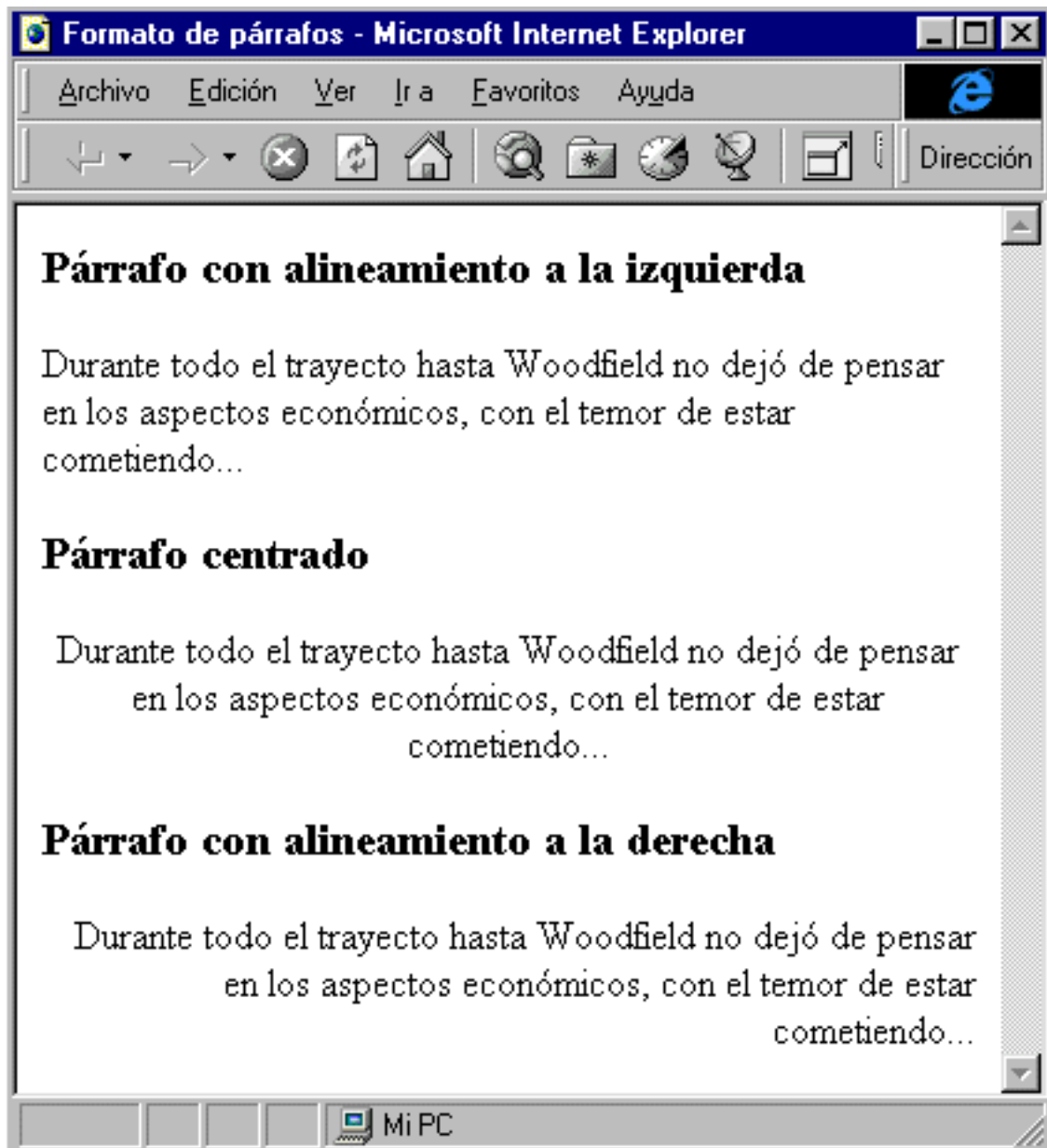
Espacio entre líneas

Para introducir un espacio vertical entre líneas debe usarse la etiqueta `
`. Si se ponen varias seguidas no se mostrará más espacio, pero esto puede solucionarse insertando entre las etiquetas la cadena " ". Por ejemplo para insertar tres líneas en blanco puede usarse:

```
<br> &nbsp;
<br> &nbsp;
<br> &nbsp;
```

El atributo 'align' en la etiqueta `<p>` permite especificar la alineación del párrafo. Puede tener tres valores: "center", "left" y "right", para obtener un párrafo centrado, alineado a la izquierda o alineado a la derecha respectivamente. En la figura 2.4 se muestra el párrafo del código anterior con los tres tipos de alineación.

Figura 2.4. Con el atributo 'align' se pueden crear párrafos alineados a la derecha, centrados o alineados a la izquierda.



Líneas horizontales para separar párrafos

Si lo que se necesita es separar dos secciones diferentes de una página, ya sea texto o no, puede emplearse una nueva etiqueta: `<hr>`. Su nombre proviene del inglés *Horizontal Rule* (línea horizontal). Esta etiqueta está formada por una única instrucción y provoca que el navegador inserte una línea horizontal como la mostrada en la figura 2.3 que separaba los dos párrafos que se estaban comparando. Existen cuatro atributos que permiten modificar su apariencia. Estos son:

-

'align': Permite cambiar la alineación de la línea. Puede tomar tres valores: "*left*", "*center*" y "*right*" para conseguir alineación a la izquierda, centro y derecha respectivamente.

•

'noshade': 'noshade', que en inglés significa *sin sombra*, existe porque algunos navegadores gráficos como Navigator™ o Explorer™ muestran las líneas horizontales con un efecto tridimensional con sombra. Al incluir este atributo se elimina este efecto de manera que el navegador sólo mostrará una línea horizontal negra.

•

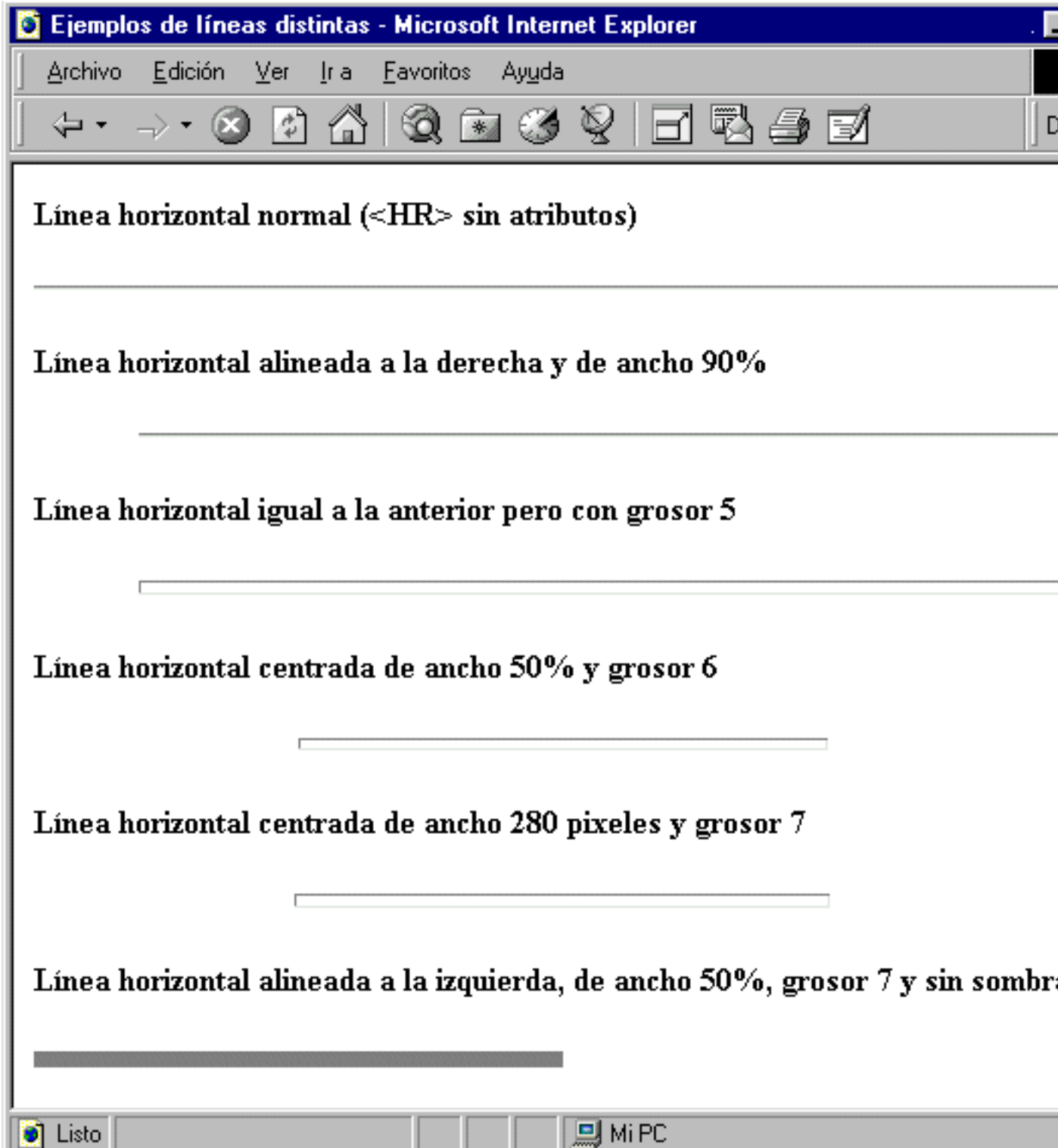
'size': Con este atributo se puede controlar el grosor de la línea. Como valor de este atributo se debe especificar el valor deseado en píxeles (puntos de pantalla).

•

'width': Este atributo permite especificar el ancho de la línea. El valor puede ser dado de una manera absoluta, especificando el ancho en píxeles (por ejemplo width=100) o de manera relativa respecto al ancho de la ventana del navegador usando porcentajes (por ejemplo width="90%"). Es más que recomendable usar este último sistema de manera que obtengamos un resultado apropiado independientemente del tamaño que cada usuario quiera dar a la ventana de su navegador.

Estos atributos pueden ser usados conjuntamente para combinar sus efectos. En la figura 2.5 podemos ver el efecto de cada uno por separado en Explorer™ junto con los valores empleados en cada caso.

Figura 2.5. Con los nuevos atributos de la etiqueta `<hr>` podemos controlar el ancho, grosor alineamiento y sombra de la línea horizontal.



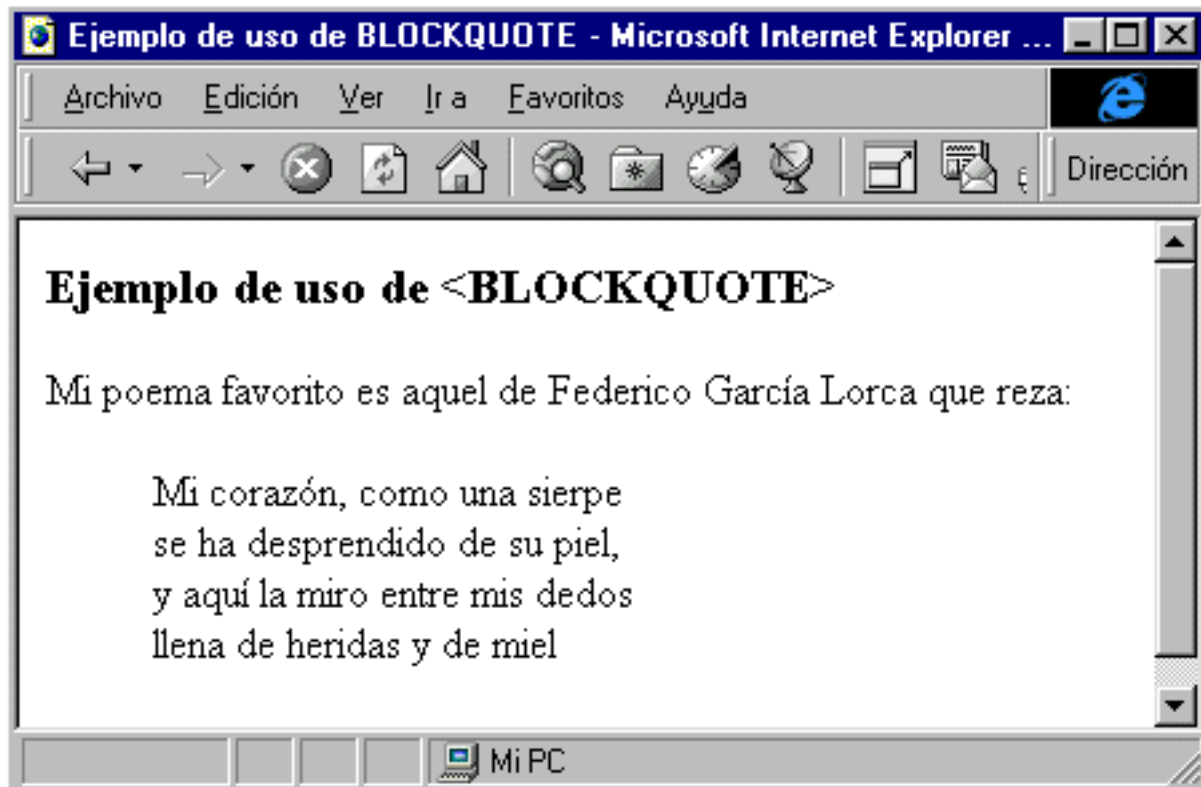
La etiqueta `<blockquote>... </blockquote>`

Otra etiqueta relacionada con el formato de párrafos en HTML es `<blockquote>`, que consta de una instrucción de inicio y una de fin y que indica que el párrafo que encierra es un texto parafraseado. Al usar esta etiqueta el navegador mostrará el texto en un nuevo párrafo con márgenes a la izquierda y a la derecha. En la figura 2.6 puede verse un uso típico de esta etiqueta.

No abusar de las líneas horizontales

Es conveniente no usar mucho las líneas horizontales, limitándolas al comienzo y fin de la página. La razón de esto es que las líneas en medio de la página dificultan la lectura de esta. Si se desea separar secciones un espacio vertical del tamaño adecuado es más efectivo.

Figura 2.6. La etiqueta `<blockquote>` es usada para insertar bloques de texto que citan a otro autor. Los navegadores suelen introducir un sangrado en este texto.



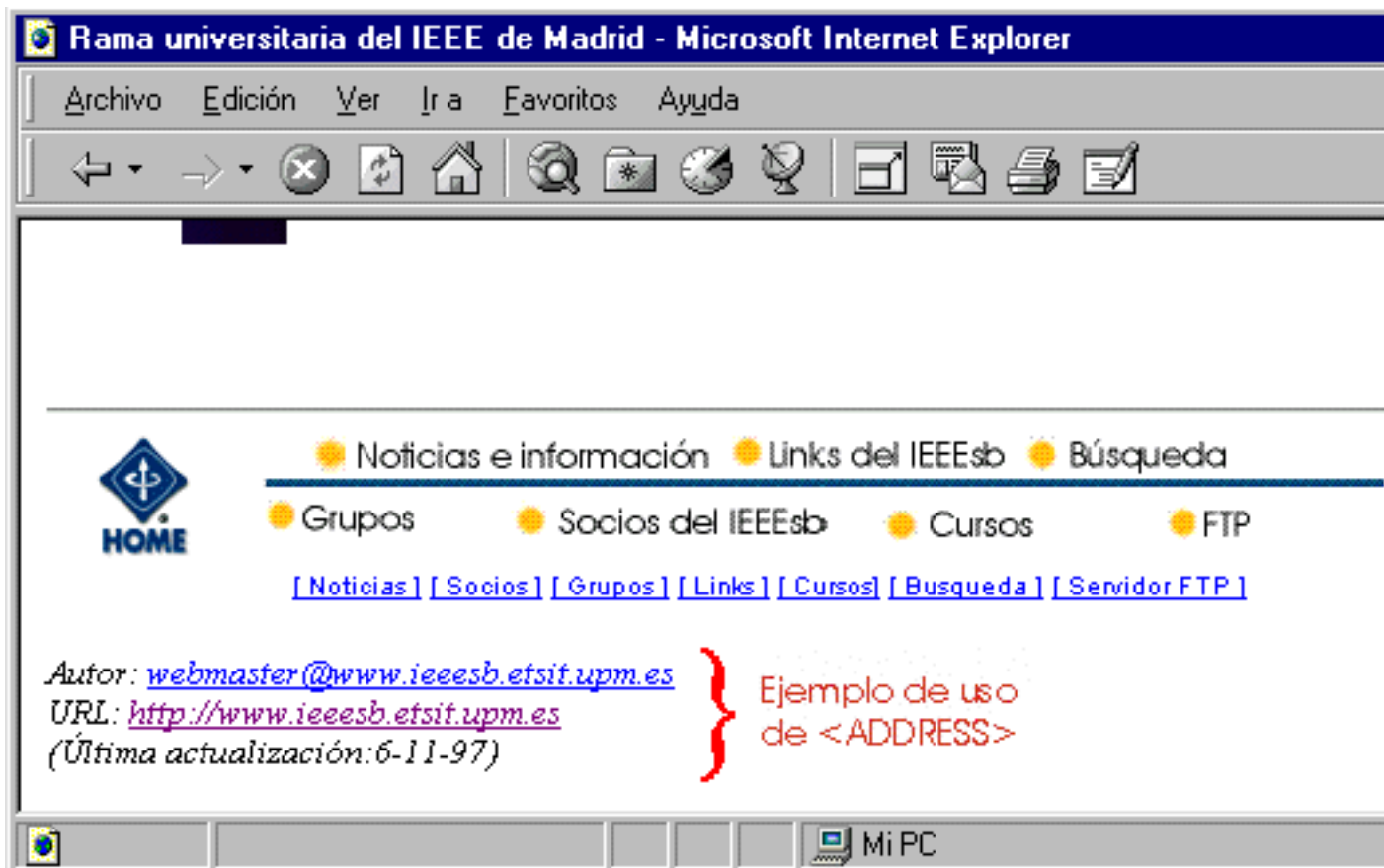
Que se corresponde con el código:

```
Mi poema favorito es aquel de Federico García Lorca que reza:  
<blockquote>  
Mi corazón, como una sierpe<br>  
se ha desprendido de su piel,<br>  
y aquí la miro entre mis dedos<br>  
llena de heridas y de miel<br>  
</blockquote>
```

La etiqueta `<address>... </address>`

Muy relacionada con la anterior, es la etiqueta `<address>`, que en inglés significa dirección. Esta etiqueta está compuesta, igualmente, por dos instrucciones y se usa para incluir información sobre el autor de una página WEB o elementos relacionados. Puede encontrarse un ejemplo de su uso en las coletillas existentes al final de muchas páginas en las que se incluye información variada para el usuario como puede ser, además del autor, la fecha de la última modificación, la fecha actual, la URL de la página, etc. como se muestra en la figura 2.7.

Figura 2.7. En esta página se observa como al final de la página se han incluido datos de los autores y de la página. Para ello se ha usado la etiqueta `<address>`.



La etiqueta `<center>... </center>`

Acabamos de ver una serie de etiquetas que permiten cambiar de línea, crear distintos tipos de párrafos o insertar líneas que les separen. También hemos visto que alguna de las etiquetas utilizadas tienen el atributo 'align' que permite indicar el alineamiento.

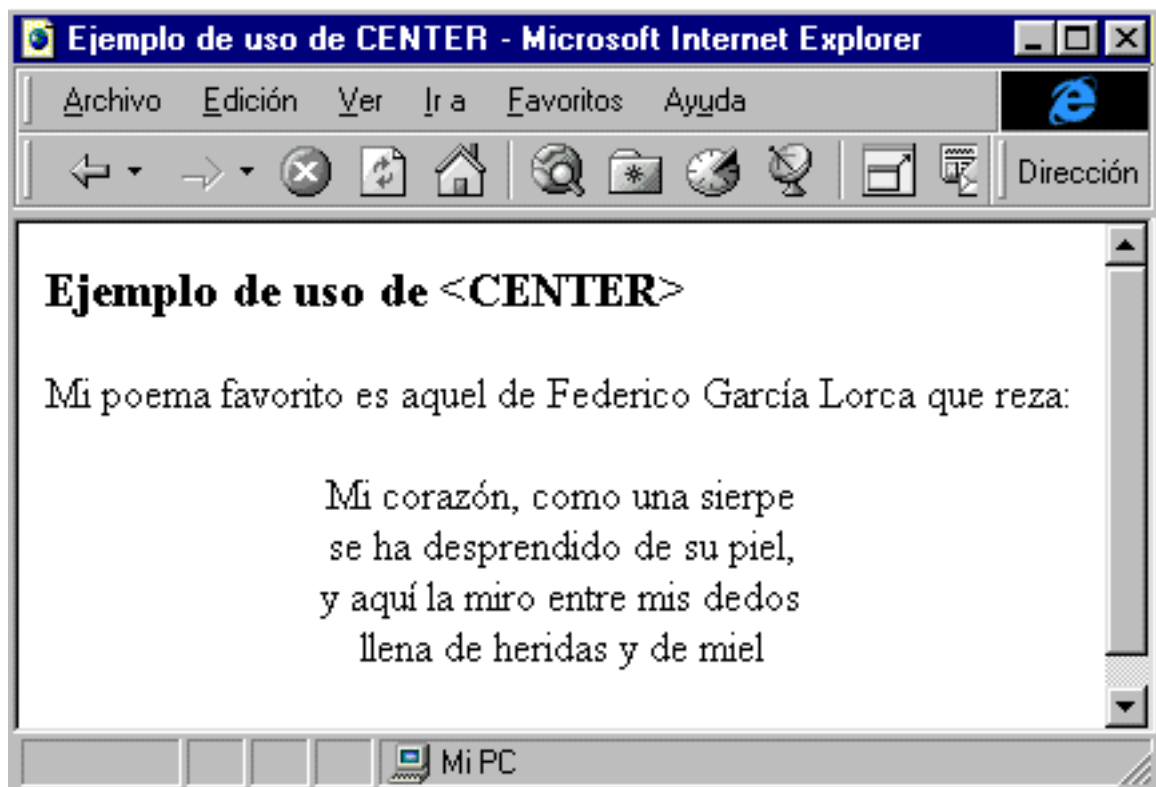
En ocasiones puede ser muy útil indicar que algún texto, imagen o algún otro elemento de la página debía ser centrado. Sin embargo no todas las etiquetas tienen el atributo 'align'. La solución es emplear una nueva etiqueta: `center`. Ésta está compuesta por una instrucción de inicio y una de fin de manera que todo lo que quedaba entre ellas, fuese lo que fuese, estuviese centrado. Por ejemplo en el ejemplo del verso de Lorca podemos centrarlo escribiendo:

```
Mi poema favorito es aquel de Federico García Lorca que reza:  
<center>
```

```
<blockquote>
Mi corazón, como una sierpe <br>
se ha desprendido de su piel,<br>
y aquí la miro entre mis dedos<br>
llena de heridas y de miel <br>
</blockquote>
</center>
```

Y el resultado obtenido hubiese sido el de la figura 2.8

Figura 2.8. Con la etiqueta `<center>` es posible centrar cualquier elemento de nuestra página como por ejemplo el verso de Lorca de la figura 2.6.



Nota

Actualmente existen alternativas mejores que la etiqueta `center` para especificar el alineamiento. En particular la etiqueta `div` y las hojas de estilo en cascada que serán tratadas a lo largo de este curso.

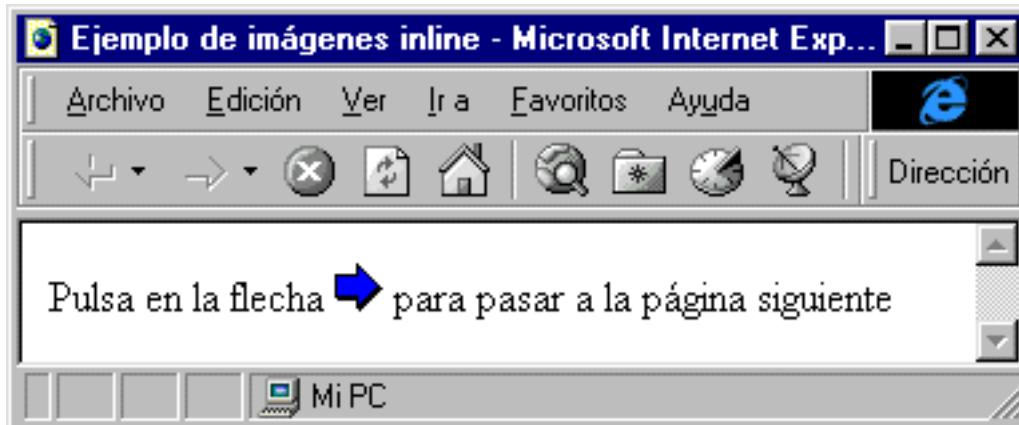
Cómo insertar imágenes en una página.

En el cuarto capítulo de la obra trataremos con detalle el mundo de las imágenes en el WWW y aprenderemos todas las posibilidades gráficas que ofrece. En este capítulo aprenderemos, como introducción, a insertar imágenes para dar mas vida a nuestra página.

La etiqueta usada para insertar imágenes es: ``. Esta etiqueta está compuesta por una única instrucción y por tanto `` no existe y su uso está prohibido. Podemos insertar una imagen en medio de una frase como si

fuese una palabra más de ella y será mostrada por el navegador a continuación del texto (o cualquier otro elemento anterior) como podemos ver en el ejemplo de la figura 2.9.

Figura 2.9. Al insertar una imagen esta se entremezcla con el texto. Cuando la imagen es pequeña se crean interesantes efectos como este.



En ella se puede observar como hemos insertado una flecha en medio de la frase y después de la imagen hemos seguido escribiendo. Aunque podemos crear este efecto con imágenes de todos los tamaños no es recomendable hacerlo con aquellas que tengan una altura mucho mayor que la del texto.

Para insertar una imagen en una página WEB necesitamos tenerla en un archivo aparte. Existen multitud de formatos para almacenar una imagen en un archivo: BMP, GIF, JPEG, XPM, XBM, PNG, TIFF, etc. cada uno de los cuales tienen unas características. El formato elegido en el World Wide Web es el formato GIF que distingue a sus archivos porque tienen la terminación ".gif". Podríamos decir que todos los navegadores son capaces de mostrar las imágenes incluidas en este formato. Los navegadores en modo texto suelen proporcionar métodos para ver las imágenes usando algún programa visor externo. Un segundo formato que se ha introducido en el WWW y que es soportado por la gran mayoría de los navegadores (incluyendo al Explorer™ y al Navigator™) es el JPEG. Este formato es especialmente útil para las fotos ya que es capaz de comprimir este tipo de imagen de manera que ocupen hasta 4 veces menos que las imágenes con formato GIF. Los archivos que contienen imágenes con formato JPEG suelen tener la terminación ".jpg".

Una vez tenemos la imagen en un archivo aparte es hora de escribir el código HTML que nos permitirá insertarla en la página. Como hemos comentado usamos la etiqueta y para indicar el nombre del archivo usamos el atributo 'src'. En el ejemplo de la figura 2.9 el código usado para insertar la flecha es:

```
Pulsa en la flecha  para pasar a la  
página siguiente.
```

Cómo crear enlaces hipertexto.

Una vez visto como introducir distintos tipos de texto y párrafos así como insertar imágenes en una página, lo único que queda por aprender para ser capaces de crear una página Web bastante completa son los enlaces hipertexto. Con ellos se pueden activar frases o palabras de la página para que al pulsar sobre ellas se salte a cualquier otra página de Internet que decidamos.

Estos enlaces hipertexto (en inglés *links*) no son más que unas zonas (habitualmente palabras o frases) *especiales* de nuestro texto que ofrecen la posibilidad de pulsar sobre ellas para ir a otras páginas. El concepto del hipertexto

no es nuevo, la mayoría de los lectores lo habrán usado ya en la ayuda de Windows™ o en otros muchos sistemas, lo realmente novedoso que ofrecen estos enlaces en el WWW es la posibilidad de pulsar y navegar hasta páginas que se encuentren en el otro extremo de la tierra. El navegante puede moverse de España a Japón con una única pulsación sobre el texto adecuado. En seguida se verá lo sencillo que es incluir esta impresionante tecnología en una página web.

Los enlaces pueden clasificarse en dos tipos, los internos o locales y los externos:

1. Los enlaces internos son aquellos que enlazan las páginas que componen un mismo sitio web (*web site*). Todas estas páginas estarán (generalmente) en el mismo servidor WWW, y por tanto estarán también en el mismo ordenador. Por esta razón sólo será necesario indicar en nuestro código el nombre del archivo donde está la página que queremos enlazar y el directorio en el que se encuentra. Un ejemplo dirección de un enlace interno es:

```
/hobbies/index.html
```

2. Los enlaces externos permiten saltar desde una página hasta otra puede estar en cualquier otro lugar del mundo en otro servidor web. Al crear estos enlaces es necesario especificar la dirección completa de Internet (la URL) de la página que queremos enlazar. Por ejemplo:

```
http://www.otroservidor.com/hobbies/index.html
```

Por último, cabe resaltar que aunque la mayoría de las veces los links se usan para enlazar una páginas con otras también pueden usarse con imágenes, sonido, vídeo y prácticamente cualquier tipo de fichero.

La etiqueta `<a>... `

Esta etiqueta, de nombre sencillo, es la que va a permitir incluir en las páginas enlaces hipertexto. Su atributo 'href' permite indicar cual es la página a la que debe saltar el usuario al pulsar sobre el texto adecuado. El texto del enlace será aquel que se introduzca entre la instrucción de inicio y la instrucción de final de la etiqueta y debe ser resaltado por el navegador para que el usuario sepa que puede pulsar en él. La estructura de un link es:

```
<a href="dirección_URL">  
Texto que será sensible (hipertexto)  
</a>
```

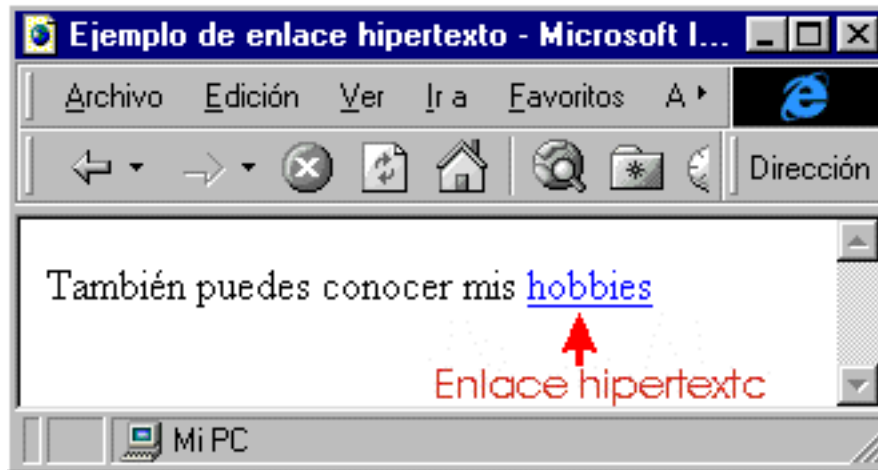
Los navegadores gráficos como Netscape Navigator™ y Microsoft Internet Explorer™ resaltan este texto mostrándolo de un color diferente y subrayándolo, además el cursor cambiará al situarlo sobre el texto. Los navegadores que funcionan en modo texto como Lynx™ resaltan los enlaces hipertexto cambiando el color del texto y de su fondo.

Veamos un ejemplo sencillo del uso de enlaces. Se ha creado un sitio web personal con 2 páginas. La página principal se encuentra en el archivo `index.html` y se quiere que en ella haya un enlace a otra página donde se han escrito sobre los hobbies del creador del sitio web y que se encuentra en el archivo `hobbies.html`. El enlace podría ser algo así:

```
También puedes conocer mis <a  
href="hobbies.html">hobbies</a>
```

El resultado es el que puede apreciarse en la figura 2.10 donde la palabra hobbies aparece resaltada.

Figura 2.10. Al pulsar sobre un enlace hipertexto *saltaremos* a otra página de Internet. Los navegadores resaltan estos enlaces para que sea fácil identificarlos.



Aviso

Si no funciona un link conviene comprobar la sintaxis del código HTML. Un error común es no cerrar las comillas que tras poner la dirección de la página enlazada. Si se comete este error, además de no funcionar el enlace, es posible que desaparezcan otros elementos de la página al visualizarla.

Páginas en otros directorios

Si la página a la que queremos crear el enlace no se encuentra en el mismo directorio que la actual es necesario indicar el directorio (o ruta de directorios, path) donde está. Si en el ejemplo anterior la página `hobbies.html` estuviese en el directorio *varios* el ejemplo anterior debería sustituirse por:

```
También puedes conocer mis
<a href="varios/hobbies.html">hobbies</a>
```

Como vemos, para separar el nombre del directorio del nombre del archivo se usa el símbolo `"/`. También se usaría, si fuese necesario, para separar varios directorios.

Enlaces externos

La verdadera fuerza de Internet estriba en la posibilidad de navegar con una sola pulsación por información que se encuentre en cualquier lugar del mundo. Para convertir esto en realidad existen los enlaces externos.

Nota

Algunos diseñadores web recomiendan que no se creen enlaces externos desde la página principal o *home page* de nuestro sitio Web para no incitar al visitante a que se vaya de la página.

La creación de uno de estos enlaces es muy similar a la creación de los enlaces locales. El único cambio radica en que el valor dado al atributo `'href'` es la dirección completa de Internet, sea otra página web, una imagen o

cualquier otro archivo. Por ejemplo, para incluir un enlace desde cualquier página a la de DMOZ (*Open Directory Project*). Para ello puede emplearse:

```
Enlace a <a href="http://www.dmoz.org">Directorio DMOZ</a>
```

Así de fácil. Para referirse a un archivo específico del servidor de DMOZ™ hay que especificar la ruta de directorios y el nombre del archivo. Un ejemplo de uso podría consistir en construir un enlace a la sección sobre Internet de este buscador. Para ello hay que incluir el siguiente link:

```
<a  
href="http://www.dmoz.org/Computers/Internet/">  
Sección de Internet de DMOZ </a>
```

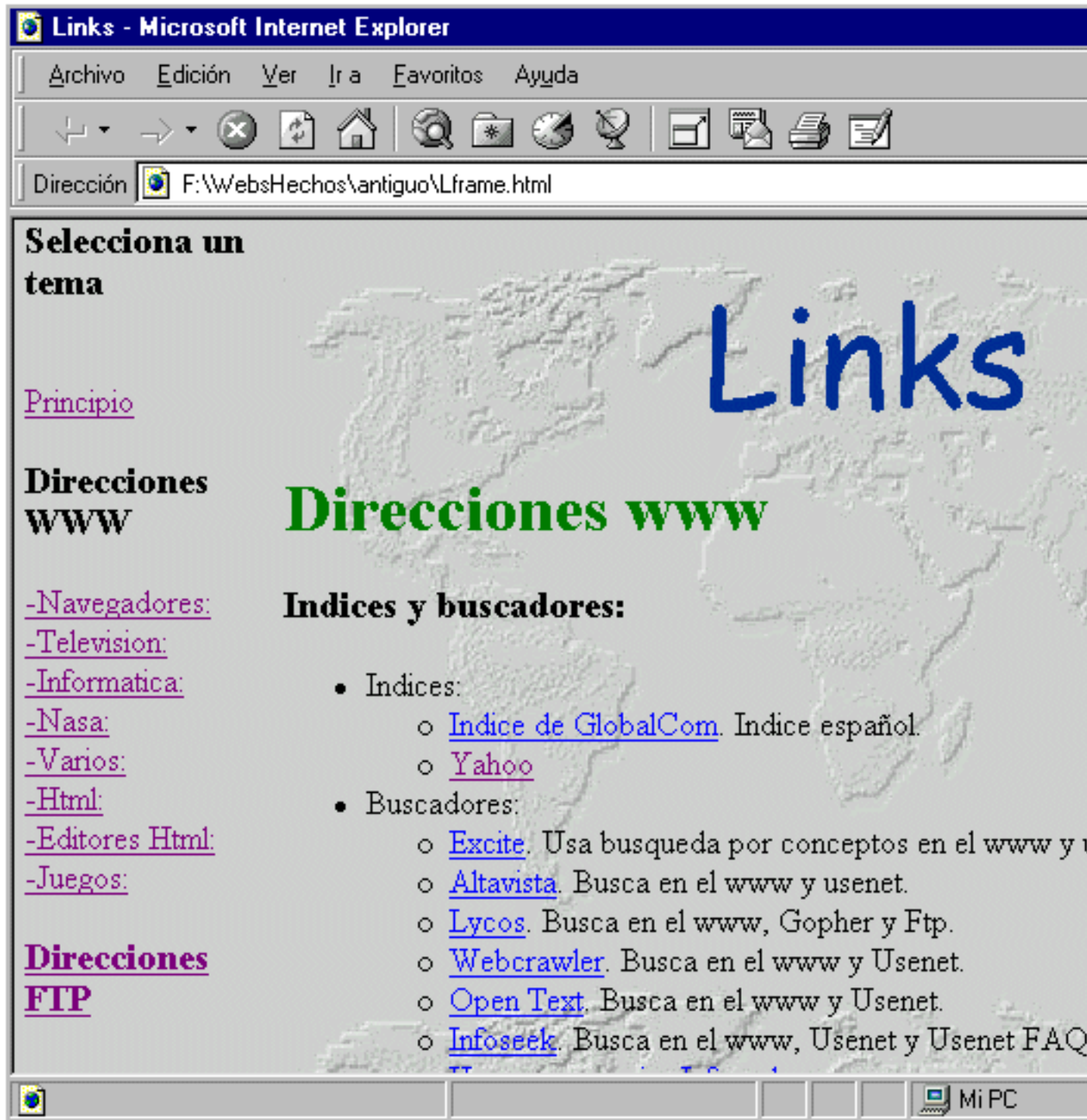
Un aspecto destacable de este ejemplo es que se ha especificado la ruta de directorios pero no el nombre del archivo. Con ello se consigue acceder al archivo por defecto de ese directorio. Todos los servidores web tienen un *archivo* por defecto de manera que si no se introduce el nombre ninguno ese archivo será el mostrado. En la mayoría de servidores el nombre elegido es `index.html` por lo que la dirección usada antes sería equivalente a:

```
http://www.yahoo.com/computers/internet/index.html
```

Nota

Si un sitio Web consta de varias páginas es usual incluir todos los enlaces externos en una página especial dedicada a ellos como la que se muestra en la figura 2.11.

Figura 2.11. Página que agrupa todos los enlaces de un sitio web



PRÁCTICA 1: La página web de "La chistera"

La propuesta

En esta primera práctica de la obra se propone crear un sitio Web REAL poniendo en práctica los conocimientos adquiridos hasta ahora. En él incluiremos imágenes, varios titulares párrafos de texto de distinto tipo y enlaces hipertexto a otras páginas.

En la figura 2.12 se muestra una captura de un navegador visitando la página tal y como debe quedar al final. Se propone al lector intentar imitar la apariencia de esta página usando únicamente las etiquetas HTML explicadas hasta ahora.

Figura 2.12. La práctica 1 propone realizar un página a partir de lo mostrado en esta figura.



LA CHISTERA - Humor

Quienes somos

La chistera es un grupo de artistas dedicados al mundo de la magia, la comedia y el encontrar humoristas, magos, showmen, etc.

Si usted o su empresa necesita crear un espectáculo y no quiere quedar decepcionado MISMO. Nuestra dirección es:

chistera@chistera.es

Un ejemplo de nuestro repertorio

Cada día le mostramos ejemplos distintos de nuestro amplísimo y novedoso repertorio

Era un tío tan gafe, tan gafe, que se sentó en un pajar y se clavó la aguja.

Doctor, Doctor, ¿cuanto tiempo me queda de vida?

5... 4... 3... 2...

Enlaces a páginas relacionadas

[¡A Reírse Toca, ¡Ja, ja, ja!](#): Si te aburres, estás deprimido, o simplemente quieres reírte de los más de 110 chistes que tienes, ¡Con posibilidad de enviar tú el tuyo!

[1er Concurso de feos por Internet](#): 1er Concurso de Feos por Internet participa y gana

Para empezar a realizar la página debe abrirse un editor de textos o de HTML (es indiferente). Algunos ejemplos de programas que pueden servir son notepad™, emacs™, vi™, HomeSite™, etc. Una vez abierto se puede empezar a escribir la página tal y como se ha explicado, comenzando por la cabecera y siguiendo con el resto de la página. A continuación se muestran las primeras líneas que podría contener el archivo a modo de ejemplo:

```
<html>
<head>
<title>
La chistera - Humor y magia
</title>
</head>
<body>
```

Tras escribir el código debe guardarse con terminación “.html” y ya está lista para ser visualizada con cualquier navegador. Tras abrir éste hay que seleccionar la opción Archivo/Abrir...Examinar... (o similar) y elegir el archivo en el que habíamos guardado la página. En ese momento debe mostrarse en pantalla la página creada. Lo habitual es tener que realizar cambios en el código HTML, volver al navegador, volver a hacer cambios y así hasta que se obtiene el resultado deseado. ¡suerte!

La solución

En esta primera práctica se va a mostrar a continuación el código completo de la página resultado. Por supuesto, conviene intentarlo antes de mirarlo. También hay que tener en cuenta que no hay un único resultado y es posible que un código ligeramente diferente de lugar a una página prácticamente idéntica al verla con el navegador.

```
<html>

<head>
<title>
La chistera - Humor y magia
</title>
</head>

<body>
<center>
<h1>
    
    LA CHISTERA - Humor y magia
</h1>
</center>
<h2>
Quienes somos
</h2>
<p>La chistera es un grupo de artistas
dedicados al mundo de la magia, la comedia y el espectáculo.
Entre nosotros podrás encontrar humoristas, magos, showmen,
etc.</p>
<p>Si usted o su empresa necesita crear
un espectáculo y no quiere quedar decepcionado no dude en
escribirnos AHORA MISMO. Nuestra dirección es:</p>
<center>
chistera@chistera.es
</center>
<h2>
Un ejemplo de nuestro repertorio
```



```
</h2>
<p>Cada día le mostramos ejemplos
distintos de nuestro amplísimo y novedoso repertorio de chistes:
<blockquote>
Era un tío tan gafe, tan gafe, que se sentó en un pajar y se
clavó la aguja.
</blockquote>
<blockquote>
Doctor, Doctor, ¿cuanto tiempo me queda de vida?
<br>
5... 4... 3... 2...
</blockquote>
</p>
<h2>
Enlaces a páginas relacionadas
</h2>
<p>
<a href="http://www.ciudadfutura.com/areirse
">¡A Reírse Toca, ¡Ja, ja, ja!
</a>: Si te aburres, estás deprimido,
o simplemente quieres reír, pásate por mi página y disfrutarás
de los más de 110 chistes que tiene. ¡Con posibilidad de enviar
tú el tuyo!
<p>
<a
 href="http://www.geocities.com/BourbonStreet/Delta/3543/"
>1er Concurso de feos por
Internet</a>: 1er Concurso de Feos por
Internet participa y gana tu premio. </p>
<p>
<a
 href="http://www.geocities.com/Hollywood/8620">Albert
Grau's Web</a>
: En esta pagina encontrarás un listado de dichos
populares, y las 50 cosas que se deberían cambiar para mejorar
la vida terrícola de nuestro querido planeta Tierra
</p>
<hr size="5">
<address>
Página creada por el equipo de programación Web de LA CHISTERA
<br> Última actualización: 32-12-1997 ;- )
</address>
</body>
</html>
```

Nota

Una vez hemos comprobado que con nuestro navegador habitual la página es mostrada correctamente conviene probarla con otros incluyendo alguno en modo texto (Lynx) y diferentes tamaños de pantalla. A veces las diferencias serán sorprendentes.

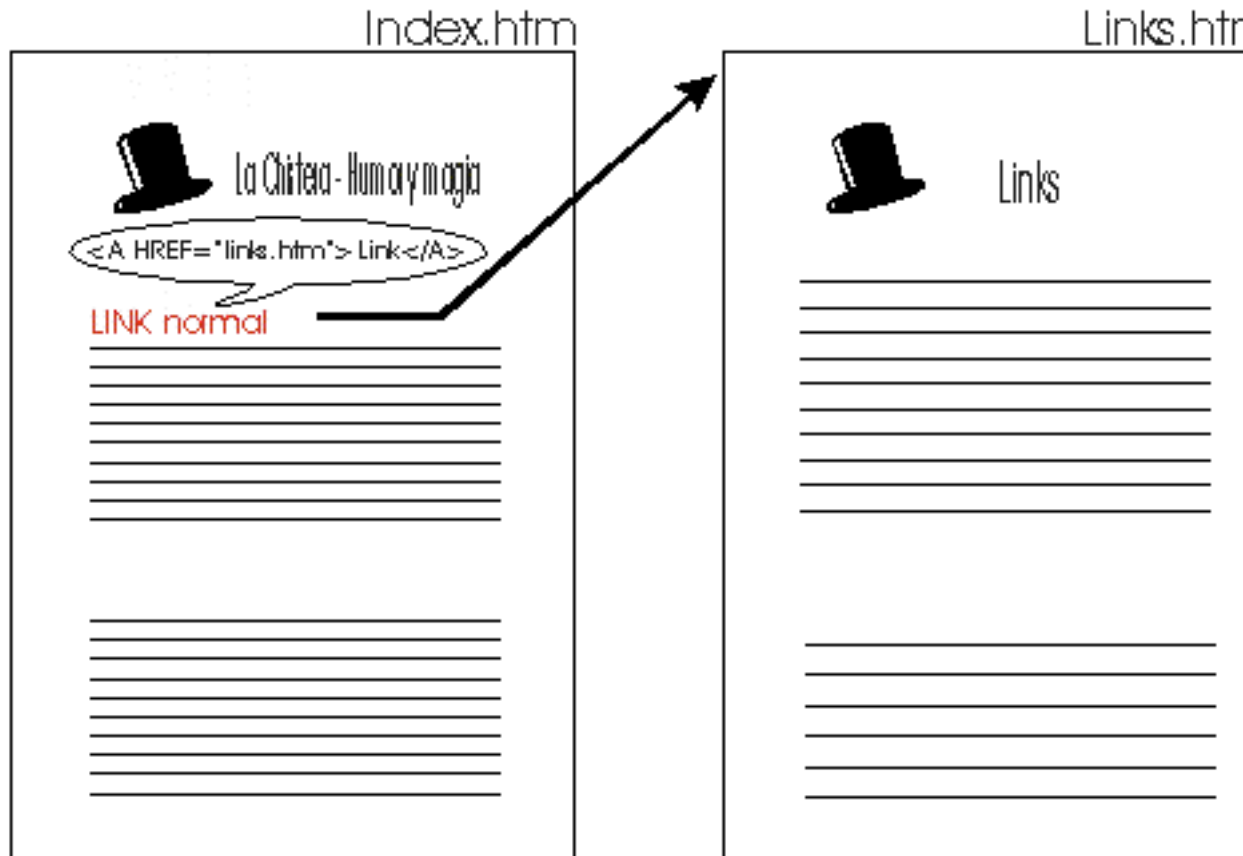
Anclas

Después de realizar nuestra primera práctica continuamos los enlaces hipertexto con unos elementos muy relacionados con ellos: las anclas, que nos permitirán crear enlaces especificando a que lugar concreto de una página queremos saltar.

¿Qué es un ancla?

Un enlace normal tiene la estructura que podemos ver en la figura 2.13.

Figura 2.13. Estructura de los enlaces normales. Al pulsar sobre el texto activo no podemos elegir el lugar de la página al que saltamos, iremos siempre al comienzo.



Desde una página cualquiera y en cualquier lugar de ella se escribe el código para crear un enlace. Cuando se pulsa sobre el texto activo se salta al comienzo de otra página, tal y como indica la flecha. Usando enlaces a anclas se puede cambiar este comportamiento.

Un ancla es una marca que se sitúa en un punto determinado de una página. Posteriormente se podrá crear un enlace a ese ancla de manera que al pulsar sobre el texto apropiado en vez de *saltar* a otra página *se salta* al lugar donde esté el ancla, es decir, ya no se va al comienzo de la página como con los enlaces vistos hasta ahora. Dicho de otra forma usando anclas es posible que un enlace apunte al lugar concreto de una página que se desee.

Creando anclas

Para crear anclas usamos, al igual que para crear enlaces, la etiqueta `<a>`, sin embargo no debe confundirse el lector porque aunque ambos elementos estén relacionados y se use la misma etiqueta para crearlos las funciones

de cada uno son radicalmente diferentes. Las anclas sólo son marcas y son invisibles cuando visualizamos la página con el navegador. Por esta razón las anclas no definen ningún texto activo.

A pesar de ello es habitual encontrar texto entre las instrucciones de inicio y de fin como en el siguiente ejemplo:

```
<a name="nombreAncla">Comienzo</a>
```

Este texto no tiene ningún significado especial y será interpretado por los navegadores de igual forma que el texto de alrededor.

Como se ha adelantado en el ejemplo anterior, al crear anclas no se emplea el atributo 'href'. En su lugar se usa otro llamado 'name'. Con este atributo se da un nombre al ancla que luego nos servirá para referirse a ella.

Un ejemplo donde la utilidad de las anclas se hace evidente es en aquellas páginas en las que nos encontramos una lista ordenada alfabéticamente. Si la lista es muy larga moverse hasta la letra S, por ejemplo puede hacerse molesto. Pero usando anclas el autor de la página puede incluir todas las letras al principio de manera que si se pulsa sobre una letra se salta al lugar de la lista donde empiecen las entradas de la letra S. En la figura 2.14 podemos ver un ejemplo real capturado de la página del buscador español OZÚ™.

Figura 2.14. En la sección superior puede verse ver una serie de enlaces (las letras mayúsculas) a anclas. En la sección superior se muestra el lugar al que se ha saltado

al pulsar sobre la letra S. Podemos deducir que antes de la palabra Salud hay un ancla.

Buscador Ozú - Microsoft Internet Explorer

Archivo Edición Ver Ir a Favoritos Ayuda

Dirección F:\tmp\la-z.html

 [Añadir](#) [Ay](#)
[E-Mail](#) [Favo](#)

[A](#) - [B](#) - [C](#) - [D](#) - [E](#) - [F](#) - [G](#) - [H](#) - [I](#) - [J](#) - [L](#) - [M](#) - [N](#) - [O](#) - [P](#) - [Q](#) - [R](#) - [S](#) - [T](#) -

- [Academias](#)
- [Acuáticos](#)
- [Aéreos](#)
- [Agencias de Viajes](#)
- [Agrupaciones Nacionales](#)

Mi PC

Dirección #S

- [Salud](#)
- [Santa Claus](#)
- [Seguridad](#)
- [Seguros](#)
- [Semana Santa](#)
- [Sexología](#)
- [Software](#)
- [Sonidos](#)
- [Inicio](#)
- [Tabaco](#)
- [Tarjetas](#)
- [Teatro](#)
- [Telecomunicaciones](#)

Mi PC

En este caso se detallan todas las secciones del Web ordenadas alfabéticamente. Al principio de la página hay un listado con todas las letras del abecedario de manera que al pulsar sobre una letra vas al lugar de la página donde están las secciones que empiezan con esa letra. ¿Cómo se hace esto? Es muy sencillo, el proceso consta de dos pasos:

1. Primero hay que insertar el ancla en el lugar de la página donde se quiere poder saltar. Tomando como ejemplo una una página web que contiene un diccionario se emplearía el siguiente código:

```
...  
<a name="S">  
</a>  
Salud  
Santa Claus  
Seguridad  
...
```

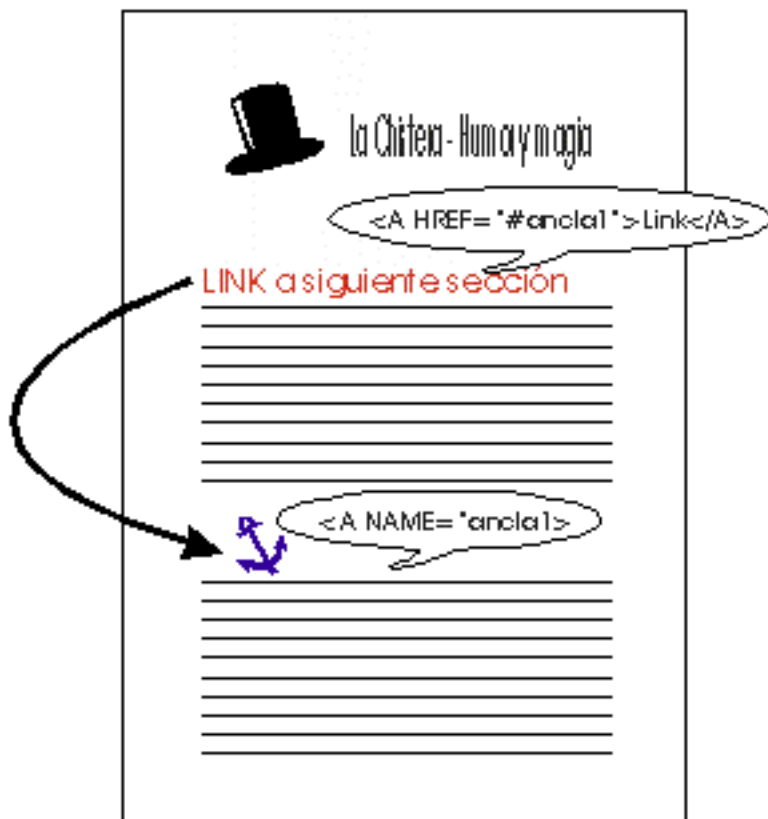
2. En segundo lugar se crea un enlace al principio de la página un que apunte al ancla. En este ejemplo el código usado es:

```
<a href="#S">S  
</a>
```

En este ejemplo al pulsar sobre la letra S se salta directamente al lugar de la página donde se encuentra la palabra *salud*, que es la primera de las que comienzan con “S”, que es el efecto deseado. Para especificar que el enlace *apunta* a un ancla se ha incluido un signo # seguido del nombre del ancla.

La estructura de un enlace a un ancla que se encuentre en la misma página como el de este ejemplo es la que puede verse en la figura 2.15.

Figura 2.15. Estructura de un enlace a un ancla de la misma página. Cuando pulsamos sobre el enlace el navegador nos *transporta* hasta el lugar de la página donde está el ancla.



En ella hemos usado un dibujo de un ancla para señalar donde existe un ancla, en un documento real en realidad no existe ningún elemento visual que la identifique, tal y como hemos visto en el ejemplo de OZÚ™.

Enlaces a anclas de otras páginas

También es posible crear un enlace a un ancla de otra página. Para ello hay que incluir la página, bien sea mediante su dirección URL o mediante su ruta de directorios, y posteriormente poner el símbolo “#” seguido del nombre del ancla. En este caso es obligatorio poner el nombre del archivo, aunque sea `index.html`. Siguiendo con el ejemplo anterior para crear un enlace a los términos que empiezan por ‘S’ de la página donde se situó el ancla puede emplearse el siguiente código:

```
<a href="http://www.ozu.es/a-z.html#S">Letra S</a>
```

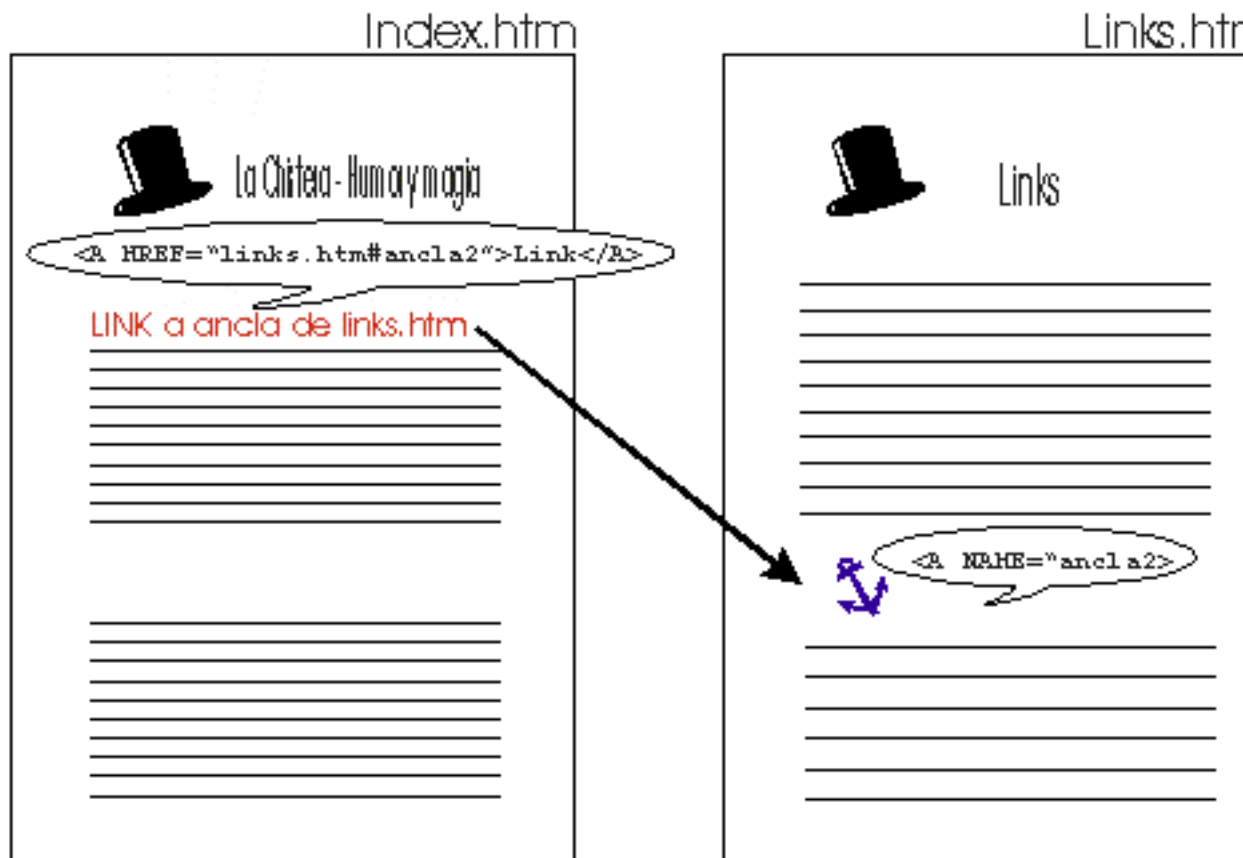
Suponiendo que el diccionario estaba en un archivo llamado `a-z.html` en el servidor de nombre `www.ozu.es` [`http://www.ozu.es`].

La estructura de los enlaces es ligeramente más complicada que los anteriores. Para crearlos hay que seguir los siguientes pasos:

1. Se parte de un ancla existente en otra página o se crea uno
2. Se crea un enlace a esa página
3. Se añade a la dirección del enlace el símbolo '#' seguido del nombre del ancla

En primer lugar deberemos crear un ancla en la página objetivo. Después deberemos escribir el enlace con la sintaxis explicada anteriormente indicando el nombre de la página y el nombre del ancla. La flecha de la figura 2.16 nos indica el salto que se producirá al pulsar sobre el enlace. Ya no iremos al comienzo de la página como pasaba con los enlaces normales sino que iremos al lugar concreto que hemos elegido (y donde hay un ancla).

Figura 2.16. Estructura de un enlace a un ancla de otra página. Cuando pulsamos sobre el enlace vamos hasta otra página, pero no al comienzo sino al lugar concreto donde está el ancla.



Sugerencia

Aunque pueden crearse enlaces a anclas que se encuentren en páginas creadas por otros, no es posible modificar las anclas existentes o añadir anclas nuevas con lo que debemos confiar que el autor haya puesto una donde interese. Para averiguar si es así se puede revisar el código de dicha página buscando una cadena de la forma ``. Hay que tener cuidado con este tipo de enlaces porque el autor de la otra página puede decidir quitar el ancla y todos los enlaces que hagan referencia a ella quedarán inutilizados.

Uso de enlace y ancla en la misma etiqueta

Hasta ahora hemos visto el uso de la etiqueta `<a ...>`; de dos maneras diferentes: para crear enlaces hipertexto o para crear anclas. En el primero de los casos se usaba el atributo 'href' para especificar la página que se querías enlazar. En el segundo caso, al crear el ancla, se usaba el atributo 'name' para darle un nombre con el que poder identificarle. Pero ¿qué ocurre si usamos los dos atributos a la vez?, ¿Está permitido? La respuesta es SÍ. En ese caso se estará creando ambas cosas a la vez un enlace o un ancla. Por ejemplo:

```
<a href="http://es.gnome.org" name="enlace_gnome_hispano">
GNOME Hispano
</a>
```

En este caso al pulsar sobre la palabra 'GNOME Hispano' se salta a la página de GNOME Hispano. Y por otro lado se puede acceder al lugar de la página donde está este enlace creando una referencia al ancla denominada 'enlace_gnome_hispano'. Es decir, la línea anterior es equivalente a poner las dos siguientes:

```
<a name="enlace_gnome_hispano">
<a href="http://es.gnome.org">
GNOME Hispano
/a>
```

Todavía puede plantearse otra pregunta: ¿para que puede servir un enlace-ancla? Esta pregunta tiene mucha lógica porque para lo único que sirve un ancla es para crear un enlace a ella. En el ejemplo anterior se podrías escribir en algún otro lugar de la página:

```
<a href="#enlace_gnome_hispano">
página de GNOME Hispano</a>
```

Pero ¿por qué no poner directamente?:

```
<a href="http://es.gnome.org">
página de GNOME Hispano™
</a>
```

Parece más lógico y así no se fuerza al visitante de la página a dar dos pasos en vez de uno. Sin embargo puede encontrar casos en que el enlace-ancla puede ser conveniente. Por ejemplo, puede ocurrir que se desee poner un enlace a la página de GNOME Hispano desde una página principal. Pero también se quiere hacer caso al consejo de ciertos diseñadores web experimentados de no poner enlaces externos en ella. ¿Cómo solucionarlo?

Seguramente el lector ya lo sabrá: se puede tener una página especial con todos los enlaces externos llamada por ejemplo *links.html* donde estuviese el enlace-ancla anterior. Entonces desde la página principal se escribiría:

```
<a href="links.html#enlace_gnome_hispano">
GNOME Hispano</a>
```

Con lo que además hemos conseguido que el *usuario* visite otra de nuestras páginas, que en definitiva es uno de los objetivos de todo programador Web: que sus páginas sean vistas.

El atributo 'title'

El lenguaje HTML define, además de los vistos, otro atributo interesante para la etiqueta `<a>..`: 'title'. Con este atributo puede escribirse un título o una descripción breve del link. Los navegadores pueden mostrar esta información al usuario de varias maneras:

- En un margen de la pantalla o mediante una caja de texto (en ocasiones llamadas *bocadillos*) que aparezca cuando se sitúen el ratón encima de la zona activa del link. Este es el método usado por Explorer™.
- Cuando el link ya ha sido pulsado y se esté esperando recibir la página puede mostrarse como texto de espera o anticipo a lo que va a llegar.
- Leyendo el texto (con un sintetizador de voz) cuando se selecciona o se pasa por encima
- etc.

Nota

No es habitual encontrar el atributo 'title' en muchas páginas, quizá porque Netscape™ ha decidido ignorarlo y sus usuarios no verán la información que introduzcamos en él de ninguna manera.

Para ver un ejemplo de su uso se utilizó la página titulada "La chistera - Humor y magia" realizada en la primera práctica. En ella se cambia la línea de código:

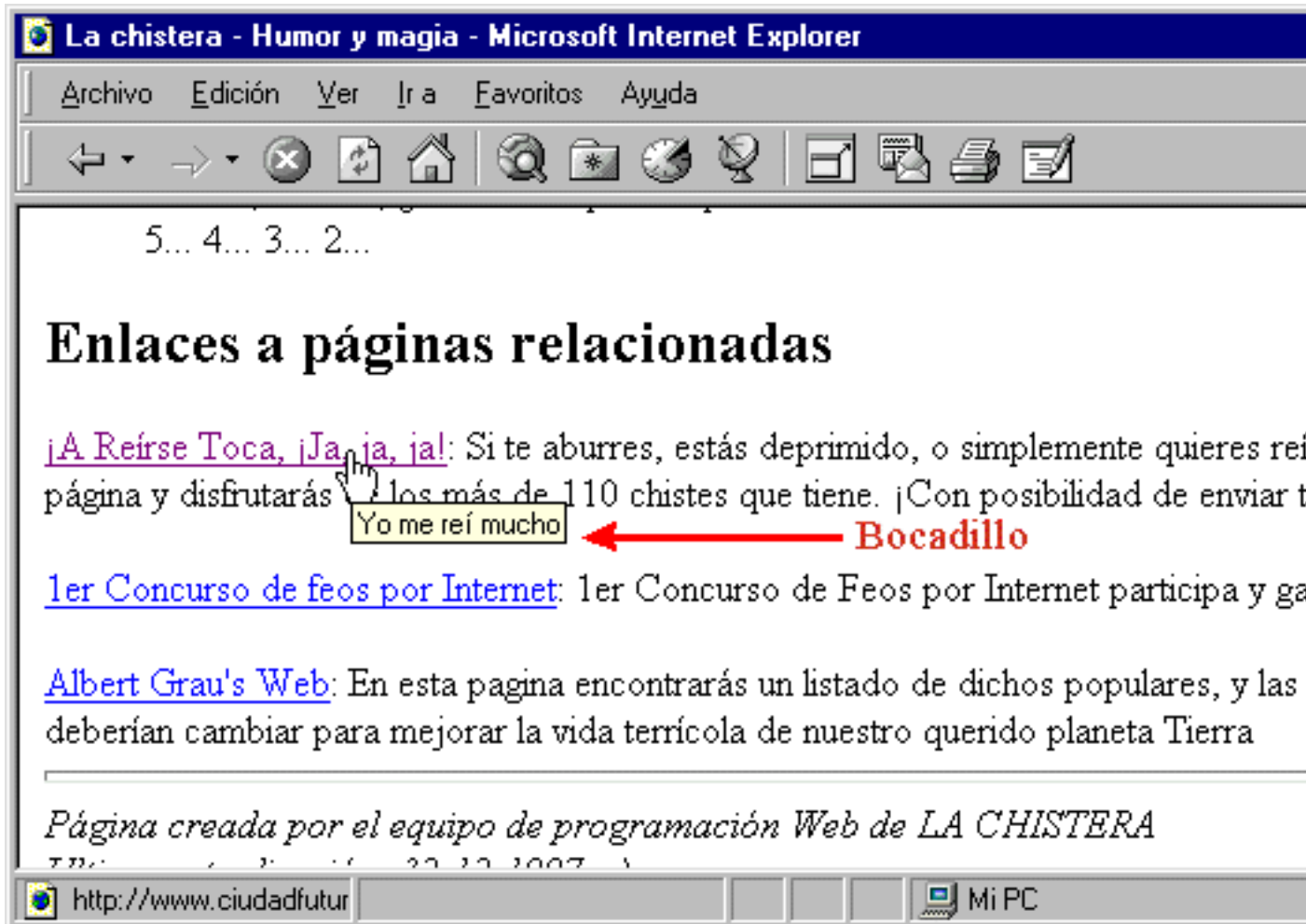
```
<p><a
 href="http://www.ciudadfutura.com/areirse">
¡A Reírse Toca, ¡Ja, ja, ja!</a>
</p>
```

Por esta otra con el atributo 'title':

```
<p><a
 href="http://www.ciudadfutura.com/areirse"
 title="Yo me reí mucho">¡A Reírse Toca, ¡Ja,
ja, ja! </a></p>.
```

El resultado puede verse en la figura 2.17. Al pasar el cursor por encima del texto activo nos aparece un bocadillo con el texto del atributo 'title'.

Figura 2.17. Con el atributo 'title' podemos incluir información adicional sobre el Link. Explorer™ mostrará esta información con un bocadillo.



Sugerencia

Es aconsejable usar 'title' siempre. De esta manera aquellos usuarios (que visiten la página) con navegadores que entiendan este atributo (como Explorer™) disfrutarán de una información extra y los que usen otros navegadores que no lo entiendan (como Netscape) se quedarán como si no hubiéramos incluido dicha información, pero en ningún caso saldrán perjudicados.

Los atributos 'rel' y 'rev'

Antes de terminar con la etiqueta `<a> ...` es conveniente comentar algunos atributos más que completan su repertorio y que no han sido mencionados hasta ahora porque no son muy usados.

Entre estos destacan 'rel' y 'rev' porque en los últimos estándares de HTML en los que han trabajado conjuntamente Microsoft™, Netscape™ y muchas otras compañías y organizaciones, se les ha dado mayor importancia de la que tienen actualmente.

Estos atributos fueron creados para establecer relaciones entre las páginas enlazadas entre sí. Sin embargo no se ha especificado todavía para que pueden servir estas relaciones. 'rel' y 'rev' sólo existen con vistas a que en un futuro se les dé algún uso. El nombre 'rel' proviene de la palabra inglesa *relation* (relación) mientras que 'rev' proviene de *REVerse relation* (relación inversa).

Un ejemplo podría consistir en un libro publicado en Internet donde cada capítulo se ha escrito en una página (`capitulo1.html`, `capitulo2.html`, etc.) Puede ser interesante establecer una relación entre cada una de las páginas. Así en el capítulo 4 se podría emplear lo siguiente:

```
<a href="capitulo3.html"
  rel="anterior">Capítulo 3
</a>
<a href="capitulo5.html" rel="siguiente">
Capítulo 5</a>
```

Es decir el capítulo 3 se relaciona con el actual (el cuarto) en que es el anterior y el 5 en que es el siguiente. Un posible uso (futuro) que podría darse a este tipo de relaciones podría ser con vistas a imprimir el libro completo.

El atributo 'rev' establece la misma relación que 'rel' pero a la inversa. En el ejemplo anterior hubiese sido equivalente escribir:

```
<a href="capitulo3.html" rev="anterior">
Capítulo 3
</a>
<a href="capitulo5.html" rev="siguiente">
Capítulo 5</a>
```

El capítulo actual es el siguiente al tercero y el anterior al quinto.

Por último, para terminar por completo la etiqueta `< a>...`, sólo nos queda comentar dos atributos: 'urn' y 'methods'. Estos dos atributos existían en los orígenes del lenguaje HTML pero fueron eliminados por su poca utilidad. Los mencionamos únicamente para que el lector no se extrañe si los ve en alguna página, pero no recomendamos su uso.

Formato de texto y Caracteres especiales

Jorge Ferrer

Victor García

Versión 1.0

Copyright Jorge Ferrer y Victor García. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation. Puede consultar una copia de la licencia en: <http://www.gnu.org/copyleft/fdl.html>

Ha llegado el momento de estudiar todas las posibilidades que ofrece el lenguaje HTML para dar formato al texto e insertar caracteres especiales, pasando por la inserción de texto preformateado.

En la página creada en el capítulo anterior se pusieron en práctica todos los conocimientos del lenguaje HTML adquiridos hasta el momento, que no son pocos. Sin embargo un aspecto que no se le habrá pasado por alto al lector es la falta entre el texto de palabras en negrita, en cursiva, con distintos tipos de letra, etc. En definitiva, se echa de menos la falta de texto con diferentes formatos. En este capítulo se aprenderán las posibilidades que nos ofrece el lenguaje de creación de páginas Web a este respecto. También se verá cómo crear un nuevo estilo de párrafos, compuesto por texto preformateado, es decir texto en el que se respetarán los espacios, tabulados y saltos de línea que se inserten.

Formato de texto

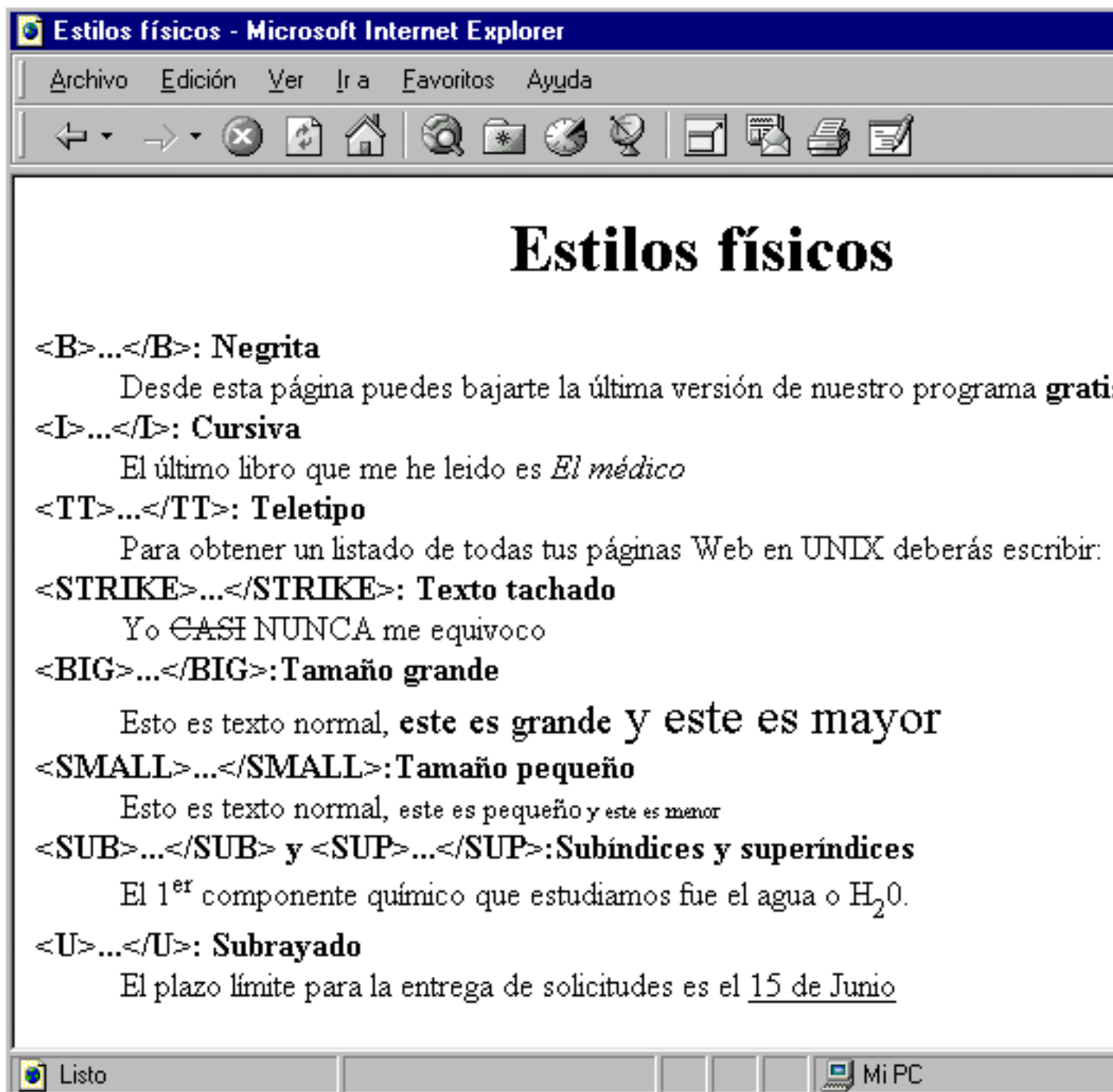
El lenguaje HTML permite dar varios tipos de formatos al texto de nuestras páginas. Para ello usa una serie de etiquetas, compuestas todas por una instrucción de inicio y otra de fin, que realizan efectos o formatos diversos en el texto que encierran. Estas etiquetas se dividen en dos grupos: las etiquetas de estilo lógico y las etiquetas de estilo físico.

Etiquetas de estilo físico

Las etiquetas de estilo físico son aquellas que especifican exactamente cómo debe ser formateado el texto que modifican. Es decir, una etiqueta de estilo físico es aquella que dice: *'este texto debe estar en negrita'* sin más, no ofrece al navegador otra posible forma de mostrar ese texto. Este tipo de etiquetas contradice el espíritu descriptivo que debe tener el HTML, por lo que son muy criticados por los puristas de este lenguaje. Estos puristas sugieren que las etiquetas de estilo físico sean sustituidas por las de estilo lógico que se verán en el siguiente apartado. A pesar de todo las etiquetas de estilo físico son de uso común y por ello las trataremos en detalle.

En la figura 3.1 se muestran ver ejemplos de las etiquetas de este tipo más usadas

Figura 3.1. Con el tipo de letra Times New Roman no todas las letras ocupan lo mismo.



que son las que sirven para...

- **Poner un texto en negrita:** para ello se usa la etiqueta `...` como abreviatura de la palabra inglesa *Bold* que significa, precisamente, negrita. La manera de usar esta etiqueta es como cualquier otra compuesta por instrucción de inicio e instrucción de fin, es decir debe introducirse el texto sobre el que se quiere actuar entre ambas instrucciones. Su forma de uso es la siguiente:

TEXTO EN NEGRITA

Por ejemplo podría usarse en la siguiente frase:

D1

Desde esta página puede adquirir la última versión de
nuestro programa

'*gratis*'

En la que la palabra '*gratis*' será mostrada en negrita tal y como podemos ver en la figura 3.1.

- *Poner un texto en cursiva*: en este caso se usa la etiqueta <I>..</I > como abreviatura de la palabra inglesa *Italics* que significa cursiva. Su uso es muy similar al anterior.

<I>
TEXTO EN CURSIVA
</I>

Esta etiqueta suele usarse para dar énfasis a alguna frase o texto pero sin que resalte demasiado. Otro uso común es para dar un formato especial a palabras de otro idioma o a títulos como por ejemplo:

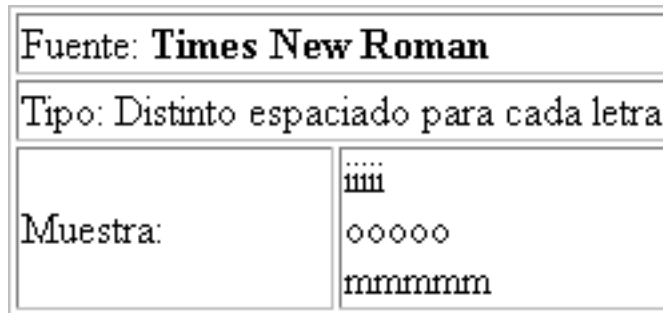
D1

El último libro que me he leído es
<I>
El médico
</I>

En el que 'El médico' será mostrado con letra cursiva como vemos también en la figura 3.1

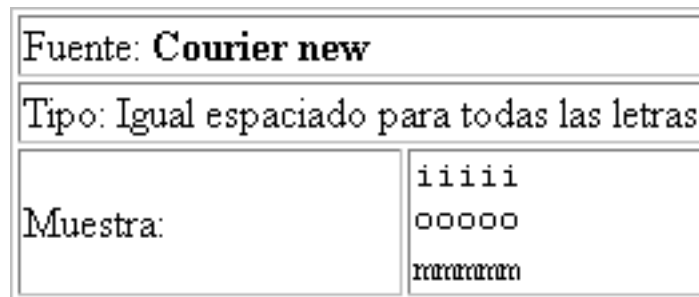
- *Poner un texto con letras de tamaño fijo*: como el lector sabrá existen diversos tipos de letra (llamadas fuentes) como *Times New Roman, Arial, Courier New*, etc. En algunos de estos tipos el tamaño de las letras no está prefijado y de esta manera la letra 'i' no ocupa lo mismo que la letra 'm'. Este es el caso de la fuente *Times New Roman* que es el usado por la mayoría de los navegadores para mostrar el texto. En la figura 3.2 se muestra ver un ejemplo de este tipo de letra.

Figura 3.2. Con el tipo de letra Courier todas las letras ocupan lo mismo (monoespaciadas)



Sin embargo en ocasiones puede ser conveniente que el tamaño de las letras esté prefijado y que todas las letras ocupen lo mismo (tipo de letra *monoespaciado*). En estas ocasiones se usa otro tipo de fuente, generalmente *Courier New*. En la figura 3.3 se observa ver la diferencia con el tipo de letra anterior.
D1

Figura 3.3. Los navegadores gráficos como Explorer [™] pueden mostrar perfectamente todos los tipos de estilos lógicos como vemos en esta figura.



La etiqueta `<TT>..</TT>` que proviene del vocablo inglés TeleType, traducido al español TeleTpo, nos permite decirle al navegador que use un tipo de letra monoespaciado. Esta etiqueta es usada habitualmente para resaltar algún tipo de comando, código, nombre de archivo o similares. Por ejemplo:

```
Para obtener un listado de todas tus páginas Web en UNIX  
deberás escribir:  
<TT>  
ls*.htm *.html  
</TT>
```


- *Tachar una palabra o frase:* aunque no es muy común en ocasiones puede ser conveniente que una palabra o frase aparezcan tachadas por una línea horizontal. La etiqueta de HTML que permite hacer esto es `<STRIKE>..</STRIKE >`, en inglés *strike* significa 'tachado' (También se puede usar la abreviatura `<S>..</S>`) En ocasiones se usa esta etiqueta para simular un texto borrado, por ejemplo:

```
Yo
<STRIKE>
CASI
</STRIKE>
NUNCA me equivoco
```

Todas las etiquetas vistas hasta ahora tienen sus etiquetas de estilo lógico equivalentes. A continuación se muestran otras para las que no ocurre igual y que realmente se muestran a medio camino entre el estilo lógico y el estilo físico.

- *Aumentar el tamaño del texto:* En ocasiones queremos resaltar una palabra o frase haciendo que esté escrita con una letra de tamaño mayor al texto adyacente. Para ello se usa la etiqueta `<BIG>..< /BIG>` con cual conseguimos un ligero aumento de tamaño en el texto que es afectado por ella. Si queremos conseguir un aumento de tamaño aún mayor podemos anidar varias de estas etiquetas como observamos en el ejemplo:

```
Esto es texto normal,
<BIG>
este es grande
<BIG>
y este es mayor
</BIG>
</BIG>
```

- *Reducir el tamaño del texto:* En otras ocasiones podemos pretender lo contrario: Hacer que una frase sea más pequeña que las anteriores. Para ello se usa la etiqueta *SMALL* que funciona de manera muy similar a la etiqueta *BIG*, como podemos ver en su ejemplo:

```
Esto es texto normal,
<SMALL>
este es pequeño
<SMALL>
  y este es menor
</SMALL>
</SMALL>
```

- *Poner subíndices y superíndices*: existen dos etiquetas que permiten poner subíndices y superíndices en una página HTML. Estas etiquetas son `SUB` y `SUP` respectivamente. Veamos un ejemplo:

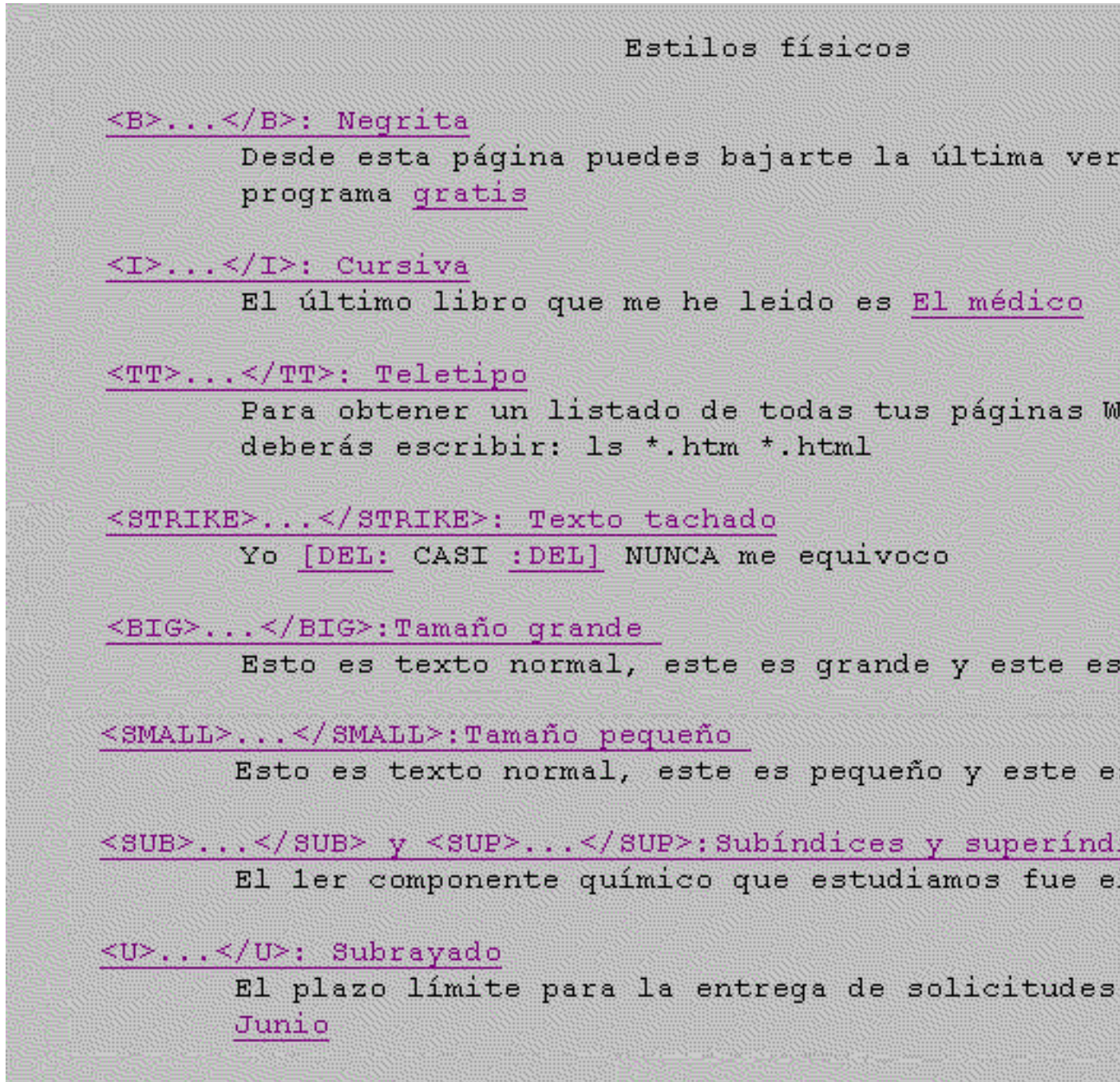
```
El 1
<SUP>
er
</SUP>
componente químico que estudiamos fue el agua o H
<SUB>
2
</SUB>O.
```

- *Y por último, poner texto subrayado*: el texto encerrado entre las instrucciones de inicio y fin de la etiqueta `U` se muestra simple. El nombre de la etiqueta proviene de la palabra inglesa *Underline* que significa precisamente subrayado. En general un texto se muestra subrayado para resaltar una información importante, por ejemplo:

```
El plazo límite para la entrega de solicitudes es el
<U>
15 de Junio
</U>
```

En la figura 3.4 se pueden apreciar todos los ejemplos de estilos físicos tal y como son vistos con un navegador de texto (en concreto Lynx™).

Figura 3.4. El navegador Lynx™ sólo puede mostrar texto y usa los medios a su alcance para formatear el texto que debería ser negrita, cursiva...



Etiquetas de estilo lógico

En contraposición con los estilos físicos están las etiquetas de *estilo lógico*. Este tipo de etiquetas está mucho más en concordancia con la filosofía del lenguaje HTML y los puristas las recomiendan como sustitutos a las de estilo físico. La función de estas etiquetas ya no será decirle al navegador 'pon esto en negrita' ni similares, sino que

simplemente *describirán* el texto, por ejemplo 'este texto es importante', 'esto es un código', etc. Usando esta descripción el navegador lo mostrará de la manera más conveniente que en gran parte de los casos será igual a si hubiésemos usado las etiquetas de estilo físico correspondientes.

Al igual que las anteriores, las etiquetas de estilo lógico están formadas por una instrucción de inicio y otra de fin que encierran un texto. Las principales etiquetas de este tipo son las siguientes :

- `...< /STRONG>`: El nombre de esta etiqueta significa fuerte o grueso, y sirve para indicar que el texto que modifique es un texto de mayor importancia que el que le rodea y es necesario resaltarlo de algún modo. Los navegadores gráficos en general muestran el texto afectado por esta etiqueta en negrita, por lo que puede usarse en sustitución de la etiqueta B. El siguiente es un ejemplo típico de su uso:

```
<STRONG>  
No puedes  
</STRONG>  
perderte la última versión de nuestro  
editor HTML.
```

- `...< /EM>`: Esta etiqueta proviene de la palabra inglesa *Emphasize*, que significa enfatizar. El texto sobre el que actúa esta etiqueta suele estar mostrado en cursiva, tal y como puede verse en la figura 3.5 como resultado del siguiente código:

```
El autor del libro es  
<EM>  
Noah Gordon  
</EM>.
```

- `<CODE>...< /CODE>`: con esta etiqueta señalamos un texto que forma parte de un código, de instrucciones de ordenador,... La palabra inglesa *code* significa, precisamente, código. Con Explorer™ y Navigator™ el texto delimitado por esta etiqueta será mostrado con un tipo de letra monoespaciada (por ejemplo Courier) y de tamaño ligeramente menor al habitual. Esta etiqueta puede usarse, por ejemplo, para insertar código en C:

```
<CODE>  
#include "stdio2.h"  
</CODE>
```

- `<SAMP>...< /SAMP>`: el nombre de esta etiqueta proviene del inglés *SAMPle* (muestra) y suele utilizarse para formatear ejemplos, muestras, códigos de scripts, etc. El texto formateado con esta etiqueta suele ser mostrado de la misma manera que el formateado con `<CODE>`.
- `<VAR>...< /VAR>`: delimita texto escrito por el usuario, variables, argumentos de comandos... Tanto el Explorer™ como el Navigator™ mostrarán este texto como cursiva. Un ejemplo de su uso podría ser:

```
La dirección de esa página es  
<VAR>  
www.towercom.es  
</VAR>
```

- `<CITE>...< /CITE>`: Esta etiqueta suele ser usada para introducir una cita o una referencia (significado de la palabra inglesa *cite*). Al igual que la etiqueta anterior provoca que el texto que encierra sea mostrado en cursiva:

```
<CITE>
Pienso, luego existo
</CITE>
dijo Descartes
```

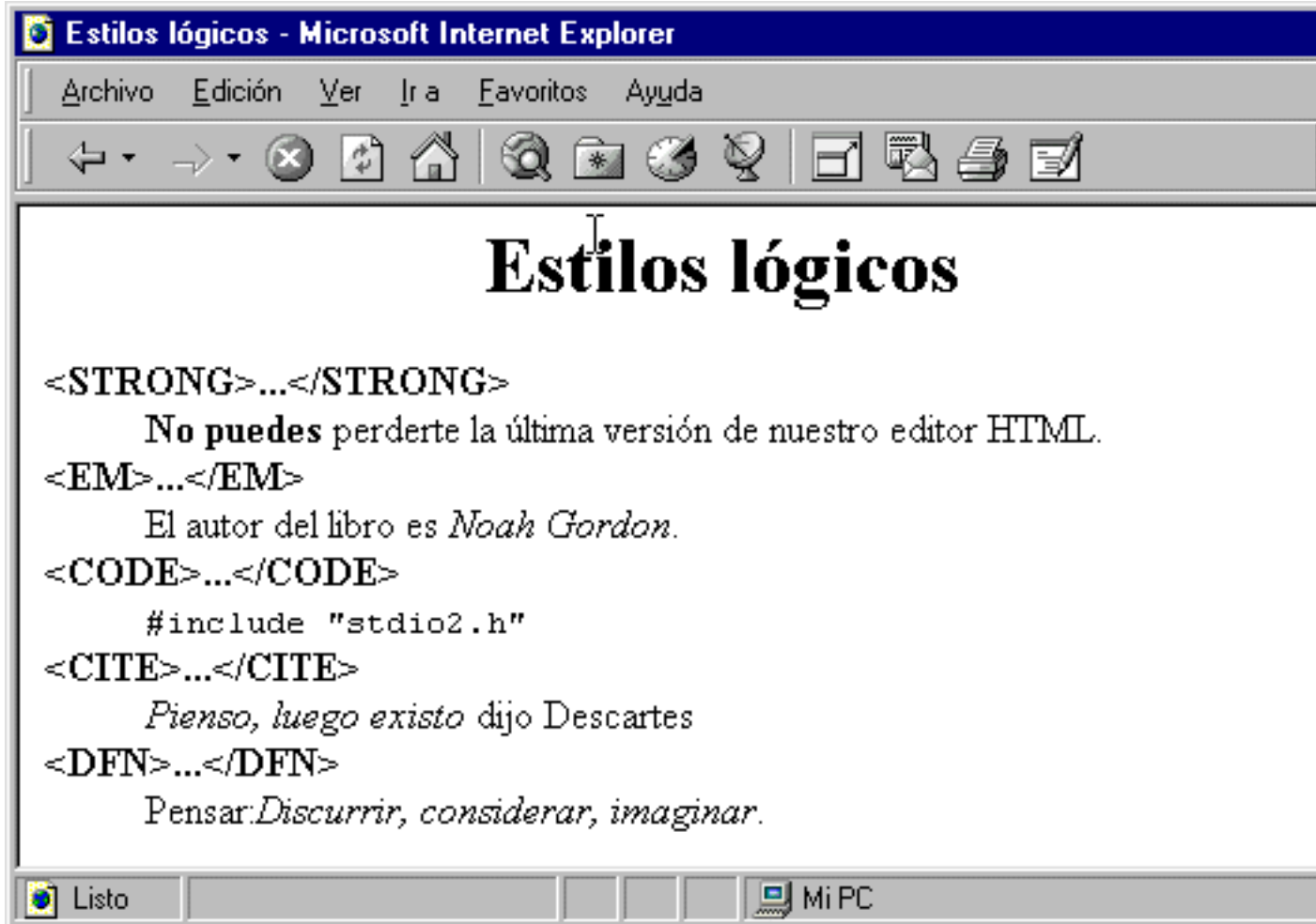
- `<DFN>...< /DFN>`: En este caso el objeto de la etiqueta es introducir una DeFiNición, pero el texto será mostrado igualmente en cursiva:

```
Pensar:
<DFN>
Discurrir, considerar, imaginar
</DFN>.
```

- `<KBD>...< /KBD>`: Tiene la función de formatear texto de entrada (input) del teclado (KeyBoarD en inglés) con un tipo de letra monoespaciado. Aunque esta etiqueta pertenece al HTML no es soportada por Explorer™. recomendable

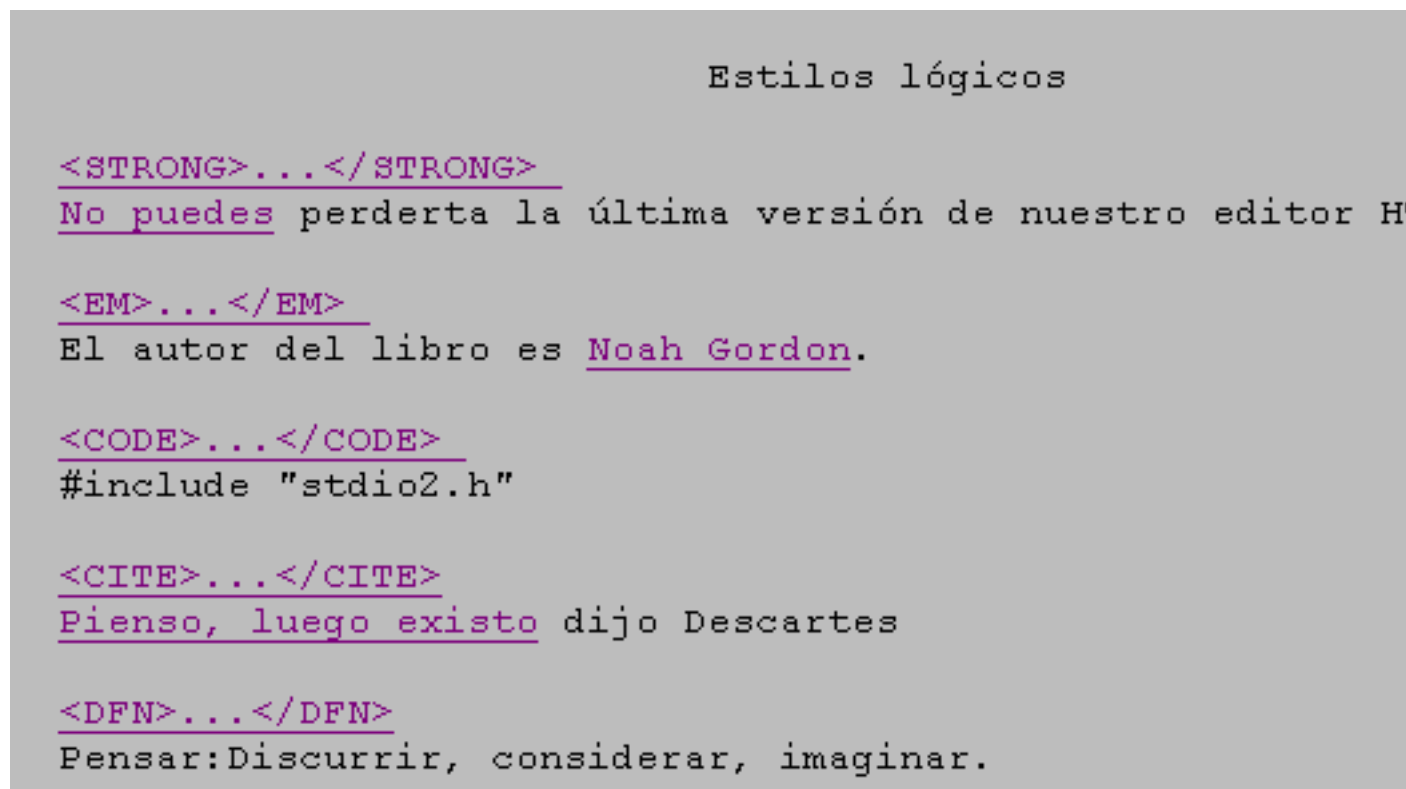
Pueden verse todos los ejemplos anteriores, tal y como los muestra Explorer™, en la figura 3.5:

Figura 3.5. En esta figura observamos los estilos lógicos con Explorer™. La gran mayoría de ellos son mostrados como letra en negrita o cursiva.



Todos estas etiquetas son entendidas por los navegadores en modo texto y usan los medios de que disponen para resaltar las frases que encierran. En la figura 3.6 se incluyen todos los ejemplos tal y como los muestra Lynx.

Figura 3.6. Usando estilos lógicos no se nota tanto la diferencia entre Explorer™ y Lynx™ aunque los medios de este sean más limitados.



Una característica que llama la atención de estas etiquetas es que varias de ellas llevan a cabo el mismo cometido, o eso parece. Esto es así porque se está describiendo el texto, no indicando como serán formateados. ¿Porqué no se unen las etiquetas que hacen lo mismo en una sola?, porque se supone que describen texto lo suficientemente diferente como para que aunque si es conveniente puedan ser formateadas de forma diferente.

Ya se han visto los estilos lógicos y los físicos, así como las diferencias entre ellas. Como se ha dicho es recomendable usar los primeros siempre que sea posible.

La etiqueta FONT.

Las etiquetas de los apartados anteriores permiten modificar ciertos aspectos de la apariencia del texto de una página, pero ¿es suficiente?. Cuando el WWW se hizo más y más popular y la apariencia de las páginas comenzó a adquirir la importancia actual se hizo evidente que no.

Para poder controlar en mayor medida aún la apariencia del texto se creó la etiqueta FONT. A continuación se explicarán sus virtudes, aunque cabe destacar que actualmente existe una alternativa mejor: las hojas de estilo en cascada o CSS que será tratada en los últimos capítulos del curso.

Cambiando el tamaño del texto.

La etiqueta FONT permite tener un gran control sobre el tamaño de un texto. Este tamaño se especifica con su atributo SIZE y afectará al texto que encierran la instrucción de inicio () y de fin ().

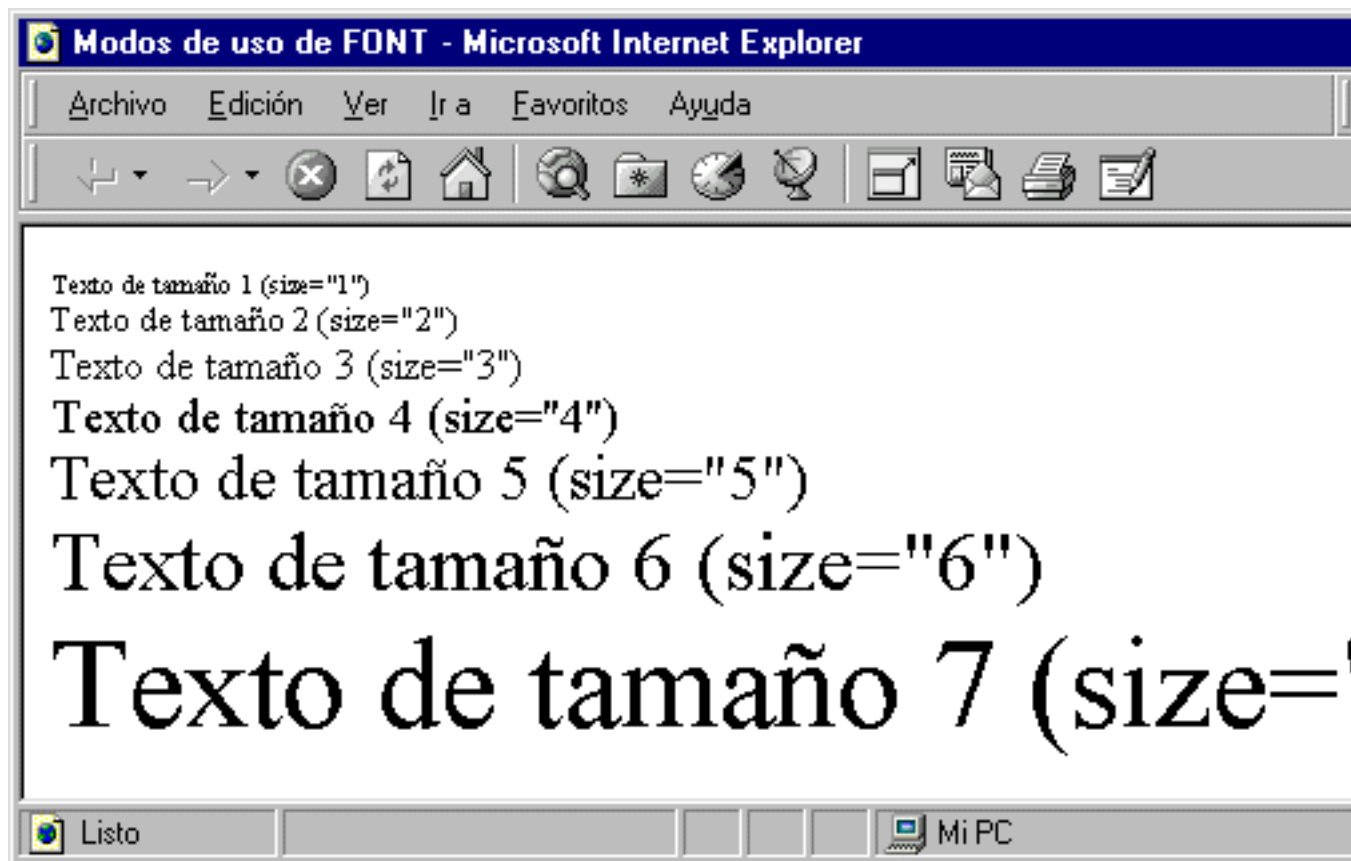
La forma de especificar el tamaño del texto consistirá en darle el valor deseado al atributo SIZE, si bien hay dos formas de hacerlo:

1. La primera de ellas es indicar un tamaño absoluto de 1 a 7. Correspondiendo 7 al tamaño mayor y 1 al menor. El tamaño por defecto del texto es el equivalente al número 3. Veamos un ejemplo:

```
Esto es texto normal  
<FONT SIZE="6">  
y esto es grande  
</FONT>
```

En la figura 3.7 podemos ver una comparación de todos los tamaños.
D1

Figura 3.7. Ejemplo de uso de la etiqueta FONT y su atributo SIZE para cambiar el tamaño del texto.



2. La segunda consistirá en indicar un *tamaño relativo* respecto al que haya por defecto. Por ejemplo, para poner un tamaño 6 y suponiendo un tamaño por defecto 3 se usaría:

```
Texto de tamaño 3  
<FONT SIZE="+3">  
Tamaño 6  
</FONT>
```

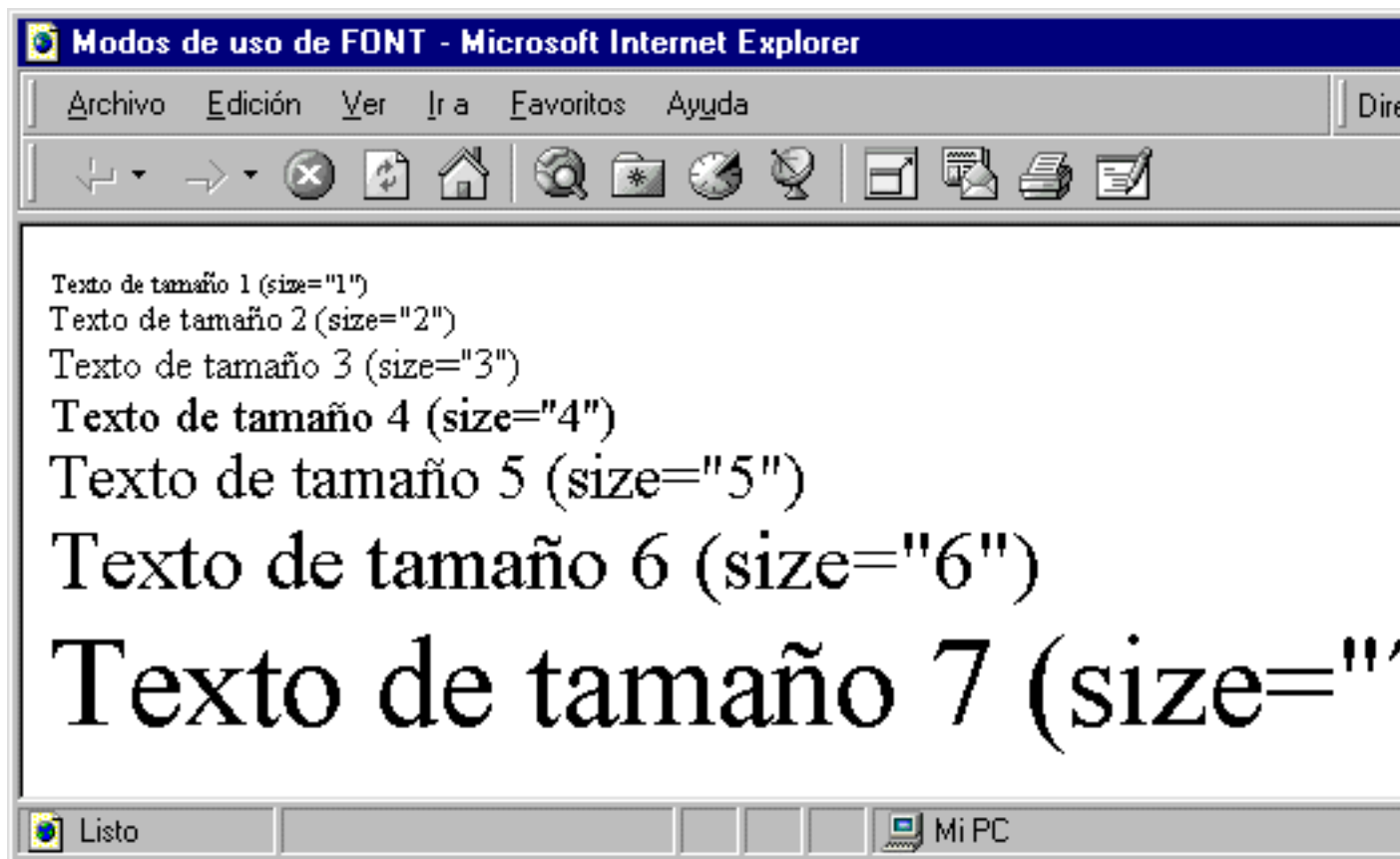
Es necesario hacer notar que el aumento de tamaño se tomará siempre respecto al que se tenga por defecto (y que será 3 mientras el usuario no cambie las preferencias de su navegador), por tanto anidando varias etiquetas FONT no se irá aumentando el tamaño del texto como ocurría con la etiqueta BIG. Por ejemplo:

D1

```
Texto normal (tamaño 3),  
<FONT SIZE="+1">  
Texto de tamaño 4 ,  
  <FONT SIZE="+1">  
    este texto sigue teniendo tamaño 4  
  </FONT>  
</FONT>
```

En la figura 3.7 se puede ver un ejemplo de ambos modos de uso.

Figura 3.7. Ejemplo de uso de la etiqueta FONT y su atributo SIZE para cambiar el tamaño del texto.



Nota

Cuando se indica el tamaño del texto usando el método relativo (+n o -n) será necesario usar comillas. Aunque algunos navegadores podrían entender el valor aunque no se usen la única manera de estar seguro de que la visualización será correcta en todos ellos será entrecomillar el valor dado al atributo SIZE.

Conocidos estos dos métodos es el momento de introducir una nueva etiqueta muy relacionada con FONT: la etiqueta BASEFONT. Esta permite modificar el tamaño por defecto del texto de la página Web. Su modo de uso es el siguiente:

```
<BASEFONT SIZE="n">
```

Donde n es un número del 1 al 7 (por defecto vale 3) y las comillas son obligatorias. Podemos observar como esta etiqueta está formada por una única instrucción cuya función será modificar el tamaño de todo el texto a partir de su inclusión. La única excepción que escapa al efecto de esta etiqueta son los encabezados, que mantienen su tamaño.

En general cuando se usa esta etiqueta se introduce como la primera después de <BODY>, no usarla es lo mismo que usar <BASEFONT SIZE="3">.

A continuación se incluyen dos ejemplos con distintos tamaños de letra por defecto:

```
<BASEFONT SIZE="3">  
Esto es texto de tamaño 3,  
<FONT SIZE="+3">  
esto es de tamaño 6,  
</FONT>  
<FONT SIZE="+4">  
y esto es de tamaño 7  
</FONT>
```

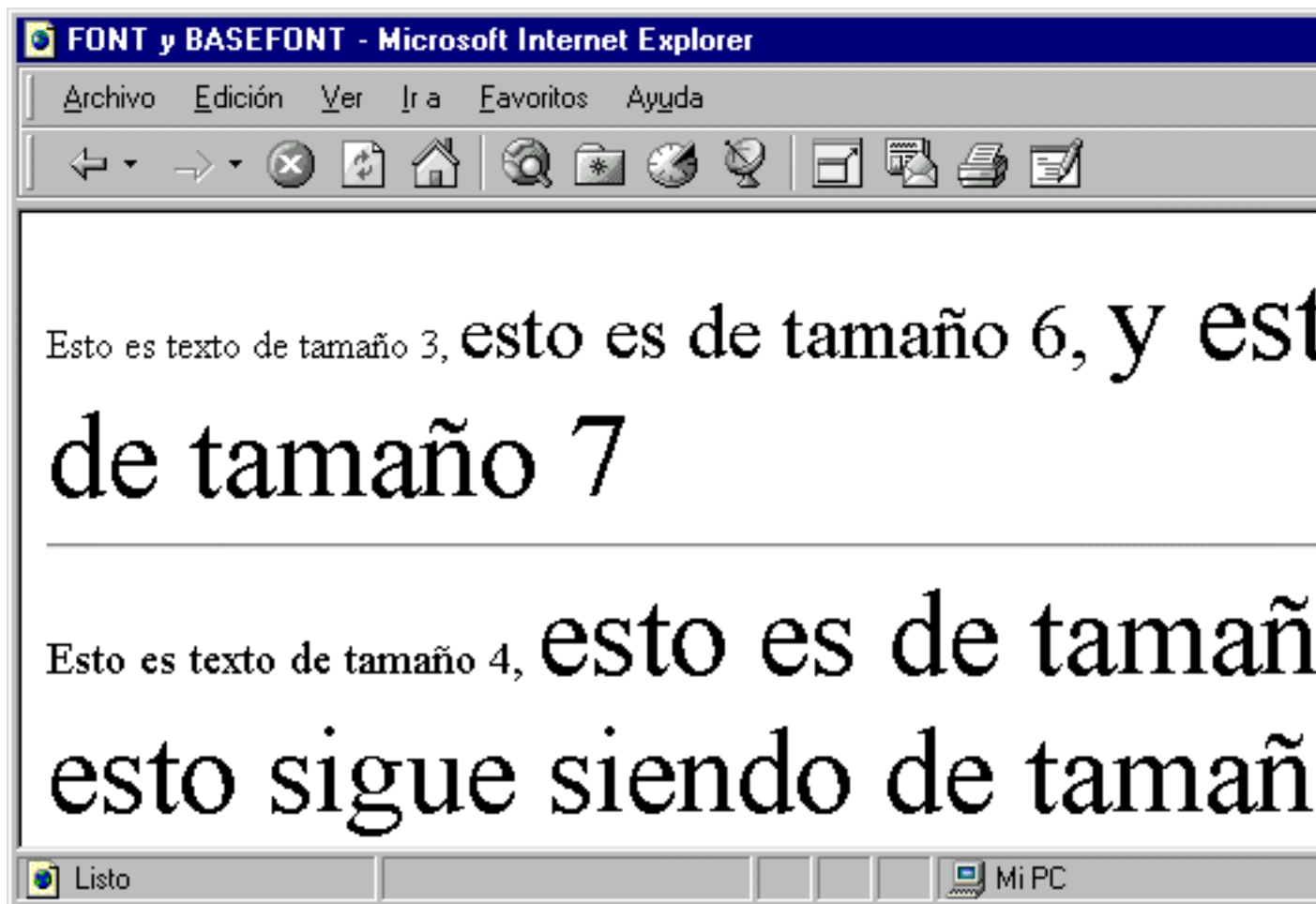
Como se ve, para hallar el tamaño de un texto hay que sumar el tamaño relativo (+3 o +4 en este caso) al tamaño por defecto del resto del texto (en este caso 2). Otro ejemplo más:

```
<BASEFONT SIZE="4">  
Esto es texto de tamaño 4,  
<FONT SIZE="+3">  
esto es de tamaño 7,  
</FONT>  
<FONT SIZE="+4">  
y esto sigue siendo de tamaño 7  
</FONT>
```

En este segundo ejemplo se ve, además, que en ningún caso se puede conseguir un tamaño superior a siete. De igual manera nunca se puede poner un tamaño menor que uno.

En la figura 3.8 podemos ver ambos ejemplos.

Figura 3.8. Los tamaños del texto que hallamos modificado usando el método relativo se calcularán a partir del al tamaño por defecto del texto normal, pero sin pasar nunca de 7.



Cambiando el color del texto

Con otro atributo de la etiqueta FONT: *COLOR* se puede cambiar el color del texto con la misma facilidad con la que hasta ahora se ha cambiado su tamaño. En un principio la única forma de indicar un color consistía en usar un código hexadecimal de la siguiente manera:

```
<FONT COLOR="#RRGGBB">
```

Dónde hay que sustituir RR por dos números hexadecimales que indican la cantidad de rojo (desde 00 que es ausencia de color a FF que es máxima cantidad de color), GG por otros dos que indican la cantidad de verde y BB por dos que indican la cantidad de azul. Combinando estos tres colores en diferentes proporciones se pueden crear hasta 16 millones de colores. Este sistema suele parecer ligeramente complicado en un principio, pero con la práctica o con ayuda de herramientas gráficas de selección de colores se convierte en un método bastante sencillo e increíblemente potente.

Existe un segundo método de indicar el color que sin duda le resultará al lector más sencillo. Este método consiste en poner como valor de COLOR el nombre del color en inglés. Los colores aceptados son: *Black* (negro), *white* (blanco), *green* (verde), *maroon* (marrón rojizo), *olive* (verde oliva), *Navy* (azul marino), *purple* (violeta), *red* (rojo), *yellow* (amarillo), *blue* (azul), *teal* (verde azulado), *lime* (verde lima), *aqua* (azul claro), *fuchsia* (fusia), *silver* (gris claro). Aunque además de estos existen otros valores, no son soportados en todos los navegadores.

Y la forma de usarlas queda clara en el siguiente ejemplo:

```
<H1>
<FONT COLOR="FUCHSIA">
Colores en el Web
</FONT>
</H1>
ya podemos decir que sabemos poner colores en las páginas Web,
<FONT COLOR="green">
verde
</FONT>,
<FONT COLOR="blue">
azul
</FONT>
<FONT COLOR="red">
rojo
</FONT>,
etc. Ninguno se nos resiste.
```

cuyo resultado podemos ver en la figura 3.9 :

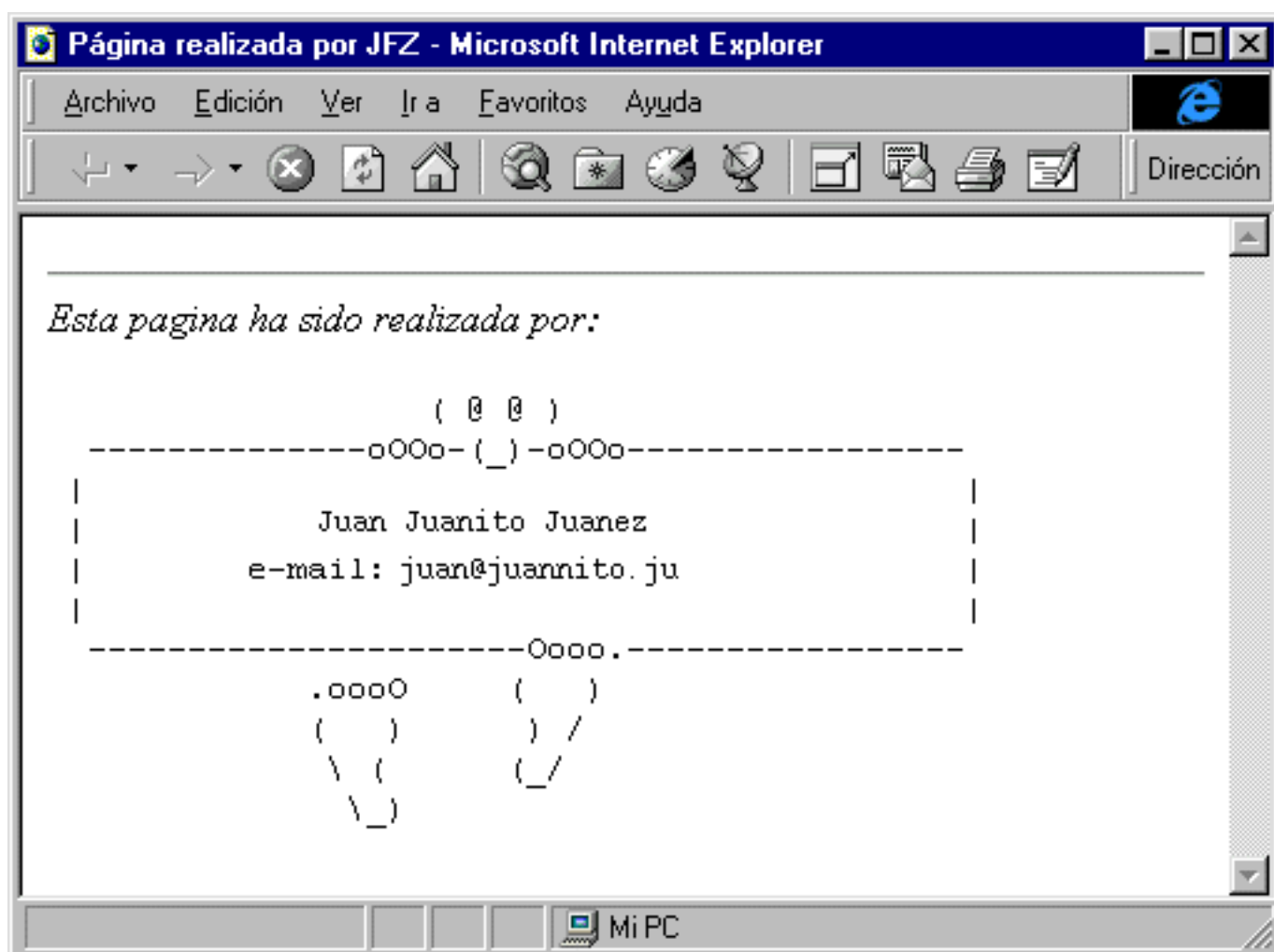
Figura 3.9. Gracias a la etiqueta FONT y a su atributo COLOR nuestras páginas tomarán un colorido totalmente diferente.



Texto preformateado.

En ocasiones es interesante mantener los espacios o saltos de línea de un párrafo tal y como están en el código HTML. Para conseguir esto hay que alterar el comportamiento habitual del navegador que, como se ha visto, ignora este formato. Un ejemplo muy simple que muestra esto es la firma de páginas con dibujos de texto como el de la figura 3.10.

Figura 3.10. Como se puede ver se pueden crear firmas muy originales usando únicamente caracteres y sin necesidad de imágenes.



Se podría crear una imagen con él e insertarla en el documento, pero este es un dibujo hecho con caracteres y podría aprovecharse para que pudiese verse también con navegadores que trabajan en modo texto como Lynx. Algo importante en este caso ya que interesa que TODO el mundo vea la firma, además de que el tenerla como texto hace más fácil su modificación.

Sin embargo todos los espacios múltiples se convertirán en un único espacio, y por tanto si *se escribiese* el dibujo junto con el código HTML el navegador lo mostraría como vemos en la figura 3.11 :

Figura 3.11. Si no usamos la etiqueta de texto preformateado la figura construida con caracteres se convertirá en algo ilegible.



que sin duda no es el resultado que esperábamos.

Aunque este es un ejemplo entre muchos, el objetivo era resaltar que en ocasiones esa característica de HTML de eliminar todo formato manual en el texto es verdaderamente incómoda. Pero todo problema tiene solución, y en este caso la solución es una nueva etiqueta: `<PRE>...</PRE>`. El nombre asignado a esta etiqueta proviene de la palabra "PREformateado" (o *PREformatted* en inglés) y su función es mantener todo formateado manual que sea hecho en el texto escrito entre la instrucción de inicio (`<PRE>`) y la instrucción de fin (`</PRE>`).

Continuando con el ejemplo de nuestra firma, el problema se podría solucionar si al insertarla la incluimos dentro de la etiqueta PRE de la siguiente manera:

```
<PRE>
      ( @ @ )
-----oOoO-( )-oOoO-----
|           |
|   Juan Juanito Juanez   |
| E-mail: juan@juanito.ju |
|           |
-----Oooo.-----
      .oooO   ( )
      ( )    ) /
      \ (    ( /
      \ )
</PRE>
```

En este caso la visualización será la esperada, es decir la de la figura 3.10.

Aviso

Es común en los nuevos programadores que descubren la etiqueta PRE usarla abundantemente, sin embargo, esto puede causar problemas de visualización de esas páginas con algunos tamaños de ventana, resoluciones de pantalla o tipos de monitor distintos del nuestro. Recordemos que esta era la razón por la que los navegadores ignoraban el formato manual y debido a ello el uso de PRE debe estar limitado a los casos en los que sea realmente necesario.

Uso de otras etiquetas dentro de <PRE>...</PRE>

A efectos de HTML estándar, el contenido de la etiqueta PRE es el mismo que el de un párrafo creado con P. Las únicas excepciones son aquellos elementos que producen cambios en el tamaño de la letra como por ejemplo IMG, BIG, SMALL, SUB, SUP y FONT. Esto excluye también a todas aquellas etiquetas que provocan un cambio de línea (son las llamadas etiquetas de nivel de bloque, *block-level tags*) como son: H1,..., H6, BLOCKQUOTE, ADDRESS, HR, etc.

Nota

Tanto Netscape™ como Explorer™ permiten el uso de todo tipo de etiquetas dentro del texto preformateado, incluso aquellas etiquetas de nivel de bloque. En estas últimas hay ligeras diferencias entre Explorer™ y Netscape™, valga como ejemplo que los encabezados en Explorer™ mantienen el tipo de letra del texto preformateado (generalmente Courier New) mientras que Netscape™ los muestra con el tipo de letra normal (generalmente Times New Roman).

Acentos y otros caracteres especiales

Algunos caracteres distintos de los habituales en los lenguajes anglosajones son considerados en el mundo de la informática como caracteres especiales. Dentro de estos se incluyen las letras acentuadas ni la letra ñe.

En los últimos años se ha avanzado mucho en conseguir que cualquier ordenador pueda leer todos los caracteres de todos los idiomas. En concreto es ya poco habitual que no se soporten los caracteres latinos. En cualquier caso es interesante saber que HTML ofrece mecanismos para insertar estos y otros caracteres especiales en aquellos sistemas donde no están soportados.

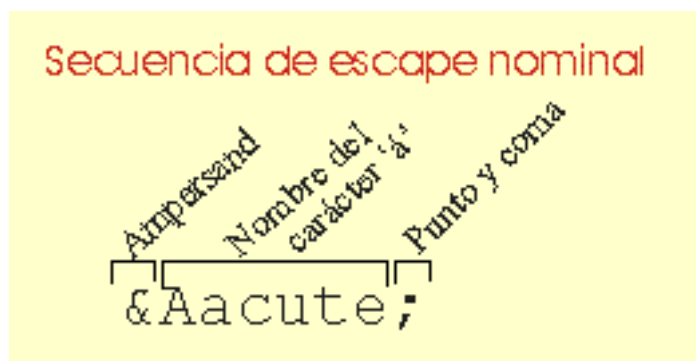
Sugerencia

Una regla mnemotécnica para recordar qué caracteres son considerados especiales por HTML es la siguiente: "*Si se puede escribir un carácter pulsando sobre una sola tecla del teclado o usando como ayuda las teclas MAYÚSCULAS o ALT-GR entonces ese carácter es válido*". En el teclado español hay que hacer tres excepciones: la letra 'ñ' y los símbolos '¿' y '¡' que no son válidas.

El mecanismo ofrecido por HTML consiste en unos códigos especiales que todos los navegadores pueden entender. A estos códigos se les llama *entidades de caracteres* o con el nombre inglés: *character entities*. Todos estos códigos comienzan por el símbolo '&' (ampersand) y terminan con el símbolo ';' (punto y coma). Puede distinguirse entre dos tipos:

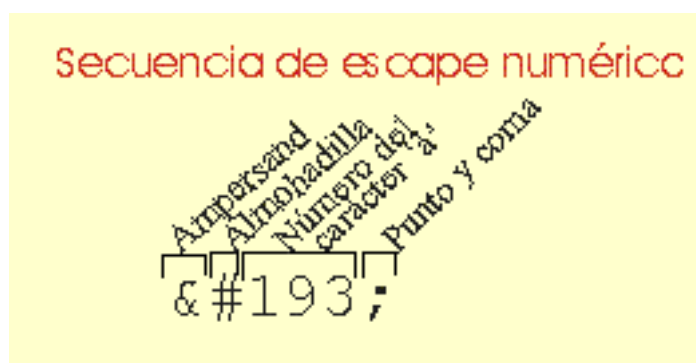
- *Entidades de caracteres con nombre*: son aquellas en las que los símbolos '&' y ';' se pone el nombre (o abreviatura) asignado a ese carácter. Sólo existen para los caracteres especiales más usados. En la figura 3.12 se puede ver un ejemplo de este tipo de entidades en el que vemos la estructura general que todas tienen.

Figura 3.12. Una entidad de caracter con nombre tiene tres partes: '&' + 'nombre' + ';'.



- *Entidades de caracteres numéricas*: en este tipo de entidades entre '&' y ';' se escribe el número asignado a ese carácter en el estándar ISO-Latin-1 precedido de una almohadilla: '#'. Este tipo de entidades son menos usadas que las anteriores aunque tienen la ventaja de abarcar cualquier letra posible en cualquier idioma. En la figura 3.13 se muestra la estructura de este tipo de entidades usando de nuevo como ejemplo la letra A mayúscula y acentuada.

Figura 3.13. Una entidad de caracter numérica tiene cuatro partes: '&' + '#' + 'número' + ';'.



Entidades de caracteres para caracteres españoles

Sin duda, en España los caracteres considerados como especiales en el resto del mundo que más se usan son los acentos, la letra ñe y los símbolos de apertura de interrogación (¿) y admiración (¡). En la tabla 3.1 se puede ver un listado con dichos caracteres en los que mostramos las entidades de caracteres con nombre y numéricas que tienen asociados.

Figura 3.1. Entidades de caracteres numéricas y nominales para caracteres acentuados, 'u' con diéresis y ñe.

Carácter	Secuencia escape	Sec. escape numérica	Carácter	Secuencia escape	Sec. escape numérica
á	á	á	Á	Á	Á
é	é	é	É	É	É
í	í	í	Í	Í	Í
ó	ó	ó	Ó	Ó	Ó
ú	ú	ú	Ú	Ú	Ú
ü	ü	ü	Ü	Ü	Ü
ñ	ñ	¤	Ñ	Ñ	¥
¡	-----	¡	¿	-----	¿

Aviso

Es muy importante respetar las mayúsculas y minúsculas en las entidades de caracteres. Si se usan indistintamente el resultado obtenido no será el deseado o bien la entidad simplemente no funcionará.

Entidades de caracteres para caracteres reservados

En el lenguaje HTML existen una serie de caracteres que tienen un significado especial. La gran mayoría de estos caracteres ya los conocemos y estamos acostumbrados a usarlos para programar un página Web. Algunos de estos caracteres son: <, >, & y ". Pero ¿qué ocurre si se quiere incluir alguno de estos caracteres en alguna página? Por ejemplo, si se desea escribir sobre los formatos lógicos en HTML y se quiere poner la cadena "" en una página. Si la se pone tal cual, el navegador la confundirá con una etiqueta más y pondrá el texto siguiente en negrita. ¿Existe entonces alguna forma de escribir el nombre de una etiqueta? La respuesta es, obviamente, Sí. Y la solución es usar las entidades de los caracteres reservados de HTML en vez del propio carácter. Estas entidades son las mostradas en la tabla 3.2.

Figura 3.2. Entidades de caracter numéricas y nominales para caracteres reservados del lenguaje HTML.

Carácter	Secuencia de escape	Sec. de esc. numérica
<	<	‹
>	>	›
&	&	&
"	"e;	"

Por ejemplo en vez de escribir: Para poner texto en negrita se usa la etiqueta

Debería escribirse: Para poner texto en negrita se usa la etiqueta

Este segundo ejemplo se visualizará correctamente en el navegador, el lector está invitado a probar los dos ejemplos.

Otras entidades de caracteres

Además de las vistas hasta ahora existen otras entidades de caracteres que son de especial utilidad en la creación de páginas Web. Estos son los de la tabla 3.3

Figura 3.3. Entidades de caracteres numéricas y nominales para algunos caracteres especiales de uso común.

Carácter	Secuencia de escape	Sec. de esc. numérica
©	©	©
®	®	®
[espacio]	 sp;	-----

El espacio requiere un comentario especial. El nombre de su entidades de caracteres, nbsp, significa *non breaking space* que quiere decir espacio que no puede ser roto (o eliminado). Este tipo de espacios no es ignorado por los navegadores. Debe recordarse que si incluimos varios espacios en nuestra página el navegador los reduce a uno sólo, pues esto no ocurre con este tipo de espacios. Si se escribe sp; cinco veces el navegador mostrará cinco espacios, con lo cual se pueden crear márgenes y sangrados manualmente.

Nota

En los archivos extra de este capítulo se incluye un listado de todos los caracteres del conjunto ISO-Latin-1 junto con sus números asociados. Este listado va en el archivo (`especial.html` [extra/especial.html]), para que pueda ser visualizado con diferentes navegadores y en dos imágenes GIF que corresponden a la visualización de dicho archivo HTML con Internet Explorer 4.0™. (`escape.gif` y `latin1.gif`). Esta información esta dividida en dos tablas, en la primera se incluyen todos aquellos caracteres que tienen una entidades de caracter nominal asociada. En una segunda tabla se incluyen todos los caracteres visualizables (por ejemplo el carácter retorno de carro no se puede ver), del ISO-Latin-1 junto con su número asociado. A partir de este número podremos construir la entidades de caracter numérica.

El porqué de los caracteres especiales. Explicación técnica

En los comienzos de la informática se creó un código llamado *ASCII* (American Standard Code for Information Interchange) para representar los caracteres. Este código asignaba a cada carácter un número (de 7 bits) que al fin y al cabo es lo único con lo que un ordenador sabe trabajar. Debido al tamaño de ese número (7 bits) ese código constaba únicamente de 128 caracteres que incluía las letras, números y algunos caracteres habituales. Estos caracteres son entendidos en cualquier ordenador y por tanto pueden ser usados con libertad en un documento HTML. Con la evolución y expansión de los ordenadores pronto surgió la necesidad de añadir nuevos caracteres, como por ejemplo aquellos específicos de cada idioma. Por esta razón se crearon extensiones del código ASCII que añadían un bit al número asociado a cada carácter con lo que el número de caracteres representable aumentaba a 256. El problema es que no hubo un acuerdo en este ASCII extendido y se crearon varias versiones en los que había ligeras diferencias. Los documentos HTML pueden tener todo tipo de caracteres pertenecientes al IS-Latin-1 (ISO-8859-1) que es una de estas extensiones. Aún así los caracteres que no pertenezcan al código ASCII de 7 bits

deben introducirse con la secuencia de escape correspondiente para evitar problemas con ordenadores que no soporten el ISO-Latin-1 sino otra extensión del ASCII. El próximo estándar del lenguaje HTML (HTML 4.0) ha llevado más lejos la extensión del código ASCII soportando el conjunto de caracteres *UNICODE*. Este conjunto incluye los caracteres del ASCII (a los que asigna el mismo número y por tanto es compatible), pero añade todos los caracteres de todos los idiomas del mundo (incluidos los chinos y japoneses). Esto es importante porque *UNICODE* se está imponiendo como uno de los estándares del presente y del futuro.

Listas y Comentarios en HTML

Jorge Ferrer
Rodrigo García

Versión 1.0

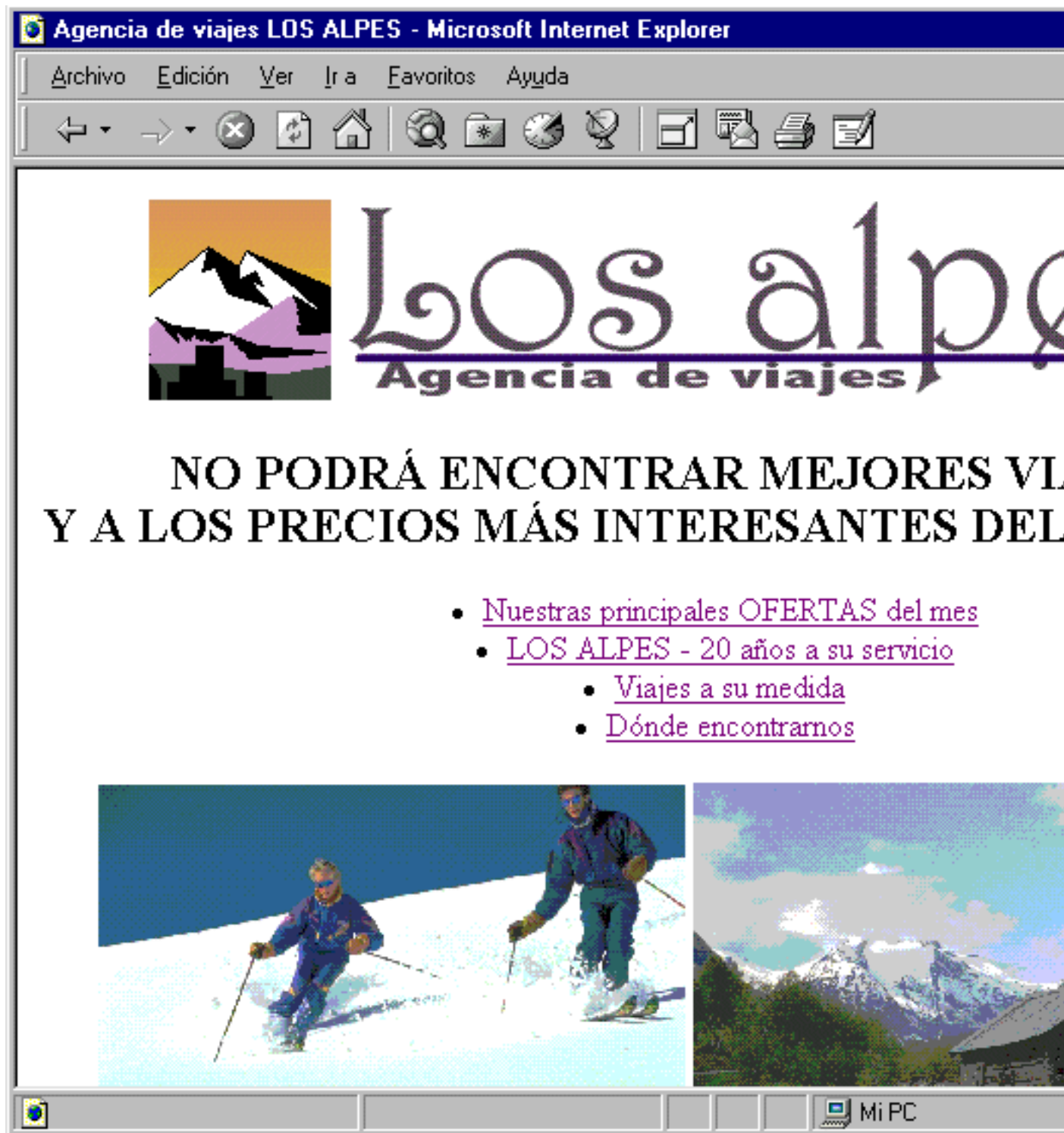
Copyright Jorge Ferrer y Victor García. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation. Puede consultar una copia de la licencia en: <http://www.gnu.org/copyleft/fdl.html>

En este capítulo nos adentramos en las posibilidades que ofrece el lenguaje HTML para la creación de listas. Es posible crear listas numeradas, no numeradas, de definición, etc. gracias a las cuales se facilitará al programador la labor de organizar los datos de un documento HTML y dar una apariencia más agradable a la página. Después crearemos una segunda página web de la que sin duda podremos presumir, pero antes aprenderemos a incluir comentarios en nuestras páginas.

Listas en HTML

Junto con los párrafos y los encabezados, las listas son uno de los elementos HTML más comúnmente usados en las páginas web. Tras este capítulo sabremos, no sólo crear listas, sino crear varios estilos de estas, una para cada ocasión. Y no tardaremos en ponerlas en práctica ya que al final del capítulo crearemos la segunda página de la cual podemos ver un anticipo en la figura 4.1.

Figura 4.1. En esta figura podemos apreciar la presentación de la segunda práctica del curso que realizaremos al final del capítulo.



Una lista, o listado, es una enumeración de dos o más elementos y suelen disponerse de tal forma que se facilite la distinción entre ellos.

El lenguaje HTML define los siguientes tipos de listas:

- Numerada u ordenada.
- No ordenada.
- Listas de glosario o de definición.
- Menús.
- Usadas con párrafos cortos.
- Listas de directorio.

Pronto sabremos como crear cada una de ellas y cuando es conveniente usar una u otras, y comprobaremos, una vez más, la gran sencillez del lenguaje HTML una vez nos hemos habituado a él.

Etiquetas necesarias para crear listas

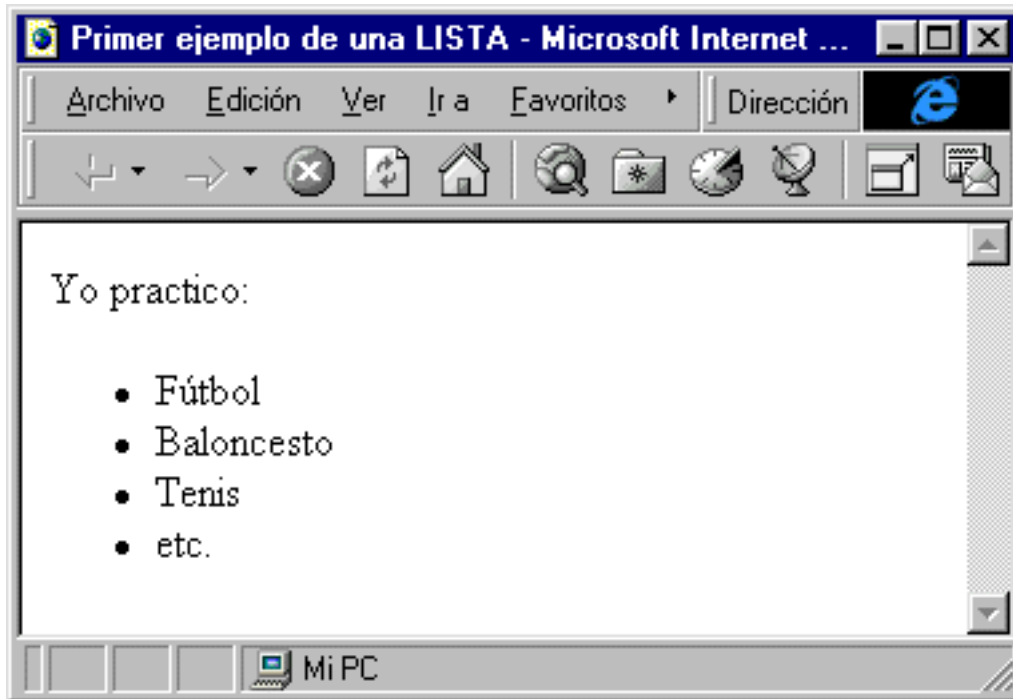
La forma de crear listas difiere ligeramente de los elementos HTML vistos hasta ahora, ya que consta de dos pasos. Primero hay que definir el tipo de lista y después deben insertarse cada uno de los elementos de lista.

Al crear una lista deberemos indicar su inicio y su fin. Para empezar la lista usaremos la instrucción de inicio de la etiqueta de esa lista. Existe una etiqueta para cada tipo distinto de lista, por ejemplo, para crear una lista no ordenada será necesario usar la etiqueta *UL* (su instrucción de inicio será ``). Una vez hemos *abierto* la lista deberemos insertar, como hemos dicho antes, cada uno de los elementos de lista. Para ello usaremos una nueva etiqueta (o dos en algún caso) que dependerá del tipo de lista. En el caso de la lista no ordenada los elementos se introducen con la etiqueta de una única instrucción *LI*, por ejemplo:

```
Yo practico:  
<UL>  
<LI>Fútbol  
<LI>Baloncesto  
<LI>Tenis  
<LI>etc.  
</UL>
```

En la figura 4.2 podemos ver como muestra Internet Explorer™ este código

Figura 4.2. Como aperitivo mostramos una lista no numerada en la que ya se pueden apreciar las características fundamentales que tienen en HTML.



El último paso, que ya hemos anticipado en este ejemplo, será *cerrar* la lista, para lo cual usaremos la instrucción de fin de UL: ``.

Y a todo esto, ¿dónde está la diferencia?. Pues bien, la diferencia es la etiqueta LI (En otras listas se usará también LI o alguna etiqueta similar, como luego veremos). La etiqueta UL no tiene ningún significado por sí sola, necesita a LI. Y por otro lado LI no puede ser usada como una etiqueta independiente sino que debe ir dentro de UL. Por esta razón en ocasiones se suele llamar a LI *subetiqueta*.

Nota

Algunos navegadores permiten el uso de `` fuera de ninguna lista como si se tratara de una etiqueta independiente. Estos navegadores mostrarán el elemento de lista como si fuese perteneciese a una lista no numerada pero sin sangrado. Sin embargo esta forma de usar `` no está admitida por el estándar y por tanto no es recomendable si queremos que todos los navegadores entiendan nuestras intenciones.

Antes de comenzar haremos un comentario más. La etiqueta `` tiene instrucción de fin (``) que delimita el final del elemento de lista actual, pero es opcional usarla o no. Si no se hace se considera que el elemento termina cuando empieza el siguiente o cuando termina la lista. De hecho algunos navegadores al interpretar el código HTML ignoran completamente esta instrucción y sólo entienden que el elemento de lista ha terminado cuando se empieza otro o se acaba la lista.

Vista la teoría nos adentramos en los distintos tipos de listas en las que veremos varios ejemplos prácticos.

Listas no ordenadas

Probablemente el tipo de listas más usados en las páginas web son las listas no ordenadas. Como hemos adelantado antes a modo de ejemplo, este tipo de lista queda delimitada por la etiqueta `...` (UL proviene de

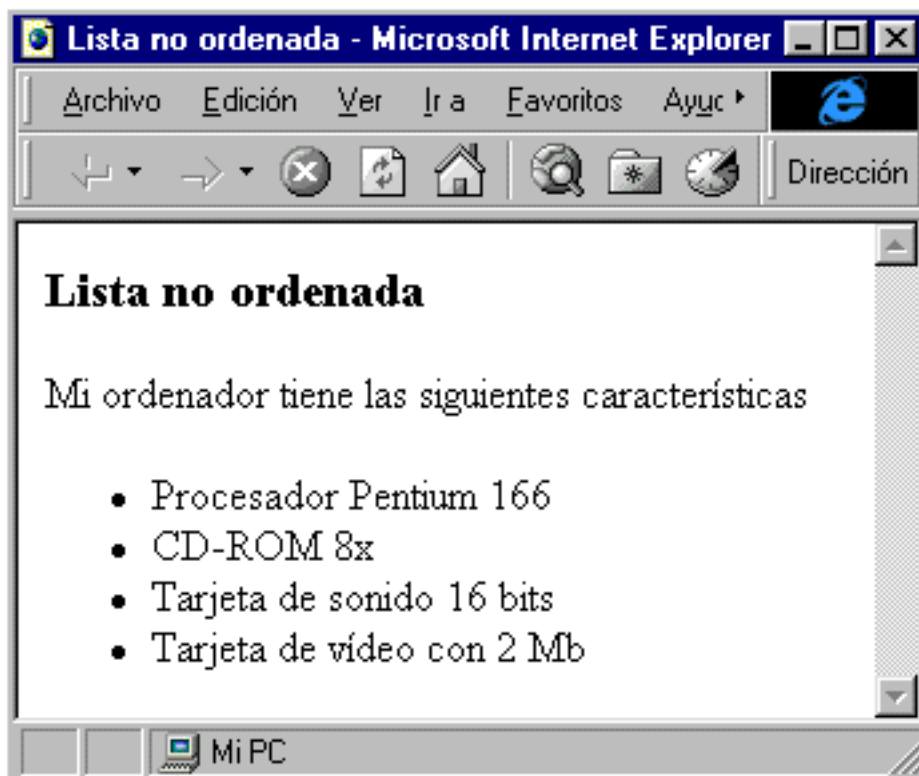
Unordered List, que significa lista no ordenada). Dentro de estas etiquetas insertaremos cada elemento con la etiqueta ``. Veamos un ejemplo ligeramente más completo que el anterior:

Mi ordenador tiene las siguientes características:

```
<UL>
  <LI>Procesador Pentium 166
  <LI>CD-ROM 8x
  <LI>Tarjeta de sonido 16 bits
  <LI>Tarjeta de vídeo con 2 Mb
</UL>
```

En la figura 4.3 podemos ver como muestra esta lista el navegador Explorer™. Vemos que está ligeramente sangrada hacia la derecha y con un símbolo gráfico para distinguir cada elemento de lista, este símbolo recibe el nombre de viñeta.

Figura 4.3. Los elementos de las listas no numeradas van precedidos de una viñeta. En Explorer™ esta viñeta es habitualmente un punto grande.



En la figura 4.4 vemos como a falta de viñetas gráficas Lynx diferencia cada elemento con un asterisco (*).

Figura 4.4. Lista no ordenada en Lynx. Cada uno de los elementos de lista comienza con un asterisco simulando a los símbolos gráfico que usan Netscape™ o Explorer™.

```
Lista no ordenada
```

```
Mi ordenador tiene las siguientes características
* Procesador Pentium 166
* CD-ROM 8x
* Tarjeta de sonido 16 bits
* Tarjeta de vídeo con 2 Mb
```

Listas ordenadas

Este tipo de listas numera los elementos de lista en el orden en el que se han introducido en el código HTML. La numeración se realiza con números empezando por el 1.

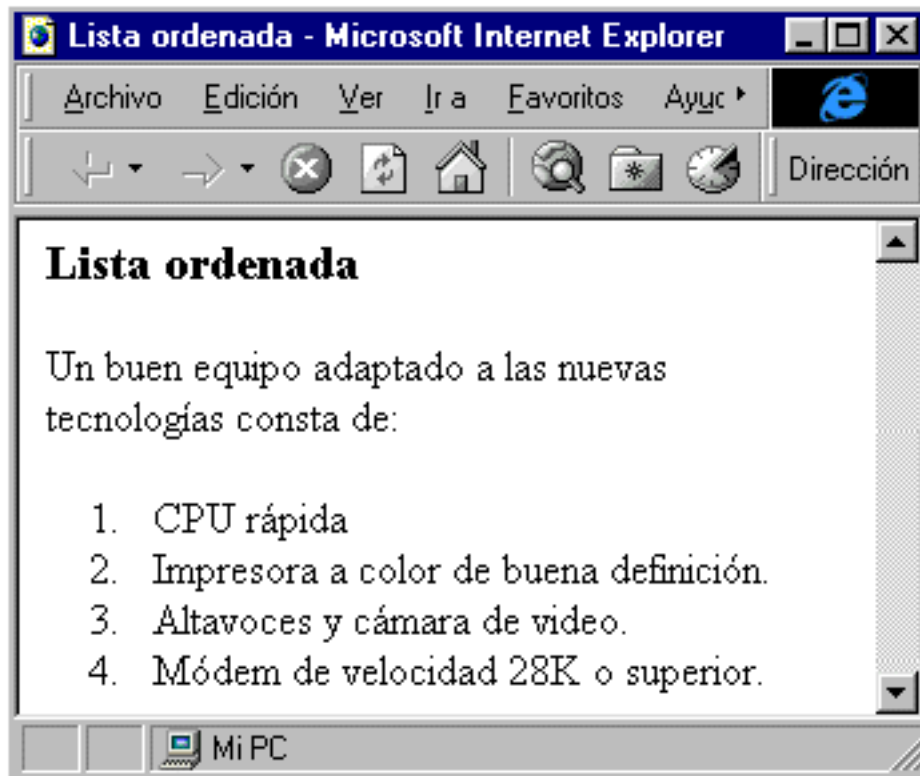
Nota

Las últimas extensiones de HTML permiten elegir entre distintos tipos de numeración para las listas ordenadas. Estas extensiones serán cubiertas en un capítulo posterior.

Este tipo de listas queda delimitado por la etiqueta `...` (OL: Ordered List, Lista ordenada), y cada elemento se inserta, al igual que en el caso de listas no ordenadas, con la etiqueta ``.

En la figura 4.5 vemos un ejemplo de uso de listas ordenadas

Figura 4.5. En las listas ordenadas cada uno de los elementos de lista aparecen numerados y sangrados hacia la derecha.

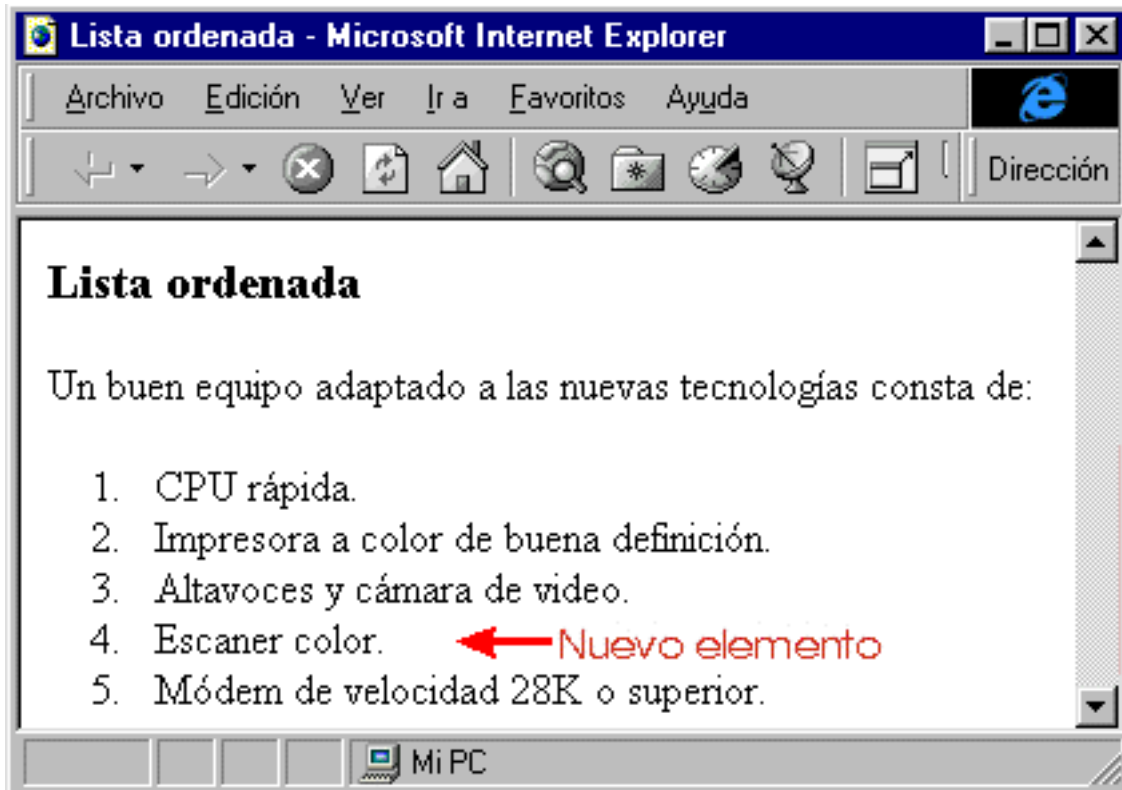


que se corresponde con el siguiente código:

```
Un buen equipo adaptado a las nuevas tecnologías consta de:  
<OL>  
  <LI>CPU rápida  
  <LI>Impresora a color de buena definición.  
  <LI>Altavoces y cámara de vídeo.  
  <LI>Módem de velocidad 28K o superior.  
</OL>
```

El navegador automáticamente numera los elementos de lista. Si en algún momento queremos insertar un nuevo elemento entre los existentes usando, de nuevo, , se realiza automáticamente una reenumeración de los elementos de la lista. Tal y como podemos ver en la figura 4.6.

Figura 4.6. Cuando insertamos un nuevo elemento de lista en una lista ordenada los elementos siguientes son automáticamente reenumerados.



que se corresponde con este código:

Un buen equipo adaptado a las nuevas tecnologías consta de:

```
<OL>
<LI>CPU rápida
<LI>Impresora a color de buena definición.
<LI>Altavoces y cámara de vídeo.
<LI>Escaner color
<LI>Módem de velocidad 28K o superior.
</OL>
```

En el navegador Lynx™ las listas numeradas se muestran como vemos en la figura 4.7.

Figura 4.7. En el caso de las listas numeradas la diferencia entre la visualización con Lynx™ y con Explorer™ es menor.

Lista ordenada

```
Un buen equipo adaptado a las nuevas tecnologías consta d
1. CPU rápida.
2. Impresora a color de buena definición.
3. Altavoces y cámara de video.
4. Escaner color.
5. Módem de velocidad 28K o superior.
```

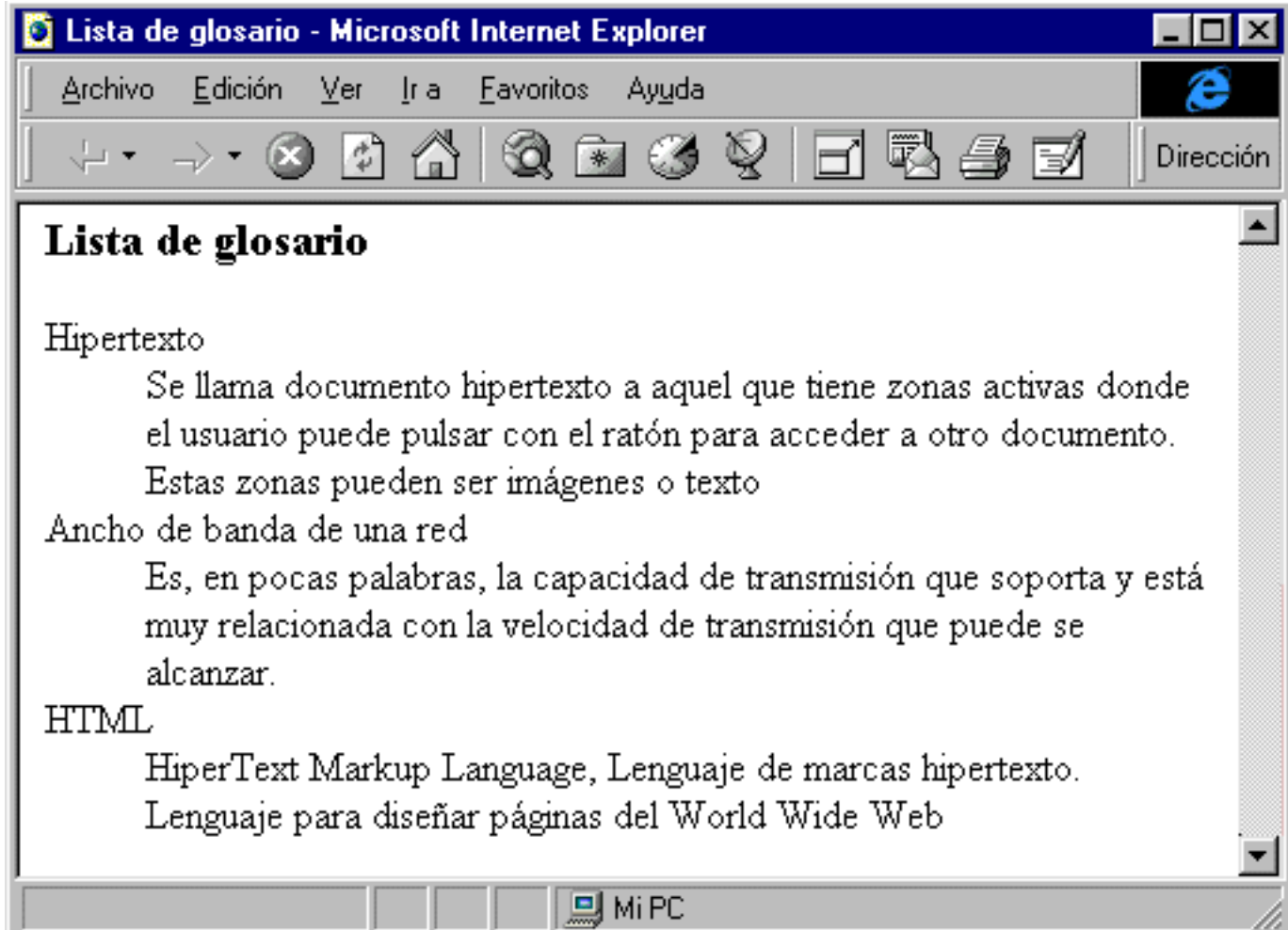
Listas de glosario

Este tipo de listas es usado, como su nombre indica, para crear glosarios, es decir listas de definiciones o de descripciones. El principio y final de este tipo de listas se delimitan con la etiqueta `<DL>...</DL >` (DL: Definition List, Lista de definición). La inserción de elemento es ligeramente distinta a la del resto de listas ya que no se usa la etiqueta `<>`. Esto es así porque en este caso cada uno de los elementos de lista está compuesta de dos partes:

- *La palabra que queremos definir:* para insertar la palabra usaremos la etiqueta `<DT>` que proviene de Definition Term (Término de definición). Esta nueva etiqueta funciona de forma muy similar a `` y la instrucción de fin (`</DT>`) es igualmente optativa.
- *La definición o descripción de esa palabra:* una vez indicada la palabra es hora de poner su definición. Para ello usamos otra nueva etiqueta: `<DD>` cuyas siglas significan Definition Descripción (parte de la definición que contiene la descripción). La definición suele ser un párrafo y será mostrado por los navegadores sangrado con respecto al texto anterior.

En la figura 4.8 podemos ver un ejemplo de listas de glosario vistas con Internet Explorer™.

Figura 4.8. Las listas de glosario constan de dos partes la palabra de definición y la propia de definición que es mostrada como un nuevo párrafo con sangrado.



En la figura 4.9 se muestra la misma vista con el navegador Lynx.

Figura 4.9. Listas de glosario en Lynx. De nuevo las diferencias respecto a la visualización con Explorer™ son escasas.

Lista de glosario

Hipertexto

Se llama documento hipertexto a aquel que tiene zonas activas donde el usuario puede pulsar con el ratón para acceder a otro documento. Estas zonas pueden ser imágenes o texto.

Ancho de banda de una red

Es, en pocas palabras, la capacidad de transmisión que soporta y está muy relacionada con la velocidad de transmisión que se puede alcanzar.

HTML

HiperText Markup Language, Lenguaje de marcas hipertexto. Lenguaje para diseñar páginas del World Wide Web.

En ambos ejemplos el código usado es el siguiente:

```
<DL>
  <DT>Hipertexto
  <DD>Se llama documento hipertexto a aquel
    que tiene zonas activas donde el
    usuario puede pulsar con el ratón
    para acceder a otro documento. Estas
    zonas pueden ser imágenes o texto.
  <DT>Ancho de banda de una red
  <DD>Es, en pocas palabras, la capacidad
    de transmisión que soporta y está
    muy relacionada con la velocidad de
    transmisión que se puede alcanzar.
  <DT>HTML
  <DD>HiperText Markup Language, Lenguaje
    de marcas hipertexto. Lenguaje para
    diseñar páginas del World Wide Web.
</DL>
```

Directorios y menús

Las listas vistas hasta ahora son, sin duda, las más usadas, pero existen otros dos tipos que están algo más olvidadas por los creadores de webs: las listas de directorio y las listas de menú.

Las listas de menú son usadas cuando los elementos de lista son frases de una sola línea mientras que las listas de directorio fueron creadas para ser usadas cuando son términos cortos que pueden ser mostrados en múltiples columnas.

Nota

A pesar de las recomendaciones del estándar HTML los principales navegadores muestran las listas de directorio y menús de igual manera (o muy similar) a las listas no ordenadas. Por esta razón es aconsejable usar únicamente los tres primeros tipos de lista vistos.

A continuación mostramos un ejemplo de listas de menú y listas de directorio. En la figura 4.10 podemos ver dos ejemplo ambos tipos en Explorer™ y en la figura 4.11 el mismo ejemplo tal y como se vería en Lynx™.

Figura 4.10. En Explorer™ las listas de menú (<MENU>) y de directorio (<DIR>) son mostradas de igual manera que las listas no numeradas ().

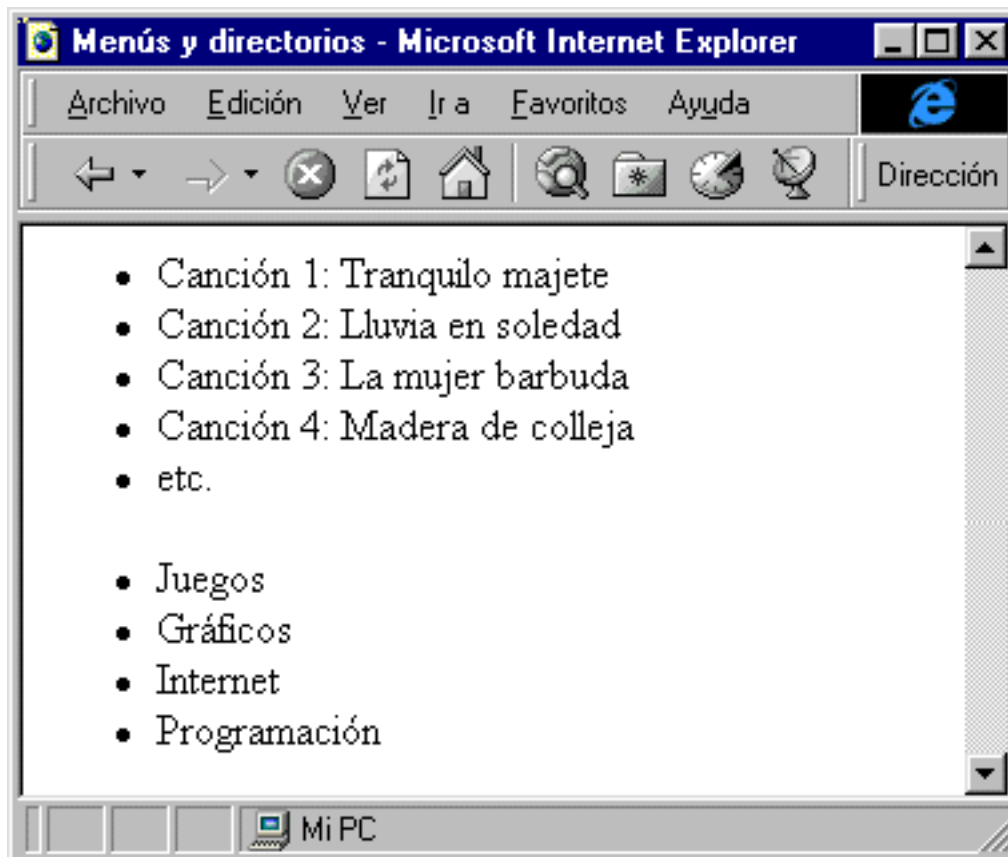


Figura 4.11. En Lynx™ las listas de menú y de directorio son mostradas simplemente insertando cada elemento de lista en una nueva línea.

```
Canción 1: Tranquilo majete
Canción 2: Lluvia en soledad
Canción 3: La mujer barbuda
Canción 4: Madera de colleja
etc.

Juegos
Gráficos
Internet
Programación
```

El código usado es:

```
<MENU>
<LI>Canción 1: Tranquilo majete
<LI>Canción 2: Lluvia en soledad
<LI>Canción 3: La mujer barbuda
<LI>Canción 4: Madera de colleja
<LI>etc.
</MENU>
<DIR>
<LI>Juegos
<LI>Gráficos
<LI>Internet
<LI>Programación
</DIR>
```

Anidamiento de listas

Vistos todos los tipos de listas se nos puede plantear una pregunta: ¿Se puede incluir una lista dentro de otra?. La respuesta es SI. El lenguaje HTML permite insertar unas listas dentro de otras ya sean del mismo o de distinto tipo. A esta posibilidad se le llama anidamiento de listas y la forma de realizarlo es incluir todo el código de una lista dentro de uno de los elementos de la otra. En el siguiente ejemplo combinaremos algunos de los ejemplos anteriores para dar lugar a un listado con varios listas anidadas unas dentro de otras:

Los temas del número de este mes de nuestra revista son:

```
<MENU>
<LI>Un nuevo Internet: IPv6
<LI>El WWW cobra vida: ha llegado el
  HTML dinámico
<LI>Qué ordenador comprarse:
  <OL>
  <LI>CPU rápida. Con las siguientes
    características:
```

```
<UL>
  <LI>Procesador Pentium 166
<LI>CD-ROM 8x
<LI>Tarjeta de sonido 16 bits
<LI>Tarjeta de vídeo con 2 Mb
  </UL>
  <LI>Impresora a color de buena
    definición.
  <LI>Altavoces y cámara de vídeo.
  <LI>Escaner color.
  <LI>Módem de velocidad 28K o
    superior.
</OL>
</MENU>
```

En las figuras 4.12 y 4.13 podemos ver como se muestra este ejemplo en los navegadores Explorer™ y Netscape Navigator™. Aunque hay ligeras diferencias, en ambos se puede observar cómo cada nuevo nivel de anidamiento se produce un sangrado en la lista. Además se observa que en la segunda lista no numerada la viñeta que precede cada elemento ha cambiado, lo que ayuda a diferenciar entre ambas listas.

Figura 4.12. Gracias al anidamiento podemos crear listas muy complejas en HTML. El segundo nivel de lista no numerada usa una viñeta distinta.

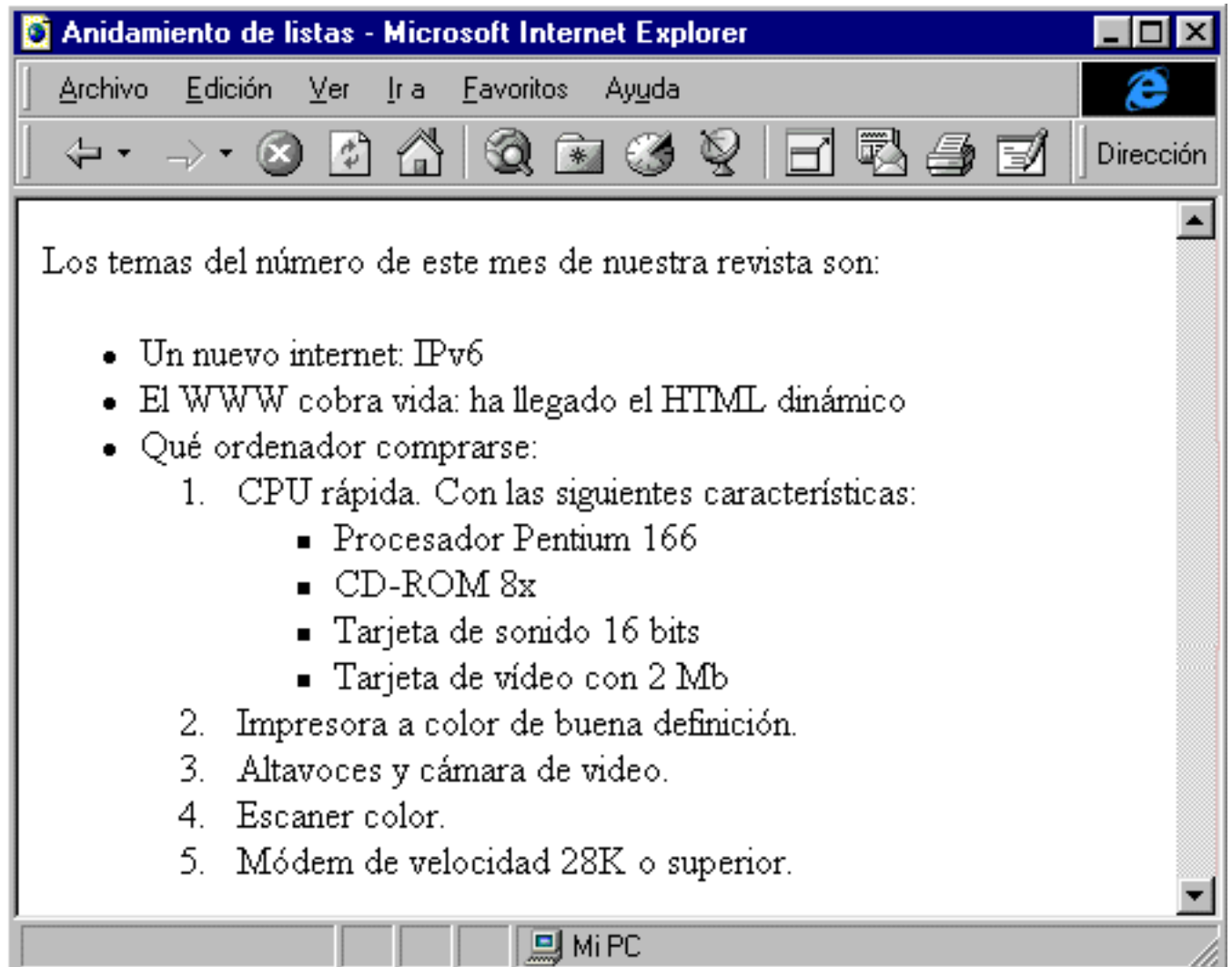
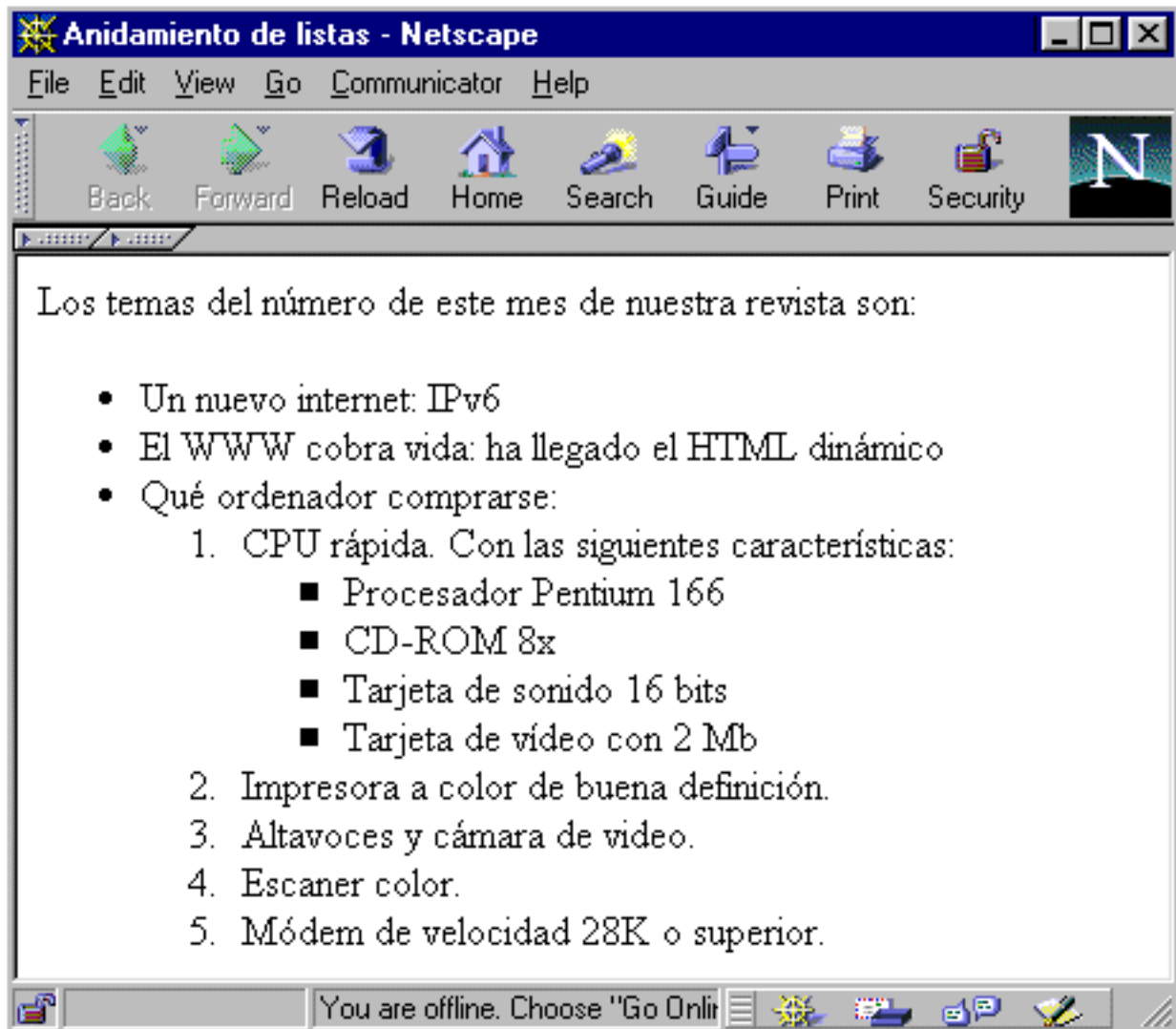


Figura 4.13. Comparando esta figura con la 4.8 podemos ver las ligeras diferencias que existen en los distintos tipos de listas entre Netscape™ y Explorer™.



Comentarios en HTML

Se llama comentario en el ámbito de HTML las notas que el autor o autores ponen en el código para facilitar su entendimiento. Estos comentarios no son mostrados por el navegador y por tanto sólo serán visibles al leer el código HTML de la página web.

En general es recomendable ir insertando comentarios al crear una página para marcar determinadas partes y así encontrarlas más fácilmente. Algunos usos que suelen darse a los comentarios son:

- *Notas para recordar detalles del código* la próxima vez que vayamos a cambiarlo, como por ejemplo para indicar por qué hemos usado una etiqueta y no otra, por qué hemos usado una lista numerada y no una no numerada, etc. Este tipo de comentarios son muy usados cuando las páginas son complejas.

- *Apuntar que queda por hacer* en una determinada sección o como es conveniente cambiarla. O bien para indicar el comienzo o fin de una determinada sección de la página.
- Para identificar fácilmente partes importantes del código o aquellas que cambian más a menudo.
- *Usos particulares de cada webmaster*. De hecho los comentarios pueden usarse para cualquier cosa y cada programador de páginas web tiene su propio modo de usarlos.

Creación de comentarios en HTML

Para crear un comentario no se usa una etiqueta, aunque es una estructura parecida. En primer lugar ponemos una cadena que indica el comienzo del comentario: `<!--`, esto es, el símbolo *menor que*, seguido del símbolo fin de exclamación y de dos guiones, todo ello sin espacios entre ellos. Todo el texto que le siga será parte de comentario, que terminará cuando insertemos la cadena de fin: `-->`, dos guiones y el símbolo *mayor que*. La estructura de un comentario es por tanto:

```
<!--Esto es un comentario-->
```

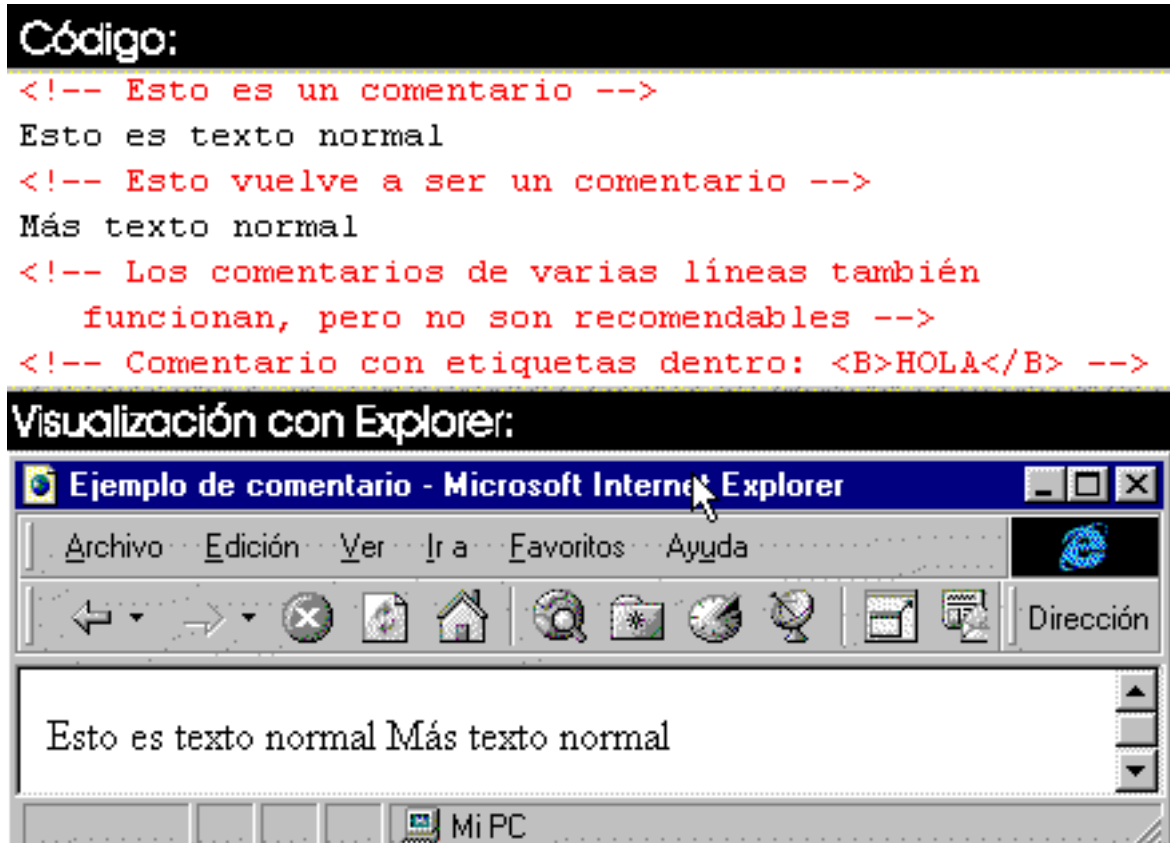
Suele ser recomendable dejar un espacio entre ambas cadenas y el texto anterior y posterior, tal y como acabamos de mostrar.

Veamos algunos ejemplos prácticos de comentarios:

```
<!--Aquí comienza el cuerpo de la página-->  
<!--Cambiar este párrafo para que se entienda mejor-->  
<!--Debería añadir más enlaces en esta página-->
```

En la figura 4.14 podemos ver un ejemplo de un comentario junto con otro código HTML y el resultado de visualizar ese código con el navegador. Como observamos el navegador ignora los contenidos del interior de los comentarios.

Figura 4.14. Al mostrar la página los navegadores actúan como si los comentarios no existieran. Como vemos Explorer™ permite comentarios multilínea y con etiquetas dentro.



Para terminar sólo queda hacer una aclaración. Tal y como hemos dicho todo el texto entre los símbolos "`<!--`" y "`-->`" es un comentario sea lo que sea y tenga el número de líneas que tenga. Sin embargo el estándar de HTML recomienda que los comentarios se limiten a una sola línea, y si debe ocupar varias necesariamente se aconseja incluir los símbolos de comentario en cada una de ellas.

Práctica 2: Un sitio web completo.

Ya podemos dar por concluida la primera fase del aprendizaje del HTML. En poco tiempo hemos aprendido:

- Como es la estructura general de una página: para crear esta estructura usábamos las etiquetas HTML, HEAD y BODY.
- Como dar título a la página con la etiqueta TITLE.
- Como crear encabezados con las etiquetas H1, H2, H3, H4, H5, H6.
- Como introducir párrafos de texto normal con la etiqueta P y alinearlos a nuestro gusto.
- Como crear otros párrafos con BLOCKQUOTE y ADDRESS, y en este mismo capítulo párrafos preformateados con PRE.

- Como dividir el texto con saltos de línea o líneas horizontales de distintos grosores y tamaños.
- También hemos aprendido a insertar en nuestras páginas los dos elementos que más fama han dado al World Wide Web: los enlaces hipertexto y las imágenes.
- Vimos también como dar distintos formatos al texto, tanto formatos lógicos (EM, STRONG, CODE, etc.) como físicos (B, I, TT, BIG, SMALL, SUB, SUP, etc.).
- Y por último en este breve resumen hemos de mencionar la creación, recién aprendida de diversos tipos de listas (UL, OL, DL, MENU y DIR)

Como vemos la lista es bien larga y ya estamos equipados con conocimientos suficientes para crear una página compleja.

Para afianzar estas bases lo más importante es la práctica, y para no quedarnos sólo en las palabras vamos a empezar ya con la segunda práctica del curso. Ésta consistirá en la realización de un *sitio web completo* que, aunque es sencillo (dos páginas), permitirá al lector hacerse una idea de los procesos que hay que llevar a cabo en la elaboración de webs en el mundo profesional actual.

El sitio web que crearemos puede ser o bien de una empresa de nuestra propiedad, o de una que ha contratado nuestros servicios. Hemos elegido como ejemplo una agencia de viajes, pero invitamos al lector a buscar otro tema, ficticio o real, y crear su propia página web, si así lo prefiere.

A lo largo de la creación de la página iremos detallando los pasos seguidos para que el lector pueda ir repitiéndola o creando otra similar en su ordenador.

Nota

En el archivo `practica-2.tgz` [extra/practica-2.tgz] se incluye el sitio web completo listo para ser visualizado y en él puede observarse el código completo. Los documentos HTML son los llamados `index.html` y `mes.html`. Las imágenes necesarias para la realización de las páginas se encuentran en el subdirectorio `img`.

Planificación

El primer paso en la creación de una página es la planificación. Si estamos creando el web de nuestra propia empresa deberemos llevarla a cabo por nosotros mismos (en nuestro caso ficticio también tendremos que realizarla nosotros), pero si creáramos el web para otra empresa deberíamos encargarle ciertos *deberes*: presentación de la empresa, presentación de su catálogo, promociones especiales, etc. Tal vez ya tenga un proyecto publicitario en marcha y quiere que el web este en línea con este.

En nuestro caso, por no aburrir al lector, no entraremos demasiado en este tipo de aspectos del diseño de páginas, pero queremos resaltar la importancia que tiene a la hora de conseguir éxito con las páginas que hemos creado.

Secciones de la página

Una vez conocidos todos los requisitos que debe cumplir el web es hora de hacer el primer borrador. En él debemos incluir las distintas secciones que debe tener el sitio web.

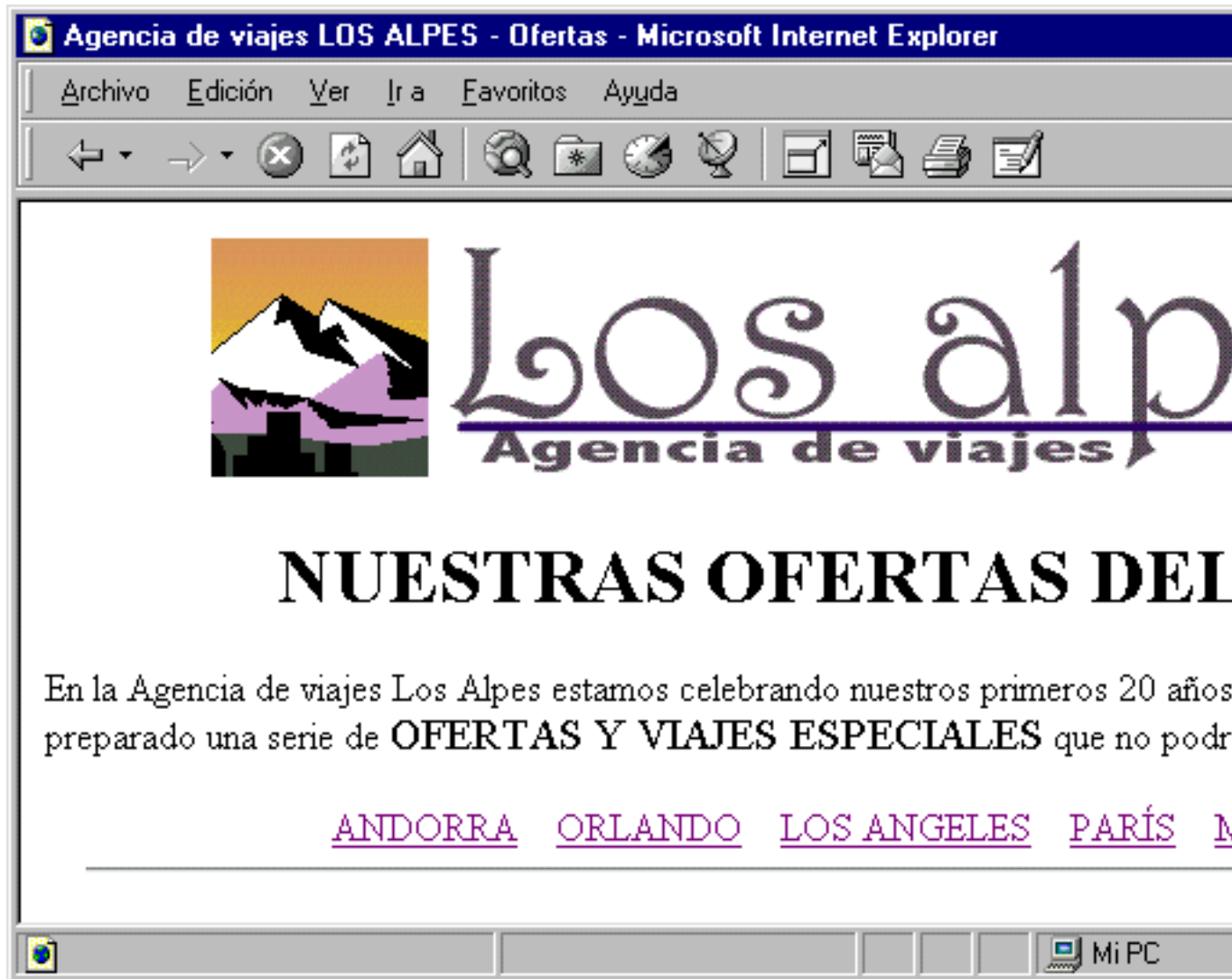
De ahora en adelante nos referiremos al conjunto de documentos HTML que estamos creando como sitio web, *website* en inglés, y llamaremos página o página web a cada uno de esos documentos.

A la hora de realizar este primer borrador hay que tener en consideración varios aspectos. La información de la página debe tener una estructura jerárquica, con una página principal con enlaces a las distintas secciones. En esta

práctica hemos decidido dividir la información en cuatro secciones y hemos creado una pantalla de presentación que nos permita acceder a ellas:

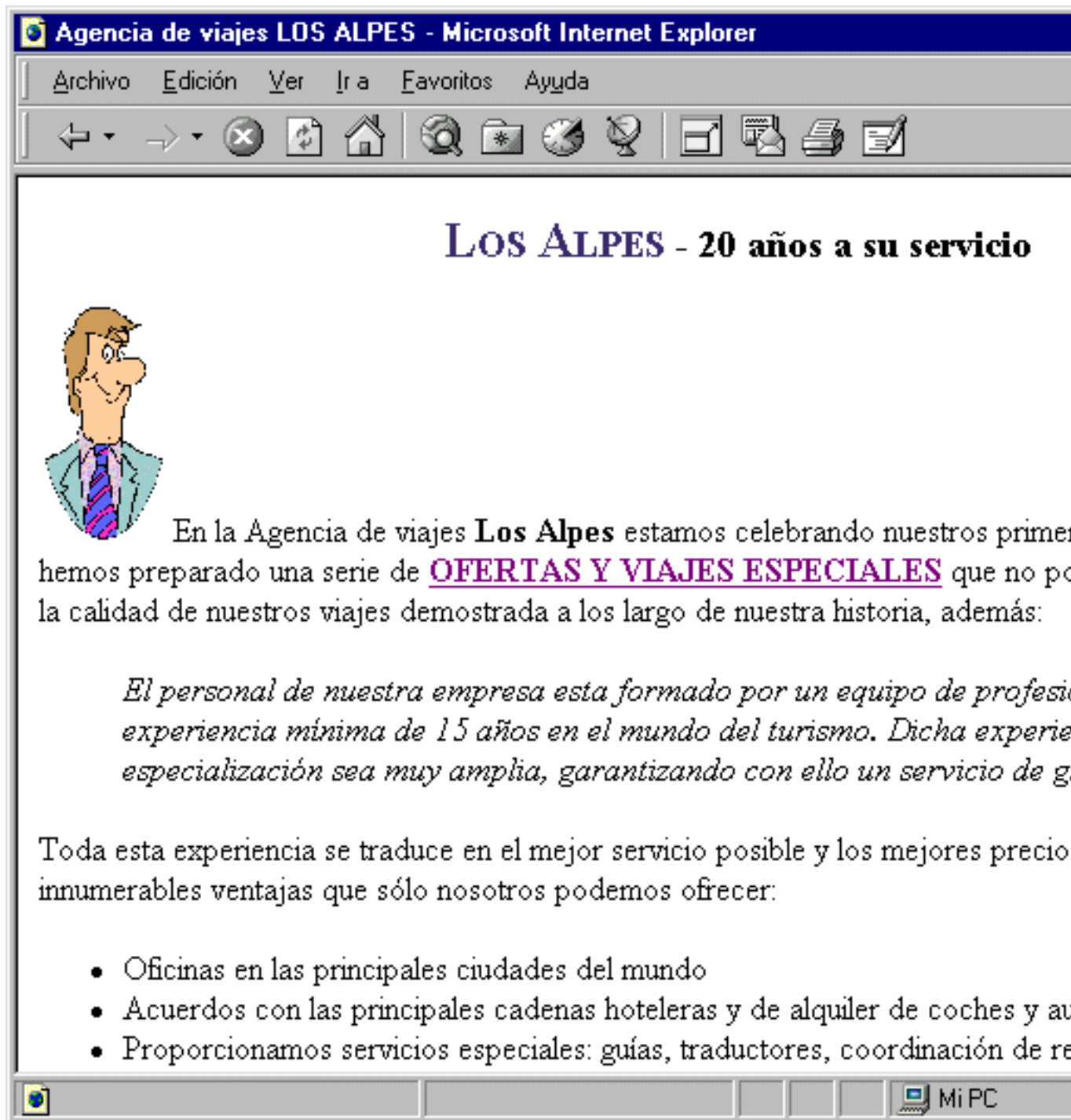
- *Pantalla de presentación*: Es lo primero que se verá al visitar la página. En ella debe verse el nombre de la agencia así como un menú con las opciones que presenta la página. Podríamos pensar en poner el titular con un encabezado de tipo H1, pero para crear un mayor impacto estético crearemos un titular gráfico e incluiremos el logotipo (ficticio) de la agencia a su izquierda. Después incluiremos un menú con enlaces a las diferentes secciones y por último alguna foto de lugares donde se pueda viajar con la agencia 'Los Alpes'. En la figura 4.1 al comienzo de este capítulo podemos ver esta pantalla de presentación.
- *Nuestras principales OFERTAS del mes*: En esta sección informaremos de los viajes que la agencia de viajes 'Los Alpes' oferta a precios especiales en el mes actual. Esta sección tiene una longitud considerable, por lo que a pesar de la sencillez de este web decidimos crearla en un documento aparte, al que llamaremos `mes.html`. En la figura 4.15 vemos la presentación de esta página que como vemos guarda una armonía con la presentación anterior.

Figura 4.15. Pantalla de presentación de la segunda página web de nuestro *website*. Es importante mantener un parecido entre todas las páginas que lo componen.



- *Los Alpes 20 años a su servicio*: En esta sección daremos información de la empresa. Esta sección es típica en todas las páginas del WWW y suele denominarse la página del *Quién Soy/Quienes somos*. Podemos apreciar esta sección en la figura 4.16

Figura 4.16. En la sección 'Los Alpes - 20 años a su servicio' podemos apreciar el uso de la etiqueta BLOCKQUOTE y de una lista no ordenada.



- *Viajes a su medida:* Uno de los puntos fuertes que pretende promocionar nuestra empresa es la creación de viajes a la medida del cliente. Si eres joven te hará descuentos especiales, los ejecutivos no tendrán que soportar tiempos de espera innecesarios,... Debemos dar constancia de ello en el web y por eso le dedicamos una sección entera. En la figura de la página completa podremos apreciar esta sección y la siguiente.

- *Dónde encontrarnos*: Esta es otra sección que debe existir obligatoriamente en las páginas de toda empresa que tenga oficinas de servicio al cliente. El objetivo de la página será abarcar una mayor cantidad de público, pero también se pretenderá que los clientes se pongan en contacto con la agencia personalmente, por ello es importante resaltar bien sus direcciones, teléfonos, fax, etc.

Las tres últimas secciones serán demasiado breves por lo que estarán incluidas en la página principal a la que llamaremos, como es común, *index.html*. Al principio de cada una de estas secciones incluiremos un ancla de manera que podamos referirnos a ellas posteriormente usando enlaces a anclas.

Una vez hecho el boceto de borrador hay que ponerse manos a la obra y diseñarlo el aspecto gráfico y realizar la programación HTML.

Diseño gráfico

Este es un aspecto fundamental al crear páginas web. En función de la apariencia de nuestro sitio web conseguiremos o no que los futuros visitantes se queden en él o incluso que lo visiten con regularidad. La presentación gráfica deberá ser lo más atractiva posible, pero sin que la página esté demasiado sobrecargada. En el siguiente capítulo trataremos a fondo el diseño gráfico y todas las posibilidades avanzadas que el lenguaje HTML nos ofrece a este respecto.

Los aspectos gráficos que más destacan en nuestra página son los que pertenecen a la pantalla de presentación entre estos destacan el LOGO de la página que vemos reproducido en la figura 4.17

Figura 4.17. Logo de los alpes.



y el título de esta, que vemos en la figura 4.18.

Figura 4.18. Letrero de 'Los Alpes' que, junto con el logo, aparecerá en todas las páginas del sitio web.



Estas dos imágenes han sido creadas con un programa de dibujo vectorial, como pueden ser Corel Draw™, Freehand™ Kontour™, etc. y retocadas con un programa de retoque fotográfico como pueden ser Photofinish™, PhotoShop™, The GIMP™ o Paint Shop Pro™ (Shareware).

Las fotos inferiores de la pantalla de presentación han sido obtenidas de alguno de los numerosos cliparts de imágenes gratuitos de los cuales podemos encontrar en Internet una buena muestra.

Programación en HTML

Una vez creadas las imágenes es hora de usarlas. El primer paso que seguiremos consistirá en la creación de una plantilla que servirá como modelo para cada uno de las páginas.

Es importante que las páginas de un sitio web sean similares o, dicho de otra forma, que tengan uniformidad. Eso le aportará una apariencia profesional y permitirá al visitante, por ejemplo, identificar las páginas como de Los Alpes y no de ninguna otra agencia.

La plantilla estará compuesta por el logo, el letrero de "Los Alpes Agencia de viajes" y de una firma de los creadores de la página. El código de esta plantilla es:

```
<HTML>

  <HEAD>
    <TITLE>
      Agencia de viajes LOS ALPES
    </TITLE>
  </HEAD>

  <BODY>
    <A NAME="arriba">

      <CENTER>
        <IMG SRC="img/logo.gif">
        <IMG SRC="img/losalpes.gif">
      </CENTER>
      Contenido de la página
      <HR>

      <ADDRESS>
        Web realizado por el equipo de
        <FONT COLOR="#800000">
          Conquista la web
        </FONT>
      <BR>
```

```
    Última actualización: Enero de 2001
  </ADDRESS>
</BODY>

</HTML>
```

Una vez tenemos hecha esta plantilla la guardamos con el nombre `plantilla.html` para utilizarla posteriormente con todos los documentos del sitio web.

El siguiente paso es programar la presentación, esta consta de un eslogan, una lista de enlaces a las secciones de la página y dos imágenes atractivas, todo ello centrado:

```
<CENTER>

<UL>

  <LI>
  <A HREF="mes.html">
  Nuestras principales OFERTAS del mes
  </A>

  <LI>
  <A HREF="#alpes">
  LOS ALPES - 20 años a su servicio
  </A>

  <LI>
  <A HREF="#medida">
  Viajes a su medida
  </A>

  <LI>
  <A HREF="#donde">
  Dónde encontrarnos
  </A>

</UL>

<IMG SRC="img/ski.gif">
<IMG SRC="img/montanas.gif">

</CENTER>
```

Como vemos el primer link nos enlaza con el archivo `mes.html` mientras que el resto son enlaces a las anclas creadas al principio de cada sección en el mismo `index.html`.

El resto de detalles no entrañan demasiada dificultad y el lector que haya seguido el curso desde el principio será capaz de realizar la página completa sin problemas fijándose en las figuras incluidas.

Como pista listaremos el resto de etiquetas y atributos (entre paréntesis) usados en la página:

```
HR (con WIDTH y SIZE), H3, BIG, STRONG, BLOCKQUOTE, CITE,
P (ALIGN), OL y PRE.
```

Imágenes y color en el WWW

Jorge Ferrer
Rodrigo García
Victor García
Versión 1.0

Copyright Jorge Ferrer, Víctor García y Rodrigo García. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation. Puede consultar una copia de la licencia en: <http://www.gnu.org/copyleft/fdl.html>

En este artículo se presenta todo el potencial que ofrece el lenguaje HTML para dar color y usar imágenes en una página web. Aprenderemos cómo especificar un determinado color de entre 16 millones posibles y ponerlo como fondo de la página, o como color del texto. También veremos como manipular imágenes cambiando su tamaño o poniendo texto a su alrededor. Después se verá como controlar los bordes y tamaño de las imágenes. Veremos como poner un texto alternativo a la imagen, crear imágenes con varias zonas activas (imágenes mapa) y cómo emplear GIFs animados. Al terminar seremos unos expertos en el arte de las imágenes y el color en el WWW.

Hasta ahora hemos aprendido a crear páginas WEB compuestas por texto, encabezados, diferentes estilos, e incluso a enlazar varias páginas entre sí. También hemos aprendido a insertar imágenes de una manera sencilla en una página. Sin embargo todavía podemos aprender mucho más aprovechar todas las capacidades que nos ofrece el lenguaje HTML para convertir nuestra página en una obra de arte gráfico.

El éxito o fracaso de un Web depende en muchas ocasiones del aspecto gráfico que ofrezca a los visitantes. Un buen ejemplo de ello lo observamos en la página de Disney

Figura 5.1. Aprovechando las capacidades del HTML para incluir colores e imágenes en un Web podemos conseguir resultados tan atractivos como el que observamos en el web

de Disney.

Disney.com -- The Web Site for Families - Microsoft Internet Explorer

Archivo Edición Ver Ir a Favoritos Ayuda

Dirección <http://www.disney.com/>

Disney Adventures The Magazine for Families

Disney.com THE WEB SITE FOR FAMILIES

Disney's Daily BLAST
THE ONLINE SERVICE FOR KIDS

READ WITH POOH TODAY!
TRY IT FREE!

RADIO Disney
Check it out!

@Disney
Go for the Goalposts!
Tony Danza is an unlikely [football hero](#) on "Wonderful World of Disney"

KIDS
Winter Sports Sweepstakes

Beauty and the Beach

SHO
New Mini Bean Bags

HIGHLIGHTS FOR FEBRUARY 15, 1998

TODAY @DISNEY: GO FOR THE GOALPOSTS!

http://www.disney.com/ddc_media/homepage/ Zona

, que es una de las más visitadas del WWW.

El diseño con programas de creación de imágenes y de retoque fotográfico está fuertemente ligado a la creación de páginas Web. Más aún cuando nuestro objetivo será, no sólo crear una página atractiva, sino de calidad técnica y que cargue en poco tiempo para no provocar esperas excesivas a los visitantes. La manera de conseguir este objetivo será un buen conocimiento del lenguaje HTML y de todas sus posibilidades, y por ello dedicaremos un buen número de páginas a este tema, empezando por los siguientes temas:

- Imágenes y colores de fondo: Extensiones de la etiqueta <BODY>
- Imágenes inline: La etiqueta
- Alineamiento de imágenes.
- Texto que envuelve a las imágenes
- Cómo controlar el tamaño de las imágenes

Usando color en el WWW.

Existen diversas etiquetas y atributos en HTML que permiten cambiar el color de los elementos de una página o incluso el color del fondo de esta. Pero antes de ver estas etiquetas es necesario explicar algunas cosas sobre como usar el color en las páginas Web.

En el tercer capítulo ya adelantamos, cuando aprendimos a dar color al texto, que existían dos métodos para indicar el color, estos métodos son:

- Usar un número hexadecimal que identifica a ese color, por ejemplo #FF0000 se refiere al color rojo.
- Usar uno de los nombres (en inglés) *predeterminados* por los navegadores para los colores más usados, por ejemplo para poner, al igual que antes, el color rojo usaríamos: *'red'*.

Por supuesto para el programador es mucho más sencillo aprenderse los nombres de los colores en inglés que no un código hexadecimal que nos resulta mucho más extraño. Sin embargo usar nombre de colores tiene algunas limitaciones. Por un lado sólo abarca unos pocos colores y por otro tiene el inconveniente de que no todos los navegadores son capaces de entenderlos, mientras que sí entienden el código hexadecimal. Aun así podemos encontrar 16 colores que son entendidos por gran parte de los navegadores y por supuesto también Netscape Navigator™ y Explorer™ que son los más usados.

En la tabla 5.1 podemos encontrar estos dieciséis colores junto con su nombre en inglés y el código hexadecimal asociado.

Figura tabla 5.1. Colores usados más frecuentemente

Negro	Black	#000000
Blanco	White	#FFFFFF
Verde	Green	#008000
Marron rojizo	Maroon	#800000
Verde oliva	Olive	#808000
Azul marino	Navy	#000080
Violeta	Purple	#800080
Gris oscuro	Gray	#808080
Rojo	Red	#FF0000
Amarillo	Yellow	#FFFF00
Azul	Blue	#0000FF
Azul verdoso	Teal	#008080
Verde lima	Lime	#00FF00
Azul claro	Aqua	#00FFFF
Fusia	Fuchsia	#FF00FF
Gris claro (plateado)	Silver	#C0C0C0

Aunque los nombres asignados a estos 16 colores son ampliamente aceptados en realidad existen muchos más, pero su uso no es recomendado ya que sólo Netscape los soporta. Entre estos otros colores están *indigo*, *ivory*, *khaki* y muchos más.

El segundo método, basado en usar el código hexadecimal asociado al color no es tan complicado como puede parecer en un principio. Y tiene dos grandes ventajas sobre usar nombres:

- Lo entienden *todos* los navegadores.
- Puedes indicar más de *16 millones* de colores.

Para construir el código de un color se usa un método que podemos comparar con el de un pintor con una paleta de tan sólo tiene 3 colores y los mezcla para conseguir el resto de los que necesita para pintar un cuadro. En el código usado por el lenguaje HTML estos tres colores son el rojo, el verde y el azul, es decir, se usa la paleta RGB (Red, Green, Blue). Para crear todos los colores se usan mezclas de estos en distintas cantidades. Estas cantidades se indican con un número del 0 al 255, correspondiendo el 255 a la máxima cantidad de color posible. Si ponemos cero de todos los colores obtendremos el color negro, si ponemos 255 de todos obtendremos el blanco. Para crear amarillo, por ejemplo, podemos mezclar 255 de rojo, 255 de verde y nada de azul; y si en vez de poner 255 de rojo y verde ponemos 230 de cada obtendremos un amarillo más oscuro.

Sin embargo todavía queda una complicación y es que el número tendremos que ponerlo en hexadecimal.

Nota

Un código hexadecimal está basada en una numeración que está compuesta por 16 símbolos en lugar de 10 como el sistema decimal. Estos 16 símbolos son 1, 2, 3, 4, 5, 6, 7, 8, 9, A (10), B (11), C (12), D (13), E (14) y F (15). Por ejemplo, el siguiente número, el 16, se escribe en hexadecimal: '10', mientras que el número hexadecimal '1F' se corresponde con el 31 decimal.

Usando el código hexadecimal podemos indicar la cantidad de color del 0 al 255 con sólo dos dígitos (del 0 al FF), y una vez que nos hemos acostumbrado no resulta tan complicado.

Nota

Afortunadamente existen editores HTML que ponen automáticamente el número hexadecimal de los colores. También existen programas independientes como HTMLib Colour Wizard™ que permite indicar color en decimal o incluso seleccionarlo de una paleta y él te genera el código HTML.

Para proseguir vamos a fijarnos en los códigos que adelantábamos en la tabla 5.1, como vemos todos empiezan por el símbolo sostenido o almohadilla: '#'. Posteriormente siguen 6 números (siempre deben ser seis) en hexadecimal. Los dos primeros corresponden a la cantidad de rojo (RR), los siguientes a la cantidad de verde (GG) y los siguientes a la cantidad de azul (BB). Es decir la estructura general es:

```
#RRGGBB
```

De ahora en adelante usaremos la cadena '#RRGGBB' para indicar un código de color en hexadecimal y debemos tener en cuenta que puede ser sustituido también por un nombre de color.

Examinemos alguno de los códigos anteriores, por ejemplo el usado para obtener verde azulado (Teal). Este código indica que el color no tiene nada (0) de rojo, 128 unidades (80 en hexadecimal) de verde y otras tantas de azul. Con lo que obtenemos el tono deseado.

Cómo cambiar la apariencia de una página.

Una vez conocemos a la perfección la manera de insertar colores ya podemos empezar a aprovechar las posibilidades que nos ofrece el lenguaje HTML para controlar la apariencia de una página Web. Como primera aplicación práctica aprenderemos a cambiar el color del fondo, luego cambiaremos el color del texto y por último crearemos el efecto más impactante de todos, pondremos un imagen de fondo.

Color de fondo.

Para cambiar el color de fondo de nuestras páginas usaremos un nuevo atributo de la, ya conocida, etiqueta BODY.

El nuevo atributo se llama *BGCOLOR* y debe ser añadido a la etiqueta BODY existente:

```
<body bgcolor="#RRGGBB">
```

Por ejemplo para crear una página con el fondo de color azul podemos usar:

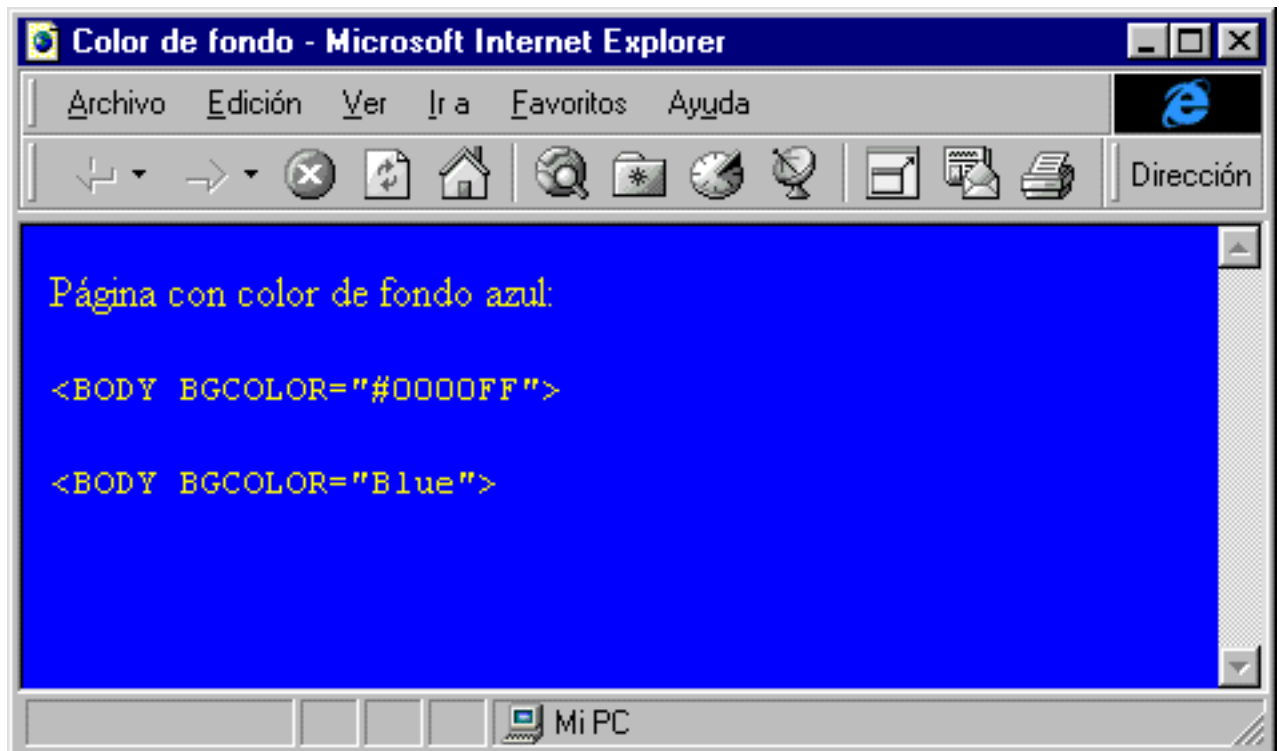
```
<body bgcolor="#0000FF">
```

O bien:

```
<body bgcolor="Blue">
```

El resultado es en ambos casos el que podemos apreciar en la figura 5.2

Figura 5.2. El lenguaje HTML nos permite cambiar el color de fondo de nuestras páginas. Podemos indicar el color por su nombre en inglés o con su número RGB.



NOTA: Algunos navegadores, como Explorer™, permiten poner el código en decimal. Para ello hay que omitir el símbolo # y usar 3 dígitos decimales (números del 0 al 255) para especificar la cantidad de cada color en el mismo orden (rojo, verde y azul). Sin embargo no recomendamos el uso de esta característica porque de esta forma provocaremos que sólo los visitantes con ese navegador puedan apreciar el colorido de nuestras páginas y realmente el esfuerzo de poner el código en hexadecimal no es tan grande.

Es importante resaltar que si no usamos el atributo BGCOLOR el color de fondo de nuestras páginas será el que cada usuario haya especificado en su navegador. En algunos navegadores el color por defecto es el blanco, mientras que en otros es el gris plateado habitual en Windows™.

Color del texto y Links.

Igual que se puede cambiar el color del fondo de la página se puede cambiar el color de su texto. Para ello se usan cuatro nuevos atributos de la etiqueta **BODY**. Lo primero que deberemos hacer es pensar en el color y el código de este, y una vez lo tengamos cambiaremos el color del texto usando:

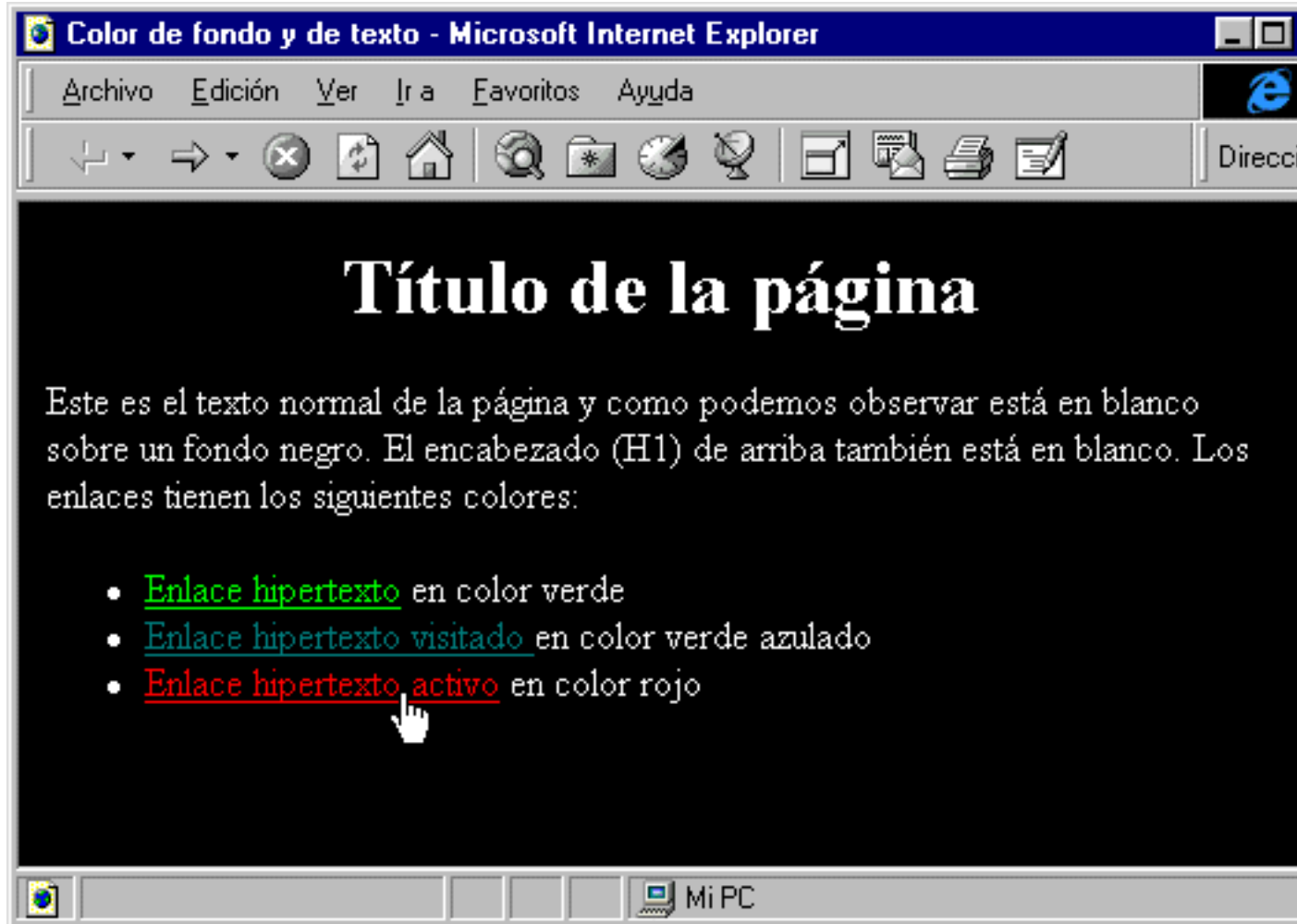
TEXT	Sirve para controlar el color que tendrá el texto normal de la página, es decir todo aquel que no sea un enlace hipertexto. Su valor por defecto es negro.
LINK	Permite cambiar el color con el que son mostrados los enlaces hipertexto o links de la página. Su valor por defecto es azul.
VLINK	Permite cambiar el color por defecto de los links visitados (Visited LINK). Los links visitados de una página son aquellos que ya hemos seleccionado alguna vez. El valor por defecto de estos enlaces es morado.
ALINK	Sirve para controlar el color de los enlaces activos (Active LINK). El significado de enlace activo varía en función del navegador. Para Netscape™ un enlace activo es aquel sobre el que se ha pulsado con el ratón pero todavía no se ha soltado. El color por defecto de los links activos es el rojo.

Vistas todas las etiquetas pasemos a crear un ejemplo, para él seleccionaremos un color de pantalla negro, texto blanco, enlaces verdes, enlaces visitados verde azulado y enlaces activos rojos. El código usado es por tanto:

```
<body bgcolor="#000000" text="#FFFFFF"  
link="#00FF00" vlink="#008080" alink="#FF0000">
```

En la figura 5.3 podemos ver el resultado de este código.

Figura 5.3. También podemos cambiar el color del texto, de los enlaces, de los enlaces visitados y de los enlaces activos a nuestro gusto, igual que hacíamos con el fondo.



Imágenes de fondo:

Las posibilidades que nos ofrece el lenguaje HTML para cambiar la apariencia de una página no se limitan a cambiar los colores, también se pueden poner imágenes ya sean dibujos o fotos como fondo de una página. Las imágenes deberán estar en formato GIF o JPEG al igual que las imágenes normales que aprendimos a insertar en el capítulo 2.

Para incluir una imagen de fondo deberemos usar el atributo `BACKGROUND` de la etiqueta `BODY`. Como habrá podido comprobar el lector a estas alturas dentro de esta etiqueta se engloban todos los parámetros que cambian las características globales de la Web, es decir aquellas que afectan a toda la página.

El funcionamiento de `BACKGROUND` es idéntico al del atributo `SRC` de la etiqueta `IMG`, esto es, deberemos indicar como valor el nombre de la imagen, junto con su ruta de directorios (path) si es necesario o bien la dirección completa de Internet (URL) si no se encuentra en nuestro sitio Web.

Veamos un primer ejemplo de una página con una imagen de fondo, para ello usamos el siguiente código en el que hemos usado la imagen `fondo1.jpg` que se encuentra en el mismo directorio que la página donde está el código:

```
<body background="fondo1.jpg">
```

En la figura 5.4 podemos ver el resultado en el que hemos incluido, además, un encabezado que contraste con el fondo.

Figura 5.4. Para crear un mayor impacto visual podemos usar como fondo una imagen, esta imagen se replicará (es puesta en mosaico) por toda la página, luego los bordes deben coincidir.



Las imágenes que insertemos como fondo de la página serán mostradas en mosaico, es decir, se replicará la imagen tanto horizontal como verticalmente hasta ocupar todo el área de la página. Por esta razón no todas las imágenes quedan bien como fondo de una página Web. De hecho se recomienda que sólo se usen dos tipos de imágenes:

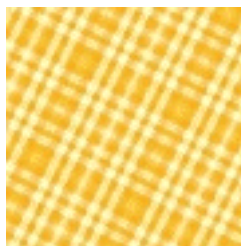
- Imágenes con pocos colores que muestran un logo sobre un relleno sólido y que interfiere poco con el contenido de la página. En la Web de la figura 5.5 podemos ver un ejemplo de este tipo de imágenes en la que no se aprecian los bordes de la imagen al ser el fondo de esta de un solo color sólido (gris oscuro en este caso).

Figura 5.5. En esta ocasión la imagen de fondo es un logo sobre un fondo de color sólido (gris). Esta es una manera sencilla para que no se noten los bordes cuando el navegador la pone en mosaico.



- Imágenes especiales cuyos bordes encajan perfectamente y parece que todo del fondo es una única imagen. La imagen de la figura 5.4 es un ejemplo de ello ya que su tamaño real es de 90x90 píxeles. En la figura 5.6 vemos esa imagen por sí sola. Es difícil apreciar que los bordes encajarán cuando se creen las réplicas, de hecho ese efecto debe ser hecho con programas especiales.

Figura 5.6. Esta es la imagen usada como fondo en la página de la figura 5.4. Como vemos, es de pequeño tamaño, pero al coincidir sus bordes simula una imagen que ocupa toda la pantalla.



Por lo general la creación de fondos requiere además de programas especiales, bastante cuidado para asegurarnos que no sobrecarguen demasiado la página, por lo que un gran número de programadores de Webs optan por usar alguno de los fondos de los cliparts gratuitos que podemos encontrar en Internet, por ejemplo en las direcciones:

- www.specialweb.com
- www.dewa.com
- www.teleportcom

Si usamos una imagen normal como fondo de una página, los bordes no encajarán y se verán claramente los límites de la imagen. Podemos ver un ejemplo de este efecto en la figura 5.7. En esta misma figura vemos también que poner una foto de fondo suele provocar que la página esté demasiado sobrecargada, además sería difícil leer el texto que hubiese en esa imagen.

Figura 5.7. También podemos poner fotos como fondo de la página, pero hay que tener cuidado porque los bordes no encajan y además la página queda demasiado sobrecargada.



Sugerencia

No use imágenes demasiado oscuras o con demasiados colores como fondo de la página. Si aún así debe usarla utilice los atributos `TEXT`, `LINK`, `VLINK` y `ALINK` de la etiquetas `BODY` para poner colores de texto que permitan que el texto sea legible.

Por último, queremos terminar este apartado con un consejo. Como hemos repetido, la creación de imágenes de fondo es complicada y delicada. Si no las hemos hecho nosotros deberemos elegir bien, en general suelen ser mas apropiadas las imágenes con tonos claros sobre los que se puede leer sin problemas texto negro. Pero antes de incluir la imagen deberemos volver a hacernos unas preguntas ¿Es realmente necesaria? ¿Mejora realmente el aspecto gráfico? que con frecuencia nos hacen ver las cosas más claras. Una vez hemos probado la imagen debemos tener mucho cuidado y hacer pruebas con otras personas para comprobar que el texto de la página se lea bien. Piense que por muy espectacular que sea la página, si no conseguimos que lea lo que tenemos que decir el objetivo no estará cumplido.

Imágenes en las páginas Web.

En el segundo capítulo: 'Nuestra primera página Web', aprendimos a insertar imágenes en una Web. Ahora empezaremos recordando brevemente aquel capítulo, para continuar con las distintas técnicas que nos ofrece el lenguaje HTML no sólo para incluir imágenes, sino para manejarlas, cambiar su apariencia o controlar su disposición respecto a los demás elementos de la página.

Insertando Imágenes.

Recordemos que para insertar una imagen en una página Web era imprescindible guardarla en un archivo independiente del documento HTML. Dicho archivo podía contener la imagen en varios formatos, siendo los de mayor uso GIF y JPEG que pueden ser vistos con la práctica totalidad de navegadores. Cuando tenemos una imagen con estos formatos suelen tener la extensión `.gif` en el caso del formato GIF y `.jpg` o `.jpeg` en el caso de imágenes JPEG.

Atención

Algunas aplicaciones de retoque de imágenes guardan las imágenes GIF y JPEG con la extensión en mayúsculas (`.GIF`, `.JPEG`). Esta terminación es válida, pero los nombres de los archivos que contienen imágenes, igual que los de los documentos HTML, son sensibles a mayúsculas/minúsculas y por tanto `.gif` no es lo mismo que `.GIF` con lo que al referirnos al archivo debemos escribir su nombre y terminación respetando las mayúsculas y minúsculas. Si las hemos confundido cuando probemos las páginas en el ordenador local probablemente funcionen, pero dejarán de hacerlo cuando las publiquemos, es decir, cuando las pongamos en el servidor Web.

Nota

El formato de una imagen es el código usado para almacenarla en un archivo. Los formatos simples simplemente guardan una tabla con el color de cada punto de pantalla (pixel) de la imagen. Sin embargo si guardamos una imagen grande con este formato ocupará demasiado. Por esta razón se crearon formatos más complicados basados en complejas fórmulas matemáticas, que consiguen reducir el tamaño de la imagen. GIF y JPEG son ejemplos de estos formatos. Los navegadores usan la extensión (los tres/cuatro últimos caracteres tras un punto) de los archivos para averiguar el formato en que están codificadas, si cambiamos manualmente esa extensión no tendrán esa información y la imagen no podrá ser visualizada.

Es importante resaltar cambiando la extensión de una imagen no cambiamos el formato de dicha imagen, pero podemos dejarla inútil.

Más adelante trataremos a fondo estos dos formatos y comentaremos cuando es conveniente usar uno u otro.

Una vez tenemos la imagen en un archivo aparte (por ejemplo `img.gif`) con el formato adecuado insertamos la imagen usando la etiqueta `IMG` de la siguiente manera:

```

```

Como ya comentamos, esta etiqueta consta de una única instrucción. Pero aunque sólo vimos uno de sus atributos, `SRC`, contiene muchos más que iremos estudiando a lo largo del capítulo.

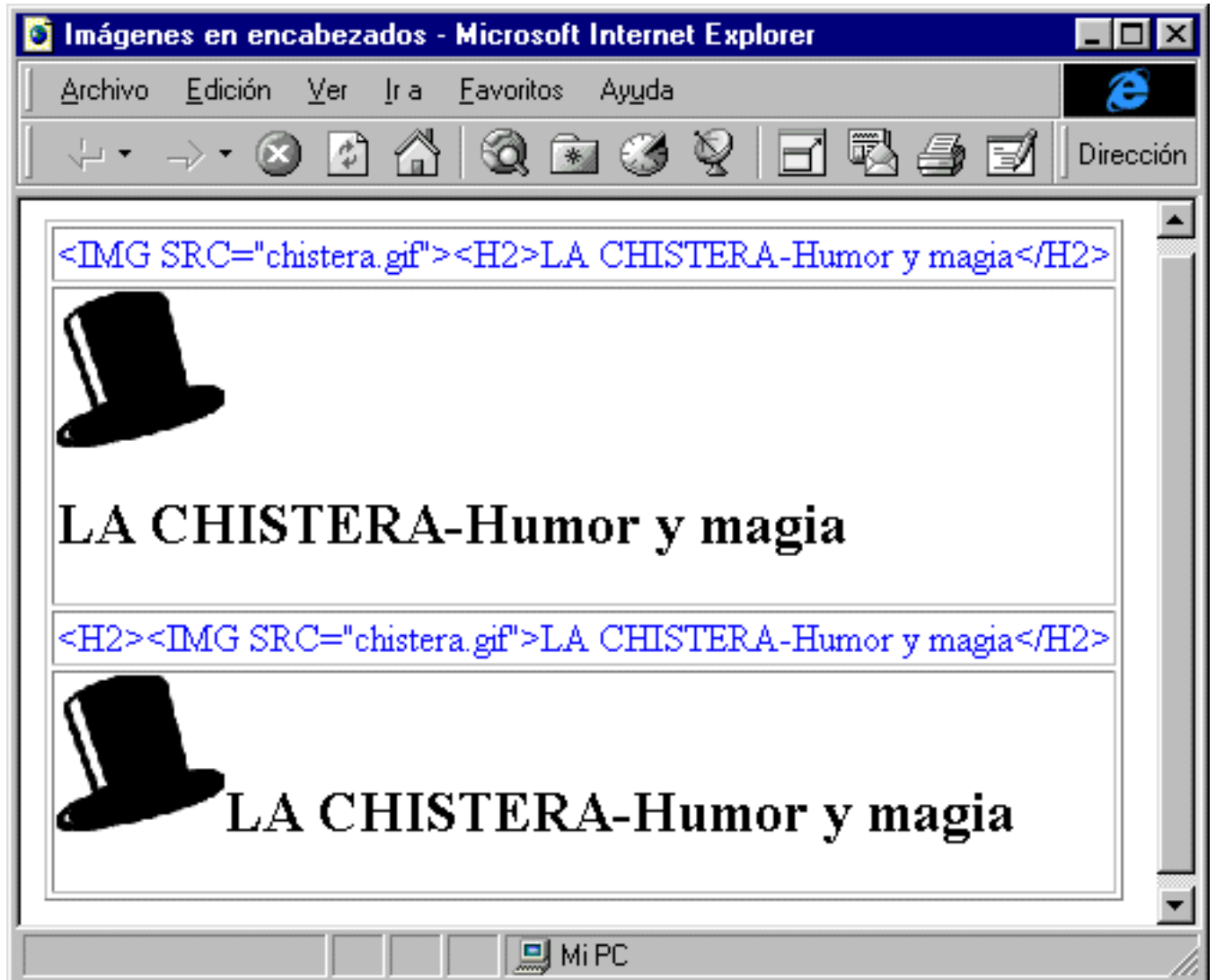
Pero antes de comenzar a ver nuevos conceptos...

Imágenes y texto.

Cuando insertamos una imagen ésta se sitúa justo en el lugar donde está la etiqueta `IMG` independientemente de si hay texto alrededor o no, por esta razón este tipo de imágenes reciben el nombre de imágenes en línea o imágenes *inline*. Usando esta característica podíamos insertar imágenes pequeñas como si fuesen parte de un párrafo.

Si queremos insertar una imagen en el interior de un elemento de nivel de bloque, pongamos por ejemplo un encabezado, no tendremos más que poner la etiqueta en el interior de ese elemento (entre las instrucciones de inicio y fin). Si no lo hiciéramos así, después de la imagen se produciría un salto de línea antes del encabezado. Para entender esto mejor podemos fijarnos en la figura 5.8 y comparar entre los siguientes códigos:

Figura 5.8. Podemos apreciar la diferencia entre insertar una imagen *dentro* del encabezado o insertarla *fuera*, en cuyo caso queda en la línea superior.



```

<h2>LA CHISTERA-Humor y magia</h2>
<h2>
LA CHISTERA-Humor y magia</h2>
```

que nos resultarán familiares de la primera práctica.

Como vemos, en el primer caso la instrucción de inicio del encabezado, `<h2>`, provocará un salto de línea y el efecto no será el deseado. En el siguiente caso hemos incluido la etiqueta `IMG` dentro del encabezado, es decir, estamos diciendo al navegador algo así: 'el encabezado consta de esta imagen y este texto'. En este caso, como resulta totalmente lógico, el resultado es el deseado.

Aún así el ejemplo anterior no resuelve todos los problemas que podrían plantearse al insertar imágenes. Como ya vimos en un ejemplo del capítulo 2 (fig. 2.9), que ahora reproducimos en la figura 5.9 cuando las imágenes

son pequeñas pueden incluirse junto con el texto sin problemas, pero ¿qué ocurre si queremos insertar una imagen de dimensiones mayores? Podemos usar el método anterior, pero entonces obtendríamos resultados como el de la figura 5.10, que sin duda no es el deseado y provoca que el texto sea difícilmente legible.

Figura 5.9. Como veíamos en el fascículo 2 (fig. 2.9) cuando la imagen es pequeña puede entremezclarse con el texto sin problema.

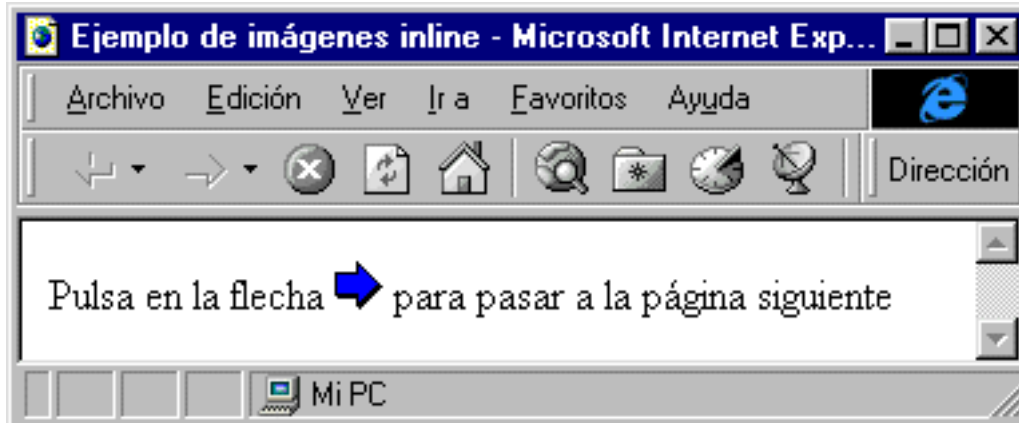
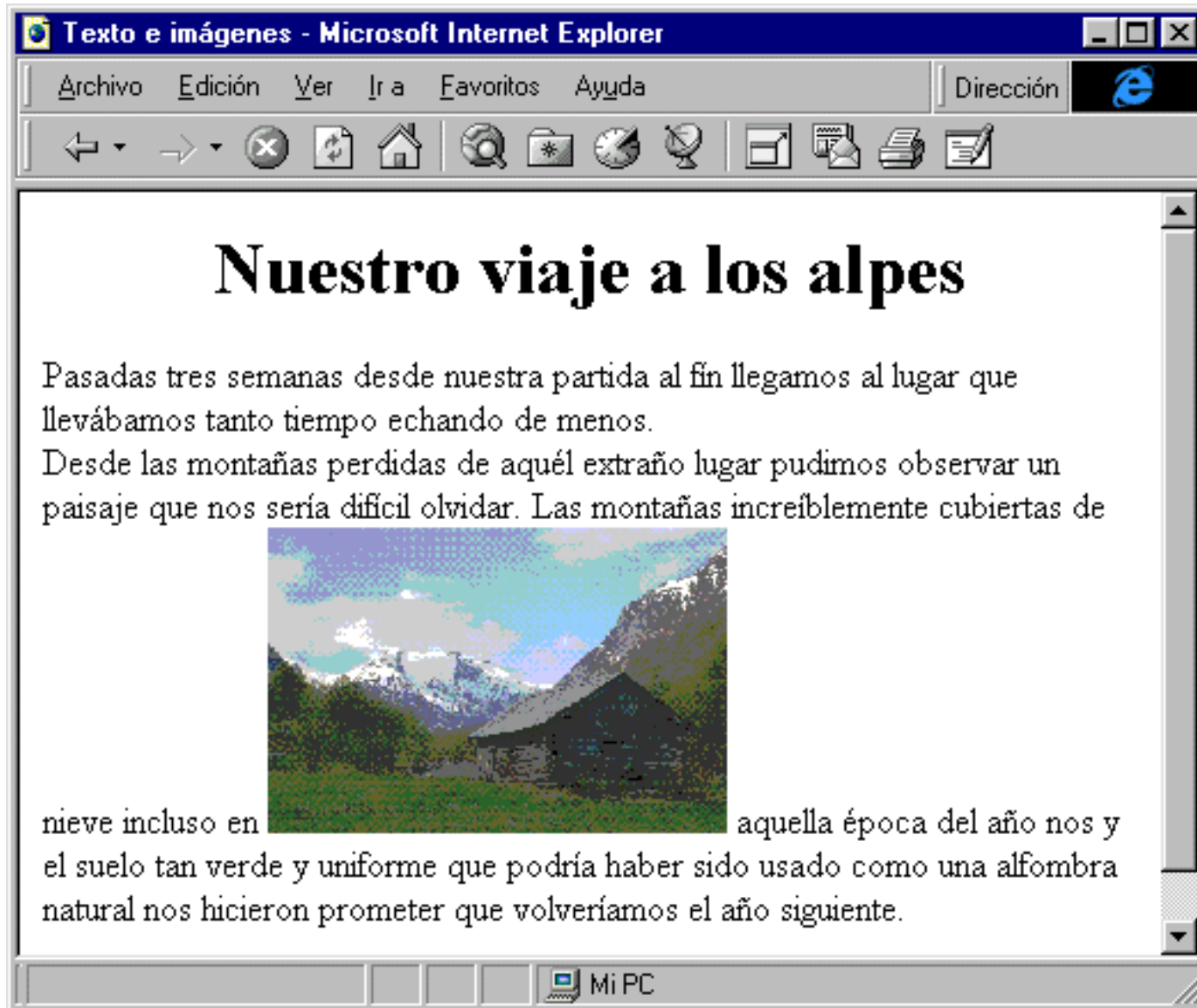


Figura 5.10. Cuando la imagen es grande, al insertarla en una línea de texto ésta queda muy separada de la línea anterior en lugar de envolver la imagen.



Una opción para solucionar este problema es provocar un salto de línea antes y después de la imagen, tal y como hacíamos en la práctica 2 para las imágenes de la página de presentación. Podemos usar diversas etiquetas para conseguir los saltos de línea: P, CENTER, BLOCKQUOTE, BR, etc. por ejemplo:

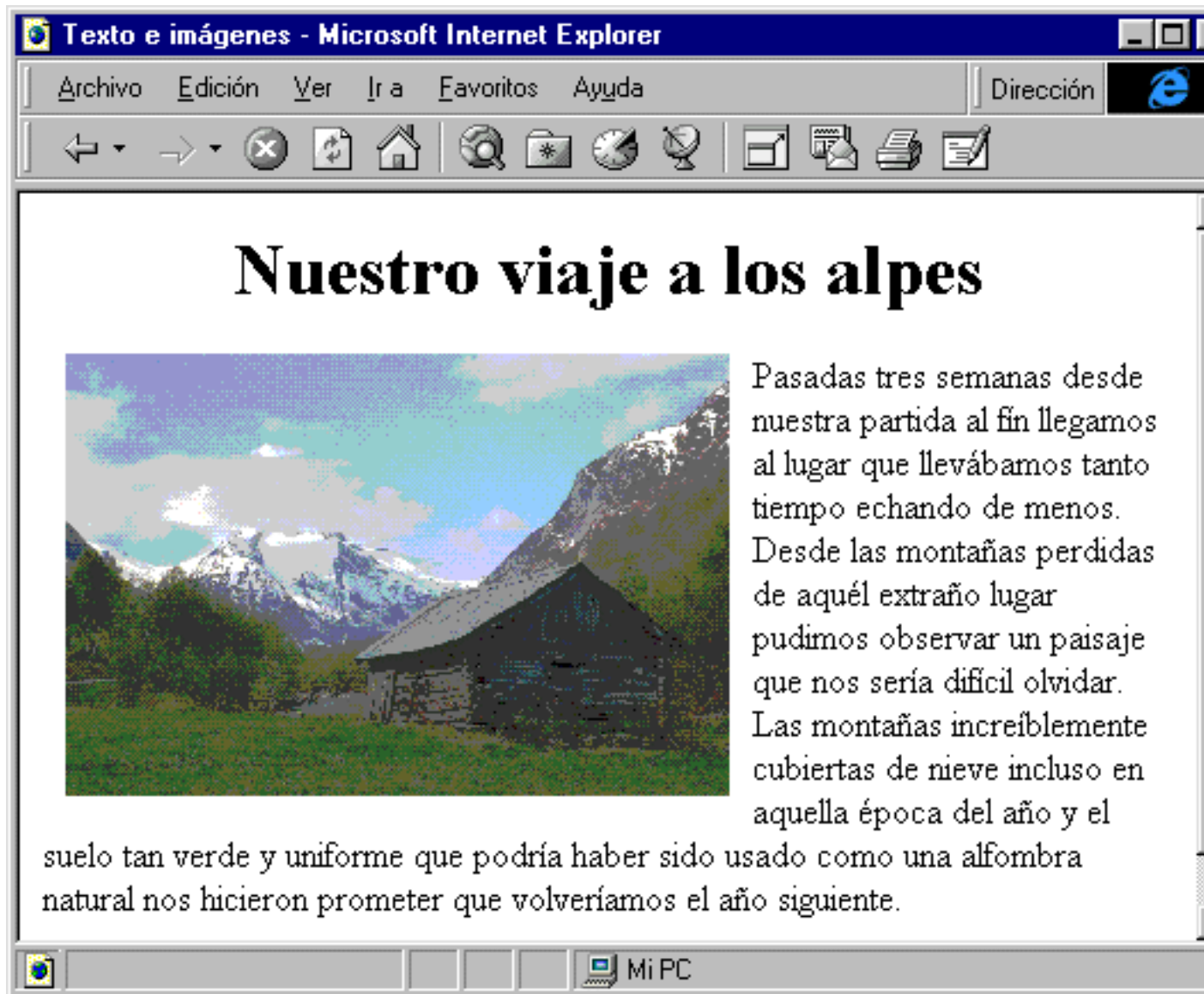
```
<p>

</p>
<center>

</center>
```

Sin embargo no siempre nos conformamos con esto. Habitualmente lo que queremos es que la imagen se inserte junto con el texto y que este se sitúe alrededor suyo, tal y como vemos en periódicos u otras publicaciones. De esta forma podríamos conseguir el efecto de la figura 5.11. Afortunadamente es posible conseguir ese efecto usando las capacidades de alineamiento de imágenes del HTML.

Figura 5.11. Con las técnicas de alineamiento del lenguaje HTML que estudiamos en este capítulo podemos crear interesantes efectos como texto envolviendo a una imagen.



Alineamiento de imágenes.

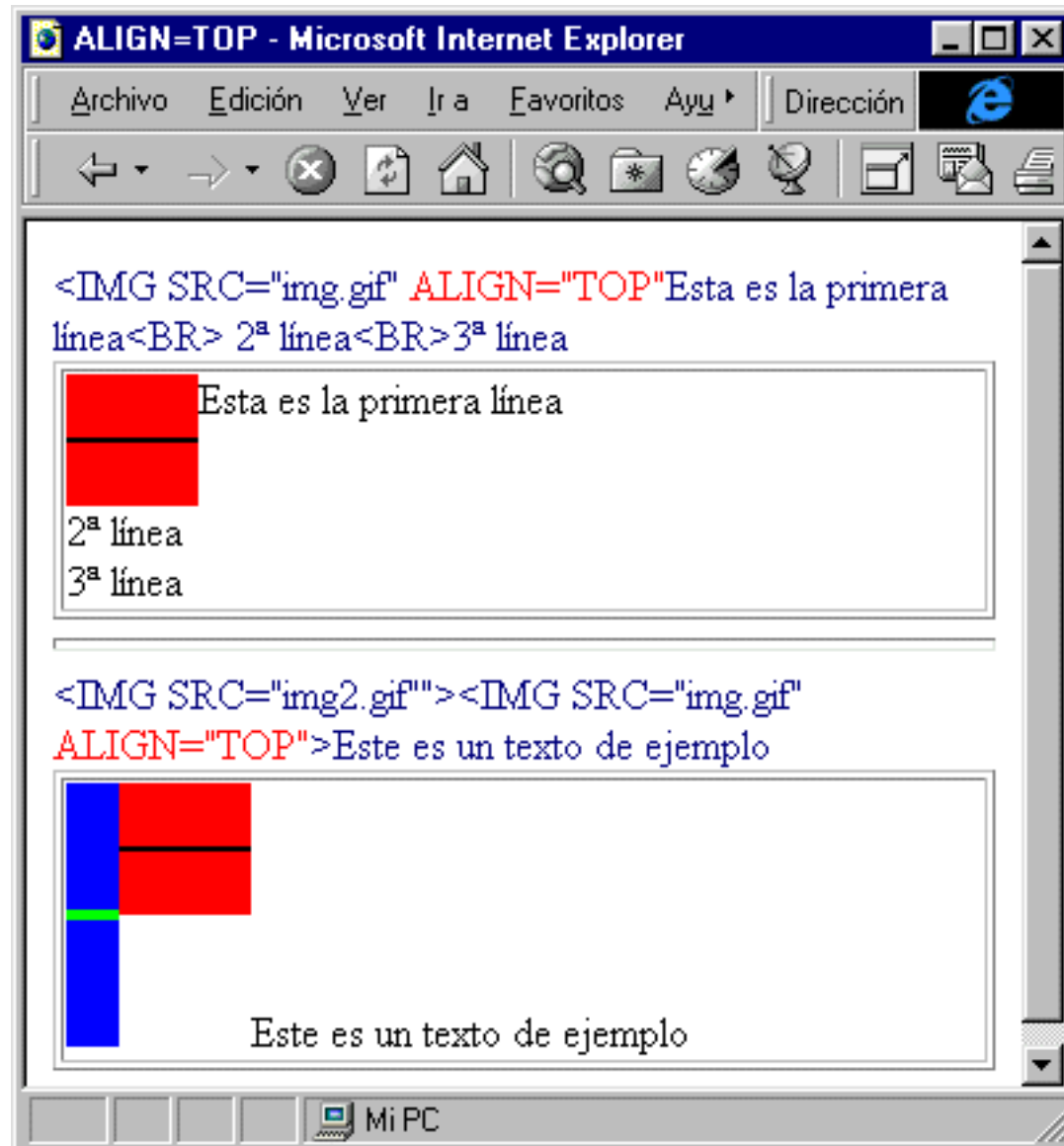
En los ejemplos vistos al principio, en los que insertábamos imágenes junto con texto normal, la parte inferior de la imagen quedaba alineada con la parte inferior del texto. La etiqueta `IMG` tiene un atributo llamado `ALIGN` que nos permite controlar el alineamiento vertical de la imagen en función del texto que le rodee o de otras imágenes de la misma línea. El atributo `ALIGN` puede tomar cinco valores diferentes. En primer lugar veremos aquellos

que nos permiten controlar la alineación vertical de la imagen. Para cada uno de ello mostraremos dos ejemplos, en ambos la imagen que será alineada es un cuadrado rojo con una línea que marca su centro. En el primero la imagen irá seguida de tres líneas de texto y en el segundo irá precedida de una imagen el doble de alto que marcará los límites verticales de la línea.

`align=top`:

Alinea la parte superior de la imagen con la parte superior de la línea. Para determinar cual es el límite superior de la línea se tendrán en cuenta todos los elementos de esta (ya sean texto, imágenes u otros), tanto los anteriores como los posteriores a la imagen que estamos insertando. En la figura 5.12 podemos ver dos ejemplos de `align=top`. El primer caso es sencillo de entender, pero el segundo requiere alguna explicación. La primera imagen se alinea (por defecto) con su parte inferior a la misma altura que la base del texto, después nuestro cuadrado rojo, alinea su parte superior con la parte superior de dicha imagen.

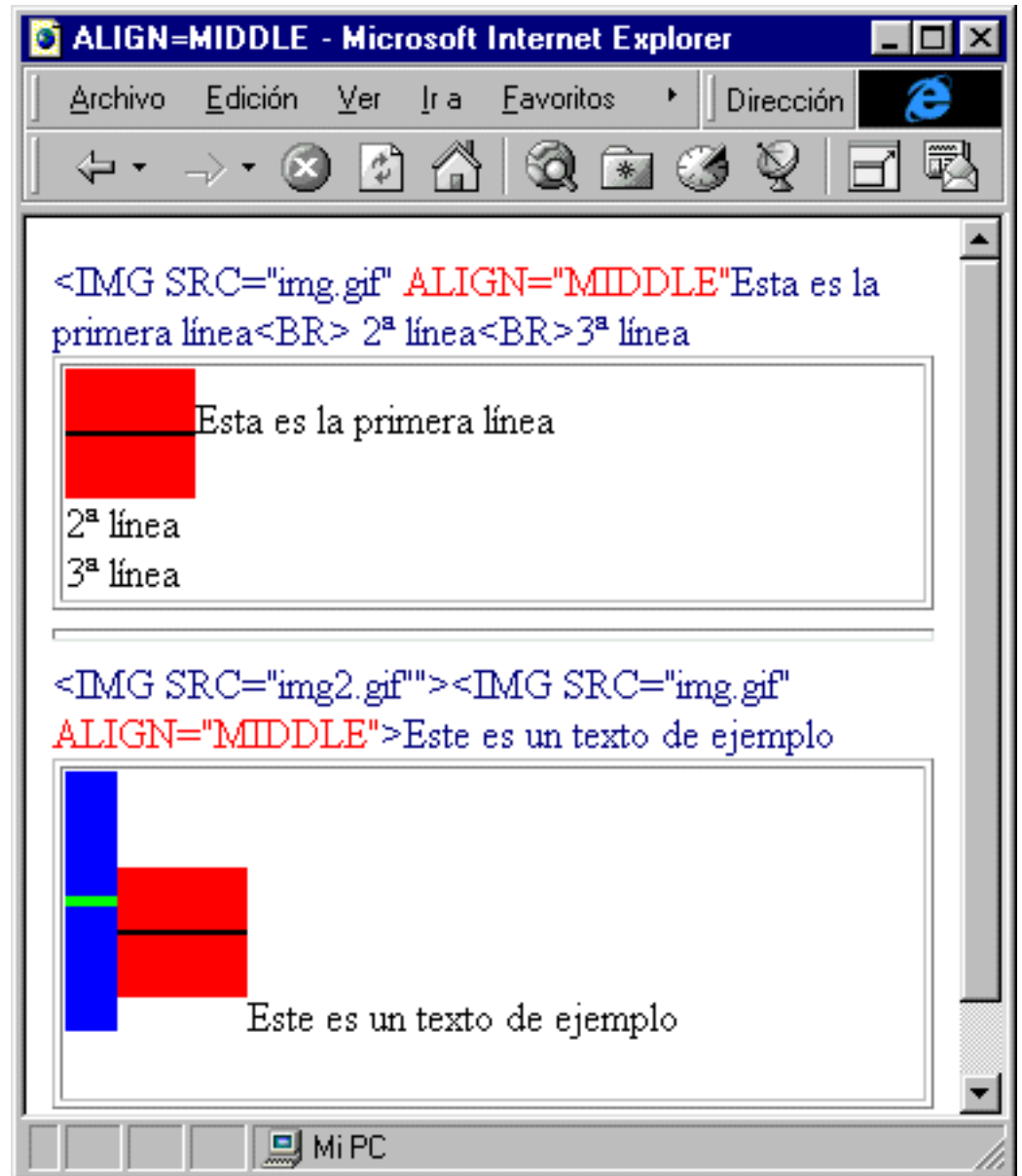
Figura 5.12. Ejemplos de `align=top`. En el primer caso la parte superior de la imagen coincide con la del texto y en el segundo con la parte superior de la imagen azul.



align=middle:

Alinea el centro (vertical) de la imagen con el centro de la línea. Si la línea esta compuesta únicamente por texto el se tomará la base de éste como centro de la línea. Si hay más elementos el navegador calcula cual es el centro de la línea en función de todos ellos. En la figura 5.13 podemos ver unos ejemplos.

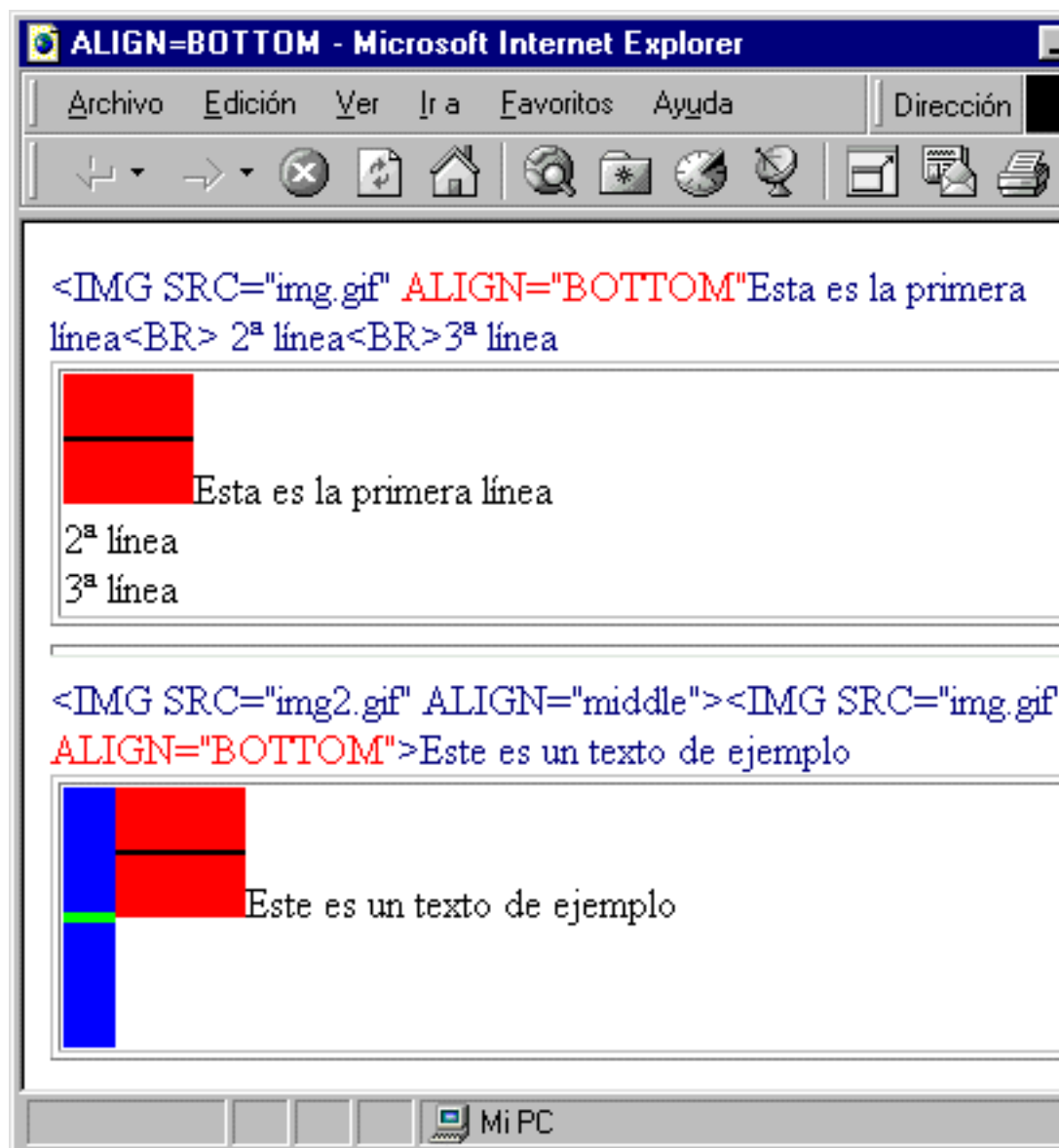
Figura 5.13. Ejemplos de align=middle. En el primer caso el centro de la imagen coincide con la base del texto, en el segundo intenta centrarse con la imagen azul.



align=bottom:

Alinea la parte inferior de la imagen con la parte inferior de la línea de texto. En este caso no se tendrán en cuenta otros elementos a parte del texto para realizar el alineamiento. De nuevo podemos ver ejemplos de este alineamiento en este caso en la figura 5.14.

Figura 5.14. Ejemplos de align=bottom. En ambos casos la parte inferior de la imagen se alinea con la base del texto, aunque la imagen azul baje por debajo.



NOTA: Es posible que nos encontremos alguna página en la que haya una imagen con el atributo `align=center`. Este nuevo valor (`center`) sólo funciona con Internet Explorer™ y tiene el mismo significado que `align=middle` y además su nombre puede confundirnos y creer que la imagen se mostrará como si estuviese encerrada por la etiqueta `<center>..</denter>`. Por estas dos razones recomendamos el uso de `align=middle` en su lugar.

Estos tres son los únicos tipos de alineamiento que incluía el lenguaje HTML en sus orígenes. Sin embargo, aunque son útiles, no nos permiten controlar la disposición de las imágenes tanto como querríamos. Por ejemplo, todavía no podemos insertar una imagen y que el texto la bordeé tal y como veíamos en la figura 5.11. De hecho en todos los valores anteriores sólo la primera línea de texto introducida después de la imagen se situaba a la derecha de esta, aunque esta línea estuviese en la parte superior de la imagen como ocurría en el primer caso de la figura 5.12. La segunda línea y las siguientes serían mostradas debajo de la imagen.

Texto que envuelve a las imágenes o alineamiento horizontal.

Vista la necesidad de nuevos métodos de alineamiento se crearon dos nuevos valores para el atributo `align`: `left` y `right`, que nos permitirán controlar el alineamiento horizontal de las imágenes (esto es, izquierda o derecha). Estos valores se han convertido en los más usados de las páginas Web por su enorme utilidad. En seguida veremos como funcionan.

NOTA: Las imágenes pueden alinearse vertical y horizontalmente, pero nunca a la vez. Esto es así porque ambos tipos de alineamiento usan el atributo `align`, y sólo puede incluirse uno de estos atributos dentro de la etiqueta `img`. Si ponemos más el navegador se quedará con el primer valor e ignorará el resto.

Antes es necesario hacer una matización. Aunque en un capítulo próximo estudiaremos con más detenimiento las diferentes versiones del lenguaje HTML, adelantaremos que los tres primeros valores estudiados: `top`, `middle` y `bottom`, pertenecen al estándar HTML 2.0. Sin embargo los dos nuevos valores que vamos a explicar fueron introducidos en la versión HTML 3.2. El lector se preguntará ¿y para que quiero saber yo eso? Pues bueno, todos los navegadores entienden HTML 2.0, pero alguno no entiende HTML 3.2 y por tanto no entenderán `align=left` ni `align=right`. Sin embargo no debemos preocuparnos en este caso, ya que es difícil a estas alturas encontrar un navegador que no soporte HTML 3.2 y por tanto podremos estar casi seguros que nuestros visitantes disfrutarán de nuestras imágenes alineadas.

Por otro lado aquellos visitantes de la página que usen un navegador muy antiguo verán lo mismo que si no hubiésemos puesto el atributo `align`, pero no tendrán ningún otro problema, por tanto concluiremos que podemos usar los nuevos valores sin demasiada preocupación.

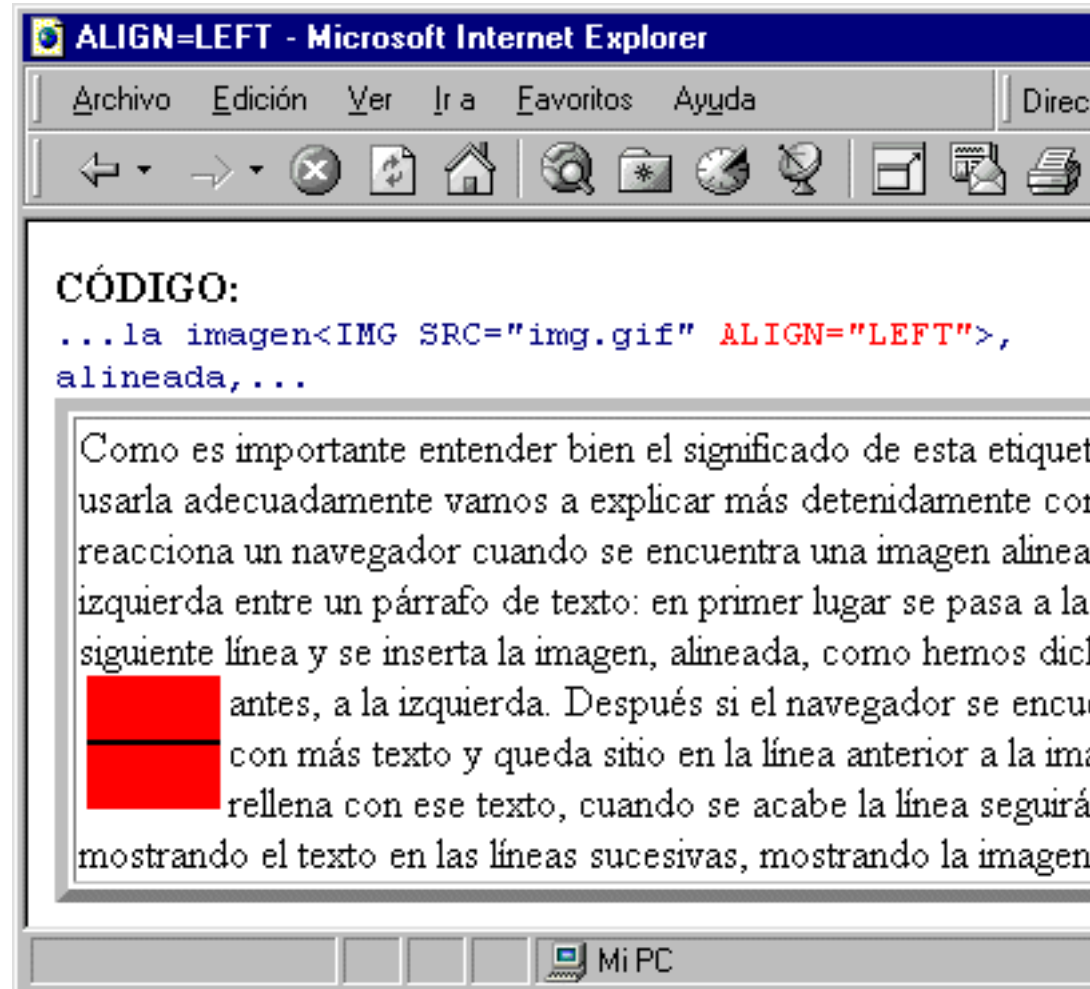
Y después de esta matización pasamos ya a explicar el significado de estos atributos.

`align=left`

La palabra *left* significa, en inglés, izquierda, por tanto al usar este valor la imagen se sitúa a la *izquierda* de la página. Pero su efecto es aún más importante, ya que si a continuación de la imagen introducimos texto o cualquier otro elemento del Web éste rodeará a la imagen, no pasará a la siguiente línea.

Como es importante entender bien el significado de esta etiqueta para usarla adecuadamente vamos a explicar más detenidamente como reacciona un navegador cuando se encuentra una imagen alineada a la izquierda entre un párrafo de texto: en primer lugar se pasa a la siguiente línea y se inserta la imagen, alineada, como hemos dicho antes, a la izquierda. Después si el navegador se encuentra con más texto y queda sitio en la línea anterior a la imagen lo rellena con ese texto, cuando se acabe la línea seguirá mostrando el texto en las líneas sucesivas, rodeando la imagen. En la figura 5.15 podemos ver un ejemplo de su funcionamiento. Como vemos aunque insertamos la etiqueta `img` tras la palabra imagen tras insertar ésta el texto posterior a esta etiqueta sigue mostrándose como si la imagen no hubiera sido insertada.

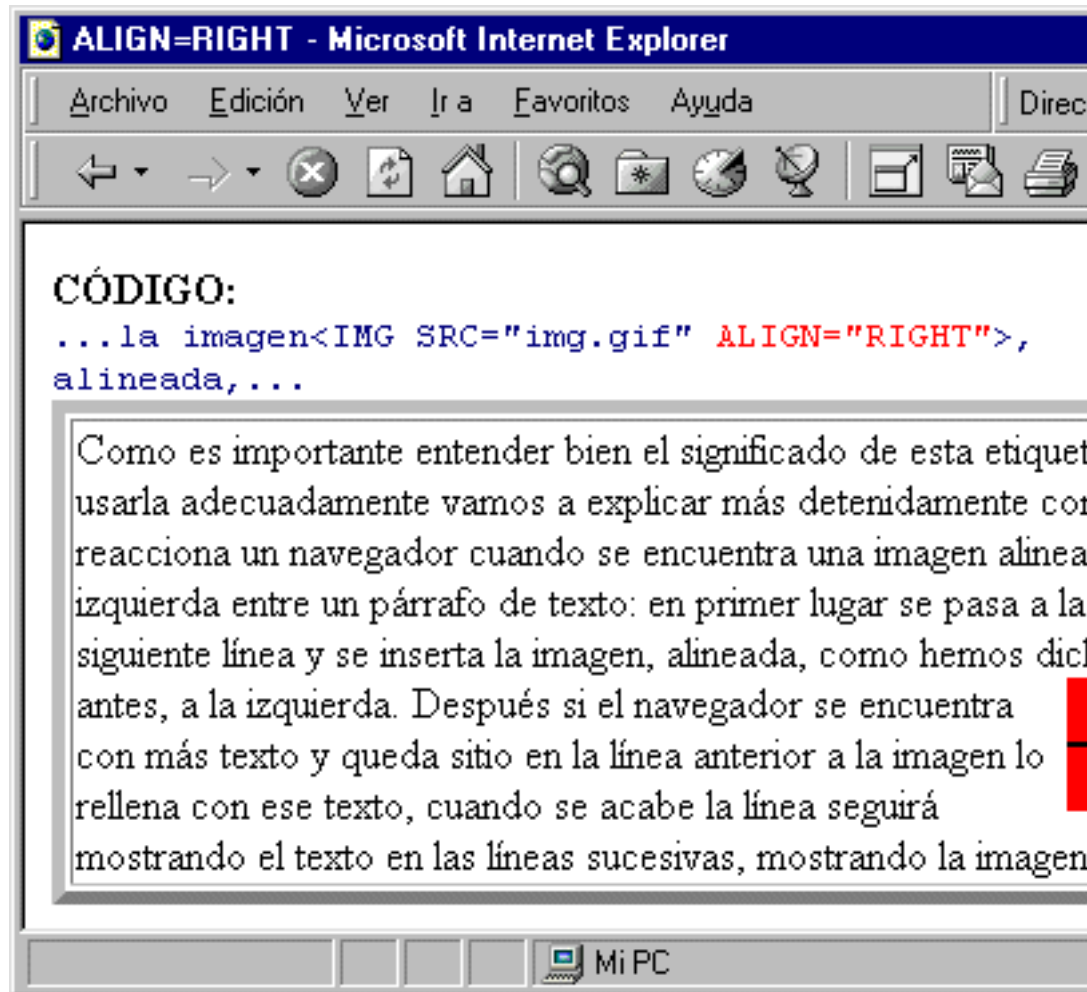
Figura 5.15. Alineando una imagen a la izquierda con `align=left` conseguiremos que el texto envuelva a la imagen.



`align=right`

Probablemente el lector ya haya deducido que este otro valor sirve para indicar que la imagen estará *alineada a la derecha* (*right* significa precisamente *derecha* en inglés). El efecto provocado por `align=right` es idéntico al anterior, solo que ahora la imagen se situará a la derecha de la página y el texto la rodeará por la izquierda. En la figura 5.16 podemos apreciar el mismo código de la figura 5.15 pero cambiando el alineamiento de la imagen.

Figura 5.16. Usando `align=right` conseguimos igualmente que el texto envuelva a la imagen pero ahora ésta se situará a la derecha de la página.



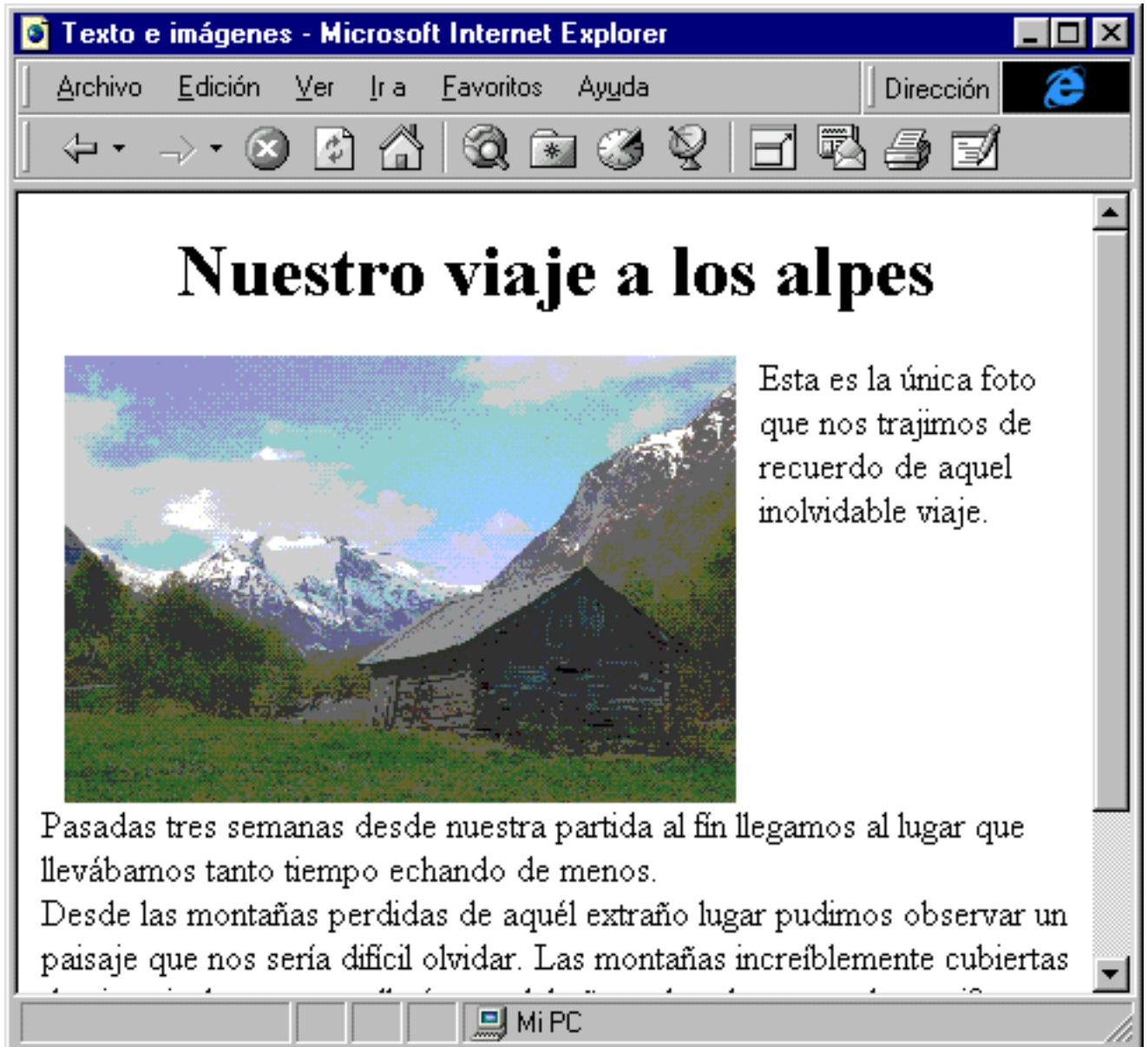
NOTA: Existen otros modos de alineamiento, pero no funcionan en todos los navegadores, ni forman parte del lenguaje HTML. Esos modos de alineamiento han sido creados por los fabricantes de un determinado navegador y por tanto no funcionan con el resto. En posteriores capítulos hablaremos de este tema y de otras *extensiones de los navegadores* y comentaremos los otros tipos de alineamiento existentes.

Como terminar el texto envolvente.

En la figura 5.11 adelantábamos un ejemplo de cómo quedaba una imagen con texto que envolviéndola. En ese caso usábamos una fotografía de tamaño considerable, y por tanto había numerosas líneas de texto a su izquierda. Después de la primera frase pusimos un punto y aparte, que como ya sabemos, se hace en HTML con la etiqueta `
`, y continuamos: "Desde las...".

Pero podemos imaginar otro caso, como el de la figura 5.17 en el que lo único que queremos es escribir unas pocas líneas que comenten la imagen pero continuar el resto del texto después (debajo) de la imagen. Si ponemos `br` sin más pasaremos a la siguiente línea, pero no al final de la imagen. Un primer método que se nos podría ocurrir para conseguir este efecto sería poner la etiqueta `br` muchas veces, tantas como líneas quepan hasta el final de la imagen. Sin embargo este sistema a parte de ser molesto, no funciona como puede comprobar el lector si decide probarlo.

Figura 5.17. Usando el atributo `clear` de la etiqueta `br` podemos terminar el texto envolvente y seguir escribiendo debajo de la imagen.



Pero entonces ¿existe alguna otra forma de conseguirlo? La respuesta es SI. Cuando se añadieron los valores `right` y `left` a los modos de alineamiento se introdujo también un nuevo atributo en la etiqueta `br` que nos permitiera tratar con este tipo de alineamiento. Ese atributo se llama `clear`, puede tomar los siguientes valores.

`left`

Provoca un salto de línea hasta el primer margen libre a la izquierda. Esto quiere decir que provocará un salto hasta la primera línea en la que no hay una imagen a la izquierda y por tanto se usa, en general, cuando hay una imagen alineada a ese lado. Este valor es el que hemos usado en el ejemplo de la figura 5.17. Su modo de uso es el siguiente:

```
<br clear=left>
```

right Provoca un salto de línea hasta el primer margen libre a la derecha, por lo demás funciona idénticamente al valor anterior:

```
<br clear=right>
```

all Provoca un salto hasta la primera línea que tiene tanto el margen izquierdo como el margen derecho libres. Es decir provoca el efecto de los dos valores anteriores combinados.

```
br clear=all
```

Ya empezamos a sentir que tenemos una gran control sobre la disposición de las imágenes en nuestra página, pero aún podemos manejar un parámetro más en relación a estas y al texto que las envuelve: el espaciado que hay entre ellas.

Márgenes alrededor de las imágenes.

Para poder controlar los márgenes se crearon dos nuevos atributos más para la etiqueta `img`: `vspace` y `hspace` que controlan el tamaño de los márgenes vertical (`vspace` proviene de *Vertical SPACE* que significa espaciado vertical) y horizontal (`hspace` proviene de *Horizontal SPACE*, espaciado horizontal) respectivamente. Estos atributos suelen acompañar al uso de `align=left` y `align=right`.

El tamaño de los márgenes se indicará en píxeles (puntos de pantalla). Veamos ahora un ejemplo con amplios márgenes para que podamos apreciar bien el efecto de estos atributos. El código usado es:

```
<img align=left vspace=35 hspace=35>
```

Y podemos ver el resultado en la figura 5.18

Figura 5.18. Usando los atributos `hspace` y `vspace` podemos controlar la separación entre la imagen y el texto que la envuelve.



Imágenes enlace.

A estas alturas del curso de HTML dominamos la creación de enlaces hipertexto y conocemos como insertar imágenes en nuestras Web. Uniendo ambos conocimientos vamos a crear imágenes que nos servirán como enlaces a otros lugares del World Wide Web.

No será necesario aprender nada nuevo, lo único que necesitaremos es combinar los efectos de las etiquetas `a` y `img`. En primer lugar deberemos insertar la instrucción de inicio de `a`, después incluir la etiqueta `img` y por último insertamos la etiqueta de fin de `a`. Un ejemplo muy típico del uso de imágenes enlace, como suelen ser denominadas, son las flechas para avanzar o volver atrás en una página. En nuestra práctica del capítulo pasado usamos estas flechas pero solo como indicación de el resultado de pulsar sobre el texto adjunto. Ahora convertiremos esas imágenes en enlace para que se pueda pulsar sobre ellas. El código a usar es:

```
<a href="index.htm">

</a>
```

Invitamos al lector a que haga el mismo cambio en el resto de imágenes usadas en aquella práctica.

Invitamos al lector a que haga el mismo cambio en el resto de imágenes usadas en aquella práctica.

Figura 5.19. Combinando las etiquetas `img` y `a` adecuadamente podemos crear *imágenes enlace*. Los navegadores indican esto mostrando un borde azul alrededor de estas imágenes.



Nuestro siguiente objetivo será aprender a controlar este borde así como crear otros muchos efectos con las imágenes.

Cómo cambiar la apariencia de las imágenes

Borde de las imágenes

El lenguaje HTML permite recuadrar a las imágenes que incluyamos en nuestras páginas con un borde de grosor variable. En general este borde es invisible, a nos ser que especifiquemos lo contrario. Sin embargo cuando vimos

las imágenes enlace observamos que alrededor de estas imágenes aparecía un borde azul indicando que se podía pulsar sobre ellas. Como enseguida veremos que es posible conseguir que las imágenes normales SI tengan ese borde y que las imágenes enlace no lo tengan o que sea de otro grosor distinto al que aparece por defecto.

La manera de controlar el borde es usar un nuevo atributo, *BORDER*, que debe ser usado dentro de la etiqueta IMG. Este atributo toma como valor el número de pixeles que debe tener el borde. Si nuestro objetivo no es añadir un borde sino quitarlo, o asegurarnos que no lo tendrá debemos usar *BORDER=0*. De esta forma podremos conseguir que desaparezca el borde en las imágenes enlace. En la figura 5.20 podemos ver tres ejemplos de imágenes enlace:

Figura 5.20. Las imágenes enlace tienen un borde azul por defecto, pero podemos quitarlo o variar su grosor usando el atributo BORDER de la etiqueta IMG.



En el primero de ellos no hemos usado el atributo BORDER:

```
<A HREF="otra.htm">  
  <IMG SRC="izda.gif">  
</A>
```

En el segundo lo hemos usado dándole el valor cero para eliminar el borde:

```
<A HREF="otra.htm">  
  <IMG SRC="izda.gif" BORDER=0>  
</A>
```

Y en el tercero le hemos dado el valor 10:

```
<A HREF="otra.htm">  
<IMG SRC="izda.gif" BORDER=10>  
</A>
```

El uso de `BORDER=0` es especialmente útil en los llamados iconos de navegación. Estos iconos son imágenes enlace que incluimos en las páginas y que suelen tener la apariencia de botones, flechas y similares y cuya función es ayudar al visitante a navegar por nuestra página. En la figura 5.21 podemos ver varios de estos iconos. En todos estos casos la aparición de un borde alrededor de la imagen perjudicaría su apariencia.

Figura 5.21. Las imágenes enlace se usan a menudo para crear iconos de navegación como los que aquí vemos. En estos casos es conveniente quitar el borde usando `BORDER=0`.



Sin embargo no debe parecer que el atributo `BORDER` sólo tiene utilidad en las imágenes enlace, ya que podemos usarlo con todo tipo de imágenes y en ocasiones nos ayuda a crear interesantes efectos. En la figura 5.22 podemos ver ejemplos de imágenes normales sin borde o con borde de distinto grosor.

Figura 5.22. Las imágenes normales también pueden tener un borde de grosor variable, aunque en este caso el color no es azul, sino negro.



En estos ejemplos observamos que si las imágenes no son links el borde es negro en vez de azul. No existe por ahora ningún medio de elegir el color del borde.

Pero además de poder añadir un borde a las imágenes, el HTML nos permite realizar alguna otra modificación en la apariencia de las imágenes más importante aún, como puede ser modificar su tamaño. De esta forma podremos usar una misma imagen varias veces y con diferentes tamaños en una página Web sin tener que crear un archivo GIF o JPG diferente para cada una.

Modificando el tamaño de las imágenes

Tanto las imágenes GIF como las JPEG son imágenes rectangulares, esto es, su tamaño está determinado únicamente por dos propiedades: su anchura y su altura. Tanto una como otra suelen tener valores dados en píxeles o puntos de pantalla.

El lenguaje HTML nos permitirá modificar estas propiedades de las imágenes a incluir en una página de manera que podamos adaptarlas al tamaño que necesitemos. La forma de hacer realidad esta *transformación* es usar dos nuevos atributos de la etiqueta IMG: *WIDTH* y *HEIGHT*, que significan en inglés anchura y altura respectivamente. A cada uno de estos atributos deberemos darle un tamaño en píxeles que suele oscilar entre 20 y 600 en los casos habituales.

Existen dos formas de usar estos atributos. La primera de ellas consiste en dar valores tanto a *WIDTH* como a *HEIGHT* con lo cual estaremos determinando por completo el tamaño con el que el navegador debe mostrar la página. El otro método consiste en usar sólo uno de ellos, o bien *WIDTH* o bien *HEIGHT*. Al hacer esto la

dimensión que no usemos se adaptará para mantener la proporción de la imagen. Por ejemplo, si tenemos una imagen de 500x250, es decir de anchura 500 y de altura 250, pero al insertarla en la página usamos el siguiente código:

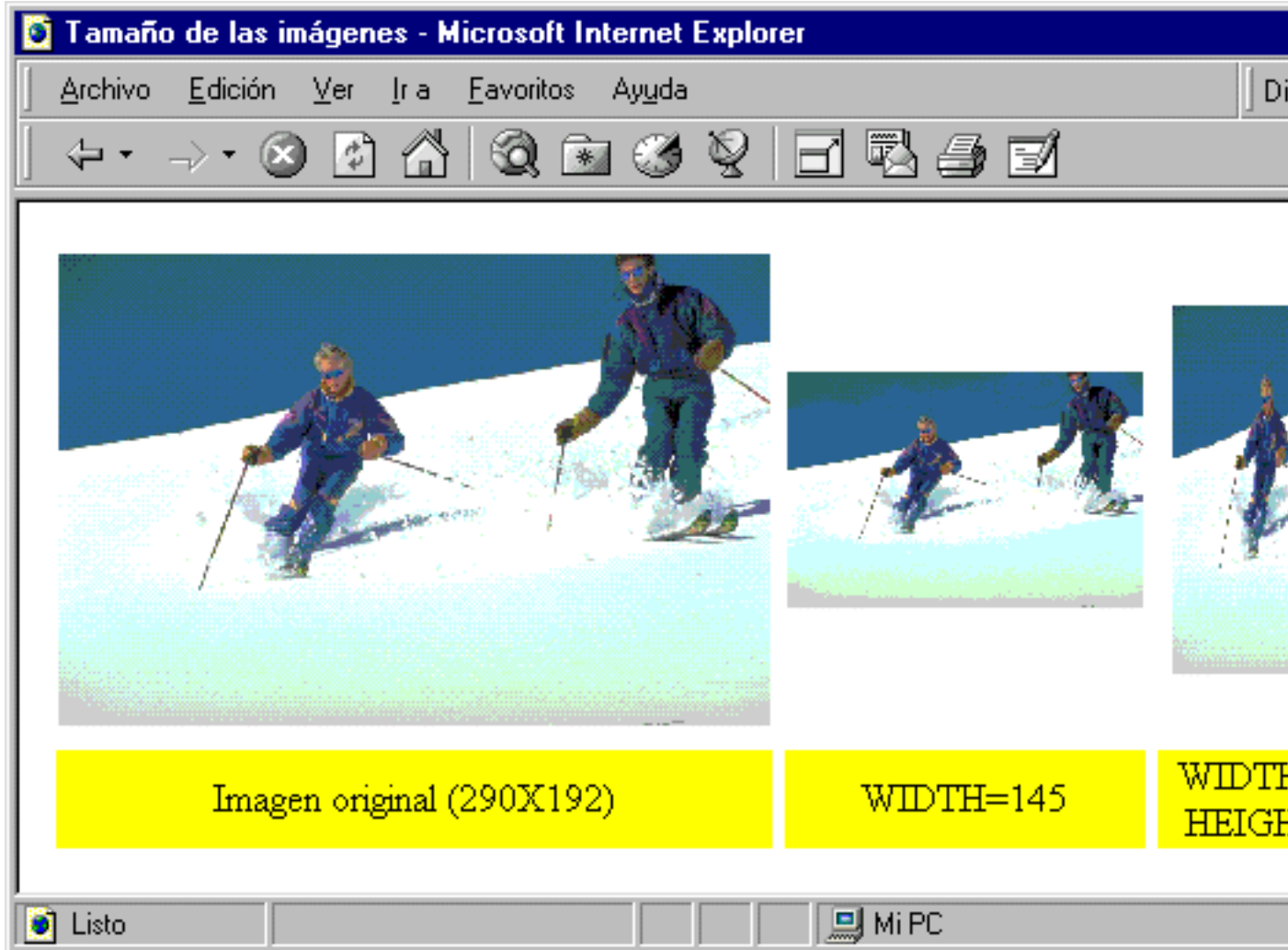
```
<IMG SRC="imagen.gif" WIDTH=250>
```

La imagen será mostrada por el navegador con una tamaño de 250x125. Como vemos la relación entre la anchura y la altura sigue siendo la misma, y por tanto la imagen no aparecerá deformada. Si lo que queríamos era conseguir un tamaño de 250x250 debíamos haber escrito el siguiente código:

```
<IMG SRC="imagen.gif" WIDTH=250 HEIGHT=250>
```

En la figura 5.23 hemos usado una imagen de tamaño original 290x192 y la hemos aplicado diferentes transformaciones.

Figura 5.23. Usando los atributos **WIDTH** y **HEIGHT** podemos variar el tamaño de las imágenes. Si sólo usamos uno la otra dimensión se ajusta para mantener las proporciones.



En la primera de ellas hemos fijado una anchura mitad de la original usando `WIDTH=145`, pero sin incluir el atributo `HEIGHT`. En este caso la altura se ajusta automáticamente a 96 pixels (la mitad del original) para que se mantengan las proporciones. En la siguiente imagen hemos usando los dos atributos, tanto `HEIGHT` como `WIDTH` con lo cual obtenemos una imagen con las dimensiones deseadas, pero deformada, ya que no se mantienen las proporciones originales.

En general no se usan los atributos `WIDTH` y `HEIGHT` para obtener imágenes de mayor tamaño que el original, ya que el resultado obtenido tendría muy poca calidad. El lector puede hacer la prueba.

Nota

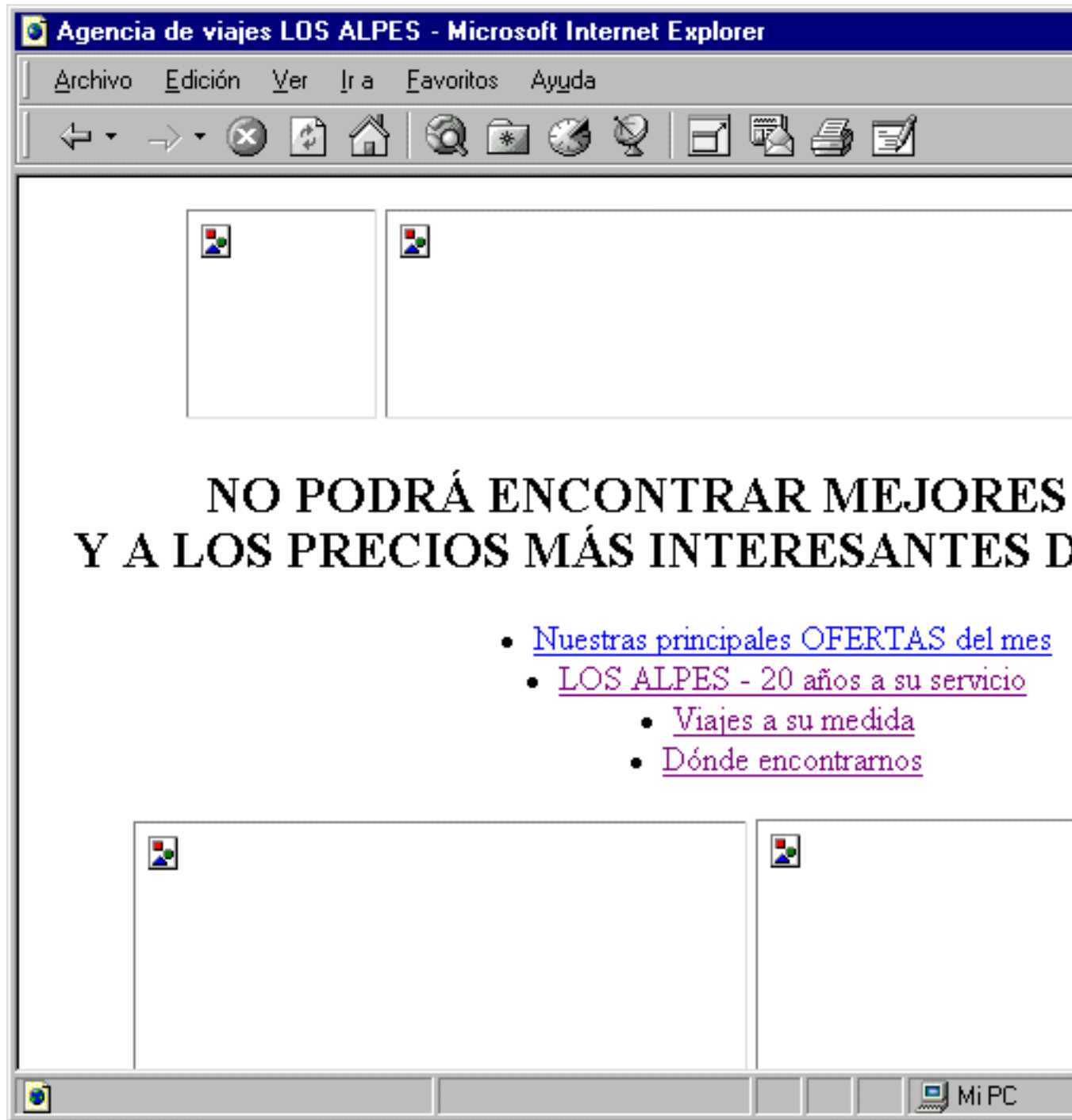
El método que usan la mayoría de los navegadores para reducir el tamaño de las imágenes consiste en eliminar filas y columnas completas de píxeles elegidas aleatoriamente. Para aumentar el tamaño se duplican filas y columnas elegidas de manera igualmente aleatoria. Este método es usado por su rapidez, pero en ocasiones la calidad ofrecida no es la deseable, especialmente si en la imagen hay texto. Por

esta razón si es importante una buena visualización de la imagen conviene usar un programa de retoque fotográfico para modificar el tamaño de la imagen en vez de usar los atributos WIDTH y HEIGHT.

Vamos a terminar esta sección dedicada a cambiar la apariencia de las imágenes de una página Web con un interesante *truco*. Gracias a él conseguiremos una visualización más rápida de nuestras páginas. La idea es simple, y consiste en incluir SIEMPRE el tamaño de las imágenes con los atributos WIDTH y HEIGHT (es necesario incluir los dos) al insertar una imagen en la página. Este truquillo que parece inocente en un principio tiene dos importantes ventajas:

1. Los navegadores cuando les llega una imagen *calculan* sus dimensiones para saber como deben mostrarlas, al dárselas directamente con el código ganaremos el tiempo que el navegador perdería realizando este cálculo.
2. La segunda, y quizá más importante ventaja, consiste en que el navegador podrá colocar todos los elementos de la página desde un principio sin esperar a que lleguen las imágenes. Para entender porque ocurre esto es necesario aprender un poco más sobre el funcionamiento del protocolo HTTP. Sin adentrarnos en detalles muy técnicos diremos que los navegadores, cuando se les pide una página, cargan (o bajan) del WWW el archivo HTML con el código. Tras leerlo y mostrar el texto empiezan a cargar los archivos con las imágenes. Cuando llega una imagen el navegador la coloca en su sitio, pero como no sabía cuanto iba a ocupar de antemano probablemente le había dejado un espacio demasiado pequeño, con lo cual es necesario recolocar todos los elementos de la página, ¡y esto ocurre con cada imagen! Al incluir los tamaños de las imágenes en el código el navegador reserva el espacio adecuado y evita tener que recolocar todo sucesivas veces con el consiguiente ahorro de tiempo. En la figura 5.24 podemos ver la página de la práctica 2 antes tras leerse el código, pero antes de mostrar las imágenes. Como vemos, el navegador (Explorer™ en este caso) ya ha reservado el tamaño para las imágenes que quedan por llegar.

Figura 5.24. Si incluimos junto con la imagen la información de sus dimensiones el navegador podrá reservar espacio para ellas con sólo leer el código.



En ocasiones este ahorro de tiempo es considerable, por lo que nuestra recomendación es que, sin duda, merece la pena usar este truco. Para averiguar el tamaño de las imágenes podemos recurrir a programas de retoque o visualización de imágenes como Paint Shop Pro™ (Shareware) o IrfanView™ (Freeware), aunque los más modernos editores de HTML como FrontPage™ o HomeSite™ realizan esta labor por nosotros.

Alternativas a las imágenes inline

El atributo ALT

Hasta ahora hemos hablado de la importancia de las imágenes y el diseño gráfico en general en el World Wide Web. Sin duda la abundancia de gráficos y otros contenidos multimedia han provocado su gran expansión. Sin embargo, como ya hemos comentado antes a lo largo de esta obra, existen algunos navegadores que sólo son capaces de mostrar texto. También hemos mencionado e incluso hemos mostrado imágenes del más conocido y usado: Lynx™. Quizá el lector se haga aún la pregunta ¿porqué querría alguien acceder al WWW para ver sólo texto? Las razones pueden ser de distinto tipo, pero podemos encontrar dos que tienen gran importancia. La primera consiste en que es posible que el *navegante* acceda a la red desde su trabajo o desde el laboratorio de una universidad donde los equipos son todavía antiguos o trabaja con terminales conectados a una estación de trabajo, que sólo son capaces de mostrar texto, en estos casos es obligatorio usar un navegador *en modo texto*. Otra razón puede ser que aún teniendo un magnífico ordenador (y monitor) último modelo, queramos acceder al WWW a toda velocidad para acceder a cierta información, en este caso los navegadores modo texto son enormemente útiles ya que alcanzan velocidades mucho mayores.

Sea como fuere nuestra labor como programadores de páginas Web será conseguir que nuestras páginas puedan ser visitadas por todo el mundo. De esta manera tendrán más éxito. Por ello debemos ofrecer alternativas a las imágenes que incluimos en nuestras páginas.

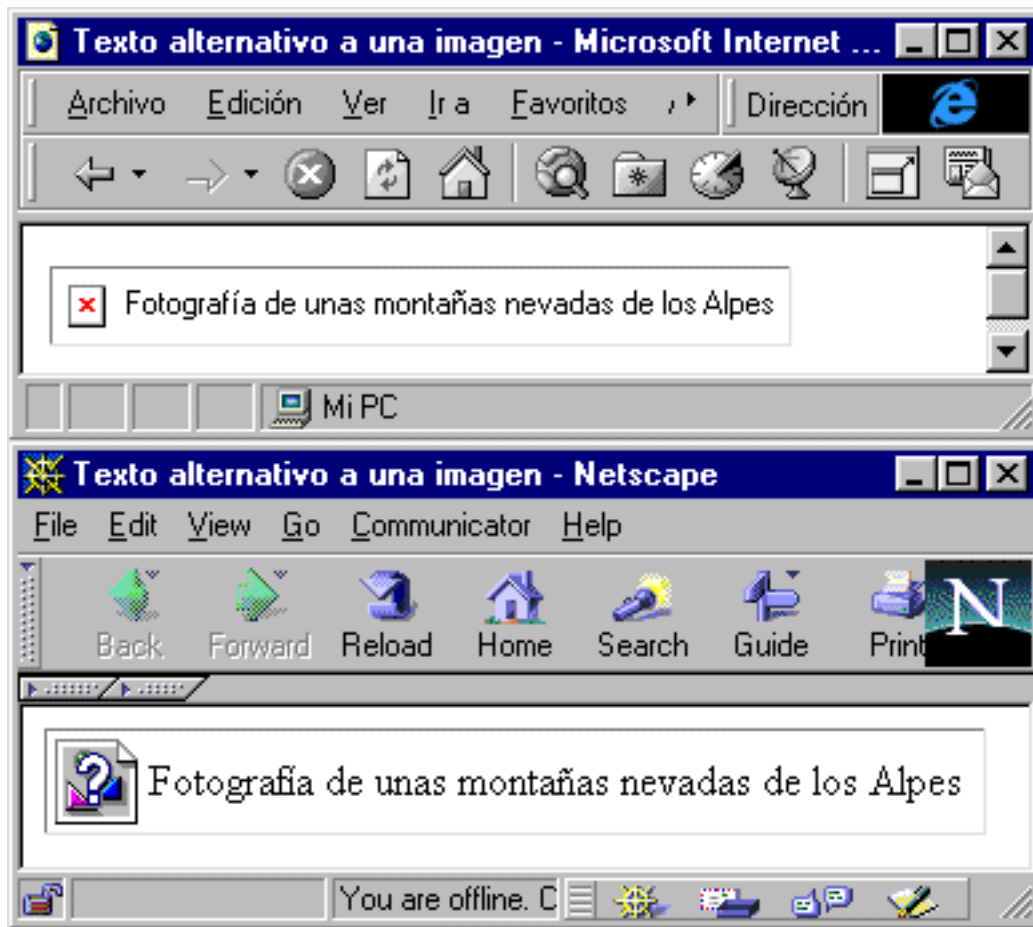
El atributo ALT, de la etiqueta IMG, permite especificar un texto alternativo que será mostrado por los navegadores en caso de que la imagen no pueda ser vista. Ese texto será introducido entrecomillado como valor del atributo ALT, de la siguiente forma:

```
<IMG SRC="montanas.gif"  
ALT="Fotografía de unas montañas nevadas de  
los Alpes">
```

Las razones por las que una imagen puede no ser mostrada son diversas pero las podemos agrupar en tres:

- El primer caso es el anteriormente comentado de navegadores en modo texto. En este caso es necesario hacer un comentario especial. Cada vez que el navegador se encuentra con una imagen (es decir con una etiqueta IMG en el código HTML) en vez de mostrarla la sustituye por el siguiente texto: [IMAGE] (*image* significa imagen en inglés). Sin duda esta palabra no es muy descriptiva, y además es la misma para todas las imágenes. Para solucionarlo el programador puede usar el atributo ALT, en ese caso el texto [IMAGE] será sustituido por el que se indique como valor de este atributo, que sin duda explicará mejor el contenido de la imagen. En ocasiones es conveniente usar el atributo ALT sin ningún texto (ALT="") para evitar la aparición de [IMAGE], pero esto no debe hacerse muy a menudo.
- Un segundo caso puede ser que el usuario haya decidido desactivar la opción de ver las imágenes de las páginas. Tanto Explorer™ (en Opciones de Internet/Opciones avanzadas/Multimedia) como Netscape™ (en Preferences/Advanced) ofrecen esta opción. Si ponemos un texto alternativo suficientemente descriptivo el visitante podrá decidir si merece la pena cargar la imagen o no.
- Un tercer caso ocurre cuando hay un problema en la transmisión de la imagen o porque se ha indicado un nombre incorrecto. En este caso los navegadores muestran un símbolo gráfico acompañado del texto alternativo. En la figura 5.25 podemos ver este caso con Explorer™ y Netscape™, donde hemos usado el código anterior usando el atributo ALT para que sea mostrado un texto alternativo.

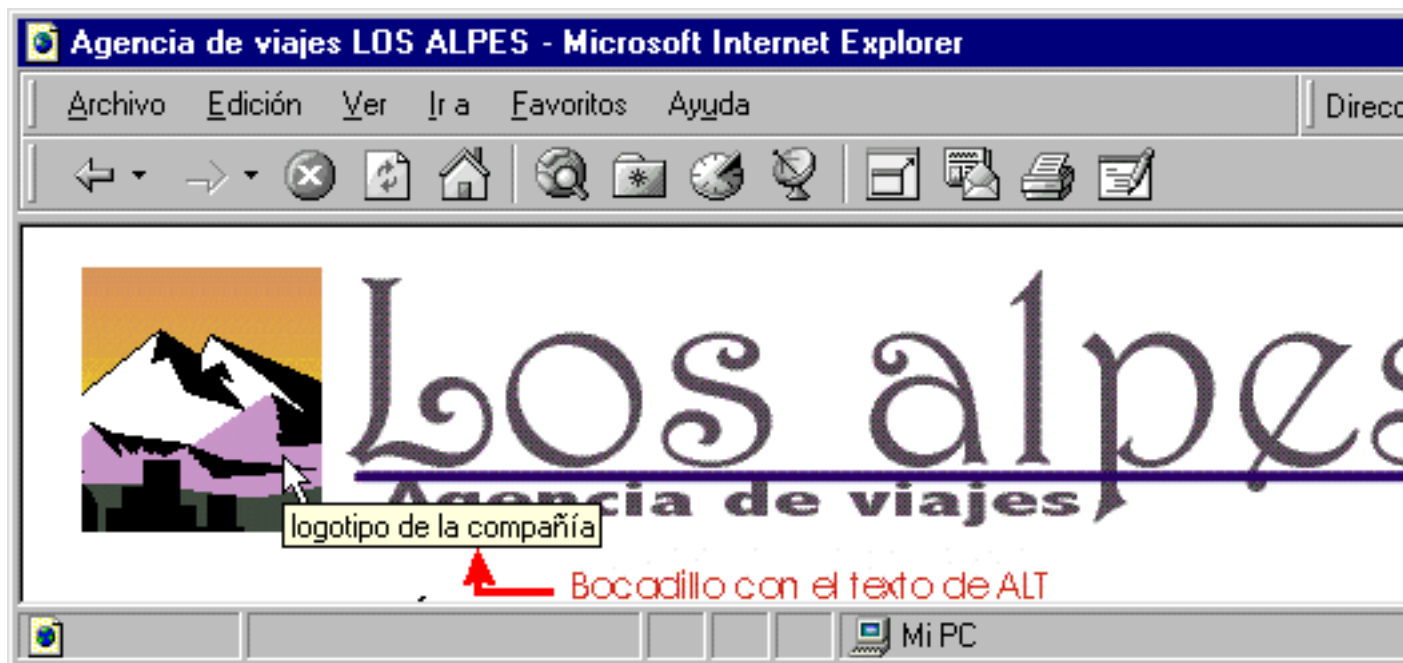
Figura 5.25. Con el atributo ALT podemos especificar un texto alternativo a las imágenes. En esta figura vemos como lo muestran Explorer™ y Netscape™.



En previsión de todos estos casos el uso del atributo ALT en las páginas que programemos es más que recomendable. De esta forma estaremos dando calidad a nuestras páginas y estaremos mostrando que nos preocupamos por todo tipo de audiencia.

Para terminar hemos de decir que los navegadores Internet Explorer™ y Netscape™ dan un segundo uso al atributo ALT. Éste consiste en mostrar su contenido en forma de bocadillo cuando el usuario sitúa el puntero del ratón encima de la imagen tal y como podemos ver en la figura 5.26:

Figura 5.26. Algunos navegadores como Explorer™ muestran el texto de ALT como un bocadillo que aparece cuando ponemos el cursor encima de la imagen.



en la que hemos modificado el código usado en la práctica 2 para añadir el atributo ALT:

```
<IMG SRC="img/logo.gif"
ALT="Logotipo de la compañía">
```

Si bien el uso del atributo ALT ofrece un interesante método para ofrecer una alternativa a las imágenes con un texto existe otro con una utilidad mucho más amplia y que consiste en usar, en lugar de las imágenes *inline* vistas hasta ahora, las llamadas imágenes externas.

Imágenes externas

Estas imágenes no se visualizan junto con el resto de la página, sino que se proporciona al visitante un enlace para que pueda acceder a ellas. Las razones por las que en ciertas ocasiones se usan las imágenes externas y por las que merece la pena mencionarlas aquí son muchas, pero las podemos resumir en cuatro:

- La gran mayoría de navegadores gráficos soportan los formatos GIF y JPEG, y algunos de ellos soportan algún otro formato pero no es lo habitual. Sin embargo gran parte de los navegadores, gráficos o no, soportan un rango mucho más amplio de formatos de imágenes con la ayuda de aplicaciones externas, por lo que usando imágenes externas podremos usar otros formatos distintos de GIF y JPEG y el navegador se encargará de llamar a la aplicación que necesite para visualizarlo.

- Como hemos comentado repetidas veces los navegadores en modo texto no pueden mostrar imágenes *inline*, sin embargo si pueden mostrar imágenes externas ayudándose de algún programa de visualización o edición gráfica.
- Mientras las imágenes *inline* deben tener un tamaño reducido para no provocar grandes esperas al navegante al visitar nuestra página, las imágenes externas pueden ser de mayor tamaño ya que sólo las verán aquellos que lo *pidan* y por tanto hemos de suponer que les interesa lo suficiente como para esperar. En todo caso si el tamaño supera los 100 Kbytes es conveniente indicar dicho tamaño entre paréntesis junto con el texto identificativo de la imagen.
- Nos ofrece la posibilidad de combinar imágenes *inline* con imágenes externas para que las páginas se carguen más rápidamente. De esta forma crearemos los llamados *thumbnails* que veremos con más detalle un poco más adelante.

La creación de imágenes externas varía de las vistas hasta ahora en que ya no se usa la etiqueta IMG. A cambio haremos uso de la etiqueta de creación de links: `<A . . . > `, pero en esta ocasión lo enlazado no será otra página HTML sino la propia imagen externa. El código que debemos usar es:

```
<A HREF="imagen.xxx">
imagen externa
</A>
```

siendo xxx la extensión correspondiente al formato de la imagen. En la tabla 5.2 podemos ver algunos formatos y extensiones comúnmente usados.

Figura 5.2. Extensiones y breve descripción de los principales formatos de imágenes usados en el WWW.

FORMATO	EXTENSIÓN	BREVE DESCRIPCIÓN
GIF	.gif	Imagen de 256 colores sin pérdidas
JPEG	.jpg, .jpeg	Imagen de 16 millones de colores. Con pérdidas
BMP	.bmp	Imagen bitmap de Windows y OS/2
XBM	.xbm	Imagen bitmap de UNIX
PICT	.pict	Imagen Bitmap de Macintosh
PNG	.png	Portable Network Graphics: 16 millones de colores
TIFF	.tif	Baja compresión pero gran calidad
XPM	.xpm	Imagen de Paintbrush

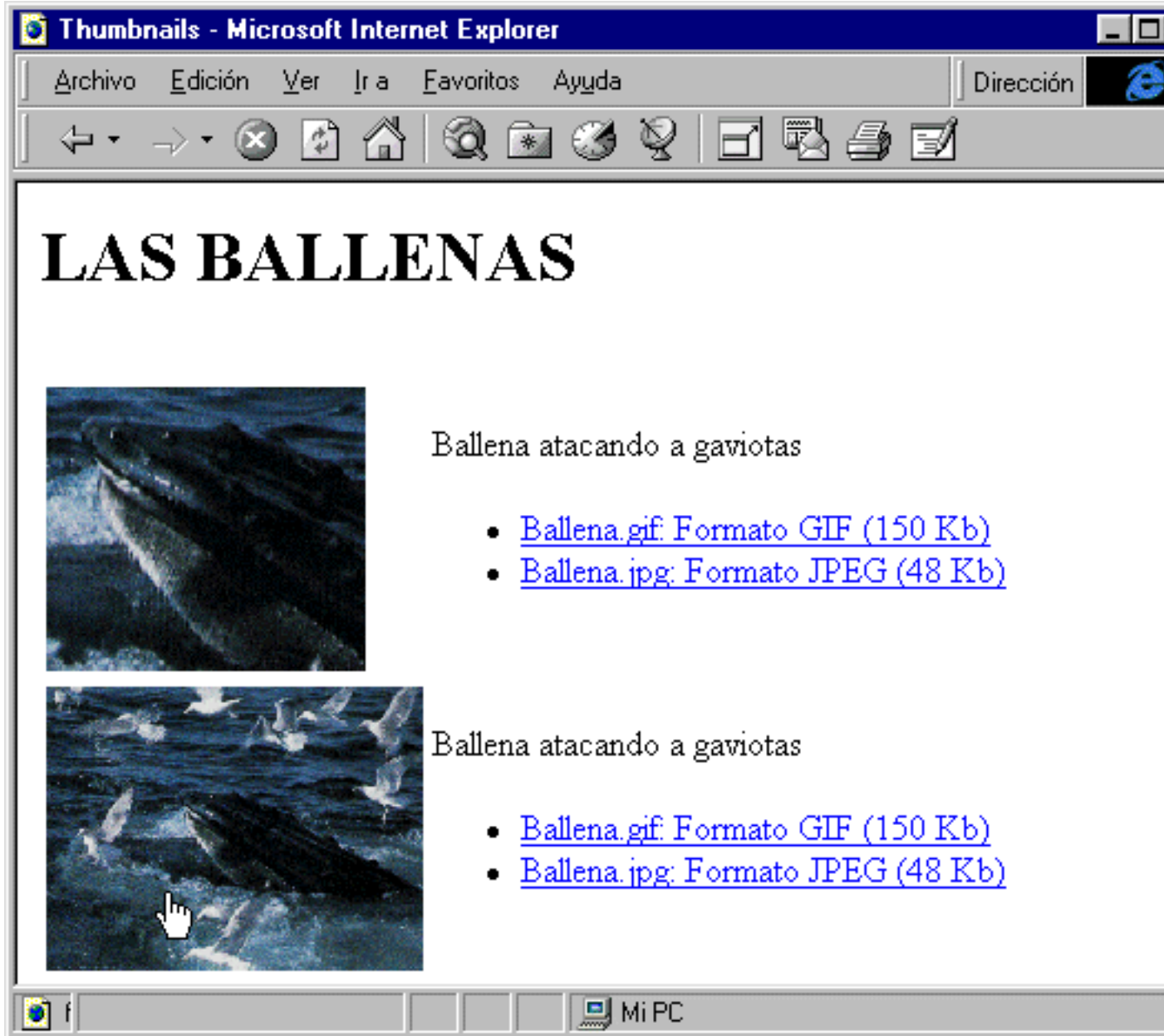
Cuando el visitante pulse sobre el texto activo, en este caso '*imagen externa*' el navegador bajará la imagen de Internet y luego la mostrará o ejecutará otra aplicación que pueda mostrarla.

Aunque hasta ahora hemos usado los links para enlazar otras páginas HTML y en este mismo apartado para insertar imágenes externas en realidad la utilidad de esta etiqueta va mucho más allá. De hecho podemos enlazar a nuestra página cualquier tipo de archivo, en la mayoría de los navegadores actuales cuando el usuario selecciona un enlace a un archivo que ellos mismos no pueden visualizar le ofrece dos opciones: guardar el archivo en el disco duro o llevar a cabo una acción en función del tipo de archivo. Esta última opción consiste en ejecutar el archivo si es ejecutable, usar un programa descompresor si es un archivo comprimido, visualizarlo con un programa adecuado si es un vídeo, etc.

Thumbnails

Como hemos adelantado antes, con este nombre se conoce al resultado de la combinación de imágenes *inline* con imágenes externas. El objetivo de los thumbnails es proporcionar una previsualización, un aperitivo, de la imagen completa (que será externa) usando una imagen *inline* de menor tamaño. Esta imagen *inline* es, de hecho, la que recibe el nombre de thumbnail, y será o bien una versión de dimensiones reducidas de la imagen final o una parte de ella. En general, o bien el propio thumbnail, o bien un texto adjunto (o ambas cosas) servirán como enlace para obtener la imagen completa tal y como podemos ver en la figura 5.27.

Figura 5.27. Ejemplos de Thumbnails. En el primer caso se muestra un trozo de la imagen y en el segundo la imagen completa reducida, el efecto es similar en ambos casos.



que muestra dos ejemplos de thumbnails de la imagen completa de la figura 5.28.

Figura 5.28. Imagen completa que será mostrada cuando el visitante seleccione los enlaces que acompañan a los thumbnails de la figura 5.27



Como vemos, el uso de thumbnails permite al visitante si decide cargar la imagen completa elegir entre el formato GIF, de mayor tamaño y calidad, o en formato JPEG con calidad ligeramente peor pero mucho menor tamaño. Sobre las imágenes *de aperitivo* hay que decir que en el primer caso se ha usado un trozo de la imagen mientras que en el segundo se ha usado la imagen entera que ha sido reducida en tamaño con un programa de edición de imágenes. En ambos casos el tamaño de la imagen a quedado reducido a unos 15-20Kb que las hace más apropiadas para incluirlas como imágenes *inline*.

Aviso

Es importante resaltar que para la creación de thumbnails no se pueden usar los atributos WIDTH y HEIGHT ya que se perdería su utilidad. El objetivo del thumbnail es tener una imagen en un archivo aparte que sea de menor tamaño en bytes que la original para que cargue más rápido. Al usar WIDTH y HEIGHT la imagen sigue siendo la misma, y por tanto su tamaño también, luego no conseguiremos reducir el tiempo de carga.

El uso apropiado imágenes externas y de thumbnails nos permitirá crear páginas de gran contenido gráfico y de pequeño tamaño.

Imágenes mapa

Empezamos a adentrarnos ya en algunas de las características más avanzadas del lenguaje HTML. En este caso es el turno de las imágenes mapa.

Con bastante probabilidad el lector habrá visitado alguna página en la que había alguna imagen en la que se podía pulsar en diferentes partes para acceder a diferentes destinos. Estas son las llamadas imágenes mapa y en la gran mayoría de páginas Web podemos encontrar una. En un principio para realizar las imágenes mapa era necesario usar determinados programas especiales que debían estar en el servidor Web, sin embargo con el estándar HTML 3.2 se incluyeron algunas etiquetas HTML que nos permitirán realizar estas imágenes mapa con mayor facilidad.

Imágenes mapa en el servidor

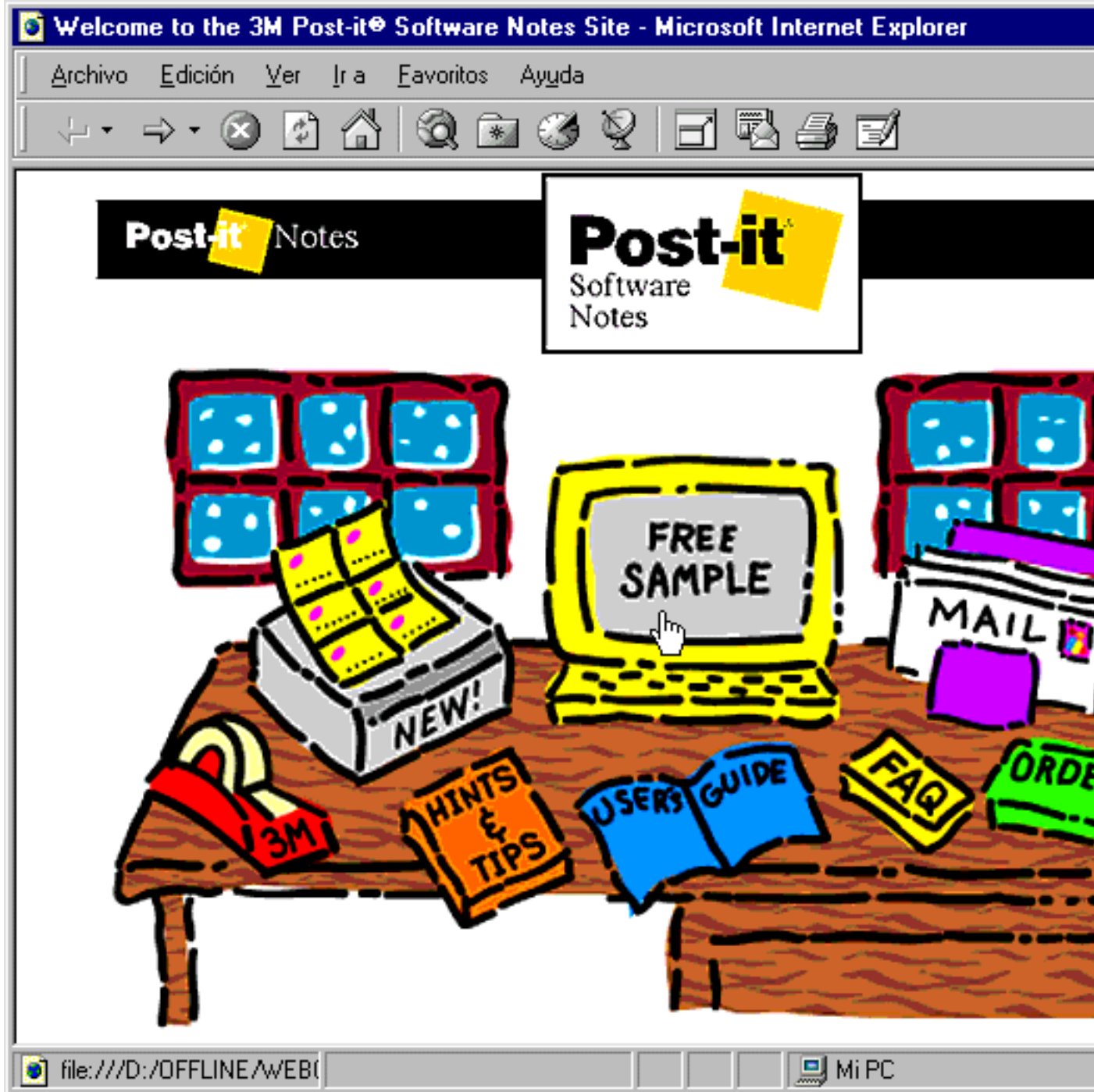
Este era el único método usado inicialmente para crear las imágenes mapa, para usarlo debemos instalar en el servidor Web (si tenemos acceso a él) algún programa dedicado a este fin. Los más usados son los del CERN y NCSA. Para usar este método debemos llevar a cabo los siguientes pasos:

1. Crear un archivo de configuración del mapa con el código del programa usado para indicar las zonas activas y donde nos dirigirán el pulsar sobre ellas. Para averiguar como es el código a usar tendremos que leer la documentación que acompañe al programa, pero en general no será muy complicado. El archivo con el código tiene en general la terminación '.map'.
2. Escribir el código HTML de manera que la imagen sea un enlace al archivo con su código. Además debemos incluir el atributo *ISMAP* en la etiqueta IMG:

```
<A HREF="img.map"><
<IMG SRC="img.gif" ISMAP>
</A>
```

En la página de POST-IT, figura 5.29, podemos ver un ejemplo de imagen mapa en el servidor.

Figura 5.29. En la página principal de POST-IT vemos un excelente ejemplo de la utilidad de las imágenes mapa.



En este caso la imagen representa una mesa con diferentes objetos encima cada uno de los cuales representa una sección del sitio Web al que saltaremos si pulsamos sobre él.

La gran ventaja de este método consiste en que funcionará con cualquier navegador, el único requisito es tener el programa en el servidor Web. El principal inconveniente es que, por lo general no todo el mundo tiene acceso al

servidor Web y no puede instalar los programas que necesita, por otro lado tener el código del mapa en un archivo aparte complica su modificación.

Para resolver estos problemas apareció el segundo método en el que el código de todas las imágenes mapa de una página podrán estar en esa misma página y además se creará usando etiquetas del lenguaje HTML.

Imágenes mapa en el cliente

Este tipo de imágenes reciben el nombre de imágenes mapa en el cliente porque será el navegador de cada usuario (cada cliente) el que se encargará de leer el código cuando se pulsa sobre la imagen mapa y averiguar la dirección de destino. No es necesario ningún programa especial en el servidor Web.

En este caso nos bastará usar la etiqueta IMG para crear la imagen, eso sí con un nuevo atributo: *USEMAP*. Por otro lado ahora deberemos usar algunas otras etiquetas nuevas para crear el mapa de la imagen. La creación de este tipo de imágenes mapa consta por tanto de dos pasos:

1. Definición de la imagen como imagen mapa, así como informar de dónde está el mapa a usar. La sintaxis es sencilla:

```
<IMG SRC="imagen.gif"  
USEMAP="#mapa1">
```

La imagen es, en este caso, una con formato GIF de nombre *imagen.gif*. En este ejemplo usamos un mapa llamado *mapa1*, como vemos la forma de referirnos a él es igual que con las anclas: anteponiendo el símbolo '#'. En esta ocasión el mapa estará en el mismo documento HTML, pero esto no tiene porque ser así. Si, por ejemplo, usamos una imagen mapa en todas y cada una de las imágenes de un sitio Web, podemos poner el mapa en la página principal (*index.htm*) y después referirnos a él como:

D1

```
<IMG SRC="imagen.gif"  
USEMAP="index.htm#mapa1">
```

Más aún, no estamos limitados a que el mapa esté en nuestro sitio Web, puede estar en cualquier lugar del WWW, para referirnos a él tendremos que poner la dirección URL de la página dónde esté seguido por un # y el nombre del mapa.

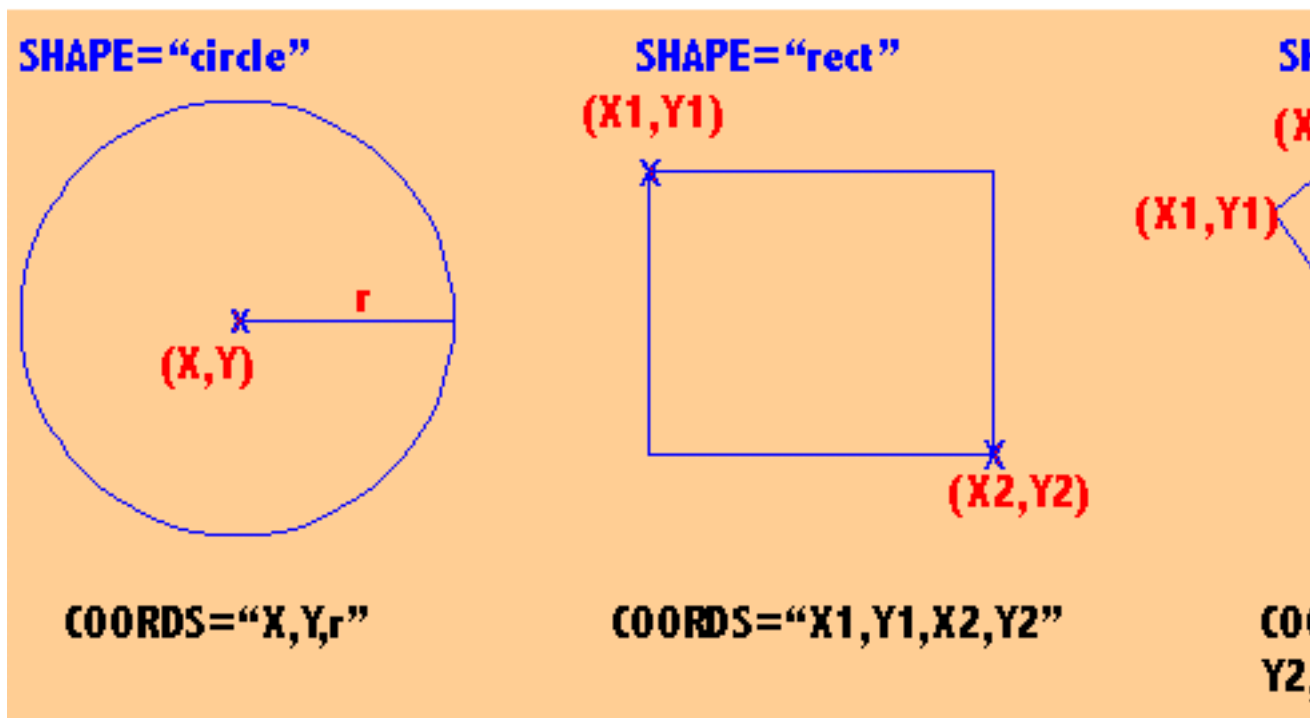
2. El segundo paso es la creación del mapa. Para ello usaremos una nueva etiqueta: *MAP*. Esta etiqueta tendrá una instrucción de inicio, `<MAP . . . >`, y otra de fin, `</MAP>` entre las cuales definiremos las diferentes *zonas activas* de la imagen mapa. El único atributo de esta etiqueta es el necesario para darle un nombre: *NAME*, y se usará de la siguiente manera:

```
<MAP NAME="mapa1">
. . .
</MAP>
```

A continuación debemos crear las formas a las que nos referíamos antes. Para llevar a cabo esta labor usaremos otra etiqueta más: *AREA* que deberá ser una vez para cada zona de la imagen mapa y que consta de cuatro atributos:

- **HREF:** Este atributo es idéntico al del mismo nombre de la etiqueta *A*. Con él debemos indicar la página o dirección URL a la que *saltaremos* cuando pulsemos sobre esta zona activa.
- **ALT:** Sirve para incluir un texto alternativo.
- **SHAPE:** Define la forma de la zona, puede tener tres valores: *rect* (zona rectangular), *circle* (zona circular) y *poly* (zona poligonal).
- **COORDS:** Coordenadas de la zona, su valor depende de la forma de esta. En la figura 5.30:

Figura 5.30. En esta figura podemos ver los valores que puede tomar el atributo COORDS en función de si la zona es rectangular, circular o poligonal.



podemos ver como será en función de si el valor de SHAPE es...

- *rect*: COORDS toma 4 valores separados por comas que representan las coordenadas de el vértice superior izquierdo y el inferior derecho.
- *circle*: COORDS toma 3 valores. Los dos primeros son las coordenadas del centro y el tercero es el radio.
- *poly*: COORDS tiene un número indefinido de valores que representan las coordenadas de cada uno de sus vértices.

Como ejemplo vamos a imaginar una imagen de 300x100 y vamos a crear un mapa que la divida en dos:

```
<MAP NAME="mapa2">

  <AREA SHAPE="rect"
  COORDS="0,0,150,100"
  HREF="izda.htm">

  <AREA SHAPE="rect"
  COORDS="150,0,300,100"
  HREF="dcha.htm">

</MAP>
```

No ha sido complicado, ¿no?

Nota

Las coordenadas vistas hasta ahora están dadas en pixeles. También pueden darse en porcentajes, aunque esto sólo suele ser útil para el caso de zonas rectangulares. La sintaxis a usar sería por ejemplo:
COORDS="0,0,50%,40%"

Existe un quinto atributo, NOHREF, que puede usarse en lugar de HREF para indicar que una determinada zona no debe conducir a ningún sitio al pulsar sobre ella. Este atributo no toma ningún valor.

Si dos o más zonas se superponen se impone la que fuera definida primero. Por esta razón si existe alguna zona con NOHREF suele ponerse la primera de la lista.

El problema las imágenes mapa en cliente radica en que no todos los navegadores están preparado para este tipo de imágenes mapa, y por tanto no podemos estar seguros de que todos los visitantes vayan a poder *disfrutar* de nuestra imagen mapa.

Nota

Siempre que usemos una imagen enlace y especialmente si esta es una imagen mapa es importante crear esos mismos enlaces con textos para que se pueda acceder a ellos con navegadores de sólo texto o por si existe algún problema con la visualización de las imágenes.

Aviso

Hay que tener mucho cuidado con el uso de los atributos `WIDTH` y `HEIGHT` con las imágenes mapa, ya que la imagen variará sus dimensiones, pero el mapa no, por tanto las zonas que hayamos definido no servirán.

EJERCICIO 5.1: Llegados a este punto queremos proponer un ejercicio sencillo a los lectores. Éste consistirá en la creación de un mapa para la imagen de la figura 5.31:

Figura 5.31. Imagen mapa que puede sustituir el menu en forma de lista de la página de la agencia de viajes Los Alpes.



incluida en el archivo `ej5-1.zip`, y usarla para sustituir el menú creado con una lista en la página de la agencia de viajes Los Alpes creada en la práctica 2. Además habrá que proporcionar enlaces en modo texto alternativos a la imagen (lo habitual es que estén en una sola línea. En el mismo archivo (`extra.zip`) se ofrece una solución a este ejercicio.

Mejoras en las imágenes

Tanto el formato GIF como el formato JPEG pueden ser vistos con casi la práctica totalidad de los navegadores. Sin embargo ¿Cuándo es conveniente usar uno y cuando es conveniente usar el otro?, ¿Puedo convertir las imágenes de un tipo a otro?, ¿Qué ventajas ofrece cada uno?. Conocer las respuestas a estas preguntas nos permitirá usar de una manera mucho más eficiente y profesional las imágenes en nuestras páginas. En este apartado nos dedicaremos a responderlas y veremos las últimas novedades del formato GIF que permite crear imágenes transparentes e imágenes con movimiento. Para terminar comentaremos algunos consejos para aprovechar al máximo y eficientemente las posibilidades del HTML en lo que a las imágenes se refiere.

Formatos GIF y JPEG

El formato *GIF* (Graphics Interchange Format, Formato de intercambio de gráficos) fue creado por CompuServe con el objetivo de que fuese entendido por todo tipo de ordenadores. La primera versión de este formato es actualmente conocida como GIF87 y sus principales características son:

- Limitada a 256 colores, si usamos menos la imagen tendrá menor tamaño.
- Compresión sin pérdidas.

Lo que le convierte en un formato idóneo para logos, iconos e todo tipo de dibujos con colores sólidos. Poco después apareció una segunda versión del formato conocido como GIF89a que añadía las siguientes características:

- Posibilidad de crear imágenes con entrelazado.
- Transparencia.
- Imágenes con movimiento.

Actualmente los principales problemas del formato GIF es su limitación 256 colores, su mala compresión de fotografías y que está patentada por CompuServe, que quiere empezar a cobrar por su uso

El formato *JPEG* (Joint Photographic Experts Group) fue creado por un grupo de expertos en fotografía por lo que está pensado especialmente para este tipo de imágenes. Sus principales características son:

- Puede almacenar 16 millones de colores. Además, al contrario de lo que ocurría con el formato GIF, el tamaño es independiente del número de colores usados.
- Ofrece un algoritmo de compresión que reduce grandemente el tamaño de las fotografías.
- Esta compresión produce pérdidas, es decir, la imagen sufre deterioro.

Esta última característica es el gran contratiempo que tiene el formato JPEG pero en la mayoría de las ocasiones es inapreciable. Al comprimir una imagen en formato JPEG se puede elegir un nivel de compresión de 1 a 99. Cuanto mayor sea este nivel mayor será la compresión pero también habrá un mayor deterioro de la señal. En la figura 5.32:

Figura 5.32. Mientras que el tamaño de la imagen JPEG es mucho menor la pérdida de calidad apenas es apreciable.



Imagen con formato GIF
Tamaño = 13 Kb



Imagen con formato JPEG
Tamaño = 3.6 Kb

podemos ver una comparación entre la imagen de la ballena en formato GIF y formato JPEG con un nivel de compresión 80 (lo habitual es 20). Existe pérdida de calidad, pero no es muy apreciable, mientras que la disminución de tamaño es impresionante y para imágenes de mayor tamaño la diferencia es aún mayor.

Una vez conocidas las características de los dos formatos ha llegado el momento de decidir cuando es conveniente usar uno y cuando es conveniente usar el otro. Por lo general:

- Conviene usar *GIF* si la imagen es pequeña y hay pocos colores, con mayor motivo aún si hay bloques de la imagen con todos los pixels del mismo color, como ocurre en los logos y dibujos similares.
- Es recomendable usar *JPEG* siempre que la imagen sea muy grande y/o tenga muchos colores como sucede en las fotografías.

EJERCICIO 5.2: Llegados a este punto podemos darnos cuenta que en la práctica 2 usamos el formato GIF en algunas fotografías y que el resultado hubiese sido mucho mejor usando el formato JPEG. Este segundo ejercicio tendrá como objetivo usar un programa de retoque fotográfico para convertir de formato estas imágenes (*montanas.gif* y *ski.gif*). Los programas que pueden usarse para llevar a cabo este propósito son muchos: PhotoShop™ y Paint Shop Pro™ son una buena muestra de ellos. En el archivo *ej5-2.zip* se incluyen las dos imágenes ya convertidas con un nivel de compresión 60. De esta forma hemos conseguido reducir *montanas.gif* de 13 Kb a 6'5 Kb y *ski.gif* de 10 Kb a 4'6 Kb. Aunque en un principio puede parecer poco cuando una página tiene bastantes imágenes escogiendo adecuadamente el formato de cada uno conseguiremos que cargue mucho más rápido.

Además de GIF y JPEG existen otros formatos que no son usados habitualmente en el WWW, pero merece la pena comentar los formatos que *entienden* los dos principales navegadores:

- Netscape™ soporta los formatos de imágenes GIF, JPEG, XPM y XBM.
- Explorer™ soporta los formatos GIF, JPEG, PNG y BMP.

Entre todos estos hay uno que destaca: *PNG* (Portable Network Graphics), porque representa el futuro de los formatos de almacenamiento de imágenes y combina las ventajas de GIF y JPEG junto con otras nuevas:

- Buena compresión tanto para logos como para fotos.
- Posibilidad de transparencia y entrelazado.
- 16 Millones de colores.
- Compresión sin pérdidas.
- Capacidad de añadir comentarios e información extra junto con la imagen.

El formato PNG está pensado como un sustituto para el formato GIF, ya que JPEG seguirá siendo mejor con las fotografías. Por otro lado todavía habrá que esperar algún tiempo para que sea ampliamente aceptado. Aquellos lectores interesados en el tema pueden obtener más información al respecto en las siguientes direcciones:

- <http://www.boutell.com/boutell/png>
- <http://quest.jpl.nasa.gov/PNG/>

Transparencia, Entrelazado y GIFs animados

Estas son las tres principales características que fueron añadidas al formato GIF en la revisión GIF89a. Vayamos una por una:

- Las imágenes transparentes consisten simplemente en elegir un color de la imagen que no será *pintado* por el visualizador (en nuestro caso será el navegador) al mostrarla. El uso de imágenes transparentes es especialmente útil cuando usamos una página con fondos. Por ejemplo, en la figura 5.33:

Figura 5.33. En ocasiones, especialmente si usamos una imagen como fondo, se consigue un efecto mucho mejor haciendo que nuestras imágenes sean transparentes.



podemos ver la imagen de Los Alpes con transparencia y sin transparencia en una página con una imagen de fondo. En general las únicas imágenes que podemos considerar como candidatas a convertirse en imágenes transparentes son aquellas con un color de fondo dominante. Si intentáramos hacer transparente una fotografía los resultados no serían los deseados.

- Las imágenes entrelazadas son imágenes muy similares a las normales. La única diferencia radica en la manera en que se va mostrando la imagen en el navegador según llega. Cuando un navegador carga una imagen normal va mostrándola desde el principio y va avanzando hacia abajo. Las imágenes entrelazadas por el contrario se muestran al principio como una imagen borrosa que se va aclarando según la imagen termina de llegar, esto ocurre por el método especial de almacenamiento que se usa en este tipo de imágenes. El entrelazado es especialmente útil cuando se aplica a las imágenes mapa, ya que en ocasiones no es necesario ver la imagen a la perfección para elegir una opción.
- La tercera novedad del formato GIF89a es probablemente la que más éxito ha tenido. Como el lector ya ha adivinado estamos hablando de las imágenes animadas o *GIFs animados*. En los primeros años del WWW la única forma de tener una animación en nuestras páginas consistía en usar un applet (programilla) de Java™ o similar, con los inconvenientes que esto representaba. Con la aparición de los GIFs animados todo se simplificó, estos en realidad no son más que varias imágenes en formato GIF almacenadas en un único archivo y que se van alternando ordenadamente para crear el efecto de la animación. Cada una de estas imágenes deben ser creadas por separado y después juntadas con algún programa especial como Microsoft

Gif Animator™, GIF Movie Gear™, etc. Otra manera de obtener imágenes es acudir a las numerosas páginas de internet con colecciones de estas imágenes a nuestra disposición. En el directorio público DMOZ puede encontrarse una gran cantidad de enlaces a estas páginas en su sección de GIFs animados [http://dmoz.org/Computers/Graphics/Web/Free/Animated_GIFs/].

Nota

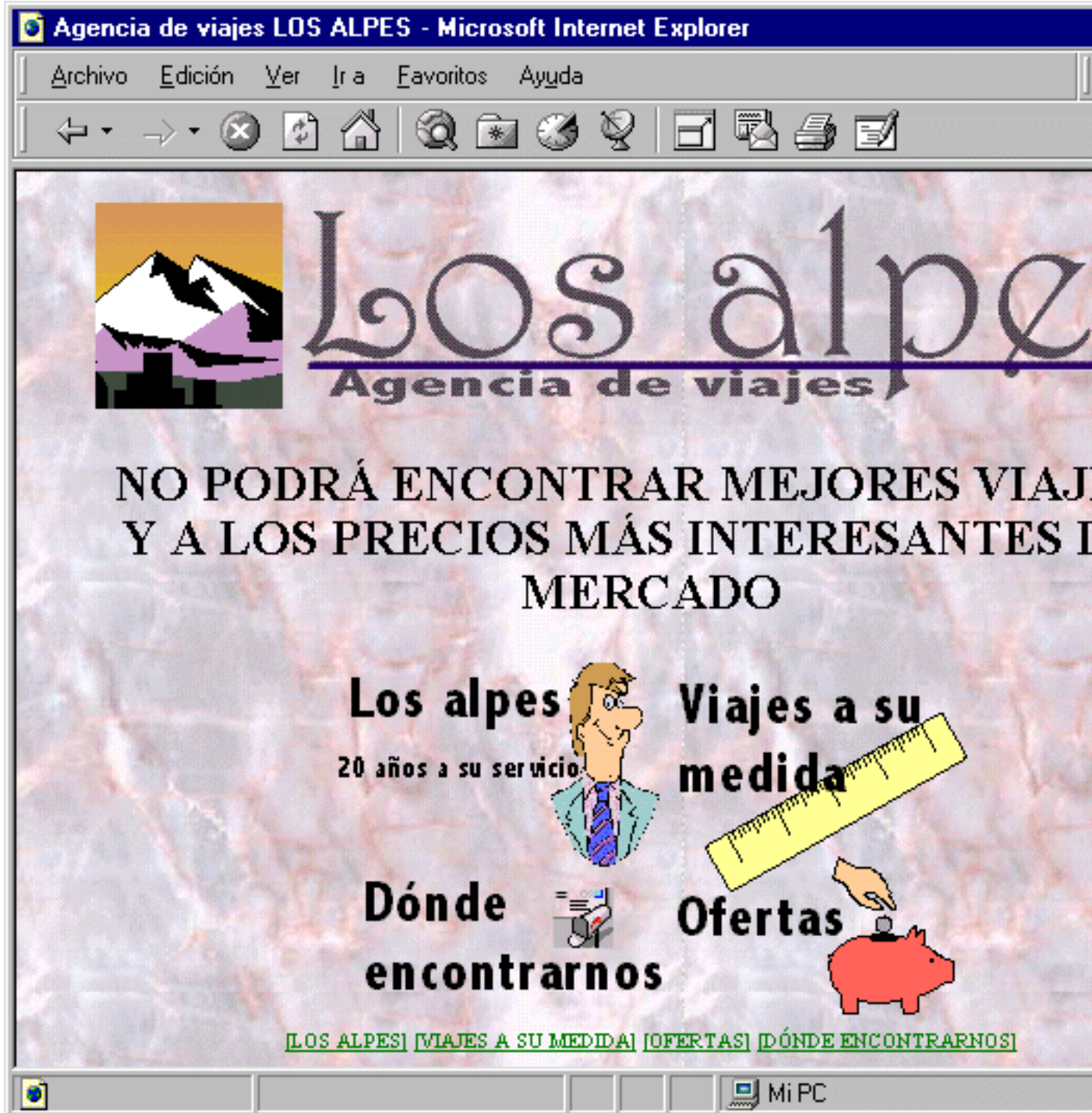
No es conveniente usar muchas imágenes animadas en una misma página. Primero porque ocupan bastante tamaño, segundo porque pueden ser mareantes para el visitante y tercero porque provocarán que el navegador vaya muy lento si el ordenador no tiene mucha memoria. Por estas razones nuestra recomendación es usar un máximo de 2 imágenes animadas por página.

EJERCICIO 5.3: Este es el ejercicio final de este largo capítulo dedicado a las imágenes y al color en las páginas Web. En él pretendemos que el lector aplique los conocimientos que ha adquirido para mejorar el aspecto gráfico de las páginas que creamos en la práctica 2. En concreto proponemos los siguientes cambios:

- Poner un fondo en ambas páginas. El fondo debe ser preferiblemente de tonos claros para que se pueda leer el texto con facilidad. Si es necesario deberemos cambiar el color de este para facilitar la legibilidad.
- Hacer que tanto los enlaces normales y los visitados sean de color verde y los activos morados.
- Añadir los tamaños de las imágenes con los atributos WIDTH y HEIGHT.
- Incluir un texto alternativo (ALT) en todas y cada una de las imágenes.
- Añadir alguna imagen externa en los viajes con temas relacionados con el destinos.
- Mejorar el aspecto gráfico general con ideas propias del lector.
- Usar adecuadamente los tipos de imágenes para conseguir una página con gran contenido gráfico de un tamaño total (el documento HTML y todas las imágenes) de 40 Kb o menor.

En el archivo `clip.zip` [extra/clip.zip] se ofrecen algunas imágenes que pueden ayudar a realizar este ejercicio y en el archivo `pract2++.zip` [extra/pract2++.zip] se muestra una posible solución a este de la que podemos ver un anticipo en la figura 5.34.

Figura 5.34. Aspecto final de la página de Los Alpes después de aplicar los conocimientos sobre imágenes que hemos adquirido en este capítulo.



Creación de tablas HTML

Jorge Ferrer

Victor García

Rodrigo García

Versión 1.0

Copyright Jorge Ferrer, Rodrigo García y Victor García. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation. Puede consultar una copia de la licencia en: <http://www.gnu.org/copyleft/fdl.html>

Con su aparición las tablas y revolucionaron el diseño de las páginas web. Las tablas eran una herramienta perfecta para organizar datos de manera ordenada, pero su utilidad no se queda ahí, ya que escondiendo los bordes podremos usarlas también para definir la estructura de las páginas. En este capítulo aprenderemos todas las etiquetas y atributos que existen en el lenguaje HTML estándar para la creación de tablas y realizaremos numerosos ejemplos y ejercicios para ir cogiendo práctica, que es, al fin y al cabo, lo más importante a la hora de sacar el máximo partido a las tablas.

Armados con nuestros conocimientos sobre la etiqueta `TABLE` y después de haber realizado unos cuantos ejemplos simples para practicar, dedicaremos el resto del capítulo a la realización de dos prácticas. Con ellas nos daremos cuenta de las grandes posibilidades que nos ofrecen las tablas en el diseño de páginas Web.

Con lo que hemos visto hasta ahora la disposición de los elementos de una página web es aún harto difícil. Tras estudiar los métodos de alineamiento empezamos a adquirir un mayor control sobre ellos, pero sin duda fueron las tablas las que van a darnos una mayor flexibilidad para decidir la disposición de los elementos en una página. En un principio las tablas pueden servir para mostrar una serie de datos de una manera ordenada, tal y como estamos acostumbrados a ver en numerosas situaciones, sin embargo las tablas en la web tienen una utilidad aún mayor. Podemos hacer que toda nuestra página, o una parte de ella, sea una tabla e ir colocando los diferentes elementos de los que conste en sus celdas. De esta forma sabremos con gran precisión donde estamos colocándolas. En la página de GNOME Hispano, figura 7.1, podemos ver un excelente ejemplo de ello.

Figura 7.1. En el web de GNOME Hispano encontramos multitud de elementos, gráficos y texto, con una disposición muy cuidada. ¿Cómo lo hacen? Con tablas.



Un inconveniente de las tablas radica en que si bien los navegadores gráficos tienen una gran libertad para colocar y distribuir los elementos, esto no ocurre así con los navegadores en modo texto. Por ello cuando vemos una página con tablas en uno de estos navegadores nuestra cuidada presentación puede caer por los suelos. Sin hablar de navegadores para ciegos o los usados en los coches, que leen las páginas. En estos casos el usar tablas para controlar la apariencia provoca que el navegador se líe y no interprete bien la página.

Por supuesto existen métodos o trucos para evitarlo en lo posible, y merecerán nuestra atención una vez nos hayamos convertido en unos expertos en la creación de tablas.

Cómo se crea una tabla

Para crear las tablas será necesario usar una nueva etiqueta: *TABLE*, que significa, en inglés, tabla. Esta etiqueta consta de instrucción de inicio, `<TABLE>`, e instrucción de fin, `</TABLE>`. Entre ambas debemos introducir otras tres etiquetas, que definirán la estructura de la tabla. Estas etiquetas son:

- **TR**: La etiqueta *Table Row* nos permitirá insertar filas en la tabla. La tabla tendrá tantas filas como apariciones de esta etiqueta haya entre `<TABLE>` y `</TABLE>`. La instrucción de inicio de TR, `<TR>`, marca el comienzo de la línea, la instrucción de fin, `</TR>`, marca su final. La instrucción de fin es optativa, si no se usa se considera que una línea ha acabado cuando comienza otra o cuando acaba la tabla. Entre el comienzo y el fin de la línea debemos insertar las celdas de la tabla.
- **TH**: Este es el primero de los dos tipos de celdas existentes en el lenguaje HTML. El nombre de la etiqueta proviene del inglés *Table Header*, encabezado de tabla. Al igual que la etiqueta anterior el comienzo del encabezado viene dado por la instrucción de inicio, `<TH>` y finaliza con la instrucción de fin, `</TH>`, o con el comienzo de otra celda. En general entre el comienzo y el fin se suele insertar texto, que será mostrado en negrita o subrayado y centrado.
- **TD**: Este es el segundo tipo de celda de las tablas HTML. El nombre de la etiqueta, *Table Data* (Datos de tabla), nos induce a pensar que su función será introducir todos los datos que queramos en las celdillas definidas de esta forma. Y así será, de hecho podemos insertar cualquier elemento de HTML: imágenes, listas, texto formateado e incluso otras tablas. La diferencia de esta celda con la anterior es pequeña, y su existencia separada sólo tiene sentido si consideramos el carácter descriptivo del HTML. En la práctica la diferencia principal es que el texto de los encabezados de tabla aparecerá resaltado y centrado y el de las celdas normales (TD) no.

Nota

El uso de las instrucciones de fin de TR, TD y TH es, como hemos dicho, optativo. Sin embargo en las primeras versiones del lenguaje HTML era obligatorio usarlas y por esta razón algunos navegadores antiguos las necesitan para poder entender las tablas. Es por ello que nuestra recomendación es usar siempre dichas instrucciones de fin para seguir con nuestro objetivo de llegar a la mayor cantidad de público posible.

Con una mera descripción de las etiquetas es difícil entender por completo como se crean tablas en HTML, así que vamos a ver unos ejemplos.

Para empezar crearemos una tabla sencilla de dos filas y dos elementos. Vayamos por pasos:

1. En primer lugar ponemos la instrucción de inicio y fin de la etiqueta TABLE dejando un espacio entre ellas para insertar posteriormente el resto de etiquetas:

```
<TABLE>
...Otras etiquetas...
</TABLE>
```

2. A continuación insertamos las filas. La tabla que nos hemos propuesto crear consta de dos filas. Por tanto debemos usar la etiqueta TR dos veces. Lo que insertemos en la primera (entre la instrucción de inicio y la instrucción de fin) será el contenido de la primera fila y lo que insertemos en la segunda será el contenido de la segunda fila:

```
<TABLE>

  <TR> </TR> <TR> </TR>
</TABLE>
```

3. El tercer paso será definir el contenido de cada fila. Como nos habíamos propuesto que la tabla tuviese dos columnas en cada fila debemos insertar dos celdas en cada una. Como hemos explicado antes, para crear una celda debemos usar la etiqueta TD. Entre la instrucción de inicio y la instrucción de fin de esta etiqueta será donde insertaremos el verdadero contenido de la tabla. En este primer ejemplo sencillo será simplemente texto, pero puede ser cualquier otro elemento web como imágenes o listas. Volviendo a nuestro ejemplo, para cada fila debemos escribir el siguiente código:

```
<TR>

  <TD> Columna1 </TD> <TD> Columna2 </TD>
</TR>
```

Nota

En los ejemplos que veremos a lo largo de este curso sangraremos las etiquetas a distintos niveles para que sea más sencillo saber que etiquetas están dentro de otras. No es necesario realizar esto para que el código funcione, de hecho podríamos escribir el código en una sola línea y también funcionaría, pero si es recomendable ya que nos facilita mucho su lectura.

4. El cuarto paso es ponerlo todo junto, repitiendo el código anterior para cada fila. El código final de la tabla será:

```
<TABLE>
<TR>
  <TD>
Fila1,Columna1
</TD>
  <TD>
Fila1,Columna2
</TD>
</TR>
<TR>
  <TD>
Fila2,Columna1
</TD>
  <TD>
Fila2,Columna2
</TD>
```



```
</TR>  
</TABLE>
```

En la figura 7.2 podemos ver como muestra Explorer [™] esta tabla.

Figura 7.2. Ya hemos creado nuestra primera tabla, aunque es sencilla ya podemos observar varias cosas curiosas en ella, por ejemplo que por defecto no se dibujan los

bordes.

Welcome to CERN – Konqueror

Dirección Editar Ver Ir Marcadores Herramientas Opciones Ventana Ayuda

Dirección <http://public.web.cern.ch/Public/>



[Version française](#)

THIS SITE

- [Languages](#)
- [Help](#)
- [Contents](#)
- [Credits](#)
- [Your comments](#)
- [Related Links](#)
- [Experiments](#)

GENERAL

- [Users Pages](#)
- [Visit CERN & Microcosm](#)
- [Jobs & Training Opportunities](#)
-
-

Welcome to CERN
European Organization for Nuclear Physics
where the Web was born

Oct 27–28: **United Nations Open Days**
Nov 8–11: **Life in the Universe**
LIVEfromCERN: watch our webcasts on antimatter

Welcome message from CERN's Director General

- [CERN in 2 minutes](#)
- [People and Organization](#)
- [Explore the Atom](#)
- [Accelerators](#)
- [Technology](#)
- [Education](#)

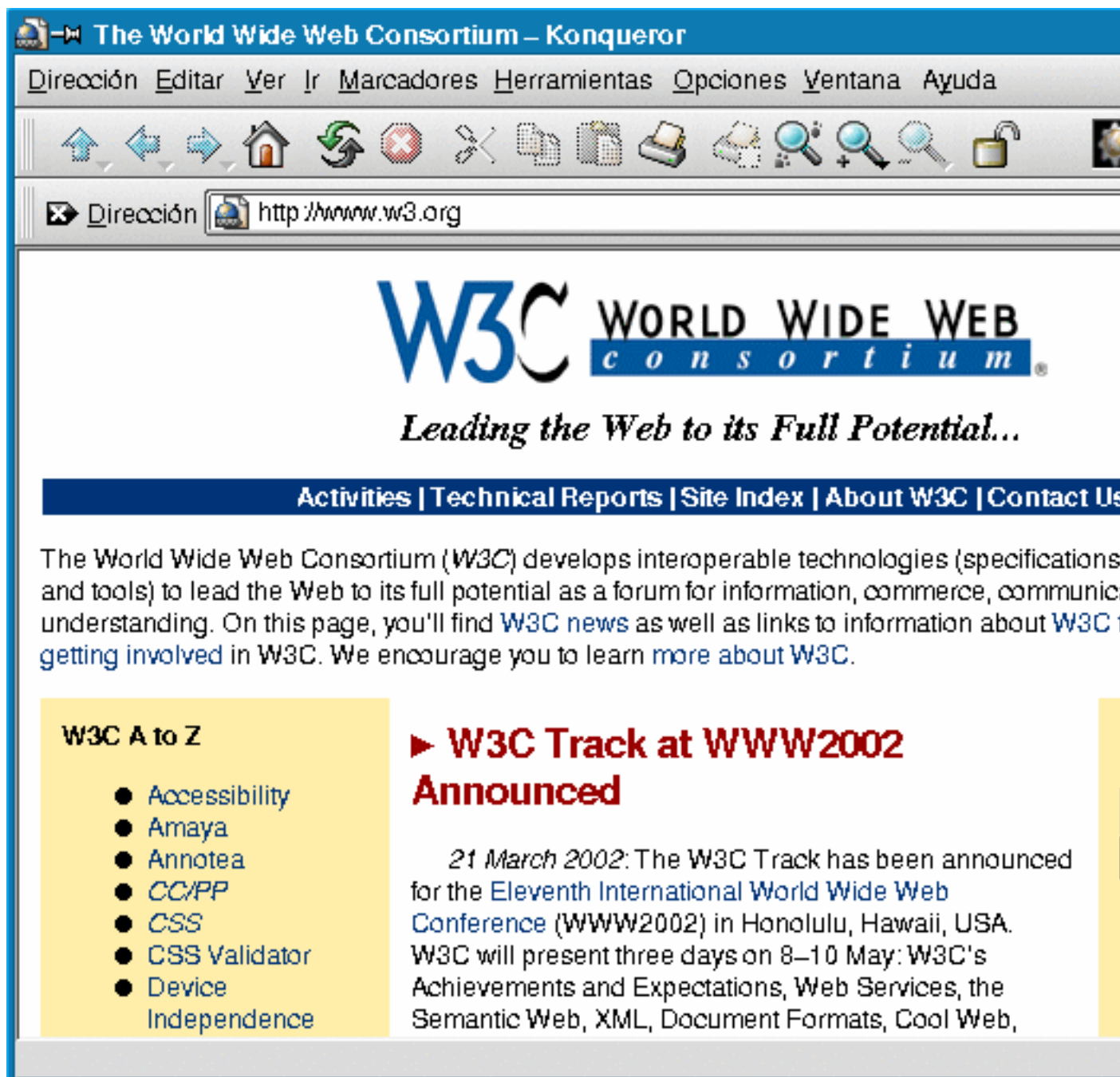
Quizá el hecho que más nos sorprenda es que la tabla no tiene bordes, y que las celdas no están dibujadas. Más adelante comentaremos algo más sobre ello y aprenderemos a incluir bordes de diferentes tamaños. Lo que más nos interesa ahora es comprender la estructura de la tabla anterior. Como hemos podido comprobar, esta tabla se va creando por filas, y dentro de cada fila se insertan las celdas, cada una de las cuales aparecerán en una columna.

A continuación mostramos una tabla ligeramente más complicada, en la que, ahora sí, haremos uso de los encabezados de tabla. En ella observaremos además que ocurre cuando una fila tiene más celdas que otra. Para la creación del código de esta tabla habrá que llevar a cabo los mismos pasos que en el ejemplo anterior. El código final que define la tabla será:

```
</TABLE>
```

Este ejemplo es mostrado en la figura 7.3.

Figura 7.3. En esta segunda tabla ya podemos ver como son mostrados los encabezados de tabla.



Como habíamos comentado, el texto de los encabezados es resaltado, en este caso (usando Explorer™) es mostrado en negrita y centrado con respecto al resto de los elementos de su columna. Por otro lado en este ejemplo observamos el efecto de la existencia de una fila con más celdas (más columnas) que el resto. En la primera fila insertamos tres celdas usando la etiqueta TH y sin embargo en las siguientes filas sólo creamos dos celdas. Aunque lo pueda parecer esto no es ningún problema. El navegador creará una tabla de 3 filas y 3 columnas, pero algunas de las celdas estarán vacías.

Borde de las tablas

El siguiente paso en el aprendizaje de las tablas consiste en añadirles un borde. Esto es tan sencillo como añadir un nuevo atributo a la instrucción de inicio de la etiqueta TABLE. El atributo del que hablamos es BORDER, que significa borde en inglés. Este atributo puede tomar un valor en pixeles que representa el grosor del borde a mostrar. Si incluimos BORDER sin ningún valor es equivalente a BORDER="1". Repitamos el segundo de los ejemplos vistos antes pero añadiéndole un borde de grosor 5.

```
<TABLE BORDER="5">
<TR>
<TH>
Encabezado1
</TH>
<TH>
Encabezado2
</TH>
<TH>
Encabezado3
</TH>
</TR>
<TR>
<TD>
Fila1,Columna1
</TD>
<TD>
Fila1,Columna2
</TD>
</TR>
<TR>
<TD>
Fila2,Columna1
</TD>
<TD>
Fila2,Columna2
</TD>
</TR>
</TABLE>
```

En la figura 7.4 podemos ver el resultado de este código. En ella observamos como Explorer™ le da un aspecto tridimensional al borde de las páginas.

Figura 7.4. Tabla con bordes. En esta tabla podemos observar lo que ocurre cuando una fila tiene más celdas que otra o cuando insertamos una celda sin nada dentro.



Un aspecto interesante a resaltar de la figura 7.4 es que las celdillas que están vacías, es decir, no tienen ni texto ni ningún otro elemento en su interior no son dibujadas por los navegadores. En este caso ni siquiera habíamos puesto las instrucciones `<TD>` y `</TD>` correspondientes a esas celdas, aunque si las hubiésemos puesto, pero sin ningún contenido dentro, es decir, si hubiéramos escrito:

```
<TR>
<TD>
Fila2,Columna1
</TD>
<TD>
Fila2,Columna2
</TD>
<TD></TD>
</TR>
```

El resultado habría sido el mismo.

Si queremos que sí sean dibujadas debemos insertar la etiqueta TD con un salto de línea (`
`) o un espacio creado con una secuencia de escape, ` `, para definirla. Esto es, en el caso anterior podíamos haber escrito.

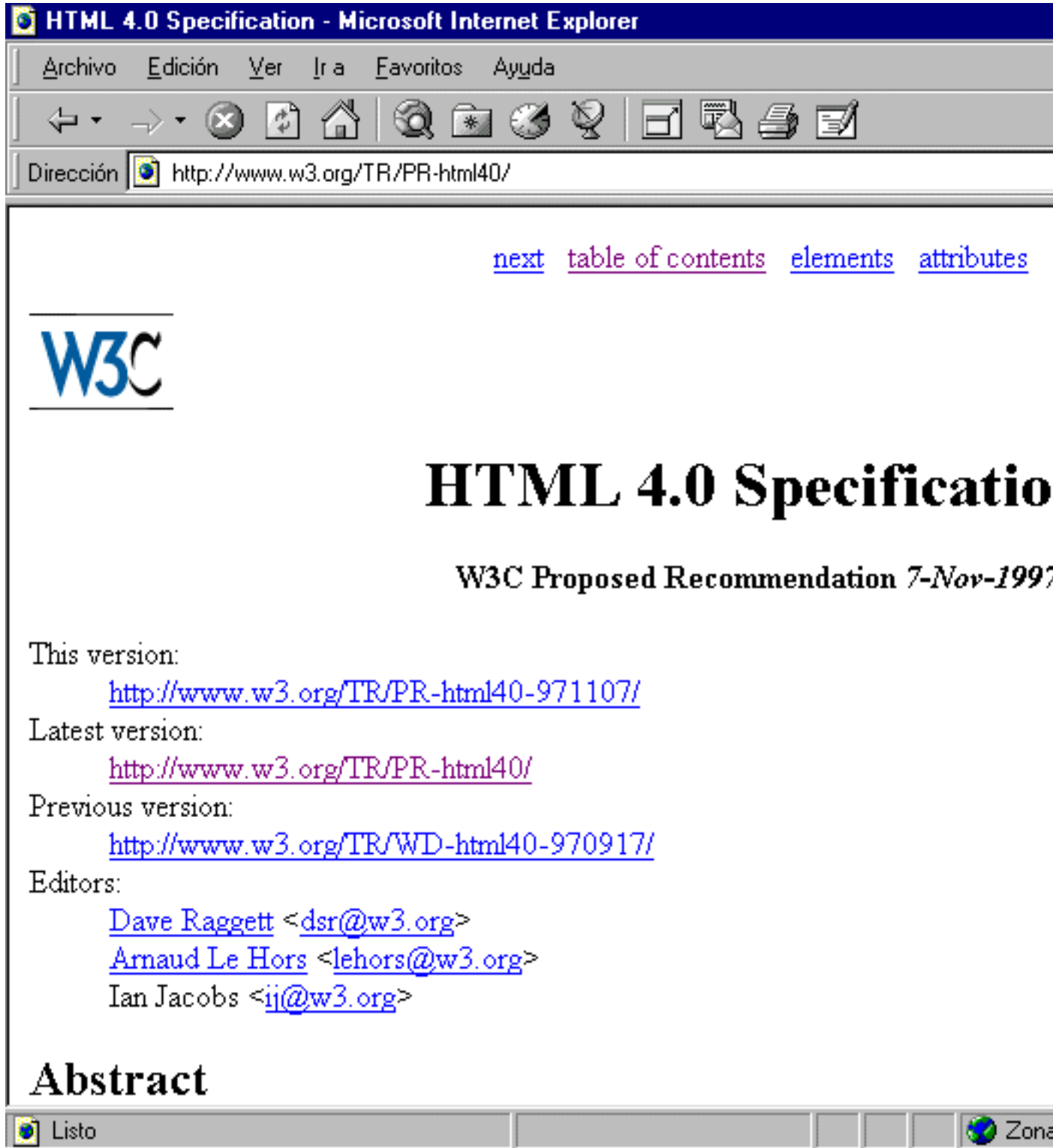
```
<TABLE BORDER="5">
<TR>
  <TH>
Encabezado1
</TH>
  <TH>
Encabezado2
</TH>
  <TH>
Encabezado3
</TH>
</TR>
<TR>
  <TD>
Fila1,Columna1
</TD>
  <TD>
Fila1,Columna2
</TD>
  <TD>
  
</TD>
</TR>
<TR>
  <TD>
Fila2,Columna1
</TD>
  <TD>
Fila2,Columna2
</TD>
  <TD>
```



```
</TD>  
</TR>  
</TABLE>
```

En este caso habríamos obtenido el resultado de la figura 7.5.

Figura 7.5. Tabla con celdas vacías. Poniendo `
` o `
` las celdas que antes no existían ahora si son dibujadas, pero sin nada dentro.



Como vemos, `
` y `
` consiguen el mismo resultado.

Insertar un título para la tabla

El siguiente elemento importante a destacar en las tablas y del que todavía no hemos hablado es el título. Este será un texto que acompaña a la tabla podrá estar situado encima o debajo de ella. Para insertarlo haremos uso de una nueva etiqueta: *CAPTION* (que se puede traducir al español como título), que deberá ser usada entre las instrucciones `<TABLE>` y `</TABLE >`, de la siguiente manera:

```
<TABLE>

  <CAPTION> Título de la tabla </CAPTION>
  ... Código de la tabla ...
</TABLE>
```

El texto que servirá de título será insertado entre las instrucciones de inicio y fin. Este texto podrá ser formateado por otras etiquetas siempre y cuando sean de nivel de texto, esto es, que su inclusión no provoque un salto de línea. Esto, desgraciadamente, excluye a los encabezados (H1 ,H2, etc.), pero no a las etiquetas de nivel de texto o a FONT.

Los títulos de la tabla serán mostrados por defecto en la parte superior de esta, pero la etiqueta *CAPTION* permite el uso de un atributo (cuyo nombre nos resultará familiar), *ALIGN* , que nos permitirá elegir que dichos títulos sean mostrados como pie de la tabla. El atributo *ALIGN*, puede tomar dos valores *top* y *bottom*. El primero de ellos, *top*, significa en inglés arriba o parte superior y su efecto será que el título sea mostrado encima de la tabla. *Bottom*, por su parte puede ser traducido como abajo o parte inferior. El uso de este atributo provocará el efecto antes comentado: el título será mostrado al pie de la tabla. El siguiente código es un ejemplo del uso este tipo de alineamiento del título:

```
<TABLE>

  <CAPTION ALIGN="bottom"> Pie de la tabla </CAPTION>
  ... Código de la tabla ...
  </TABLE>
```

A continuación veremos un ejemplo completo de tabla con título. Es necesario resaltar dos aspectos de este ejemplo. El primero es que aunque usemos *ALIGN="top"* para el título, en realidad no es necesario, ya que este es el alineamiento que aparece por defecto. El segundo es que para resaltar el título nos hemos ayudado de las etiquetas *BIG* y *B* que ya conocemos, algo perfectamente válido dentro de la etiqueta *CAPTION*.

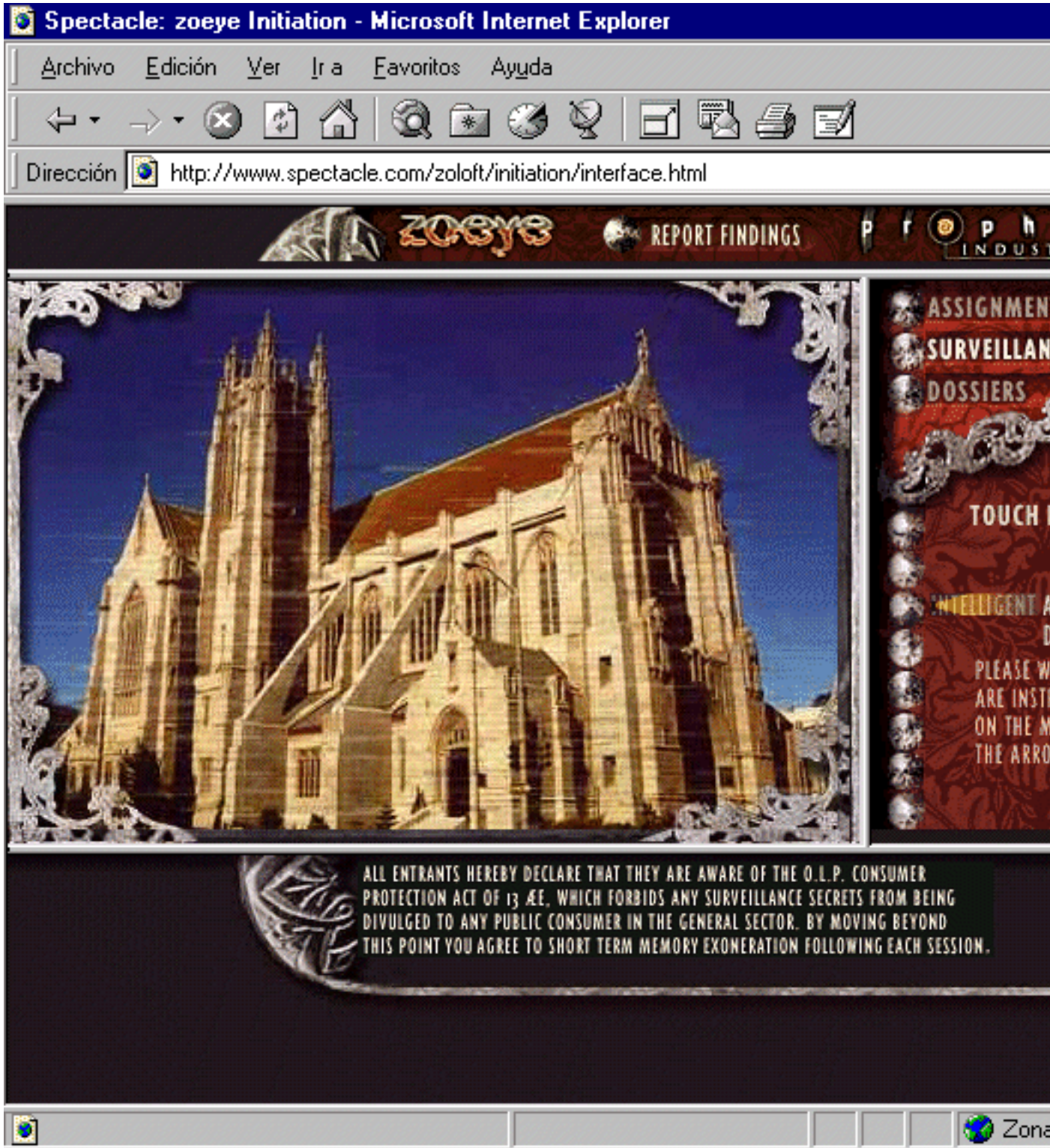
Un ejemplo de inclusión de un encabezado es el siguiente:

```
<TABLE>
<CAPTION ALIGN="top">
  <BIG>
  <B>
Tabla de ejemplo
  </B>
  </BIG>
</CAPTION>
<TR>
```

```
<TD>
Fila1,Columna1
</TD>
<TD>
Fila1,Columna2
</TD>
<TR>
<TD>
Fila2,Columna1
</TD>
<TD>
Fila2,Columna2
</TD>
</TABLE>
```

Cuyo resultado podemos apreciar en la tabla izquierda de la figura 7.6.

Figura 7.6. Con la etiqueta CAPTION podemos poner un título a la tabla. Este título puede situarse encima (ALIGN="top") o debajo (ALIGN="bottom") de la tabla.



La tabla de la derecha corresponde al mismo código pero con ALIGN="bottom".

Tamaño de las tablas

Todavía queda un tema de gran importancia en la creación de tablas que debemos comentar: su tamaño. Quizá tras haber hecho algunos ejemplos el lector se habrá dado cuenta que el tamaño de las tablas no es siempre el mismo. Lo más probable es que este hecho no nos haya sorprendido, aún así es importante la manera en que los navegadores *calculan* el tamaño de las tablas y sus celdas no es tan sencilla y suele ser muy útil conocerla bien. Posteriormente aprenderemos a cambiar este tamaño *a mano*.

Cuando al leer una página el navegador encuentra una tabla calcula cual es el tamaño tanto horizontal como vertical necesario para mostrar el contenido de cada una de las celdas. En este momento hay que tener en cuenta que todas las celdas de una misma columna deben tener un mismo ancho y que todas las celdas de una misma fila deben tener la misma altura. La altura de cada fila quedará determinada por el elemento más alto (o el texto con más líneas) de todas las celdas de la fila. De manera similar se calcula el ancho de las columnas. Vistos estos conceptos probablemente se haya quedado el lector un poco confundido, esta parte es un poco complicada al principio pero se comprende rápidamente una vez hemos practicado un poco con tablas.

Llegados a este punto ya podemos afirmar que sabemos todo lo necesario para crear tablas. Ahora nos queda conocer algunos aspectos más avanzados y realizar algunos ejemplos para adquirir destreza en su uso. Pero antes, practiquemos un poco.

Ejercicio 7.1

En este ejercicio se propone la creación de tablas con elementos de diferentes tamaños en sus celdas para observar como varían sus tamaños. En el archivo `extra.zip` [ejemplos/extra.zip] se ofrecen imágenes de diferentes tamaños como ayuda. Las tablas que se proponen realizar son:

1. Una tabla de 2 filas y 3 columnas. El contenido de todas las celdas será la palabra: TABLA.
2. Se sustituye la palabra TABLA en cada una de las celdas anteriores por un texto lo suficientemente largo para que ocupe varias líneas.
3. Crear dos tablas iguales a las dos anteriores pero con la imagen `montana.gif` en la segunda celda de la primera fila.
4. Combinar el texto escrito en el segundo caso, la imagen `montana.gif`, y la imagen `spot.gif` en diferentes celdas para ver el cambio de tamaño que sufren.

Alineamiento de tablas

Hasta ahora las tablas que hemos visto no tenían ningún tipo de alineamiento especial. Se situaban en el margen izquierdo y se extendían hasta donde ocuparan por la derecha. Pero por supuesto el lenguaje HTML nos permite tener un mayor control sobre la disposición de las tablas con el uso de un nuevo atributo de la etiqueta TABLE: *ALIGN*. Este atributo nos permitirá alinear la tabla a la derecha, a la izquierda o centrada respecto a la ventana del navegador, para ello toma los siguientes valores:

- *ALIGN="left"*: Alinea la tabla a la izquierda de la ventana del navegador. La manera de actuar del navegador cuando se encuentra con una tabla alineada a la izquierda es idéntica a su comportamiento con imágenes con este mismo tipo de alineamiento, y por tanto también podemos conseguir que un texto envuelva la tabla tal y como hacíamos con las figuras y como podemos ver en la figura 7.7.

Figura 7.7. Con el atributo **ALIGN** podemos controlar el alineamiento de la tabla. Si elegimos **ALIGN="left"** el texto bordeará a la tabla al igual que ocurría con las imágenes.



La forma de usar este tipo de alineamiento es:
D1

```
<TABLE ALIGN="left">
...Código de la tabla...
</TABLE>
```

Para escribir texto al final de una tabla alineada a derecha o izquierda deberemos usar, al igual que hacíamos con las imágenes, la etiqueta **BR** con el atributo **CLEAR**. El siguiente ejemplo funcionaría en ambos casos de alineamiento:

```
<BR CLEAR="all">
```

- **ALIGN="right"**: En esta ocasión la tabla será alineada a la derecha. El texto también la envolverá como ocurría con las imágenes alineadas a este lado. En la figura 7.8:

Figura 7.8. Usando alineamiento a la derecha (ALIGN="right") conseguimos un efecto muy similar al alineamiento a la izquierda.



podemos ver el efecto de usar este modo de alineamiento:

D1

```
<TABLE ALIGN="right">
...Código de la tabla...
</TABLE>
```

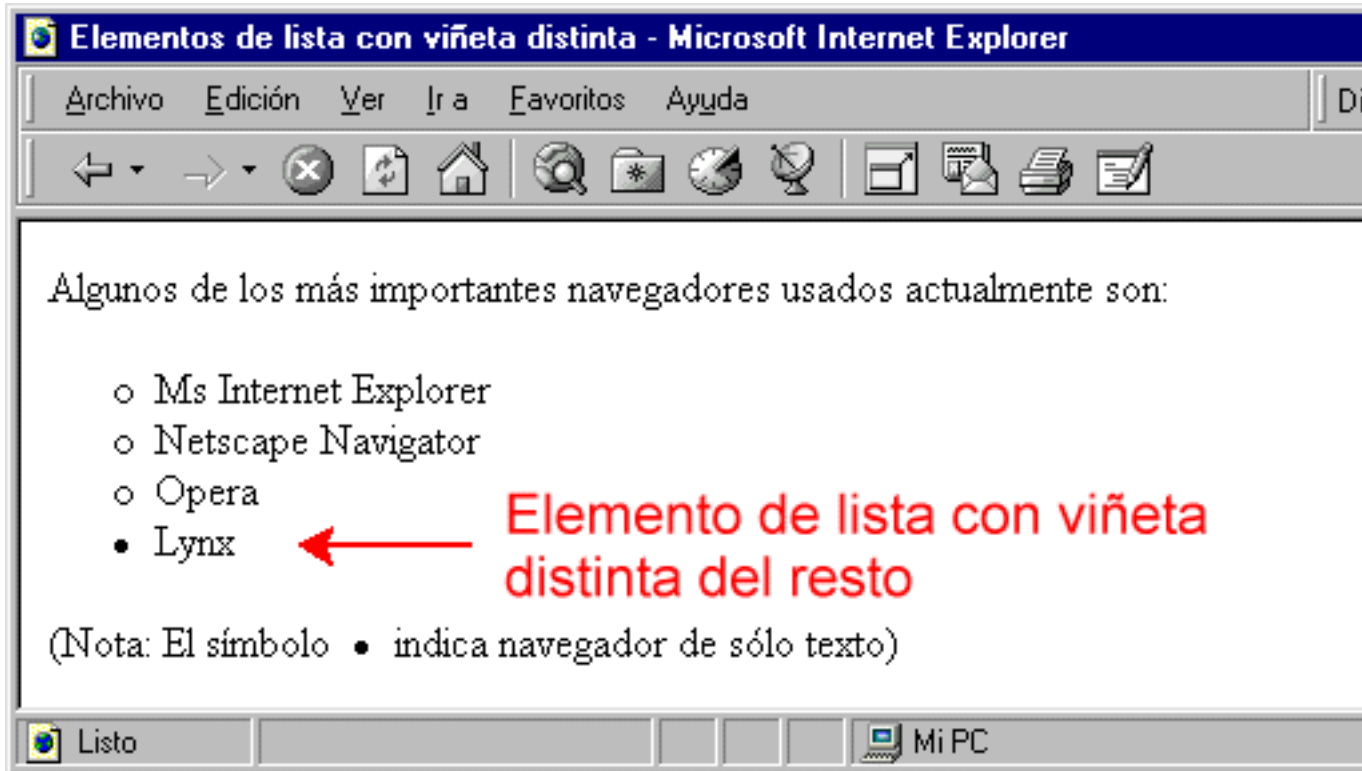
- **ALIGN="center"**: Usando este método de alineamiento obtendremos una tabla centrada respecto al ancho total de la ventana del navegador. El resultado de usar ALIGN="center" es idéntico al que obtendríamos si introdujéramos la tabla entre las instrucciones <CENTER> y </CENTER >. En este tipo de alineamiento

```
<TABLE ALIGN="center">
...Código de la tabla...
</TABLE>
```

el texto no envuelve a la tabla como podemos apreciar en la figura 7.9.

D1

Figura 7.9. Con `ALIGN="right"` la tabla se sitúa en el centro de la ventana del navegador y el texto ya no la envuelve.



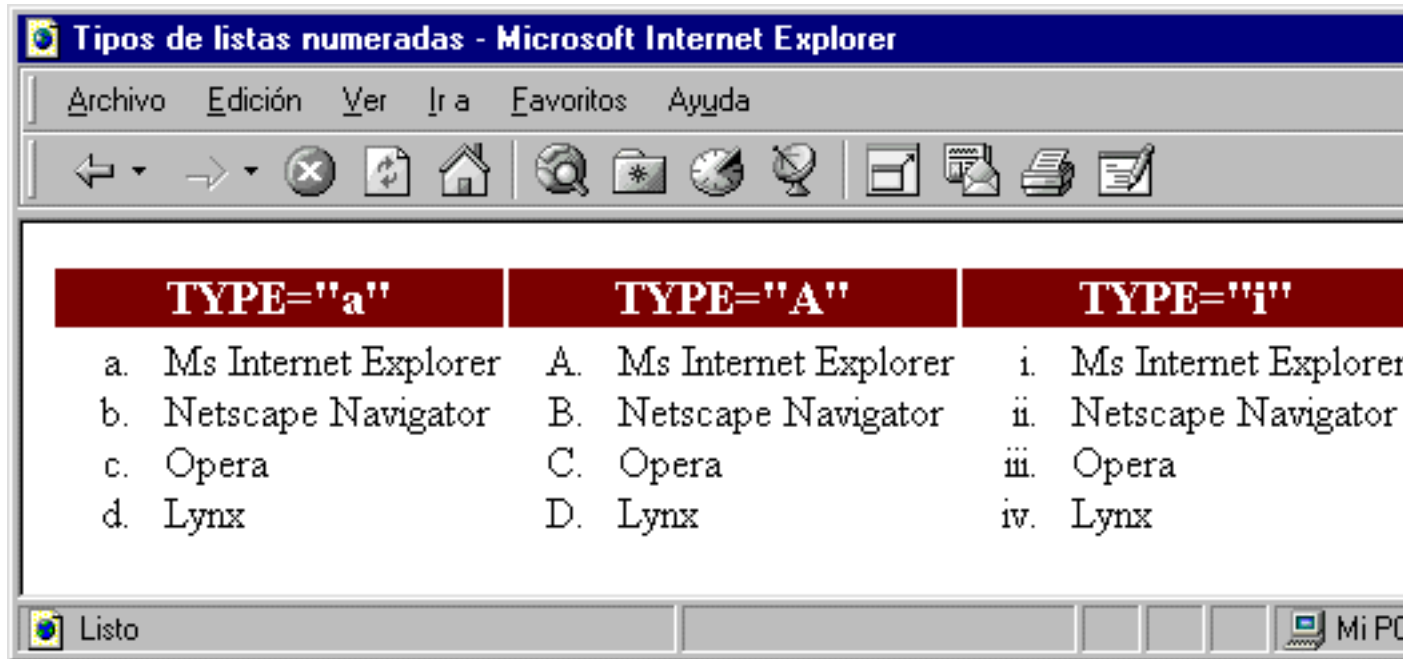
Aviso

Hasta la aparición de las versiones 4 de Explorer™ y Navigator™ ningún navegador soportaba el alineamiento `ALIGN="center"` en las tablas. Por esta razón es recomendable usar además la etiqueta `CENTER` encerrando la tabla para asegurarnos que esta aparecerá centrada. En este caso le estaremos diciendo al navegador dos veces que alineamiento tiene la tabla, esto no es problema siempre que los dos alineamientos coincidan, y por otro lado conseguiremos que todos los navegadores centren la tabla ya que si no entienden `ALIGN="center"` entenderán la etiqueta `CENTER` y viceversa.

Alineamiento de celdas y filas

Hasta ahora hemos estudiado el alineamiento de las tablas con respecto a la ventana del navegador. El siguiente paso será aprender los diferentes tipos de alineamiento de los elementos de las diferentes celdas con respecto a estas. Estos tipos de alineamiento se dividen en dos: alineamiento vertical y alineamiento horizontal. En la figura 7.10 podemos ver una tabla en la que se ilustran las posibilidades de tipos de alineamiento.

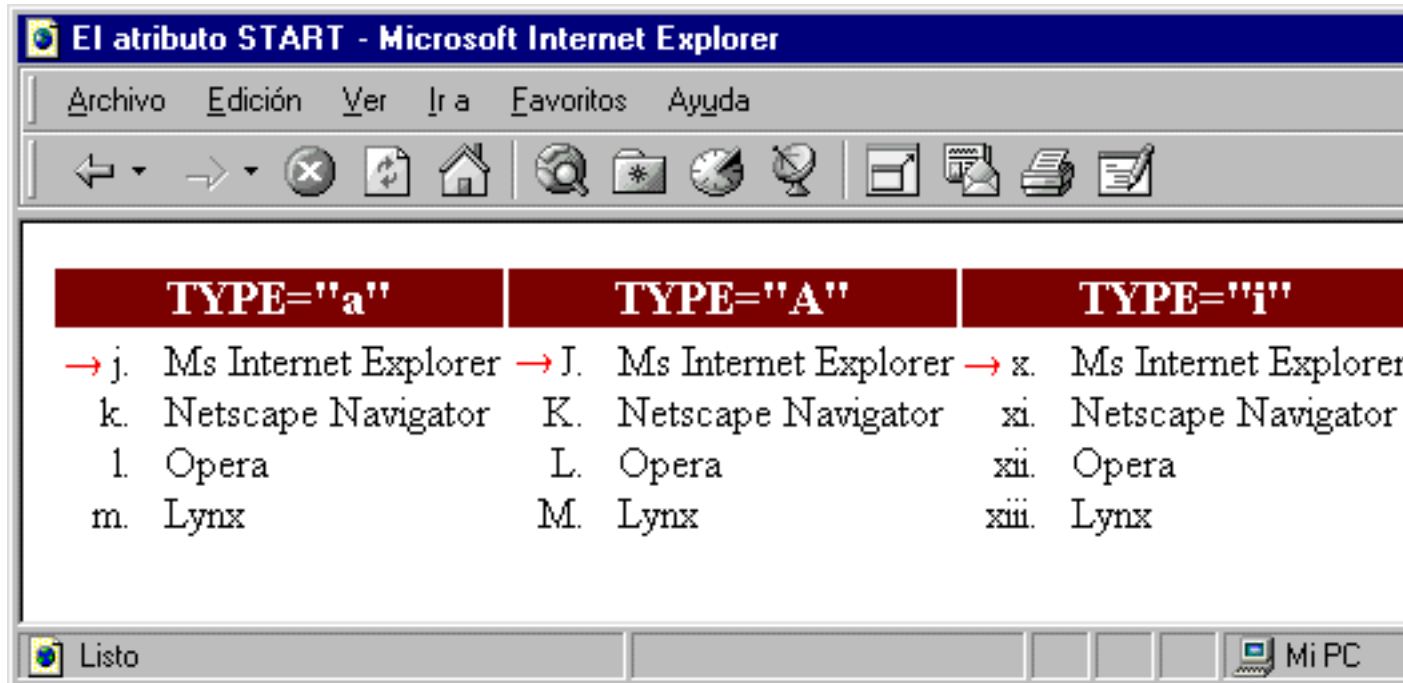
Figura 7.10. Distintas posibilidades de los alineamientos vertical y horizontal de las celdas de una tabla.



- *Alineamiento horizontal:* Para seleccionar el tipo de alineamiento horizontal de las celdas hay que usar el atributo *ALIGN* de las etiquetas TH o TD en función del tipo de celda. Este atributo permitirá seleccionar entre un alineamiento a la izquierda (*ALIGN="left"*), centrado (*ALIGN="center"*) o a la derecha (*ALIGN="right"*), tal y como veíamos en la figura 7.10.
- *Alineamiento vertical:* En este caso usaremos un nuevo atributo: *VALIGN* (Vertical Align, alineamiento vertical). Las diferentes opciones con este tipo de alineamiento son las que veíamos en la figura 7.10: Alineamiento con la parte superior de la celda (*VALIGN="top"*), alineamiento en medio de la celda (*VALIGN="middle"*) y alineamiento con la parte inferior de la celda (*VALIGN="bottom"*).

Por defecto las celdas de encabezado (TH) están centradas tanto horizontal como verticalmente, mientras que las celdas normales (TD) están alineadas a la izquierda y verticalmente se sitúan en el centro de la celda. El punto azul de la figura 7.11 representa el alineamiento por defecto en ambos tipos de celdas.

Figura 7.11. El punto azul de esta figura nos permite observar el alineamiento por defecto de los encabezados de tabla (TH) y de las celdas normales (TD).



Este alineamiento por defecto puede ser cambiado para toda una fila usando los atributos ALIGN y VALIGN en la etiqueta TR. Por ejemplo si creamos una fila con el siguiente código:

```
<TR ALIGN="right"
VALIGN="bottom">
... Etiquetas para crear las celdas ...
</TR>
```

Todas las celdas de esa fila tendrán se alinearán por defecto a la derecha y abajo. Por supuesto si en alguna de las celdas (TD o TH) de esa fila se vuelven a usar ALIGN y VALIGN con unos valores distintos serán estos últimos los que determinarán el alineamiento de esa celda en concreto. Por ejemplo:

```
<TABLE BORDER="1">

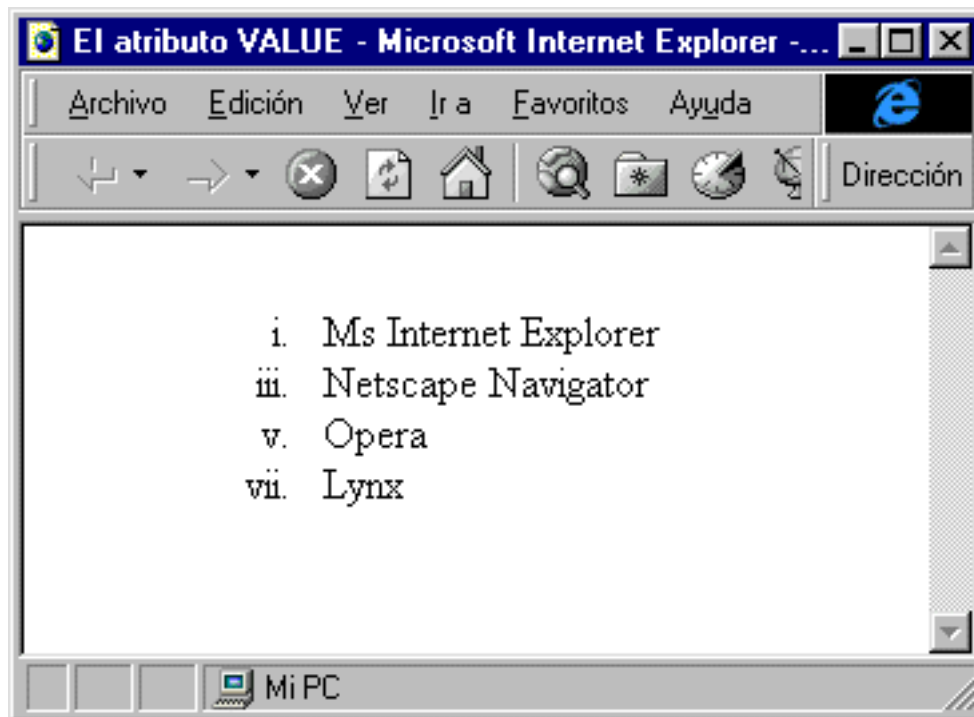
<TR ALIGN="right"
VALIGN="bottom">
<TD>
Celda1
</TD>
<TD>
Celda2
</TD>

<TD ALIGN="left">
```

```
VALIGN="top"> Celda3 </TD>  
</TR>  
</TABLE>
```

El resultado de este código es que todas las celdas de esa fila tendrán como alineamiento por defecto `ALIGN="right"` y `VALIGN="bottom"`, pero en la tercera celda especificamos `ALIGN="left"` `VALIGN="top"` y cambiamos el alineamiento de esa celda en concreto como podemos ver en la figura 7.12

Figura 7.12. Usando los atributos `ALIGN` y `VALIGN` en la etiqueta `TR` cambiamos el alineamiento por defecto de las celdas de esa fila, pero aún podemos modificar ese alineamiento usando los mismos atributos en cada celda.



Nota

Si el lector prueba este último ejemplo probablemente no pueda apreciar el efecto del cambio de alineamiento porque la tabla es demasiado pequeña. Para poder apreciar dicho efecto será necesario aumentar el tamaño de la tabla con los métodos que aprenderemos en el siguiente apartado.

Ancho de tablas y celdas

Como vimos antes en este capítulo, el ancho de las tablas queda determinado por el tamaño necesario para mostrar el contenido de sus celdas. Sin embargo el lenguaje HTML nos permite especificar el tamaño que deseemos por medio de un nuevo atributo de la etiqueta `TABLE`: `WIDTH` (que traducido significa *ancho*). Este atributo se usa igual que los vistos hasta ahora, y su valor representará el ancho en píxeles de la tabla:

```
<TABLE WIDTH="500">
... Código de la tabla ...
</TABLE>
```

Es importante resaltar que si con ese tamaño no se pueden mostrar todos los elementos de la tabla, por ejemplo si en la tabla anterior hay una imagen de ancho 550 que no cabe en una tabla de 500 pixeles de ancho, el navegador ignorará el atributo WIDTH. Si la tabla únicamente contiene texto la etiqueta WIDTH tiene una mayor utilidad, ya que el texto se adaptará a ese ancho, ocupando más líneas si es necesario.

Nota

El tamaño del ancho de una tabla también se puede especificar en porcentaje respecto al ancho total de la ventana del navegador. Por ejemplo:

```
<TABLE WIDTH="50%">
... Código de la tabla ...
</TABLE>
```

hace que la tabla sea la mitad de la ventana. Si el usuario hace más pequeña la ventana la tabla también se hará más pequeña.

Tamaño de las celdas

Al igual que hemos modificado el ancho de la tabla completa también podremos modificar el tamaño de cada celda por separado. En este caso tendremos más posibilidades aún, ya que podremos modificar también la altura de la celda usando el atributo HEIGHT.

Al igual que antes, las dimensiones que especifiquemos con WIDTH y HEIGHT sólo serán respetadas si en ese espacio cabe su contenido. Por ejemplo en el siguiente caso:

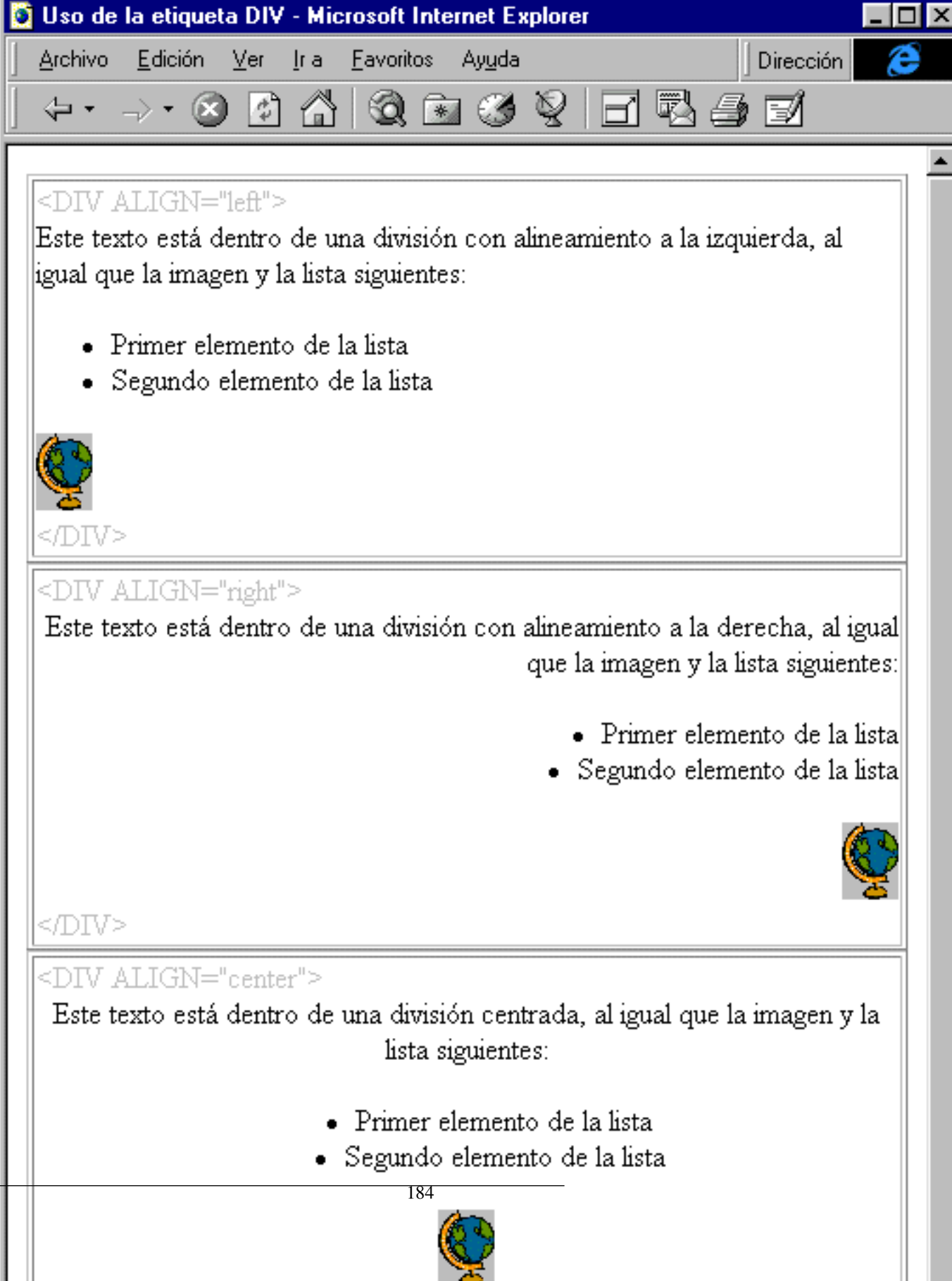
```
<TABLE>
  <TR> <TD WIDTH="50"
    HEIGHT="50">
    <IMG SRC="img.gif"
      WIDTH="100"
      HEIGHT="100">
  </TD>
</TR>
</TABLE>
```

la única celda de la tabla tendrá un tamaño de 100 por 100, que es el tamaño de la imagen que hemos insertado en ella. En este caso no pueden respetarse los atributos WIDTH y HEIGHT de la etiqueta TD ya que en una celda de las dimensiones indicadas en ellas no cabría la imagen. Si en su lugar hubiésemos usado `< TD WIDTH="150" HEIGHT="150">` estas dimensiones SI se hubiesen respetado.

EJERCICIO 7-2

Como la mejor manera de aprender es practicar vamos a realizar un sencillo ejercicio pero que agrupa gran parte de los conocimientos adquiridos hasta ahora sobre tablas. El objetivo será crear una tabla con dos celdas y dos columnas de tamaño 100x100 todas ellas. Debemos insertar en cada una de ellas la imagen `spot.gif` y alinearla de tal manera que consigamos el resultado de la figura 7.13.

Figura 7.13. Este será el objetivo del ejercicio 7.2. Una pista: debemos usar los atributos WIDTH, HEIGHT, ALIGN y VALIGN.



The screenshot shows a Microsoft Internet Explorer window titled "Uso de la etiqueta DIV - Microsoft Internet Explorer". The browser's address bar is empty, and the menu bar includes "Archivo", "Edición", "Ver", "Ir a", "Favoritos", and "Ayuda". The toolbar contains various navigation and utility icons. The main content area is divided into three sections, each demonstrating a different alignment for a DIV element:

- Left Alignment:** The first section shows the HTML code `<DIV ALIGN="left">` followed by the text "Este texto está dentro de una división con alineamiento a la izquierda, al igual que la imagen y la lista siguientes:". Below the text is a bulleted list with two items: "Primer elemento de la lista" and "Segundo elemento de la lista". At the bottom left of this section is a small globe icon, followed by the closing tag `</DIV>`.
- Right Alignment:** The second section shows the HTML code `<DIV ALIGN="right">` followed by the text "Este texto está dentro de una división con alineamiento a la derecha, al igual que la imagen y la lista siguientes:". Below the text is a bulleted list with two items: "Primer elemento de la lista" and "Segundo elemento de la lista". At the bottom right of this section is a small globe icon, followed by the closing tag `</DIV>`.
- Center Alignment:** The third section shows the HTML code `<DIV ALIGN="center">` followed by the text "Este texto está dentro de una división centrada, al igual que la imagen y la lista siguientes:". Below the text is a bulleted list with two items: "Primer elemento de la lista" and "Segundo elemento de la lista". At the bottom center of this section is a small globe icon.

At the bottom of the page, the number "184" is centered, and a small globe icon is positioned below it.

La tabla deberá estar centrada y tener un grosor de 5 píxeles. [En el archivo `ej7-2.zip` se ofrece la solución de este ejercicio]

El atributo NOWRAP

Para completar nuestro aprendizaje sobre el tamaño de tablas y celdas y como modificarlo nos queda por comentar un último atributo: *NOWRAP*. Este atributo no toma ningún valor, es decir, su forma de uso es:

```
<TABLE>
<TR>

    <TD NOWRAP> Texto </TD>
</TR>
</TABLE>
```

Su efecto será que el texto de esa celda no puede ser *partido* en varias líneas, para adecuarse al tamaño de la celda. Cuando esto sucede se dice que la tabla ha envuelto (*wrap* en inglés) el texto. El significado de *NOWRAP* es por tanto *no envolver* el texto. Si es necesario para mantener el texto en una sola línea se ignorarán los atributos *WIDTH* y *HEIGHT* si fueron usados. En la figura 7.14 observamos una primera tabla con una única celda en la que no hemos usado el atributo *NOWRAP*, en la segunda tabla mostramos la misma tabla pero poniendo *NOWRAP* en dicha celda.

Figura 7.14. Usando el atributo NOWRAP evitamos que el texto sea partido en varias líneas, sin embargo no es recomendable usarlo en líneas muy largas como esta.



Como vemos en este segundo caso la línea de texto no cabe en el tamaño de la ventana y por tanto se extiende más allá de esta.

Aviso

Hay que tener mucho cuidado usando el atributo NOWRAP, ya que si después escribimos un texto muy largo sin incluir ningún salto de línea (BR) todo él será mostrado en una única línea, que no cabrá en

pantalla y por tanto obligará al visitante a usar las barras de desplazamiento.

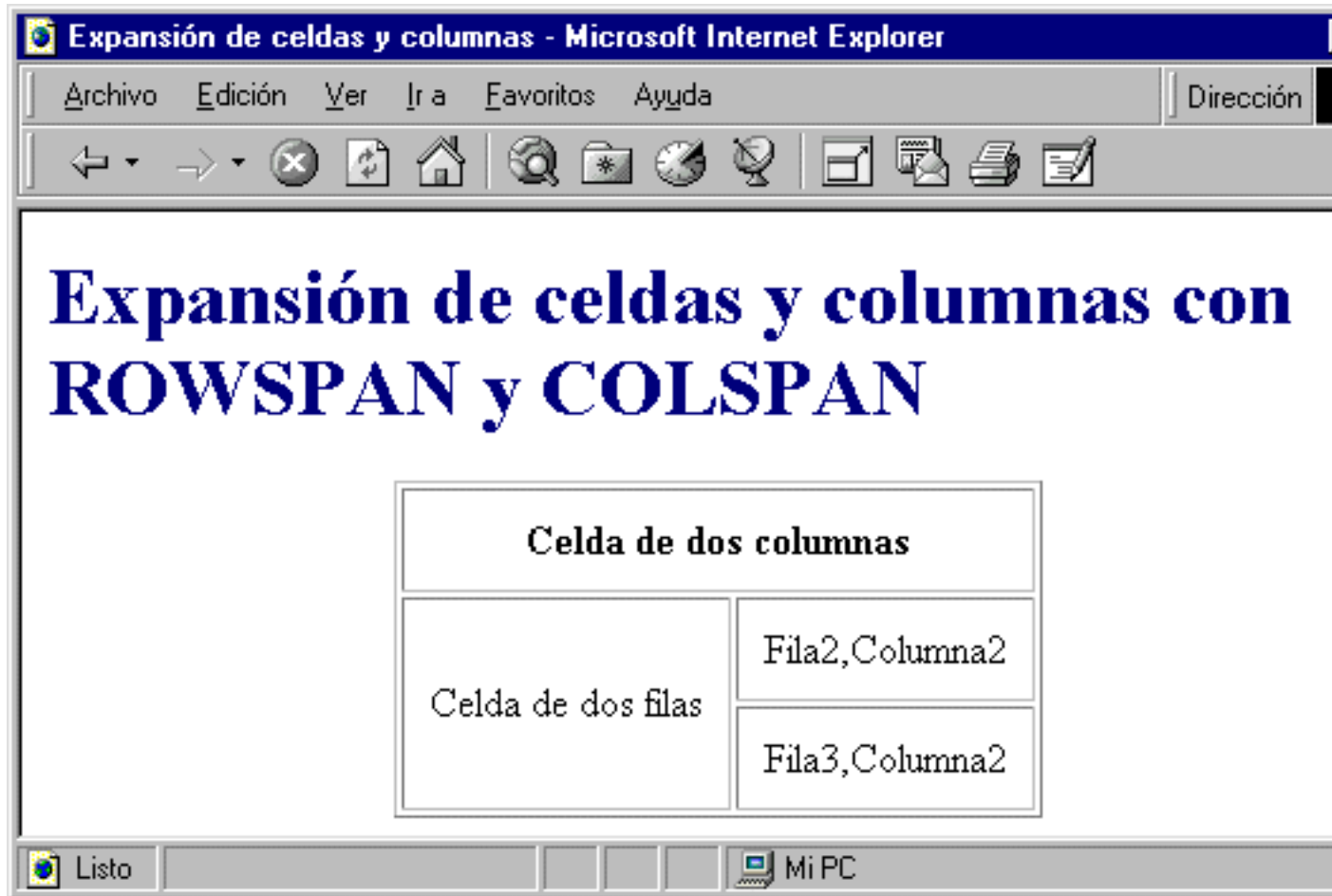
Celdas que abarcan varias filas o columnas

Es hora ya de comenzar con dos de los atributos más interesantes en la creación de tablas: *ROWSPAN* y *COLSPAN*. Gracias a estos atributos podremos crear tablas mucho más complejas y que se adapten a nuestras necesidades. Estos dos atributos sirven tanto para las celdas normales, TD, como para las celdas encabezado, TH, y su función será extender una fila (*ROWSPAN*) o una columna (*COLSPAN*) para que ocupen el tamaño de varias. Para comprender el significado del concepto *extender* lo mejor es ver un ejemplo:

```
<TABLE BORDER="1">
<TR>
<TH COLSPAN="2">
Celda de dos columnas
</TH>
</TR>
<TR>
<TD ROWSPAN="2">
Celda de dos filas
</TD>
<TD>
Fila2, Columna2
</TD>
</TR>
<TR>
<TD>
Fila3, Columna2
</TD>
</TR>
</TABLE>
```

En la figura 7.15 podemos ver la tabla resultado de este código:

Figura 7.15. Los atributos COLSPAN y ROWSPAN nos permiten crear celdas que se extiendan varias columnas o varias filas.



como vemos la primera celda se extiende para ocupar dos columnas y la primera celda de la segunda fila ocupa dos filas. En este momento invitamos al lector a practicar con los atributos ROWSPAN y COLSPAN incluso con celdas de mayor tamaño, ya que esta es la mejor manera de habituarse a ellos, más adelante seguiremos realizando ejemplos prácticos con estos y otros atributos de las tablas.

Márgenes y separación de celdas

El único aspecto que nos queda por aprender sobre las tablas en HTML es el correspondiente a los márgenes en las celdas y la separación entre ellas. Antes de nada es conveniente comentar que el lenguaje HTML considera a cada celda como un elemento individual con sus propios bordes que no comparte con las celdas adyacentes. Basándonos en esta forma de ver las tablas podemos cambiar la *distancia* que existe entre los bordes de las células adyacentes usando el atributo CELLSPACING (Espaciado de celdas). El valor por defecto de este atributo es 1. En general se suele usar para eliminar el espaciado entre celdas (como veremos en el ejercicio 7-3) o para aumentarlo para crear un efecto como el del siguiente ejemplo:

```
<TABLE CELLSPACING="10">
```

```
BORDER="1">
... Código de la tabla ...
</TABLE>
```

cuyo resultado podemos observar en la figura 7.16:

Figura 7.16. El atributo **CELLSPACING** controla el espaciado entre los bordes de cada una de las celdas de la tabla.

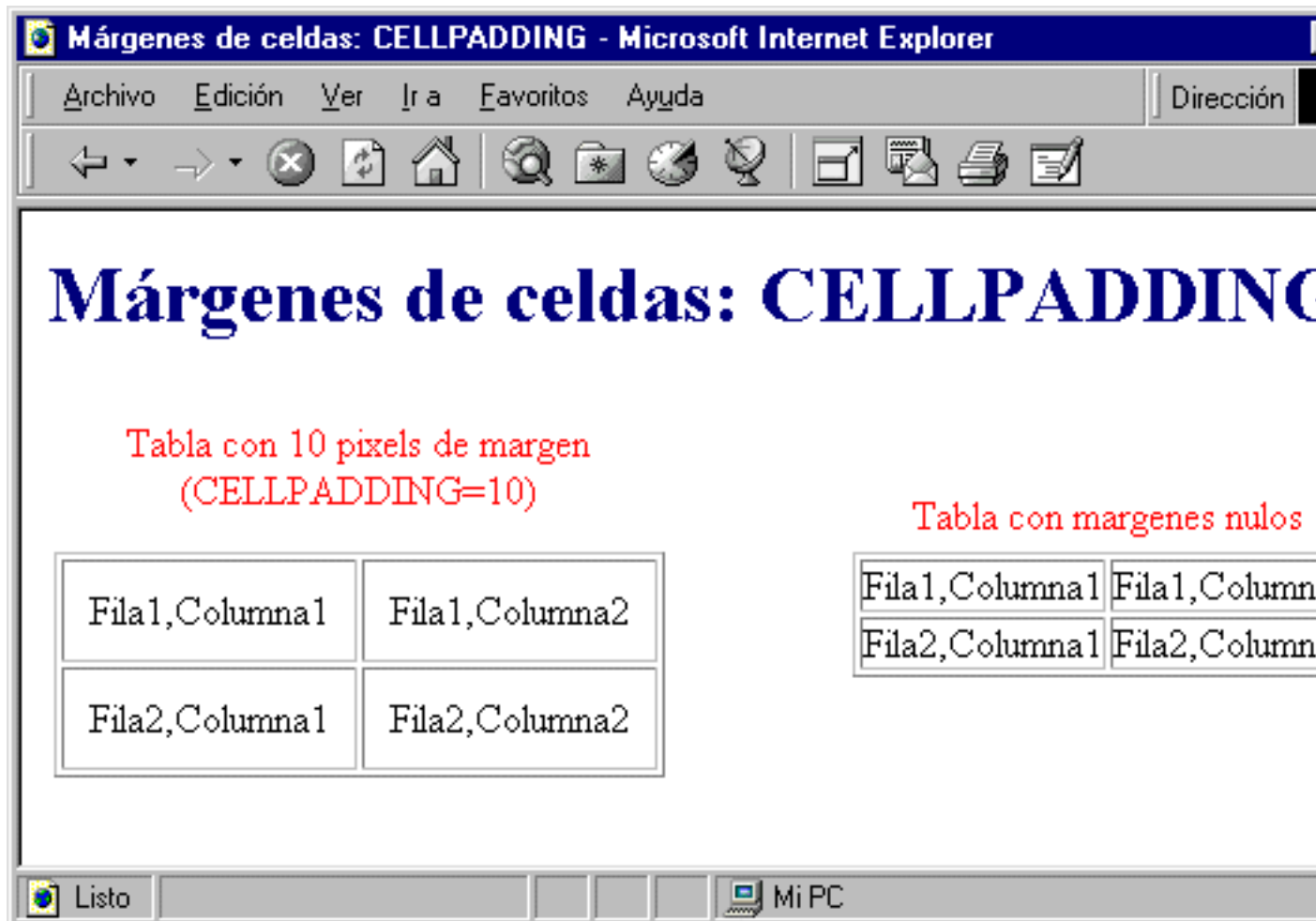


Una segunda propiedad de las tablas de gran importancia es el margen que existe entre los elementos de las celdas y sus bordes. Esta separación es nula por defecto, pero puede aumentarse usando el atributo *CELLPADDING*. La forma de usar este atributo queda descrita en el siguiente ejemplo:

```
<TABLE CELLPADDING="10"
BORDER="1"
ALIGN="LEFT">
... Código de la tabla ...
</TABLE>
```

y en la figura 7.17:

Figura 7.17. Con el atributo `CELLPADDING` podemos aumentar los márgenes entre el contenido de las celdas y su borde.



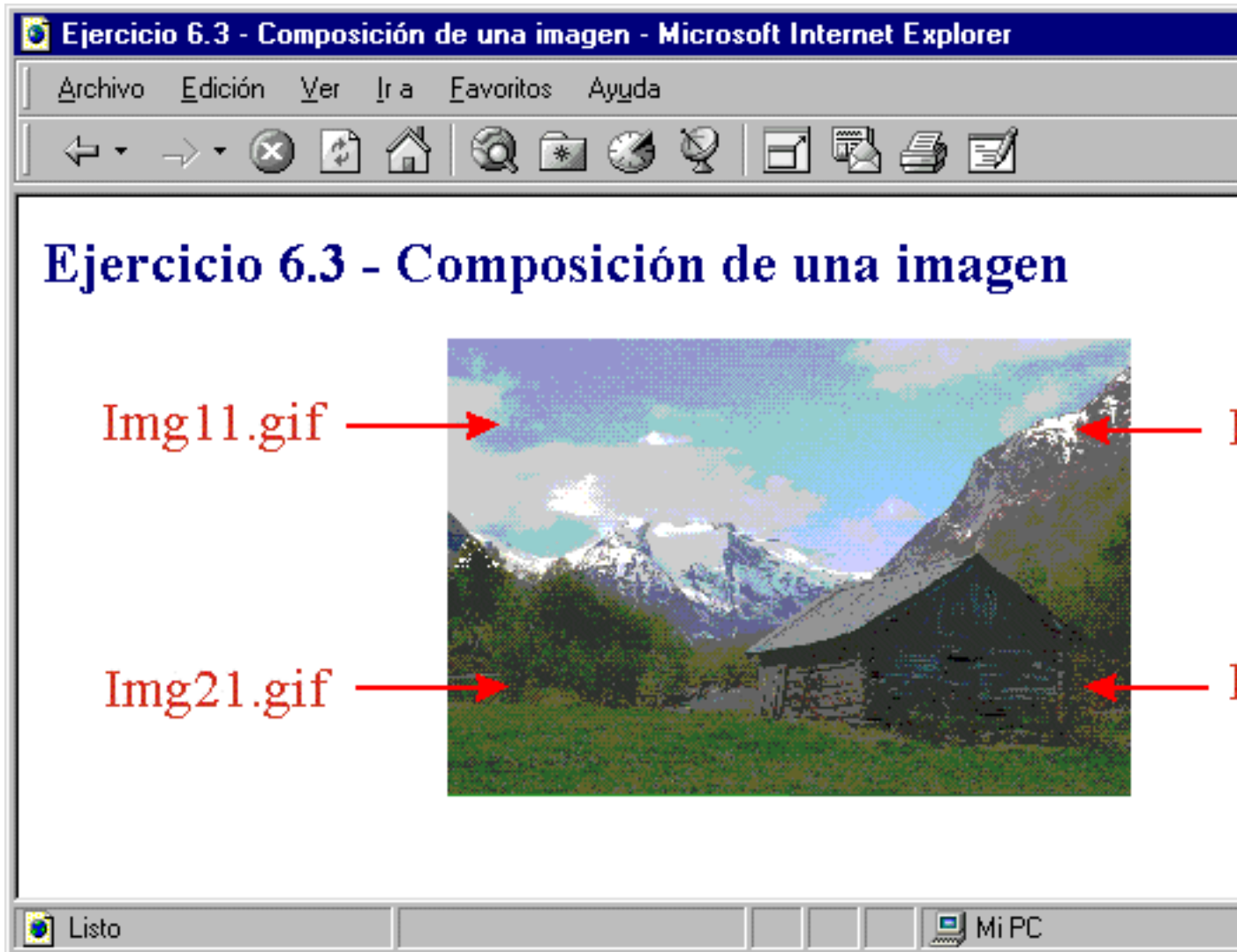
Tanto en el atributo `CELLPADDING` como en el atributo `CELLSPACING` el valor entre comillas representa un tamaño en píxeles.

Para practicar un poco el uso de estos dos atributos se propone un ejercicio con el que conseguiremos crear el efecto de una imagen entera a partir de sus trozos.

Ejercicio 7.3

Este ejercicio consiste en la división de una imagen grande en cuatro partes de igual tamaño. El objetivo será volver a juntarlas. Para ello usaremos una tabla de 2x2 con una de las partes en cada celda, sin borde y con márgenes y espaciado de las celdas nulo. El resultado buscado debe parecerse al de la figura 7.18 (exceptuando las flechas y las letras rojas que son indicativas):

Figura 7.18. Ejercicio 7.3. Aunque pueda parecer que hay una sola imagen, en realidad hay cuatro cuidadosamente colocadas.



Las imágenes a usar, así como una posible solución, a este ejercicio pueden encontrarse en el archivo `ej7-3.zip`.

Conclusión

Bueno, ya podemos respirar tranquilos, ya sabemos todo lo que se puede saber sobre la creación de tablas en el lenguaje HTML™ estándar. Sin duda es como para estar orgullosos, porque el método que usa este lenguaje no es excesivamente sencillo. Por esta razón no debe desanimarse el lector que llegado a este punto no recuerde parte de las etiquetas y atributos descritos en este capítulo. Nuestro objetivo no era ese, sino que como buenos creadores de páginas web conociéramos las etiquetas del lenguaje HTML sin llegar a saberlas de memoria pero lo suficiente para reconocer su función, conocer las posibilidades de las tablas y poder crear rápidamente tablas sencillas. Para la creación de tablas muy complicadas es recomendable usar un asistente como los que incluyen FrontPage™ o HomeSite™ (la versión shareware será incluida en el archivo `FrontPage.zip` y `HomeSite.zip`). Aunque estos asistentes no siempre nos ofrecen justo lo que queremos, nosotros ya conocemos lo suficiente para retocar el código de la tabla una vez creada.

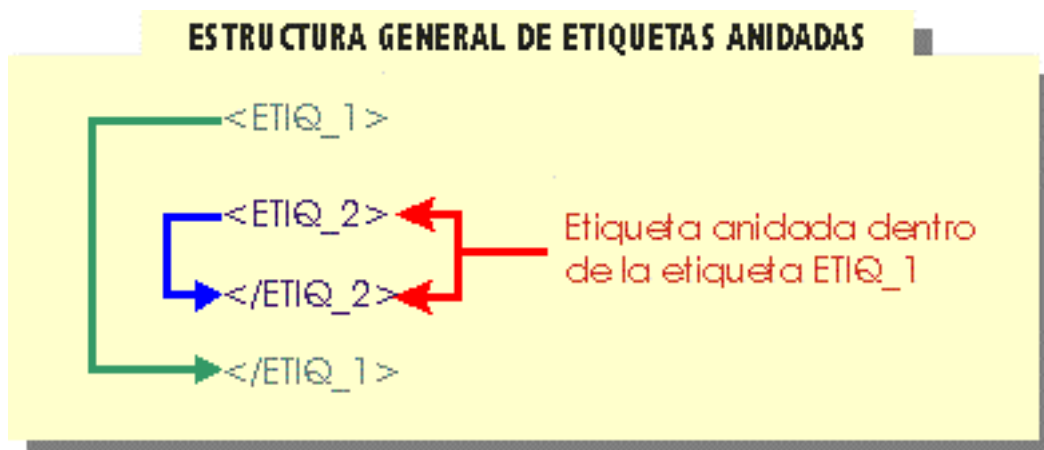
Anidamiento

Lo que vamos a contar ahora no es nuevo, ya lo hemos comentado antes a lo largo de nuestro aprendizaje del lenguaje HTML y de las tablas, pero dado que es una de sus características más usadas es importante repetirla y ensalzarla, estamos hablando del anidamiento de tablas. ¿Y qué es el anidamiento?. Ya hemos usado este nombre anteriormente en el curso, con lo que es probable que el lector ya tenga una idea de a qué nos referimos con este nombre, pero es conveniente dar una descripción más detallada de en que consiste.

Una definición técnica sería: "El anidamiento consiste en el uso de una etiqueta *dentro* (entre la instrucción de inicio y la instrucción de fin, [Ver figura 6.20]) de otra, de manera que los efectos de ambas se sumen o combinen". Las imágenes enlace eran un perfecto ejemplo del uso del anidamiento. En ese caso usábamos la etiqueta `IMG` para insertar la imagen y después usábamos la etiqueta `A` para convertirla en un enlace. Se combinaban los efectos de las dos etiquetas:

```
<A HREF="enlace.htm">
<IMG SRC="img.gif"> <!-- Imagen dentro de la etiqueta A -->
</A>
```

Figura 6.20. Las etiquetas anidadas son aquellas que se encuentran entre la instrucción de inicio y la instrucción de fin de otra etiqueta. De esta forma conseguimos que los efectos de ambas se sumen.



Otro ejemplo que vimos anteriormente consistía en el uso de dos o más etiquetas formateadoras de texto que anidábamos para sumar sus efectos. Por ejemplo:

```
<B> <!--Negrita -->
<U> <!-- Subrayado -->
Texto en negrita y subrayado
</U> <!-- Fin subrayado -->
</B> <!-- Fin negrita -->
```

Este caso es ligeramente más complicado que el anterior, ya que ambas etiquetas tienen instrucción de inicio e instrucción de fin. Es por ello que tendremos que tener un cuidado especial. Si no insertamos las instrucciones en el orden correcto los resultados serán imprevisibles. Un ejemplo de uso *incorrecto* del anidamiento sería:

```
<B> <!--Negrita -->
```



```

<U> <!-- Subrayado -->
Texto en negrita y subrayado
</B> <!-- Fin negrita -->
</U> <!-- Fin subrayado -->

```

Una vez aclarado el concepto de anidamiento estamos en disposición de aplicarlo a las tablas. Cuando vimos la etiqueta de definición de celdas, TD, comentamos que podríamos insertar en ellas prácticamente cualquier otra etiqueta del lenguaje HTML. Si lo que insertamos es otra etiqueta estaremos ante un caso de anidamiento de tablas. En la figura 6.21 vemos un ejemplo donde se observa la utilidad del anidamiento de tablas. El código usado es:

Figura 6.21. En este ejemplo vemos como anidando una tabla dentro de otra podemos crear interesantes efectos en las páginas Web.



```

<TABLE BORDER="1">
<TR>
<TD>
<!-- TABLA ANIDADA DENTRO DE OTRA-->
<TABLE BORDER="1">
<TR>
<TD><A HREF="mes.htm">Nuestras principales OFERTAS del mes</A>
</TD>
</TR>
<TR>
<TD><A HREF="#alpes">LOS ALPES - 20 años a su servicio</A>

```

```
</TD>
</TR>
<TR>
<TD><A HREF="#medida">Viajes a su medida</A>
</TD>
</TR>
<TR>
<TD><A HREF="#donde">Dónde encontrarnos</A>
</TD>
</TR>
</TABLE>
<!-- FIN TABLA ANIDADA -->
</TD>
<TD><IMG SRC="losalpes.gif"></TD>
</TR>
</TABLE>
```

Como vemos, anidando dos tablas conseguimos crear efectos complejos obteniendo como resultado un gran control sobre la disposición los elementos de nuestra página.

Nota

El ejemplo de la figura 6.21 podía haberse realizado usando una única tabla de 4 filas y el atributo `ROWSPAN` en la segunda columna, sin embargo en ocasiones resulta más fácil usar el anidamiento de tablas y además da lugar a un código ligeramente más intuitivo.

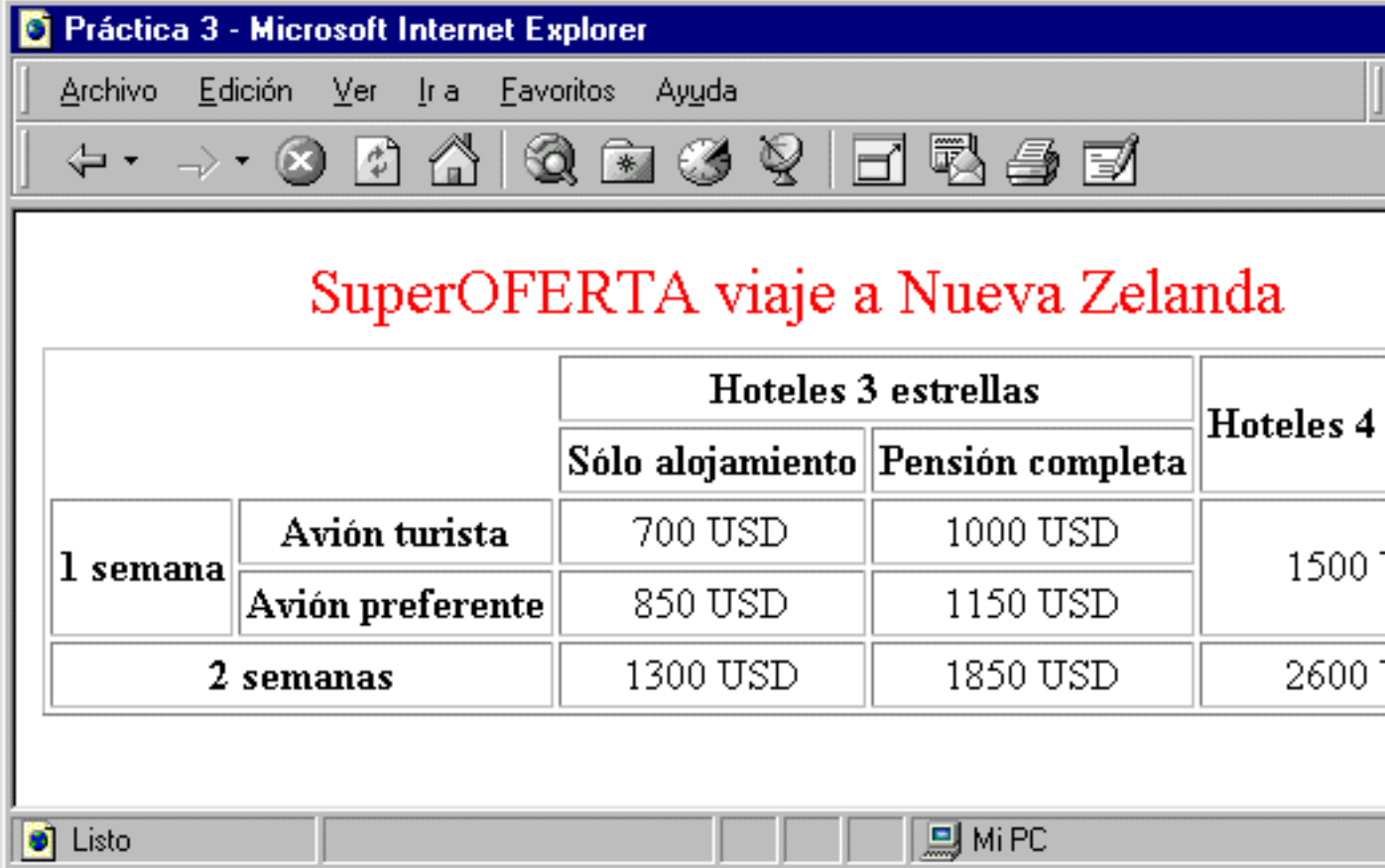
Práctica 3 - Tabla de Datos.

La primera de las prácticas consistirá en la creación de una tabla de datos. Será más complicada que los ejemplos vistos hasta ahora y se parecerá más a lo que necesitaremos en el mundo real.

Nuestro objetivo será crear la tabla de la figura 6.22, pero vayamos paso por paso.

Antes de empezar será útil desglosar la tabla objetivo en sus columnas y filas. Esto nos será de gran ayuda según vayamos creándola y nos permitirá hacernos una idea inicial de la estructura de código que necesitamos. En la figura 6.23 podemos ver como hemos realizado este desglose, en este caso vemos que será necesaria una tabla de 5 filas y 5 columnas, existiendo celdas que se extienden varias columnas (`COLSPAN`) y otras que se extienden varias filas (`ROWSPAN`). Debemos recordar en este momento que las celdas que se expanden deben ser definidas (con `TD` o `TH`) en la primera de las filas y/o primera de las columnas que ocupan.

Figura 6.22. Nuestro objetivo en esta práctica 3 será realizar esta tabla. Para su realización necesitaremos usar gran parte de los atributos que hemos aprendido.



The screenshot shows a Microsoft Internet Explorer browser window titled "Práctica 3 - Microsoft Internet Explorer". The browser's menu bar includes "Archivo", "Edición", "Ver", "Ir a", "Favoritos", and "Ayuda". The toolbar contains various navigation and utility icons. The main content area displays a red heading "SuperOFERTA viaje a Nueva Zelanda" above a table. The table is structured as follows:

		Hoteles 3 estrellas		Hoteles 4
		Sólo alojamiento	Pensión completa	
1 semana	Avión turista	700 USD	1000 USD	1500
	Avión preferente	850 USD	1150 USD	
2 semanas		1300 USD	1850 USD	2600

The status bar at the bottom of the browser shows "Listo" and "Mi PC".

Figura 6.23. Una vez que sabemos como queremos que quede la tabla es muy útil desglosarla en columnas y filas para no liarnos al escribir el código.

SuperOFERTA viaje a Nueva Zelanda

		Hoteles 3 estrellas		Hoteles 4
		Sólo alojamiento	Pensión completa	
1 semana	Avión turista	700 USD	1000 USD	1500 USD
	Avión preferente	850 USD	1150 USD	
2 semanas		1300 USD	1850 USD	2600 USD

Columna 1 Columna 2 Columna 3 Columna 4 Columna 5

Vamos allá con la creación de la tabla, para seguir mejor el proceso recomendamos al lector que se ponga de su editor de texto o editor de HTML favorito y vaya escribiendo el código.

Paso 1.

En primer lugar crearemos la estructura básica de la página y de la tabla, incluyendo, de paso, el título de ésta con la etiqueta `CAPTION`:

```
<HTML>
<HEAD>
<TITLE>Practica 3</TITLE>
</HEAD>
```

```
<BODY>
<TABLE BORDER=1>
<CAPTION>SuperOFERTA viaje a Nueva Zelanda</CAPTION>
... Aquí iremos insertando el resto del código ...
</TABLE>
</BODY>
</HTML>
```

Paso 2.

A continuación vamos a crear la primera de las filas. La primera de las celdas, que estará vacía ocupa dos filas y dos columnas. El código para definirla debemos ponerlo en la celda de la primera columna de la primera fila, y será:

```
<TR><TH ROWSPAN=2 COLSPAN=2></TH>
```

A continuación sigue una celda que ocupa 2 columnas y otra que ocupa dos filas. En definitiva el código necesario para crear la primera fila será:

```
<TR>
<TH ROWSPAN=2 COLSPAN=2></TH>
<TH COLSPAN=2>Hoteles 3 estrellas</TH>
<TH ROWSPAN=2>Hoteles 4 estrellas</TH>
</TR>
```

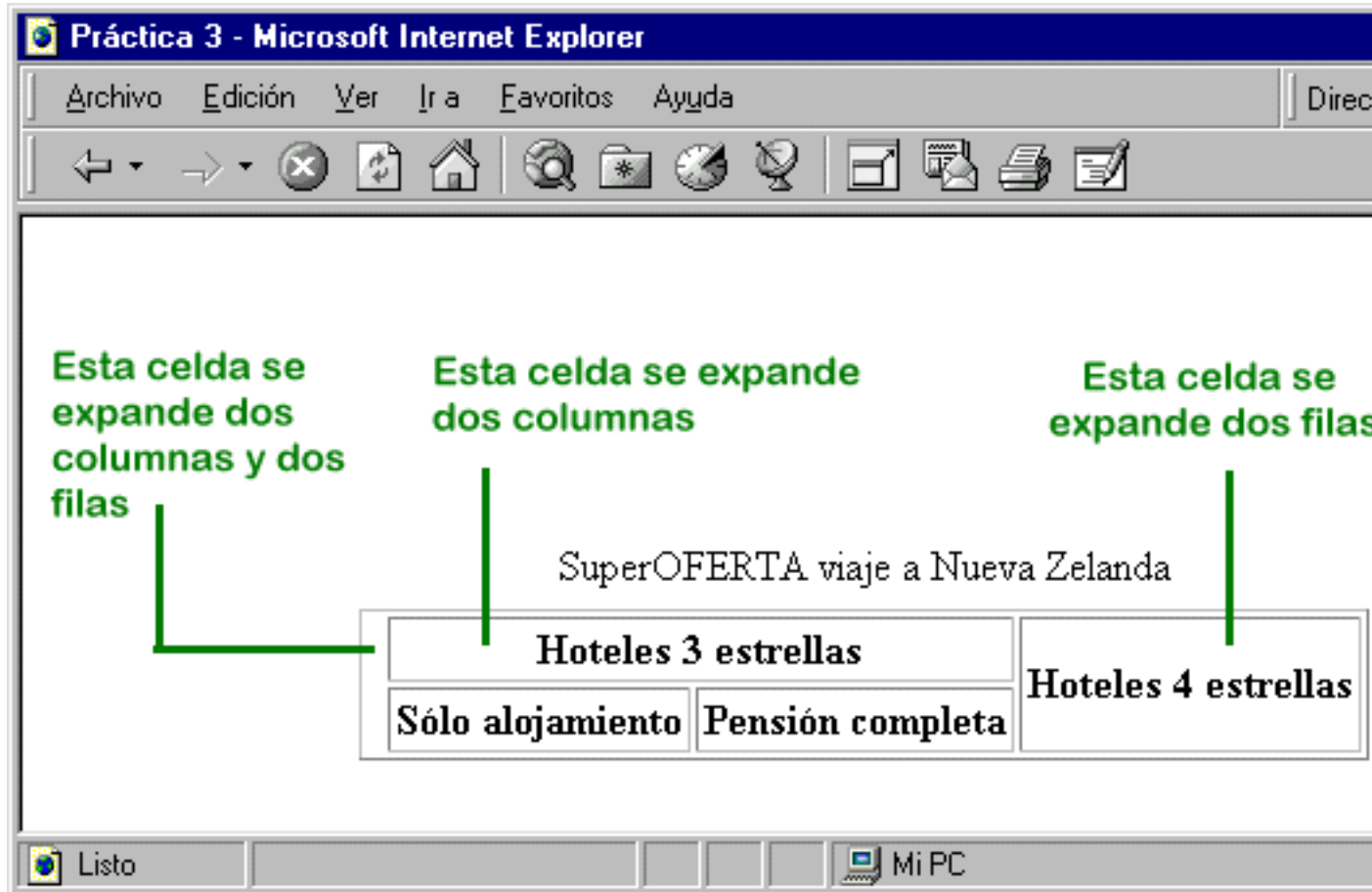
Paso 3.

Ahora crearemos la segunda de las filas, esta no es la que comienza con la celda "1 semana", sino la anterior porque la primera celda en la fila anterior se extendió dos filas. En este caso sólo debemos escribir el código de dos celdas, ya que el resto son celdas que comenzaron en la primera fila y se expanden a esta segunda. No será necesario definir esas celdas, ni siquiera para dejarlas vacías, simplemente hay que ser consciente del espacio que ocuparán. Teniendo esto en cuenta el código usado será:

```
<TR>
<TH>Sólo alojamiento</TH>
<TH>Pensión completa</TH>
</TR>
```

Es posible que resulte un poco difícil ver que de sólo con este código las dos únicas celdas que hemos insertado se colocarán en la tercera y cuarta columna, pero recordemos las dos primeras ya están ocupadas por la primera celda de la primera fila. Este es un buen momento para probar con un navegador lo que hemos hecho hasta ahora. Guardemos el código y abramos el archivo HTML con un navegador. En la figura 6.24 podemos ver la tabla tal y como la muestra Internet Explorer™.

Figura 6.24. Ya tenemos hecha una parte de la tabla, algunas de las celdas se expanden 2 filas, 2 columnas o ambas, aunque en algún caso todavía no es apreciable.



Paso 4.

Siguiente fila. La primera celda y la última se expanden 2 filas, por lo que usaremos el atributo `ROWSPAN=2` en ellas:

```
<TR>
<TH ROWSPAN="2">1 semana</TH>
<TH>Avión turista</TH>
<TD>700 USD</TD>
<TD>1000 USD</TD>
<TD ROWSPAN="2" ALIGN="center">1500 USD</TD>
</TR>
```

La siguiente fila, la cuarta, será muy similar a la segunda, ya que sus celdas no usan `ROWSPAN` ni `COLSPAN` y tendremos que tener en cuenta las celdas de la fila anterior que se expanden hasta la cuarta. El código que debemos usar será:

```
<TR>
```

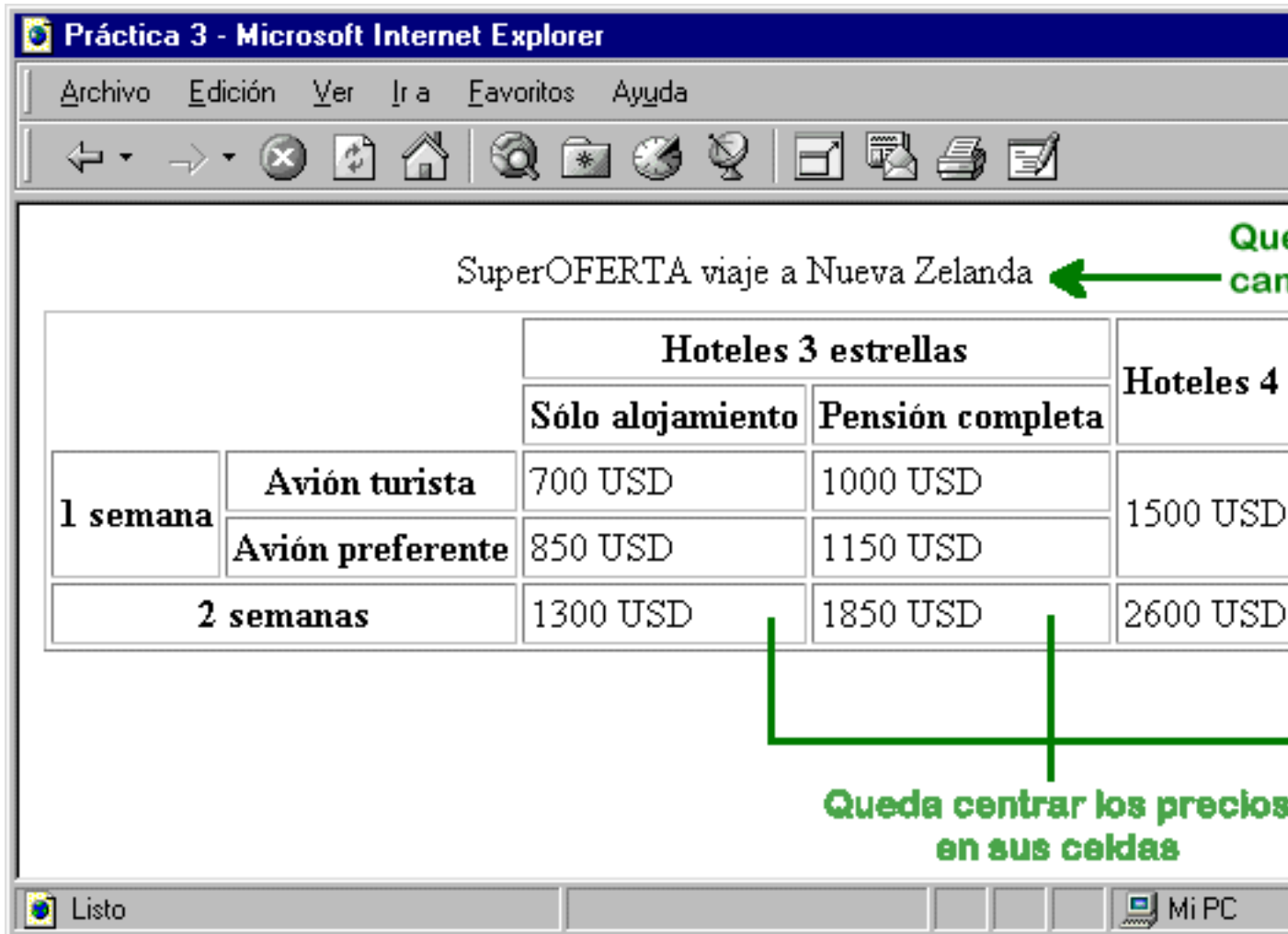
```
<TH>Avión preferente</TH>  
<TD>850 USD</TD>  
<TD>1150 USD</TD>  
</TR>
```

Ya hemos llegado a la última de las filas. A estas alturas ya nos resultará fácil. La primera celda se expande dos columnas, con lo que tendremos que definir un total de 4 celdas para completar las 5 columnas:

```
<TR>  
<TH COLSPAN="2">2 semanas</TH>  
<TD>1300 USD</TD>  
<TD>1850 USD</TD>  
<TD>2600 USD</TD>  
</TR>
```

Tras haber escrito todo este código es momento de volver a ver el resultado con un navegador. En este caso obtendremos la tabla de la figura 6.25. Ya está prácticamente terminada nuestra tabla objetivo, sólo nos quedan algunos detalles.

Figura 6.25. Ya tenemos la tabla prácticamente terminada. Sólo quedan algunos detalles para darla por concluida.



Paso 5.

En este quinto y último paso daremos los últimos acabados a la página. Primero daremos formato al texto del título. Para ello usaremos la etiqueta `FONT`. Para elegir el color rojo (`#FF0000` usando el código hexadecimal) usaremos el atributo `COLOR` y para aumentar el tamaño usaremos el atributo `SIZE`:

```
<CAPTION>
<FONT SIZE="+2" COLOR="#FF0000">
SuperOFERTA viaje a Nueva Zelanda
</FONT>
</CAPTION>
```

Por último debemos centrar las celdas donde están los precios de los distintos viajes. Estas celdas son las creadas con la etiqueta `TD` y deberemos usar el atributo `ALIGN` de la siguiente manera:

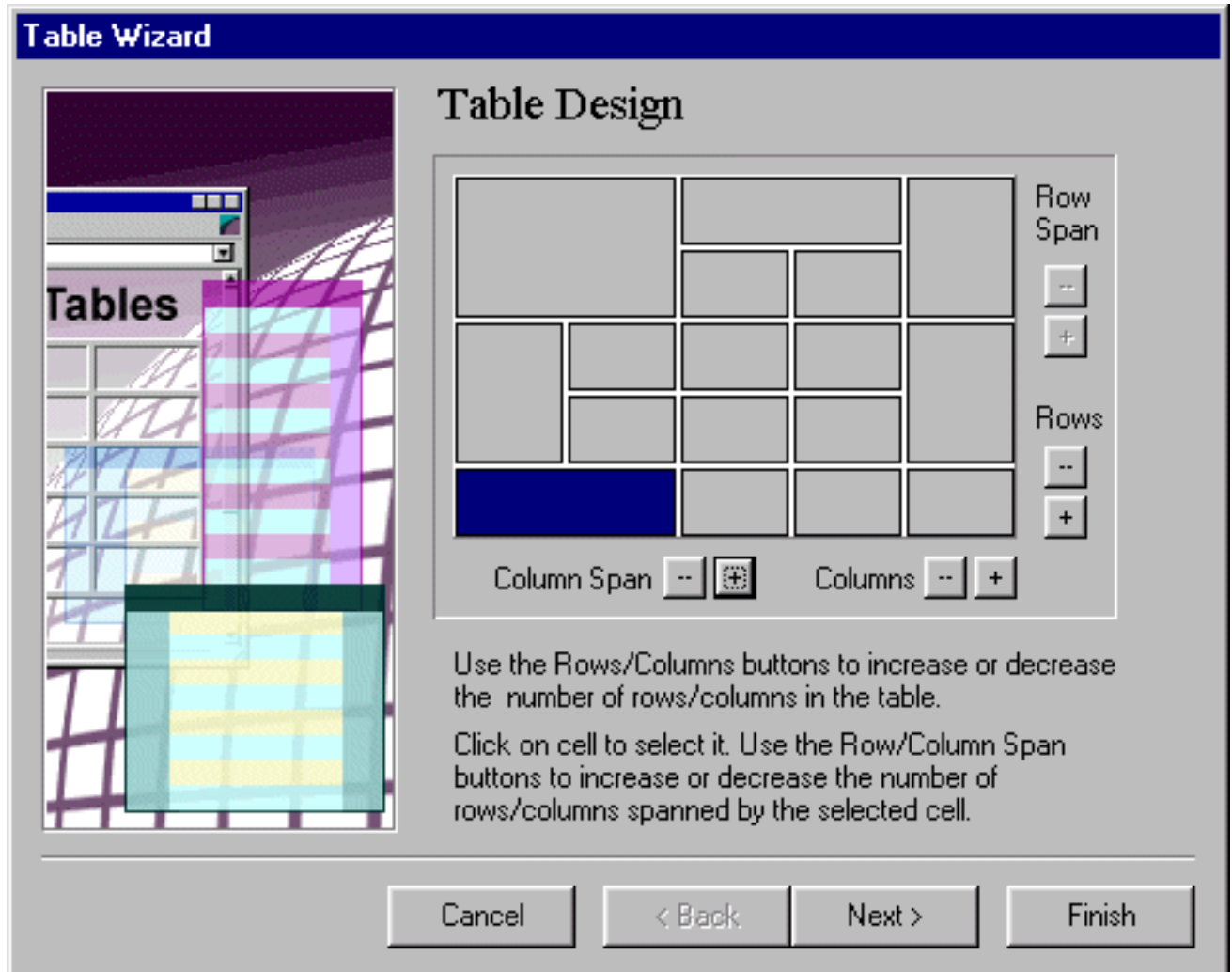

```
<TD ALIGN="center">700 USD</TD>
```

Repitiendo esto para el resto de las celdas ya tenemos la tabla terminada. El código final será:

```
<TABLE CELLSPACING="2" CELLPADDING="2" BORDER="1">
<CAPTION><FONT SIZE="+2" COLOR="#FF0000">SuperOFERTA viaje a Nueva
Zelanda</FONT></CAPTION>
<TR><TH ROWSPAN="2" COLSPAN="2"></TH>
<TH COLSPAN="2">Hoteles 3 estrellas</TH>
<TH ROWSPAN="2">Hoteles 4 estrellas</TH>
</TR>
<TR><TH>Sólo alojamiento</TH>
<TH>Pensión completa</TH>
</TR>
<TR><TH ROWSPAN="2">1 semana</TH>
<TH>Avión turista</TH>
<ALIGN="center">700 USD</TD>
<TD ALIGN="center">1000 USD</TD>
<TD ROWSPAN="2" ALIGN="center">1500 USD</TD>
</TR>
<TR><TH>Avión preferente</TH>
<TD ALIGN="center">850 USD</TD>
<TD ALIGN="center">1150 USD</TD>
</TR>
<TR><TH COLSPAN="2">2 semanas</TH>
<TD ALIGN="center">1300 USD</TD>
<TD ALIGN="center">1850 USD</TD>
<TD ALIGN="center">2600 USD</TD>
</TR>
</TABLE>
```

Una vez escrito por completo guardamos el archivo y volvemos a verlo con el navegador eligiendo la opción (o pulsando el botón) actualizar o *reload*. Si hemos conseguido crear esta tabla es indudable que somos capaces de crear cualquier otra. Como ya comentamos anteriormente el método usado por el lenguaje HTML para crear tablas no es muy sencillo y el proceso se hace algo pesado. Por esta razón en ocasiones es conveniente usar algún asistente o *wizard* de creación de tablas. Un excelente ejemplo de este tipo de asistentes es el que acompaña al editor de HTML HomeSite 3.0™ y que podemos ver en la figura 6.26. El problema de este tipo de asistentes es que no siempre permiten aprovechar todas las posibilidades del lenguaje HTML y además en ocasiones no obtenemos con ellos el resultado deseado. Pero ahora nosotros ya conocemos las etiquetas y atributos de creación de tablas del lenguaje HTML, con lo que no nos resultará complicado retocar el código creado con el asistente.

Figura 6.26. Los asistente (wizard) para la creación de tablas como este, incluido con el editor de HTML HomeSite™, son de gran ayuda en la creación de tablas complicadas.



Sugerencia

Un buen truco para hacer compatibles las con los navegadores que no entiendan la etiqueta `TABLE` ni sus *subetiquetas* es incluir la etiqueta `
` al final de cada fila (antes de `</TR>`). De esta manera forzaremos un cambio de línea que simula, a grosso modo, un cambio de fila.

Práctica 4 - Página Web con una estructura creada con tablas.

Es el momento de realizar de la cuarta práctica del curso de HTML. En ella seguiremos desarrollando el sitio Web de la agencia de viajes Los Alpes, pero esta vez añadiremos un gran avance de complejidad de acuerdo con los conocimientos que ahora tenemos.

El sitio Web constará de un total de cuatro páginas, y contará con gran contenido gráfico. Además usaremos las tablas tanto para insertar datos como para controlar la disposición de los elementos de las páginas, con lo que nos servirá para practicar lo recién aprendido en este capítulo.

Pero sin perder más tiempo vamos a comenzar con la realización de las nuevas páginas. Vayamos por pasos:

Paso 1 - Planificación.

En todo proceso de creación y diseño de páginas Web el primer paso a llevar a cabo será la planificación, antes incluso de escribir la primera línea de código. En esta planificación deberemos:

1. Fijar cuales son nuestros objetivos con la creación las páginas
2. Pensar que tipo de personas visitarán nuestra página para adaptar su contenido a ellas.
3. Planear el contenido que queremos incluir en las páginas.
4. Planificar la estructura general del sitio Web y la información que debe haber en cada una de las páginas.
5. Diseñar la disposición de los elementos en cada una de las páginas. En general la página principal será ligeramente distinta al resto pero debe mantener suficientes similitudes.
6. Lo más probable es que existan otras tareas a llevar a cabo específicas del Web en concreto que estemos realizando.

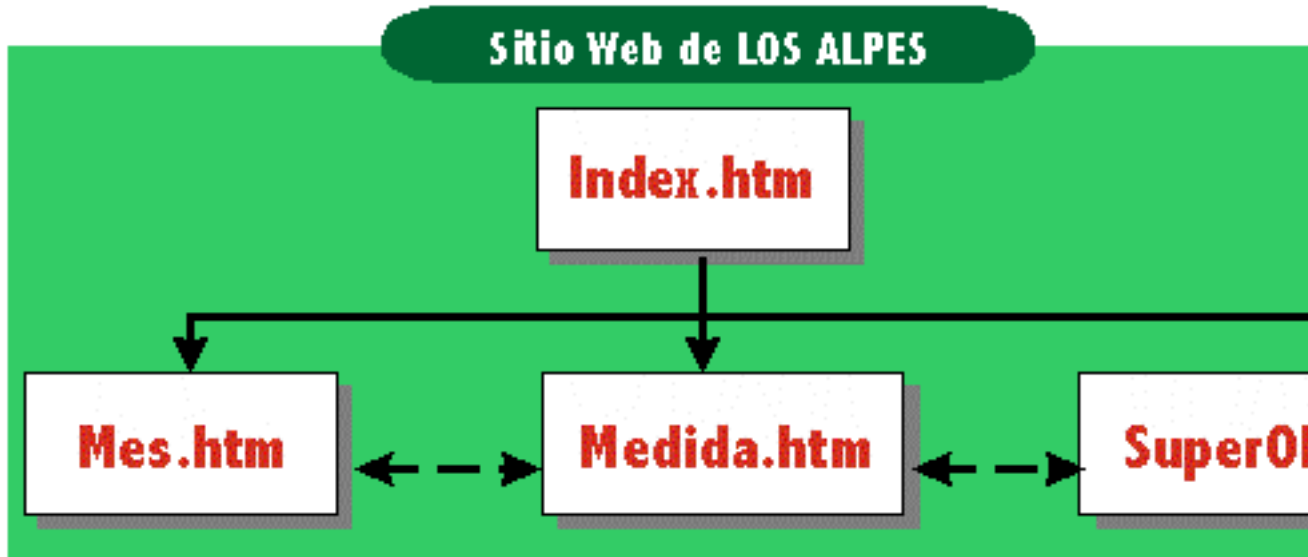
Nota

Para la realización de las tareas cuarta y quinta debemos ser conscientes de los conocimientos y herramientas con los que contamos para no proponernos objetivos irrealizables. Tener un buen conocimiento del lenguaje HTML, como ocurre en nuestro caso, nos permitirá eliminar muchas limitaciones que aparecen en este punto de la planificación.

Vayamos paso por paso realizando cada una de las tareas que acabamos de nombrar:

1. Nuestro objetivo será que las personas que visiten las páginas se interesen por las ofertas y se pongan en contacto con la agencia para contratar el viaje. Un segundo objetivo será dar una buena imagen de la compañía.
2. Todo tipo de personas podrán acceder a nuestras páginas por lo que será conveniente que puedan ser vistas con cualquier navegador.
3. En las páginas informaremos sobre el prestigio de la agencia y sus ofertas de viajes más atractivos.
4. La página principal dará información general sobre la empresa y como contactar con ella y ofrecerá enlaces para ir a las distintas secciones del sitio Web: Ofertas, Viajes a su medida y la superOferta. En total el sitio Web quedará compuesta por cuatro páginas: la principal y tres páginas de contenido que estarán además interconectadas entre sí. En la figura 6.27 podemos ver la estructura del sitio Web de Los Alpes tal y como lo hemos planeado.

Figura 6.27. Estructura del sitio Web de Los Alpes. La página principal tendrá enlaces a las otras tres que además estarán enlazadas entre sí.



5. Es hora de diseñar la disposición de los elementos en cada una de las páginas. Aquí es donde más variarán las páginas con respecto a las de las prácticas anteriores. Usando tablas para insertar el contenido tendremos una gran libertad en la colocación de los distintos elementos de la página. Para la estructura de la página principal usaremos dos tablas, la primera para la cabecera con el logo e imagen de Los Alpes y la segunda para dividir el cuerpo en dos columnas. En la de la izquierda pondremos las diferentes opciones del sitio Web y a la derecha el texto. En la figura 6.28 podemos ver un boceto de cómo quedará la página. Las líneas azules discontinuas representan los bordes de las tablas que necesitaremos usar.

Figura 6.28. Boceto de la disposición de los elementos en la página principal y las tablas que vamos a necesitar crear para conseguir esa visualización.

Sugerencia

Un buen truco para cuando creamos la estructura de un documento HTML basándonos en tablas es mantener el borde visible, `BORDER="1"`, para asegurarnos que toma la forma que deseamos. Cuando ya sepamos que funciona bien eliminaremos el borde, `BORDER="0"`.

El resto de páginas serán ligeramente distintas. La tabla del encabezado tendrá una fila más con las opciones del sitio Web y con un *botón* para volver a la página principal. El cuerpo se dividirá en el texto (derecha), y una serie de anclas a las distintas secciones (izquierda) de ese mismo documento HTML. En la figura 6.29 podemos ver la estructura general de estas páginas.

Figura 6.29. Boceto de la disposición de los elementos del resto de páginas. Aunque conserva un parecido con la página principal hay algunos cambios.

Mes.htm



Los Alpes

Principal Opción1 Opción2 Opción3

OFERTAS DEL MES

Sección 1

Sección 2

Sección 3

Una vez hemos creado la estructura del sitio Web y ya tenemos una idea de cómo va a ser la disposición de los elementos en el sitio Web es hora de empezar a escribir el código.

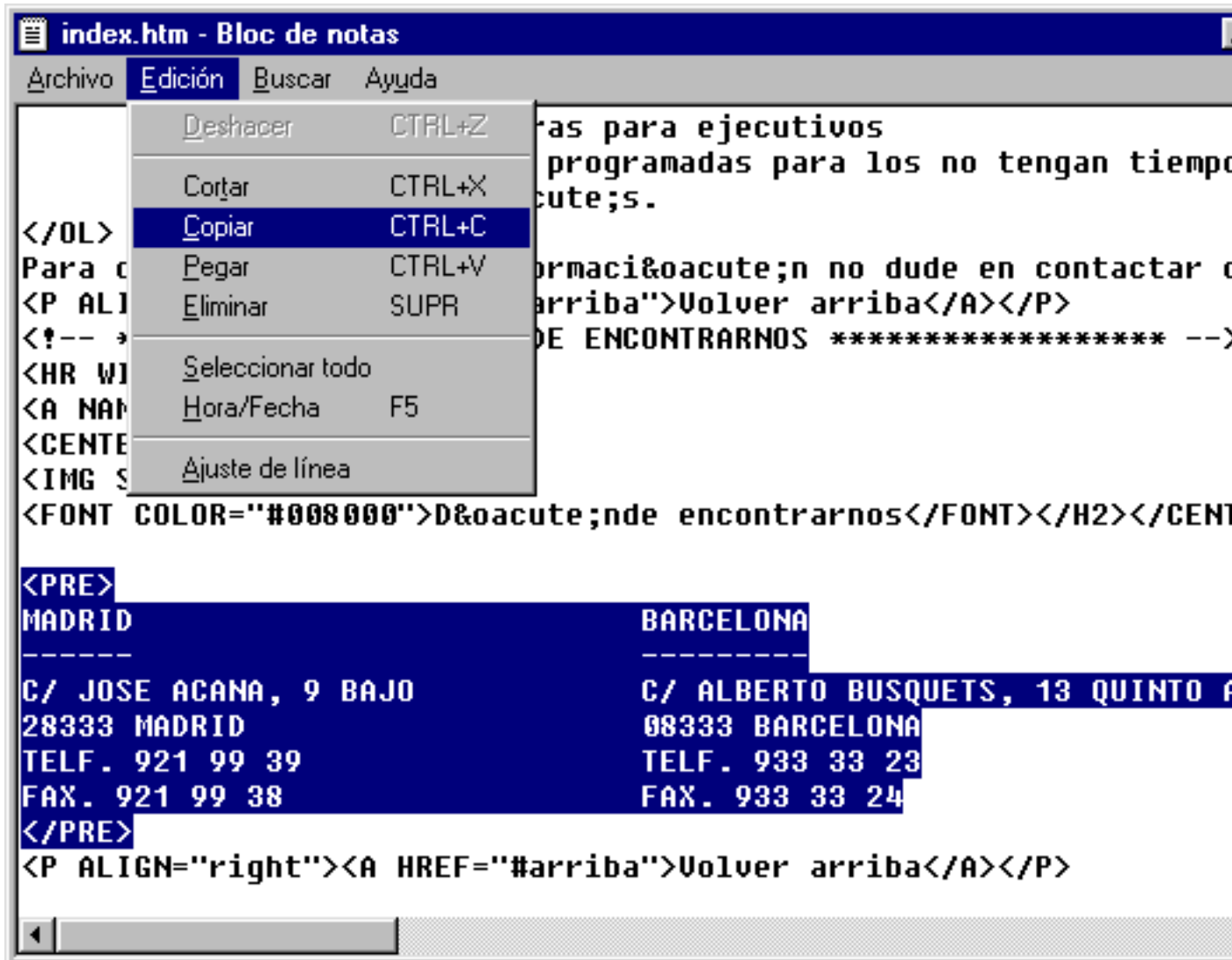
Paso 2 - Código básico.

En primer lugar ejecutamos nuestro editor texto o editor de HTML favorito, ya sea el bloc de notas, el editor HTML de FrontPage™ o cualquier otro, y creamos un archivo nuevo. Es recomendable crear un archivo nuevo porque aunque usaremos parte del código escrito en la práctica 2 y 3 los cambios serán significativos y modificar ese código sería algo complicado.

Nota

Como parte del código que usaremos será igual al que utilizamos para las prácticas 2 y 3 recomendamos a aquellos lectores que tengan esta posibilidad que en vez de escribir de nuevo dicho código lo copien de lo que ya tienen. Para ello lo más cómodo es usar las opciones de cortar y pegar que ofrecen tanto Windows™ como MacOS™ y Unix™. Será necesario tener tanto el documento HTML nuevo como el archivo del que queremos copiar el código abierto. Estando activo este último seleccionamos la sección de código a copiar y elegimos en el menú archivo la opción copiar (figura 6.30). En algunos programas la manera de copiar será distinta pero en todo caso será muy similar. Después vamos al archivo que estamos escribiendo, situamos el cursor en el punto donde queremos insertar el código copiado y elegimos la opción pegar del menú edición. Ya hemos copiado el código y nos hemos ahorrado el trabajo de volver a escribirlo.

Figura 6.30. Usando las opciones de cortar y pegar nos ahorraremos el proceso reescribir el código de las prácticas anteriores.



Una vez tenemos delante un archivo nuevo escribimos las etiquetas que definen la estructura del documento HTML:

```
<HTML>
<HEAD>
<TITLE>Agencia de viajes LOS ALPES</TITLE>
</HEAD>
<BODY>
<!-- Resto del código -->
</BODY>
</HTML>
```

Paso 3 - Estructura de la página principal.

El siguiente paso consistirá en dar la estructura a las páginas y aquí es donde empezamos a poner en práctica nuestros conocimientos sobre tablas. Por ahora vamos a centrarnos en crear la página principal, `index.htm`. En la figura 6.28 veíamos cómo planeamos colocar los distintos elementos. Es hora de escribir el código necesario para llevar esa planificación a la práctica. Serán necesarias dos tablas ambas de una fila y dos columnas.

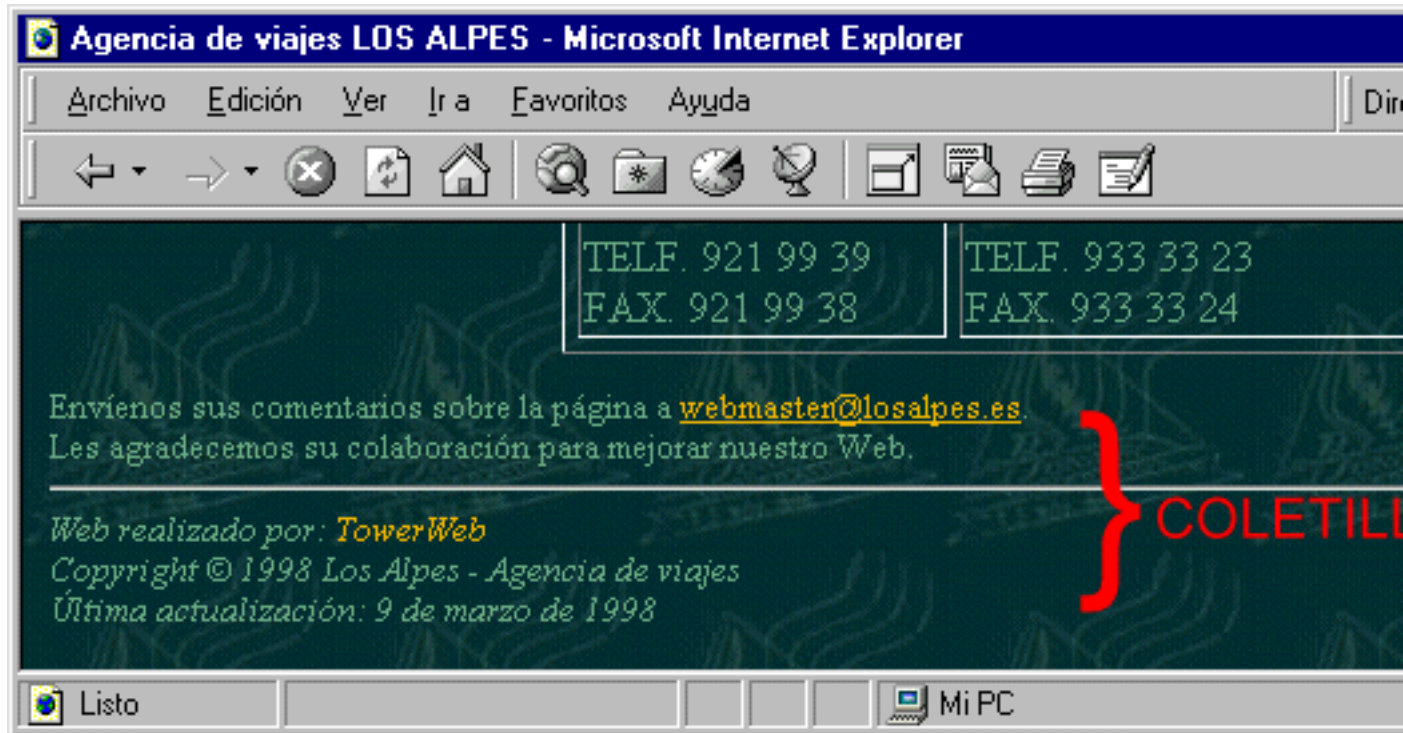
```
<TABLE ALIGN="center">
<TR>
<TD><!-- Logo --></TD>
<TD><!-- Los alpes --></TD>
</TR>
</TABLE>
<TABLE WIDTH="100%">
<TR>
<TD><!-- Opciones --></TD>
<TD><!-- texto --></TD>
</TR>
</TABLE>
```

La primera de las tablas la hemos centrado usando el atributo `ALIGN`. La segunda nos convendrá que ocupe todo el ancho posible, por lo que usamos `WIDTH="100%"`. Después de la segunda tabla simplemente pondremos datos relacionados con la elaboración de la página (dirección Webmaster, Copyright, fecha actualización, etc.). A este tipo de información incluida generalmente al final de las páginas se le denomina en ocasiones *coletilla*. A continuación podemos ver el código de nuestra coletilla.

```
<ADDRESS>
<SMALL>
Web realizado por: <FONT COLOR="#EDB301">TowerWeb</FONT>
<BR>
Copyright © 1998 Los Alpes - Agencia de viajes<BR>
Última actualización: 9 de marzo de 1998
</SMALL>
</ADDRESS>
```

En la figura 6.31 vemos el resultado de este código. En esta figura vemos un adelanto de la imagen de fondo y color del texto que más adelante incluiremos en la página, además hemos incluido un texto adicional (sólo en la página principal) para animar a los visitantes a hacer sugerencias sobre la página.

Figura 6.31. La coletilla aporta un toque de profesionalidad a nuestra página además de ofrecer información interesante a los visitantes.



Como vemos, hemos usado la etiqueta `SMALL` para que el texto de la coletilla sea menor que el del resto de la página.

Esta coletilla la repetiremos en cada una de nuestras páginas, y posteriormente iremos cambiando la fecha de la última actualización en cada una individualmente.

Paso 4 - Estructura del resto de páginas.

La disposición de los elementos Web en el resto de páginas será muy similar a la de la página principal. En este caso crearemos en primer lugar una página con el código que deben compartir todas las páginas del sitio Web y lo guardaremos en un archivo de nombre `plantilla.htm`. Para crear el resto de páginas usaremos como base el código de este archivo, que será: el código base que escribimos en el paso 2 más:

```
<TABLE ALIGN="center">
<TR>
<TD><!-- Logo --></TD>
<TD><!-- Los alpes --></TD>
</TR>
<TR>
<TD><!-- Opciones --></TD>
</TR>
</TABLE>
<TABLE WIDTH="100%">
<TR>
<TD><!-- Secciones --></TD>
<TD><!-- texto --></TD>
```

```
</TR>
</TABLE>
```

También podremos añadir una *coletilla* con datos sobre la página Web al igual que hacíamos en el caso de la página principal.

Paso 5 - Contenido de la página principal.

Terminada la planificación y una vez creada la estructura de las páginas ahora es cuando empezaremos a ver los resultados de nuestro trabajo. Empezaremos poniéndole contenido a la página principal. En este momento recuperamos el código que escribimos en el paso 3 y empezamos a rellenar las celdas de las tablas. Empezamos con la primera tabla. En la celda de la derecha ponemos el logo de Los Alpes y en la de la derecha ponemos el letrero. Como en ambos casos debemos insertar imágenes usamos la etiqueta `IMG` de la siguiente manera:

```
<CENTER>
<TABLE ALIGN="center">
<TR>
<TD><IMG SRC="img/logo.gif" WIDTH=90 HEIGHT=99 ALT="[LOGO]"></TD>
<TD><IMG SRC="img/losalpes.gif" WIDTH=440 HEIGHT=99 ALT="LOS ALPES - Agencia de
Viajes"></TD>
</TR>
</TABLE>
</CENTER>
```

Nota

Para preparar las páginas para los visitantes que usen navegadores en modo texto hemos rellenado el atributo `ALT` de la etiqueta `IMG`. Además hemos usado los atributos `WIDTH` y `HEIGHT`, pero no para modificar su tamaño, sino porque de esta manera conseguimos una carga más rápida de las páginas, como comentamos en el capítulo 5.

Es el momento de rellenar la segunda de las tablas. La columna de la derecha la planeamos para insertar el texto de la página. En ella insertamos una breve descripción de la Agencia de viajes Los Alpes para darla a conocer (podremos ver este texto completo en la figura 6.32). A continuación ponemos las direcciones de las sedes en Madrid y Barcelona de los Alpes. Si recordamos en la práctica 2 usamos la etiqueta `PRE` para introducir dichas direcciones. Ahora podemos usar las tablas para conseguir el mismo efecto y con mejores resultados:

Figura 6.32. Página principal de la agencia de viajes Los Alpes. Aunque no podemos ver los bordes de las tablas podemos intuir por donde irían.

Agencia de viajes LOS ALPES - Microsoft Internet Explorer

Archivo Edición Ver Ir a Favoritos Ayuda

Los alpes
Agencia de viajes

Viajes a su medida

Ofertas

SuperOFERTA

La agencia de viajes Los Alpes goza de un gran prestigio. Numerosas empresas confían en nosotros, conociendo nuestra dedicación. Con nosotros los viajes de la empresa de

Pero no sólo las empresas nos ofrecen su apoyo, cada día confían en nosotros para organizar sus vacaciones y t...
¿Por qué? porque en Los Alpes ofrecemos lo que la empresa desea sin preocupaciones, atención personal, calidad de los servicios a un precio muy económico.

Dónde encontrarlos

MADRID C/JOSE ACANA, 9 BAJO 28333 MADRID TELF. 921 99 39 FAX. 921 99 38	BARCELONA C/ALBERTO BUSQUETS 08333 BARCELONA TELF. 933 33 23 FAX. 933 33 24
--	--

Envíenos sus comentarios sobre la página a webmaster@losalpes.es.
Les agradecemos su colaboración para mejorar nuestro Web.

Web realizado por: **TowerWeb**
Copyright © 1998 Los Alpes - Agencia de viajes
Última actualización: 9 de marzo de 1998

```

<TABLE WIDTH="100%">
<TR>
<TD><!-- Opciones --></TD>
<TD><!-- texto -->
<P>La agencia de viajes Los Alpes goza de un gran prestigio en el mundo de los ...
<TABLE BORDER="1" CELSPACING=5>
<TR><TD>
<STRONG><U>MADRID</U></STRONG><BR>
C/JOSE ACANA, 9 BAJO<BR>
28333 MADRID<BR>
TELF. 921 99 39<BR>
FAX. 921 99 38<BR>
</TD><TD>
<STRONG><U>BARCELONA</U></STRONG><BR>
C/ALBERTO BUSQUETS, 13 QUINTO A<BR>
08333 BARCELONA<BR>
TELF. 933 33 23<BR>>
FAX. 933 33 24<BR></TD></TR>
</TABLE>
</TD>
</TR>
</TABLE>

```

Ya podemos dar por terminada la columna de la derecha de la página principal. La columna de la izquierda tendrá enlaces a las distintas páginas del sitio Web y volveremos sobre el código usado en el paso 6 al hablar del contenido gráfico. Por ahora podemos ver un adelanto de cómo va a quedar esta página principal mostrado en la figura 6.32.

Paso 6 - Contenido gráfico.

La apariencia de un Web es uno de los aspectos más importantes en la creación de este. Por ello dedicamos un largo capítulo 5 a él. Ahora es el momento de poner en práctica de nuestros conocimientos. A la hora de insertar gráficos tenemos dos posibilidades:

1. Crear nuestros propios gráficos usando un programa de dibujo vectorial y/o un programa de retoque fotográfico. Esta es, sin duda, la mejor de las dos opciones, pero también es la que lleva más tiempo así como varios programas y amplios conocimientos de su uso por parte del creador de los gráficos. Por esta razón es común recurrir a la segunda de las posibilidades.
2. Crear nuestros propios gráficos usando un programa de dibujo vectorial y/o un programa de retoque fotográfico. Esta es, sin duda, la mejor de las dos opciones, pero también es la que lleva más tiempo así como varios programas y amplios conocimientos de su uso por parte del creador de los gráficos. Por esta razón es común recurrir a la segunda de las posibilidades.

En nuestro caso vamos a optar por la segunda opción. Hemos usado algunos gráficos del clipart que acompaña a Microsoft FrontPage™, en concreto al tema *Travel*. Posteriormente hemos retocado alguno de estos gráficos con Paint Shop Pro™ (podríamos haber usado cualquier otro programa de retoque fotográfico, pero hemos elegido este por ser shareware). Usando este programa hemos creado una especie de botones (tranque servirán como enlace a las distintas páginas de la Web. Ahora ya podemos escribir el código usado:

```

<TABLE ALIGN="center">
<TR>
<TD><IMG SRC="img/logo.gif" WIDTH=90 HEIGHT=99 BORDER=0 ALT="[LOGO]"></TD>

```

```
<TD><IMG SRC="img/losalpes.gif" WIDTH=440 HEIGHT=99 BORDER=0 ALT="LOS ALPES -
Agencia de Viajes"></TD>
</TR>
</TABLE>
</CENTER>

<TABLE CELLPADDING="10">
<TR>
<TD VALIGN="top">
<!-- Opciones -->
<A HREF="medida.htm"><IMG SRC="img/tranvha1.jpg" WIDTH=136 HEIGHT=48 BORDER=0
ALT="Viajes a su medida"></A>
<A HREF="mes.htm"><IMG SRC="img/tranvha2.jpg" WIDTH=136 HEIGHT=48 BORDER=0
ALT="Ofertas del Mes"></A>
<A HREF="superofe.htm"><IMG SRC="img/tranvha3.jpg" WIDTH=136 HEIGHT=48 BORDER=0
ALT="La superOFERTA"></A>
</TD>
<TD>
<!-- texto -->
<P><FONT SIZE="+2">L</FONT>a agencia de viajes Los Alpes goza de un gran prestigio
en el mundo de...
```

Nota

También podíamos usar una tabla de tres filas dentro de esta celda para tener un mayor control sobre la disposición de los botones, poniendo cada uno en una de las filas. Dejamos la realización de esta mejora como ejercicio para el lector.

Figura 6.33. Visualización de medida.htm

Los Alpes
Agencia de viajes

Principal Viajes a su medida Ofertas

VIAJES A SU MEDIDA

EMPRESAS

TURISTAS

ESTUDIANTES

JUBILADOS

Cada persona tiene unas características especiales y unos requisitos. En la agencia de viajes LOS ALPES no tratamos a todo el mundo de la misma manera, cada persona de manera personalizada. Una prueba de ello son nuestros servicios para USTED:

EMPRESAS

Porque sabemos lo que su empresa quiere hemos creado una serie de servicios que resolverán todos sus problemas. Lea sus características y no dude en contactar con nosotros mismo.

- Los viajes para empresas incluyen transporte de ida, con alojamiento y hospedaje en hoteles de nuestra cadena.
- Cuando el viaje es en avión un mensajero parará a recogerlo en el aeropuerto. Más tarde enviaremos un chófer a recogerle, para que usted sólo tenga que coger el avión y volar.
- En la estación o aeropuerto de destino le estarán esperando.
- Si tenía equipaje un mensajero lo recogerá en el aeropuerto.

TURISTAS

Paso 7 - Cambiar el fondo y color del texto.

Hemos elegido usar una imagen de fondo, y como esta imagen es oscura debemos además cambiar el color del texto y los enlaces. Todo ello lo hacemos con los atributos de la etiqueta BODY:

```
<BODY LINK="#FED354" VLINK="#EDB301" ALINK="#FFFF00" BGCOLOR="#335142" ↵  
TEXT="#71AC84" BACKGROUND="img/tratilea.jpg">
```

Este código deberemos usarlo en todas y cada una de las páginas. En él hemos usado el atributo BGCOLOR para poner un color de fondo similar al color de la imagen, de esta forma si hay algún problema con la imagen el color de fondo especificado se usará en su lugar y la página no perderá mucho.

Figura 6.34. Resultado final de la página mes .htm



Los Alpes

Agencia de viajes

[Principal](#) [Viajes a su medida](#) [Ofertas](#)

NUESTRAS OFERTAS DE

En la Agencia de viajes Los Alpes estamos celebrando nuestros primeros 20 años, y **OFERTAS Y VIAJES ESPECIALES** que no podrá resistir. Entre ellas destaca

- [ANDORRA](#)
- [ORLANDO](#)
- [Los Angeles](#)
- [PARÍS](#)
- [MOSCÚ](#)

ANDORRA (Salidas desde Madrid y

Viajes de una semana de duración
Desde el 12/17 de Enero al 16/21 de Marzo y del 31 de Ma

Hotel Florida, o similar ***	36
Semana Carnaval, 9/14 Febrero	39

Información y reservas: 902 333 333

ORLANDO (Salidas desde Ma

Salida 7 Enero al 11 Febrero Hotel Travelodge Mail

7 noches **104.000 pesetas**, niños 2 a 11 años con 2 adultos **59.000**

Incluye avión ida y vuelta US Airways traslados aeropuerto hotel aerop

Paso 8 - Realización del resto de las páginas.

Lo que queda es ya muy similar a lo que llevamos hecho. Usando la plantilla creada en el paso 4 y ciertas imágenes (aunque invitamos al lector a crear las suyas) no nos será difícil. Para que sea más fácil ahí van algunas pistas:

- En las figuras 6.33, 6.34 y 6.35 hemos ido mostrando el resultado final de estas páginas.
- Para la página `mes.htm` hemos creado 5 nuevos botones para cada uno de los viajes y los hemos situado en la columna de la derecha de la segunda tabla, tal y como habíamos planeado.
- Igualmente hemos creado botones para la página `medida.htm`.
- El último comentario es que para el archivo `SuperOFE.htm` al final suprimimos la segunda de las tablas por ser innecesaria. En esta página usamos la tabla creada en la práctica 3.

Figura 6.35. Resultado final de la página superOFE.htm



Los Alpes
Agencia de viajes

Principal Viajes a su medida Ofertas

LA SUPEROFERTA

En la agencia de viajes Los Alpes no nos cansamos de buscar nuevas fórmulas para nuestros clientes.

De ahora en adelante todos los meses ofreceremos nuestra **SuperOFERTA** que ofrece numerosas opciones a elegir y con visitas turísticas guiadas totalmente gratis.

Viaje a Nueva Zelanda

		Hoteles 3 estrellas	
		Sólo alojamiento	Pensión completa
1 semana	Avión turista	700 USD	1000 USD
	Avión preferente	850 USD	1150 USD
2 semanas		1300 USD	1850 USD

No lo dude más y [póngase en contacto](#) con nosotros para reservar su plaza antes de

Principal Arriba

Paso 9 - Retoques, pruebas, revisión y mantenimiento.

El último paso es el más largo de todos y consta de varias tareas. En primer lugar se pueden retocar las imágenes para que queden mejor y/o ocupen menos se puede aumentar el tamaño de la primera letra de cada párrafo (como hemos hecho nosotros) y en definitiva cualquier mejora del aspecto de las páginas. Después hay que hacer pruebas de la página. Son especialmente importantes las pruebas con otros navegadores, con varias resoluciones de pantalla (600x480 y 800x600 especialmente) y con distintas cantidades de colores, en concreto debemos ver si con 256 colores los gráficos todavía se ven bien.

Si este fuese un sitio Web real deberíamos además deberíamos realizar continuas revisiones de las páginas y mantener un contenido actualizado constantemente.

Una vez hemos creado el Web completo podemos tomarnos un descanso y enseñar las páginas a nuestros conocidos. Sin duda es como para estar orgullosos de lo que ya somos capaces de hacer. De ahora en adelante aprenderemos los aspectos más avanzados, y a la vez más interesantes, del lenguaje HTML.

HTML avanzado

Jorge Ferrer

Victor García

Versión 1.0

Copyright Jorge Ferrer y Victor García. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation. Puede consultar una copia de la licencia en: <http://www.gnu.org/copyleft/fdl.html>

Ya hemos terminado la primera fase en el aprendizaje de la programación de documentos HTML. Dominamos todas las etiquetas necesarias para crear páginas de gran calidad. Es hora de comenzar con una segunda fase dedicada al HTML avanzado. En esta fase aprenderemos etiquetas para crear efectos muy llamativos en las páginas Web e iremos conociendo mejor cómo funcionan los navegadores para crear páginas que se ven correctamente en todos.

A estas alturas del curso ya conocemos una gran cantidad de etiquetas. Sabemos controlar el tamaño, color y formato del texto, incluir imágenes y todo tipo de contenido gráfico, crear listas, disponer datos en tablas e incluso, usando estas, tener un mayor control sobre la disposición de los elementos de una página. En definitiva, somos capaces de crear páginas Web como el mejor de los programadores del WWW. Con estos datos en la mano podríamos pensar que no queda nada por aprender, pero nada más lejos de la realidad, todavía nos queda lo mejor. Las etiquetas vistas hasta ahora son las más usadas desde hace ya tiempo en las páginas Web, pero con el tiempo y debido a las necesidades que acusaban los programadores han ido apareciendo nuevas etiquetas que nos permitirán crear efectos realmente atractivos. A estas etiquetas y a su correcto uso en las páginas Web es a lo que hemos llamado HTML avanzado y gracias a él aprenderemos a crear los llamados Webs de tercera generación como el que podemos ver en la figura 7.1:

Figura 7.1. Usando adecuadamente las etiquetas del lenguaje HTML y con algunos conocimientos de diseño se pueden crear webs tan atractivos como este.



Entre los temas que cubriremos de ahora en adelante destacan: mayor control en la creación de listas y en el alineamiento de los elementos de la página, inserción de scripts y hojas de estilo, inserción de applets de Java, formularios interactivos, páginas con varias ventanas (los marcos o frames), HTML dinámico, y más.

En definitiva todas las novedades que han ido apareciendo en estos últimos años el mundo de la programación de páginas web.

Pero todo lo nuevo tiene, además de grandes ventajas, algunos inconvenientes. Al hablar de HTML, los inconvenientes de usar algunas de las etiquetas avanzadas es que no son entendidas por los navegadores y por tanto debemos tener cuidado al usarlas. Esto no quiere decir que no las usemos, ya que si no lo hiciéramos estaríamos desperdiciando algunas de las posibilidades que nos ofrece el lenguaje HTML. La solución es simplemente conocer las ventajas e inconvenientes y como subsanar estos últimos de la mejor manera posible. A esto es a lo que dedicaremos gran parte de este capítulo. Para empezar, es interesante (incluso necesario) conocer un poco la historia del lenguaje HTML a través de sus diferentes versiones.

Estándares y extensiones del lenguaje HTML

El lenguaje HTML empezó a usarse a gran escala a partir de 1990 cuando Tim Berners Lee, mientras trabajaba en el laboratorio Europeo de Física de Partículas (CERN) en Suiza (ver figura 7.2) desarrolló un novedoso sistema de distribución de información basado en páginas independientes, llamadas páginas web (*web pages*) o simplemente webs, con posibilidad de hipertexto. Era el nacimiento del World Wide web (de ahora en adelante WWW), que podría traducirse como la *telaraña mundial*.

Figura 7.2. En el CERN (<http://www.cern.ch> [<http://www.cern.ch>]), un instituto de física con poca relación con Internet, se engendró la idea del World Wide Web.



A partir de entonces el WWW no ha parado de crecer y lo que empezó como un pequeño proyecto se convirtió en un sistema usado en todo el mundo a través de la red Internet. En 1995 el WWW se convirtió en el servicio de esta red más utilizado por sus usuarios por delante incluso del e-mail o el FTP (sistema usado para transporte de ficheros).

HTML era el lenguaje usado para crear las páginas individuales que formaban esa telaraña mundial. Este era un lenguaje muy sencillo que permitía incluir texto, encabezados, imágenes, formularios simples y muy poco más. Debido a la gran expansión del WWW el HTML empezó a ser usado ampliamente y diversas empresas se dieron cuenta del potencial que representaba y empezaron a comercializar programas, posteriormente llamados navegadores, exploradores o browsers, capaces de encontrar páginas de esta telaraña en Internet e interpretar el código HTML para presentar la información de las páginas.

Pero entonces surgió el desorden. Cada una de las empresas u organismos que elaboraban los navegadores proponía su propia interpretación del lenguaje y programaba sus aplicaciones para que fuesen fieles a esa interpretación. Esto causaba verdaderos quebraderos de cabeza a los programadores de webs ya que si preparaba su página para que funcionase con un navegador no funcionaba con otro o las diferencias eran más importantes de lo se puede considerar como aceptable.

El estándar HTML 2.0

Para solucionar este desorden se creó un organismo, el World Wide Web Consortium (de ahora en adelante W3C), que se encargaría de estandarizar todos los temas relacionados con el WWW y por supuesto el lenguaje HTML entraba en sus objetivos. En la figura 7.3 podemos ver la página de esta organización.

Figura 7.3. El World Wide Web Consortium, o W3C, fue creado para regular el World Wide Web. Esta regulación incluye una estandarización del lenguaje HTML.



El primer estándar serio que vio la luz fue el HTML 2.0. Este estándar pretendía contentar a todos los navegadores existentes y en un principio lo consiguió. De hecho podemos considerar que todos los navegadores existentes actualmente soportan este estándar, por lo tanto cualquier página escrita usando código HTML 2.0 podrá ser visualizado en cualquiera de ellos. Este estándar incluía:

- Texto normal y algunos pocos estilos físicos y lógicos, pero no se podía variar ni el tamaño ni el color. También incluía alguna etiqueta de párrafos como `<P>` o `
` pero sin ningún atributo.
- Seis niveles de encabezados.
- Formularios básicos.
- Enlaces (también llamados links).
- Imágenes.

Y muy poco más. Pronto se demostró que este estándar era insuficiente, y muy pronto las principales empresas que habían creado navegadores, como Netscape Navigator™ y más tarde Internet Explorer™. Empezaron a inventarse nuevas etiquetas HTML que, por supuesto, sólo serían válidas si se usaba su navegador. A estas nuevas etiquetas se las llamó extensiones y entre las más importantes destacan las tablas, frames, imágenes de fondo, etc. El desorden surgió de nuevo.

El estándar HTML 3.2

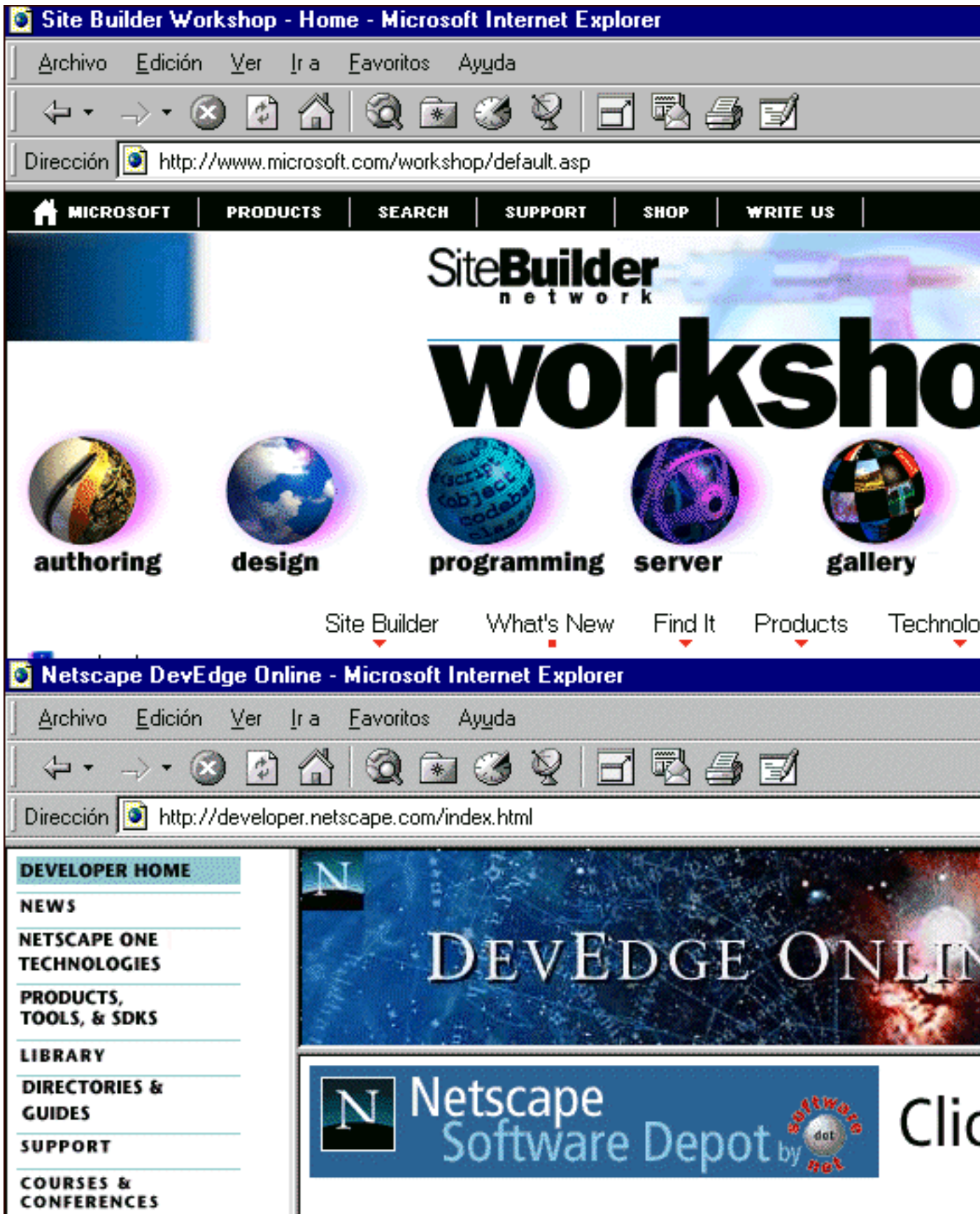
El W3C se apresuró en publicar un nuevo estándar, HTML 3.0, que incluía algunas de las extensiones de mayor éxito. Sin embargo este estándar era excesivamente diferente a HTML 2.0. Introducía muchas etiquetas nuevas, y algunas de dudosa utilidad. Por esta razón el propio Web Consortium aconsejó que dejase de usarse. Poco tiempo después apareció una revisión de este estándar el HTML 3.2 que es el que rige actualmente. En este estándar se incluyeron:

- Nuevos elementos para introducir en la cabecera como `<SCRIPT>` para incluir código de JavaScript o VBScript y otros como `<STYLE>`, `<META>`, `<BASE >` y `<LINK>`
- Nuevos atributos para la etiqueta `BODY` para poner color o imágenes de fondo y para cambiar el color del texto y links.
- Nuevos atributos para la mayoría de las etiquetas existentes,
- La posibilidad de cambiar el tamaño y color del texto con la etiqueta ``.
- Y por último, y probablemente la más importante, la posibilidad de usar tablas.

Sin embargo no todas las extensiones que habían creado los fabricantes de navegadores fueron introducidas en el estándar. Algunas de ellas muy extendidas, como por ejemplo los frames (explicados más adelante) se dejaron fuera, con lo que no se consiguió satisfacer a todo el mundo. Además, Microsoft™ ya había entrado de lleno en el mundo Internet con su navegador Internet Explorer™, lo que propició una explosión de extensiones. Hoy en día los programadores de páginas web deben tener en cuenta no sólo el lenguaje HTML estándar sino las etiquetas y atributos que van incluyendo Microsoft™ y Netscape™. Ambas compañías crearon páginas con información sobre cómo hacer páginas web aprovechando todas las características de sus navegadores, al margen de los estándares. La figura 7.4 muestra estas páginas tal y como se presentaban hace algunos años.

Figura 7.4. Tanto Microsoft™ como Netscape™ crearon sitios web especialmente dedicados a programadores de Internet en el que informan de las tecnologías propietarias

soportadas por sus navegadores.



HTML 4.0 y XHTML

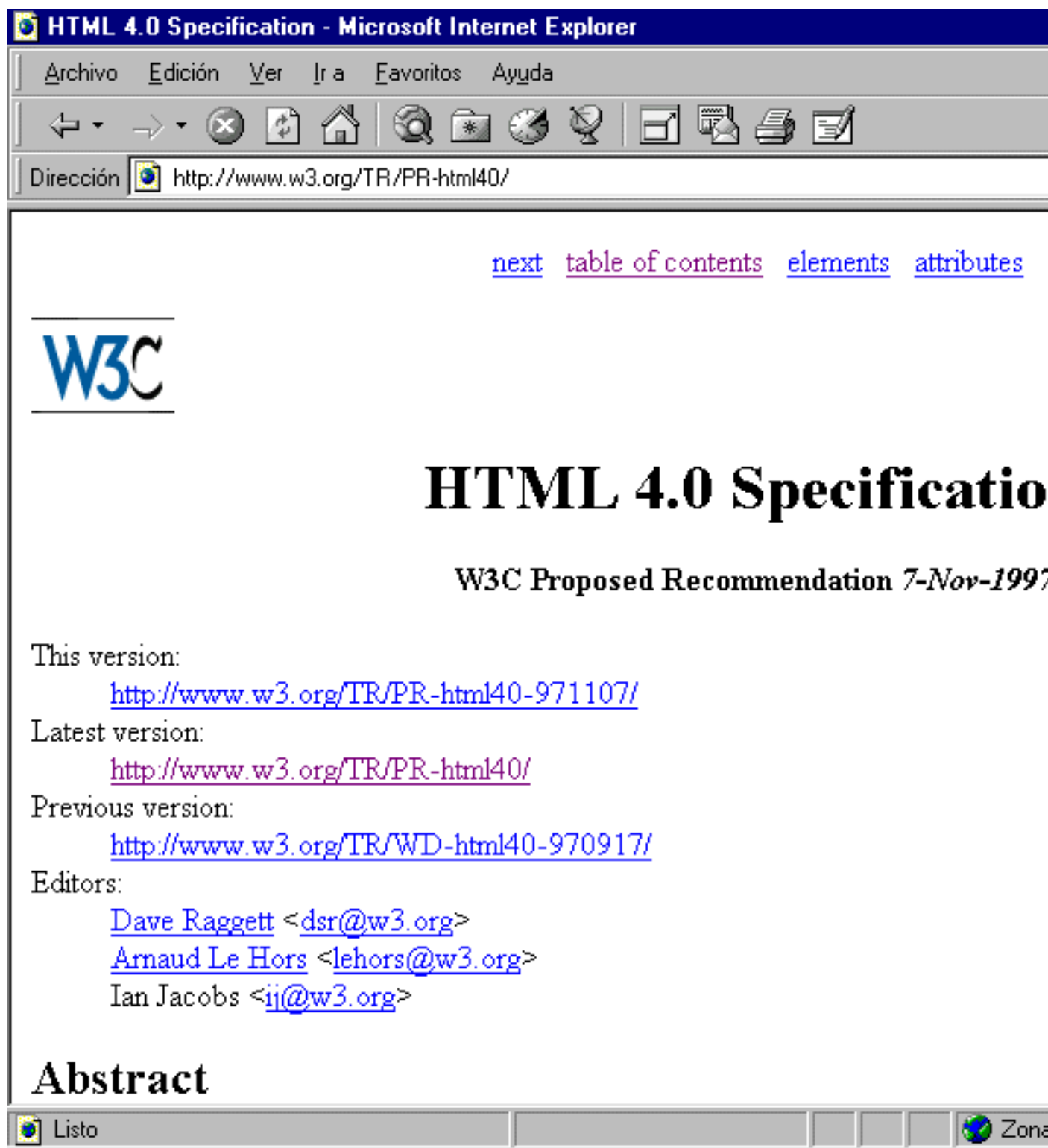
Dado el gran incremento de incompatibilidades entre los diferentes navegadores, el W3C hizo un gran esfuerzo para reunir a las principales compañías y crear un nuevo estándar que todas se comprometían a cumplir: HTML 4.0. Este estándar incluye muchas novedades, algunas muy usadas como los frames.

HTML 4.0 es tremendamente completo y hay pocas excusas para crear páginas web que usen extensiones. Parece que los creadores de navegadores han empezado a dar una mayor importancia al soporte de estándares, lo que permite crear páginas web HTML que se vean correctamente por todo el mundo. En particular es destacable el esfuerzo del proyecto Mozilla (<http://mozilla.org>), que ha tomado las riendas del desarrollo del navegador Netscape Navigator como software libre. En este proyecto han demostrado un fuerte compromiso por el soporte de los estándares y el abandono de las extensiones propietarias.

En la figura 7.5 podemos ver el comienzo del documento que define este estándar así como la dirección donde podemos encontrarlo:

Figura 7.5. En el servidor web del W3C podemos ver la especificación del nuevo estándar HTML 4.0. La dirección concreta es: <http://www.w3c/TR/PR-html40/>

[<http://www.w3c/TR/PR-html40/>]



Desde hace ya algún tiempo todos los navegadores soportan el estándar HTML 4.0: Internet Explorer, Netscape Navigator, Mozilla, Konqueror, Opera, etc. Su uso será una garantía de compatibilidad para nuestros sitios webs.

Atención

Si bien es cierto que usando únicamente etiquetas de HTML 4.0 conseguimos crear páginas que se vean en cualquier navegador han aparecido otras tecnologías como CSS, JavaScript, Applets Java, Flash, etc cuyo soporte varía mucho de unos navegadores a otros. Estas tecnologías no serán tratadas en este curso, pero si queremos advertir de la importancia de que sean usadas de forma que no se obligue al visitante a tener un navegador concreto.

¿Y qué es exactamente XHTML? Pues es una variante de HTML que tiene las mismas etiquetas que HTML 4.0 (en concreto es idéntica a la revisión HTML 4.0.1), pero con una serie de restricciones de sintaxis para ser compatible con XML. Para un navegador no hay diferencias entre usar HTML o XHTML, pero este último puede ser útil para poder emplear herramientas de tratamiento de XML al crear una página web. Las principales normas que debemos cumplir para que una página HTML sea compatible con XHTML (y por tanto compatible con XML) son:

- La primera línea del archivo debe ser:

```
<?xml version="1.0"?>
```

- Todas las etiquetas deben tener instrucción de inicio e instrucción de fin. No está permitido no poner `</p>` al final de un párrafo o `` al final de un elemento de una lista, como es habitual en HTML.
- Las etiquetas de una única instrucción deben terminar con `</>`. Por ejemplo:

```

```

- Todos los valores de los atributos deben ir entrecomillados.
-

El futuro de HTML

El lenguaje HTML seguirá evolucionando y continuará siendo, con toda probabilidad, la base de las páginas web. Conocerlo será imprescindible para poder aprovechar todas sus ventajas. Por ello seguirá usándose directamente para crear páginas web muy cuidadas mientras que en otros casos será ocultado por herramientas que permitan crear páginas al estilo de un procesador de texto. Además de esto seguirán adquiriendo mayor y mayor importancia nuevas tecnologías que añaden nuevas características a las páginas web como dinamismo o interacción con el usuario.

Entre estas tecnologías nos encontramos con los ya habituales lenguajes de Script (JavaScript y VBScript). Estos son lenguajes de programación que se insertan en las páginas web junto con el código HTML y que nos permitirá aumentar la interactividad con el visitante respondiendo a algunas de sus acciones como pulsar un botón, pasar con el ratón por encima de una imagen, etc.

Por supuesto también hemos de mencionar las hojas de estilo que facilitan el mantenimiento de la apariencia en las páginas web además de darnos más potencia.

La suma de HTML con los lenguajes de script y las hojas de estilo ha sido denominado *HTML dinámico*. Desgraciadamente este ha sido el comienzo de una nueva serie de características incompatibles entre navegadores. En particular es habitual encontrar páginas que sólo pueden verse correctamente en el navegador dominante en la actualidad: Internet Explorer™. Esperemos que entre todos los creadores de páginas web podamos presionar

para que se tienda a una mayor compatibilidad entre navegadores. Mientras tanto tendremos que tener cuidado en hacer páginas que puedan verse en todos ellos.

Conclusiones

Bien, ya sabemos todo lo que necesitamos saber sobre la historia del lenguaje HTML, sus estándares y las extensiones de los distintos navegadores. Es probable que el lector se haga dos preguntas en este momento ¿a qué estándar pertenecen las etiquetas vistas hasta ahora? y ¿Cuáles puedo usar libremente?. Todas las etiquetas vistas hasta ahora pertenecen en su mayor parte al estándar HTML 2.0. De hecho siempre que no hemos hecho mención explícita de que cierta etiqueta o atributo era nuevo pertenecían a esta versión. También hemos visto gran parte de las etiquetas del estándar HTML 3.2, en general las más importantes y de mayor uso. En ocasiones hemos dicho explícitamente que esta o aquella etiqueta pertenecían al HTML 3.2, ahora sabemos que sentido tenía remarcarlo.

¿Porqué lo hemos hecho así? Uno de los objetivos que nos planteamos desde el principio fue que nuestras páginas pudiesen ser vistas por todo el mundo, de esta manera podríamos llegar a la mayor cantidad de público posible. Con esta finalidad hemos aprendido a usar etiquetas que pueden ser vistas por casi todo el mundo. Las etiquetas de HTML 2.0 pueden ser vistas por todo el mundo mientras que las etiquetas del lenguaje HTML 3.2 vistas hasta ahora están muy extendidas y causan problemas al ser vistas con navegadores antiguos. En definitiva, todas las etiquetas que ya conocemos pueden ser usadas con gran libertad con la garantía de no estar limitando que nuestras páginas puedan ser vistas únicamente por los usuarios de las últimas versiones de Navigator™ o Explorer™. Si tenemos mayores requerimientos y queremos estar seguros de que *nadie* tendrá problemas al ver nuestras páginas debemos considerar algunos aspectos más.

El W3C ha puesto un servicio público que permite ver si una determinada página cumple los estándares o no. Está disponible en <http://validator.w3.org/>

También hay un programa, llamado Tidy™ que permite hacer la validación en un archivo local, sin tener que publicarlo en Internet. Este programa es software libre y está disponible en <http://tidy.sourceforge.net/>.

¿Qué etiquetas debo usar y cómo?

Podemos asumir que todas las etiquetas del estándar HTML 4.0 son entendidas por cualquier navegador. Eso sí debemos tener en cuenta que el uso de frames o tablas complicadas puede dificultar su interpretación correcta en navegadores de texto o hablados (como los usados por personas ciegas o en coches).

Si optamos por emplear una extensión hay que ser consciente de qué navegadores lo soportan y cómo afectará a los usuarios de el resto. Teniendo esto en cuenta hay que considerar:

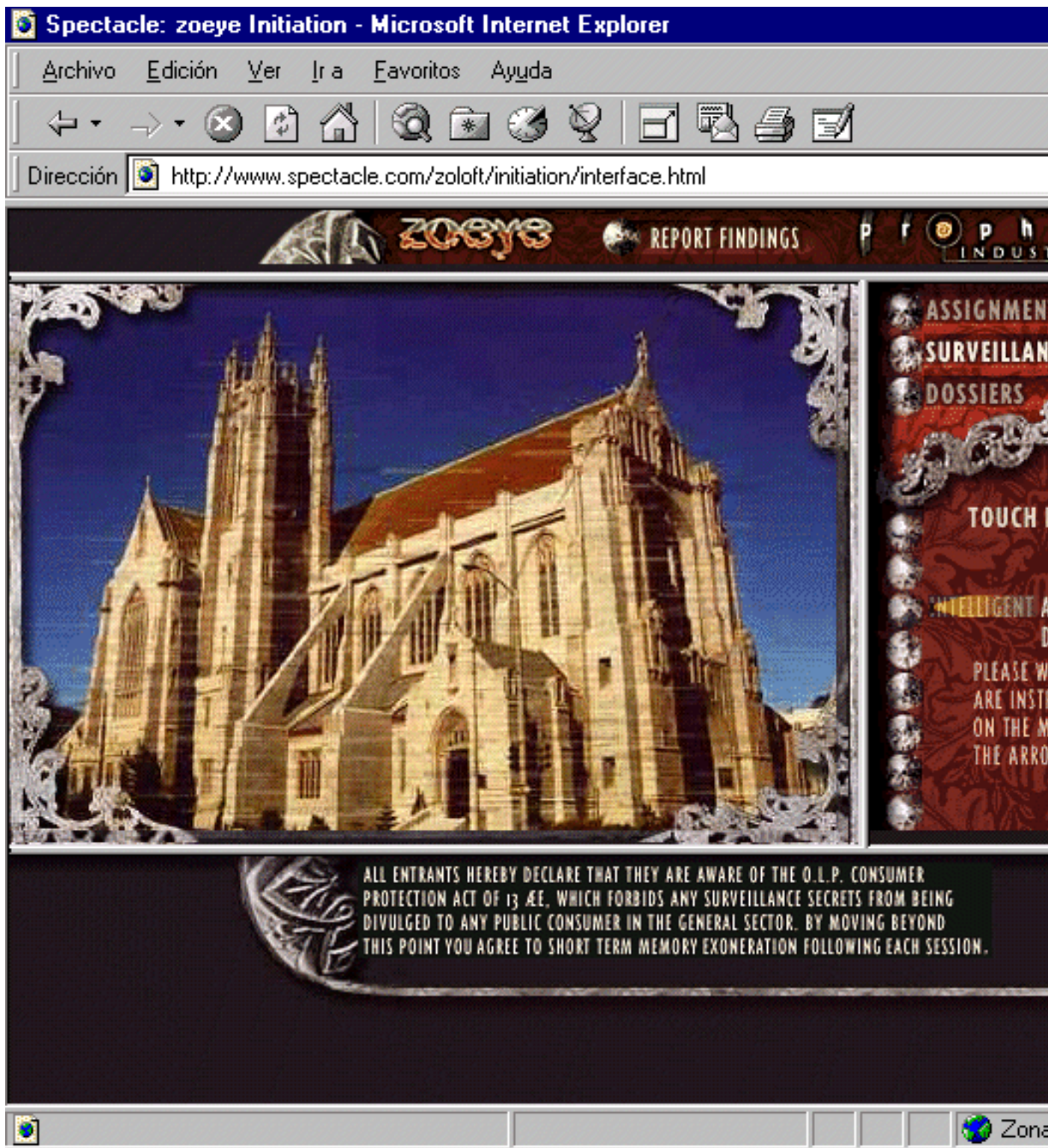
- Si los usuarios a los que va dirigido el web poseen alguno de dichos navegadores.
- Que perjuicios tendrán aquellos usuarios que no lo tengan. Por ejemplo, nunca debemos incluir información vital usando una extensión si existe el riesgo de que cierto tipo de usuarios con navegadores antiguos no puedan acceder a ella. Es decir, la información muy importante debe ser incluida con etiquetas de HTML 2.0 para que todo el mundo pueda verla.

Si usamos con cuidado las extensiones, podemos evitar que causen mucho perjuicio a los usuarios que usen navegadores que no las soporten, y los usuarios que si que tengan los navegadores adecuados saldrán beneficiados. Para conseguir esto es necesario conocer bien estas extensiones y el comportamiento de un navegador cuando se encuentra con alguna etiqueta o atributo que no soporta (que no entiende). Esto último no es difícil de entender: si un navegador se encuentra con una etiqueta que no conoce simplemente la ignora y lo mismo ocurre con los atributos.

¿Qué queda por aprender?

En este mismo capítulo veremos las etiquetas del estándar HTML 3.2 que todavía nos queda por ver a excepción de los formularios que serán vistos próximamente. En los siguientes capítulos aprenderemos a usar las extensiones de Microsoft Explorer™ y Netscape Navigator™ para aprovechar las últimas capacidades de estos navegadores pero sin causar problemas a los usuarios de los navegadores más antiguos o navegadores en modo texto. Según vayamos estudiando estas nuevas etiquetas aprenderemos también a usarlas adecuadamente. Un buen ejemplo de una de las más importantes extensiones del lenguaje HTML, los frames, es la página web de la figura 7.6:

Figura 7.6. En esta figura vemos un excelente uso de los frames o marcos, que dividen la pantalla en cuatro partes para facilitar la navegación.



En ella podemos ver como adelanto al capítulo posterior en el que trataremos esta extensión a fondo, que los

frames sirven para dividir la pantalla en varios marcos con una página diferente en cada una. En este caso se ha dividido la pantalla en cuatro marcos.

En los capítulos que seguirán trataremos las técnicas para incluir contenido dinámico en nuestras páginas, incluido el nuevo lenguaje HTML Dinámico que constituye por sí sólo una nueva revolución en el World Wide Web.

En definitiva nos queda por estudiar las etiquetas más novedosas del lenguaje HTML y otros acompañantes que nos ayudarán en la programación de páginas web y nos permitirán crear efectos realmente impactantes y sorprendentes. Empecemos con ello sin perder más tiempo.

Extensiones a las listas

Las etiquetas y atributos que aprendimos a crear en el capítulo 4 pertenecen al estándar HTML 2.0. Con ellas podíamos crear listas de diferentes tipos y anidarlas para crear interesantes efectos. Sin embargo tras unos años trabajando con estas etiquetas los programadores de HTML empezaron a sugerir algunas ideas para conseguir un mayor control sobre las listas.

El primer paso para satisfacer estas ideas fue la inserción de nuevos atributos en el estándar HTML 3.2 y estos son los que veremos a continuación.

Nota

El siguiente paso se ha dado con la creación de las hojas de estilo. Estas no son más que un método nuevo con el que se nos ofrece a los programadores la posibilidad de especificar las características de cada uno de los elementos de nuestras páginas web. En lo que a listas se refiere podemos indicar que la viñeta de una lista no numerada sea de una manera determinada, que el tamaño de los elementos de la lista vayan decreciendo, etc. O incluso podemos crear diferentes subtipos de listas. Las hojas de estilo son de reciente aparición y por tanto no están bien soportadas por los navegadores aunque ya podemos aprovechar gran parte de sus ventajas y por ello serán adecuadamente cubiertas en un capítulo posterior.

Las novedades que esta nueva versión del estándar introdujo se refieren a las listas ordenadas, OL, y a las listas no ordenadas, UL, que son en definitiva las más usadas. Estas novedades afectan también a la subetiqueta LI.

Listas no ordenadas

Para aumentar el control sobre la apariencia de las listas no numeradas o no ordenadas se introdujo un nuevo atributo: *TYPE*. Con él podremos cambiar el tipo de viñeta de los elementos de la lista. Los tipos de viñetas disponibles son:

- *TYPE="Circle"*: El símbolo es un círculo, es decir, una circunferencia no rellena.
- *TYPE="Square"*: La viñeta es un cuadrado.
- *TYPE="Disk"*: Similar a Circle, pero ahora la circunferencia si está rellena. Este tipo es el que aparecía por defecto cuando no usábamos el atributo *TYPE*.

Quizá al lector que haya practicado con anidamiento de listas le resulten familiares estos tipos de viñetas. Si recordamos al anidar listas el tipo de viñeta de cada nivel de anidamiento era distinto. Pero ahora nosotros tendremos libertad para elegir *cuando* debe usarse cada uno.

Veamos algunos ejemplos. El primero de ellos será una lista de tipo Disk. En él se muestra un listado de algunos de los principales navegadores, y el código usado es:

```
<UL TYPE="circle">

<LI>
Ms Internet Explorer™
<LI>
Netscape Navigator™
<LI>
Opera™
<LI>
Lynx™

</UL>
```

Si queremos hacer esta misma lista pero con viñetas cuadradas tendremos que escribir:

```
<UL TYPE="square">

<LI>
Ms Internet Explorer™
<LI>
Netscape Navigator™
<LI>
Opera™
<LI>
Lynx™

</UL>
```

En la figura 7.7 podemos ver ambos ejemplos junto con un tercero que se obtiene no usando el atributo TYPE o usándolo con el valor TYPE="disc".

Figura 7.7. En HTML 3.2 existen 3 tipos de viñetas para las listas no ordenadas. Con el atributo TYPE podemos elegir entre ellos.



En la figura 7.8 vemos estos tres ejemplos de nuevo, pero esta vez visualizándolos con Netscape Navigator™.

Figura 7.8. Comparando esta figura con la anterior vemos la similitud entre las distintas viñetas de los elementos de las listas usadas por Netscape Navigator™ e Internet

Explorer™.

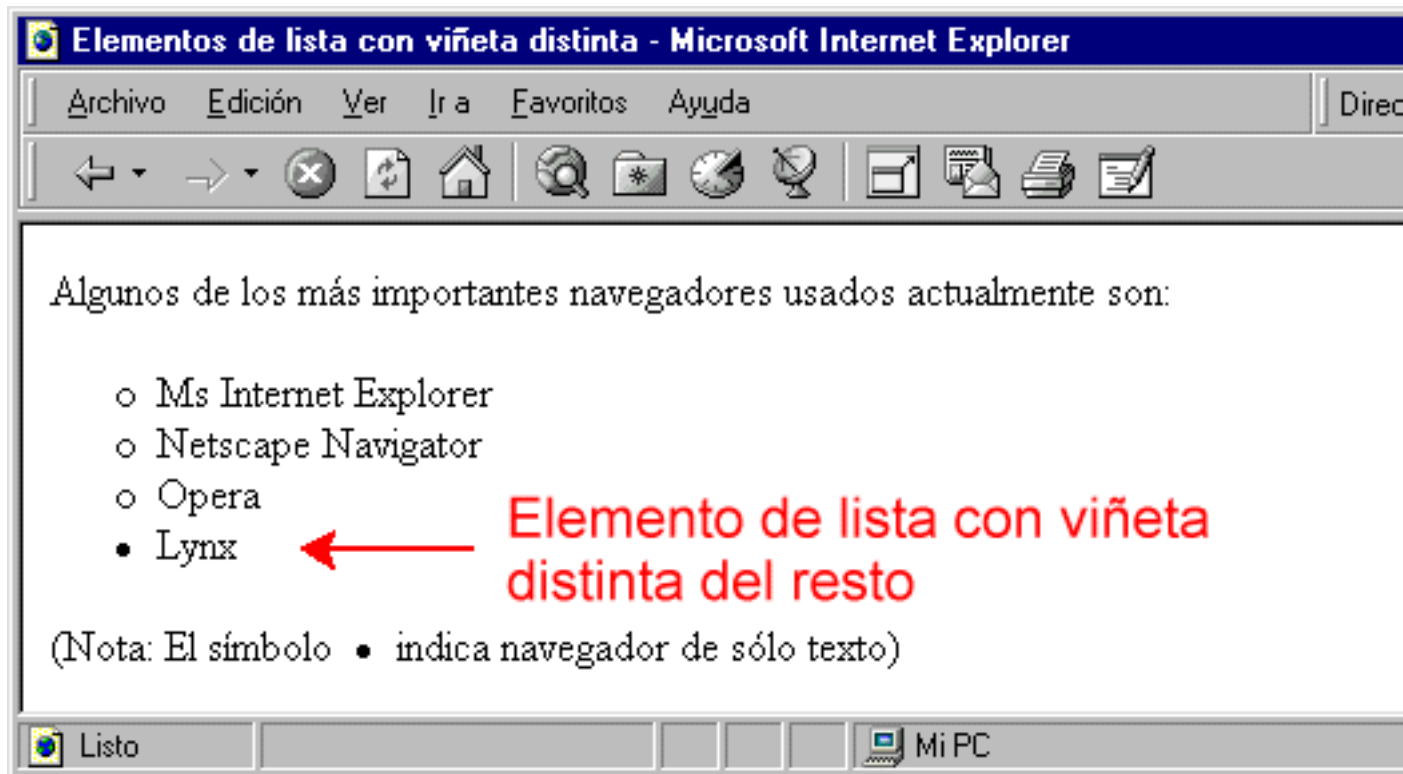
Es importante apreciar que no existen prácticamente diferencias, quizá la única sea la variación de tamaño de los símbolos de algunas viñetas.

Hasta ahora hemos usado el atributo TYPE en la etiqueta que definía el tipo de lista, es decir, en la etiqueta UL. De esta forma el cambio en el tipo de viñeta afectaba a todos los elementos de lista. Sin embargo también puede usarse en la etiqueta LI, y de esta forma conseguimos que el cambio de viñeta sólo afecte a un elemento de lista concreto. La mejor manera de ver este efecto es con un ejemplo y por ello vamos a retomar el primero de los tres vistos antes pero cambiando el tipo de viñeta del navegador Lynx™. Esto podría servirnos en un ejemplo real para indicar que este navegador es un navegador en modo texto, mientras que los demás son navegadores gráficos. El código que debemos usar es:

```
<UL TYPE="circle">
<LI>
Ms Internet Explorer™
<LI>
Netscape Navigator™
<LI>
Opera™
<LI TYPE="disc">
Lynx™
</UL>
```

En la figura 7.9 podemos ver el resultado.

Figura 7.9. Usando el atributo TYPE en una etiqueta LI podemos cambiar la viñeta de un único elemento de lista sin afectar a los demás.



Con esto podemos dar por terminadas las novedades de las listas no ordenadas, pasemos a las ordenadas.

Listas ordenadas

Este tipo de listas tiene alguna novedad más que las anteriores. Para empezar también le ha sido añadido un atributo cuyo nombre es *TYPE*. Pero ahora la función que desempeña es ligeramente distinta, si antes cambiaba el tipo de viñeta ahora cambiará el tipo de numeración. En la tabla 7.1 podemos ver los cinco tipos distintos que hay y el valor que debe tomar el atributo *TYPE* para elegir cada uno:

Figura 7.1. Los cinco tipos distintos que puede tomar el atributo TYPE para listas ordenadas

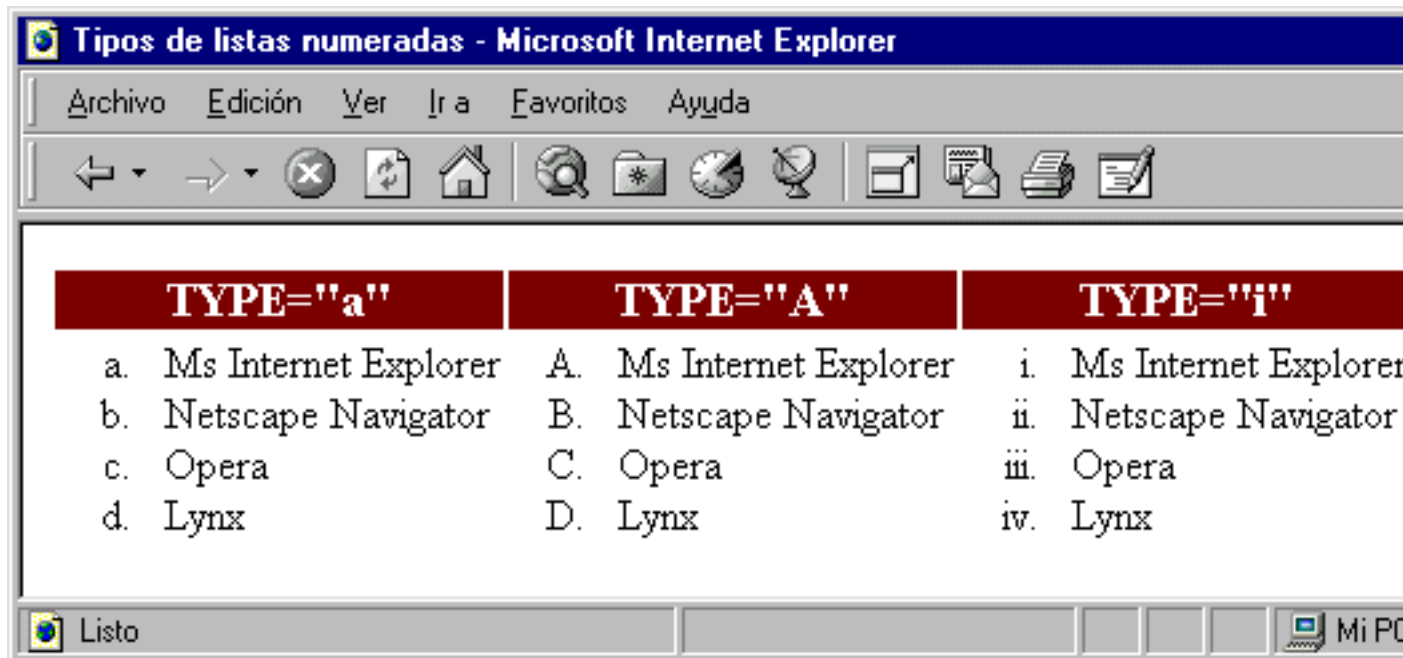
TYPE=	Estilos de numeración	
	Tipos	Ejemplo
1	Números árabes	1, 2, 3, ...
a	Alfabético en minúsculas	a, b, c, ...
A	Alfabético en mayúsculas	A, B, C, ...
i	Romano en minúsculas	i, ii, iii, ...
I	Romano en mayúsculas	I, II, III,

Siguiendo con la lista usada como ejemplo anteriormente, ahora podríamos numerar los elementos de lista con letras escribiendo el siguiente código:

```
<OL TYPE="a">
<LI>
Ms Internet Explorer™
<LI>
Netscape Navigator™
<LI>
Opera™
<LI;>
Lynx™
</OL>
```

Cambiando el valor del atributo TYPE por cada uno de los que veíamos en la tabla 7.1 obtendremos todos los tipos de numeración tal y como apreciamos en la figura 7.10:

Figura 7.10. Con el atributo **TYPE** podemos elegir entre 5 tipos distintos de numeración en las listas de HTML. En esta figura vemos los 4 nuevos.

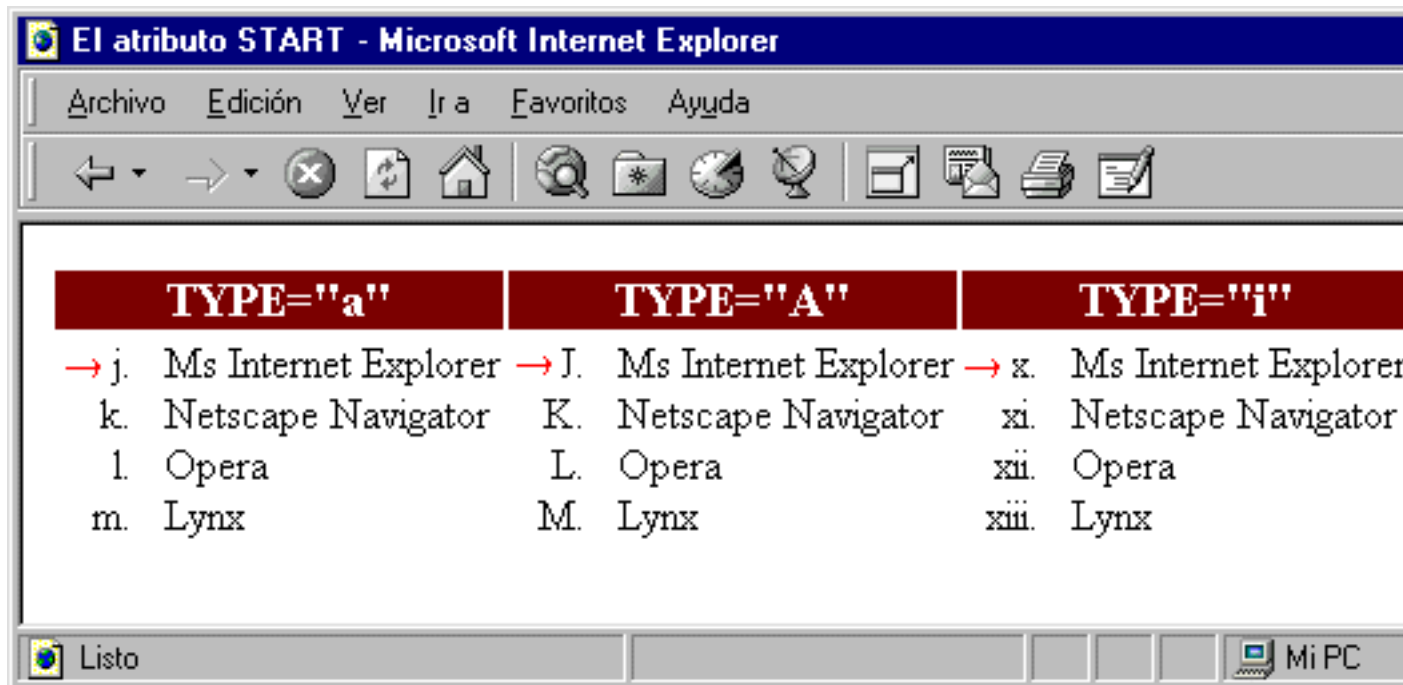


Al igual que en el caso de las listas no ordenadas podemos usar el atributo **TYPE** con cada una de las etiquetas **LI**. Sin embargo ahora el efecto es menos útil que antes, ya que, en general, cambiar el tipo de numeración (OJO, no se vuelve a empezar desde cero) para algunos de los elementos de lista confunde al navegante que vaya a verla.

La siguiente novedad introducida por la versión 3.2 del estándar del lenguaje HTML para las listas ordenadas es el atributo **START**. Si bien las listas que hemos creado hasta ahora comenzaban siempre por 1 (o por a, A, i, I, según el tipo de numeración) con este atributo podemos elegir por qué número (o letra) comenzará el primer elemento de lista. Esto suele ser útil cuando tenemos que partir una lista en dos porque queremos insertar algo en medio y queremos retomar la numeración después. Este atributo tiene como valores posibles todos los números naturales (1, 2, 3, ...)

Si en cada uno de los ejemplos de la figura 7.10 ponemos el atributo **START** con el valor 10 obtendremos el resultado de la figura 7.11:

Figura 7.11. Si usamos el atributo **START** podemos variar el número (o letra en estos casos) por los que se empiezan a numerar los elementos de la lista.



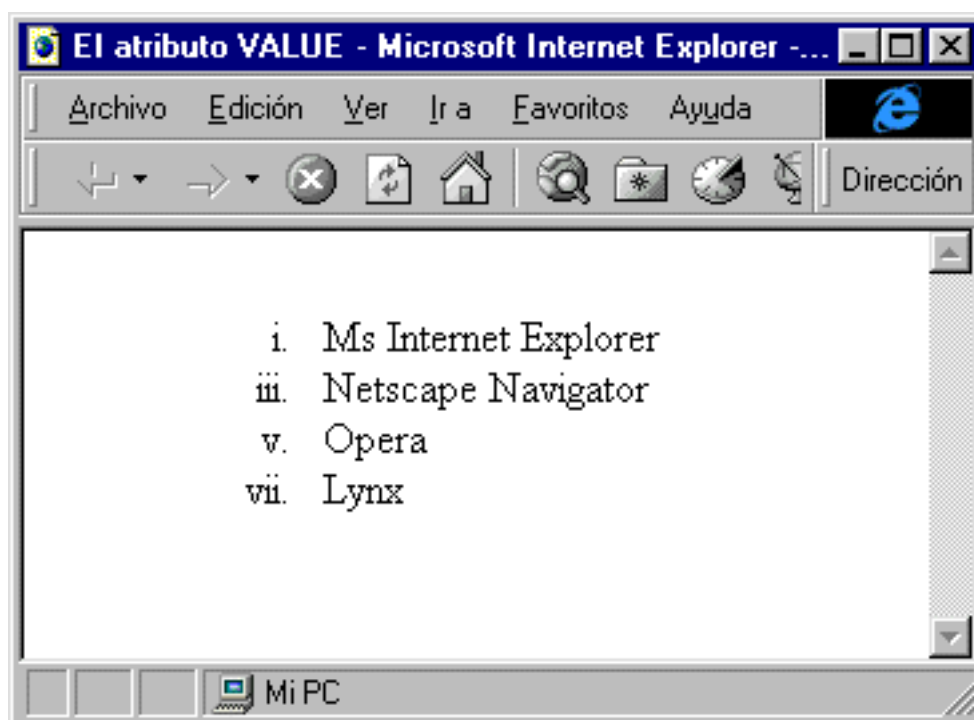
Aunque en esta figura no está el tipo de numeración arábigo (1, 2, 3, ...) los resultados de este tipo de numeración son fácilmente previsibles.

El atributo **START** no puede ser usado de manera independiente en cada elemento de lista, **LI**. Si embargo si existe otro atributo que nos permitirá cambiar el número que precede a cada elemento de lista: **VALUE**. Este atributo funciona de manera muy similar a **START**, toma como valor un número natural y lo convierte al tipo de numeración que usa esa lista. Por ejemplo podríamos crear una lista con numeración romana impar de la siguiente manera:

```
<OL TYPE="i">
<LI VALUE=1>
Ms Internet Explorer™
<LI VALUE=3>
Netscape Navigator™
<LI VALUE=5>
Opera™
<LI VALUE=7;>
Lynx™
</OL>
```

Que nos da el resultado de la figura 7.12:

Figura 7.12. Con el atributo VALUE podemos indicar el valor de la numeración de cada uno de los elementos de las listas numeradas.



Con este atributo ya podemos dar por terminadas las novedades introducidas por HTML 3.2 en la creación de listas. Pero aún queda por comentar un último atributo que existe desde las primeras versiones del lenguaje HTML: *COMPACT*. Este atributo puede ser usado en cualquiera tipo de lista: UL, OL, MENU, DIR y DL, y su función es decirle al navegador que intente comprimir el tamaño que ocupa la lista al máximo, por ejemplo reduciendo el espacio entre cada uno de los elementos de lista. La forma de usar este atributo es simple y no toma ningún valor. Siguiendo con el ejemplo usado hasta ahora podríamos usar el siguiente código:

```
<OL COMPACT;>

<LI>
Ms Internet Explorer™
<LI>
Netscape Navigator™
<LI>
Opera™
<LI>
Lynx™

</OL>
```

El problema de este atributo es que en general los navegadores lo ignoran y muestran la lista exactamente igual que las demás. Si el lector prueba el código de este último ejemplo en Navigator™ o Explorer™ probablemente no notará ninguna diferencia por haber usado COMPACT, aun así hemos considerado recomendable comentar su existencia.

Alineamiento

Este no es un tema nuevo. En gran parte de las etiquetas que hemos visto hasta ahora existía el atributo llamado ALIGN, que aunque variaba en su funcionamiento ligeramente de unas a otras cumplía la función principal de permitirnos elegir el alineamiento de los elementos en la página. Algunos ejemplos de etiquetas donde nos encontramos con este atributo son IMG, P y TABLE. También vimos la etiqueta CENTER. Con ella podíamos centrar cualquier elemento que insertáramos en una página web sin más que situar el código correspondiente entre la instrucción de inicio <CENTER > y la instrucción de fin </CENTER> . De esta forma conseguíamos alinear elementos cuyas etiquetas no tuviesen el atributo ALIGN.

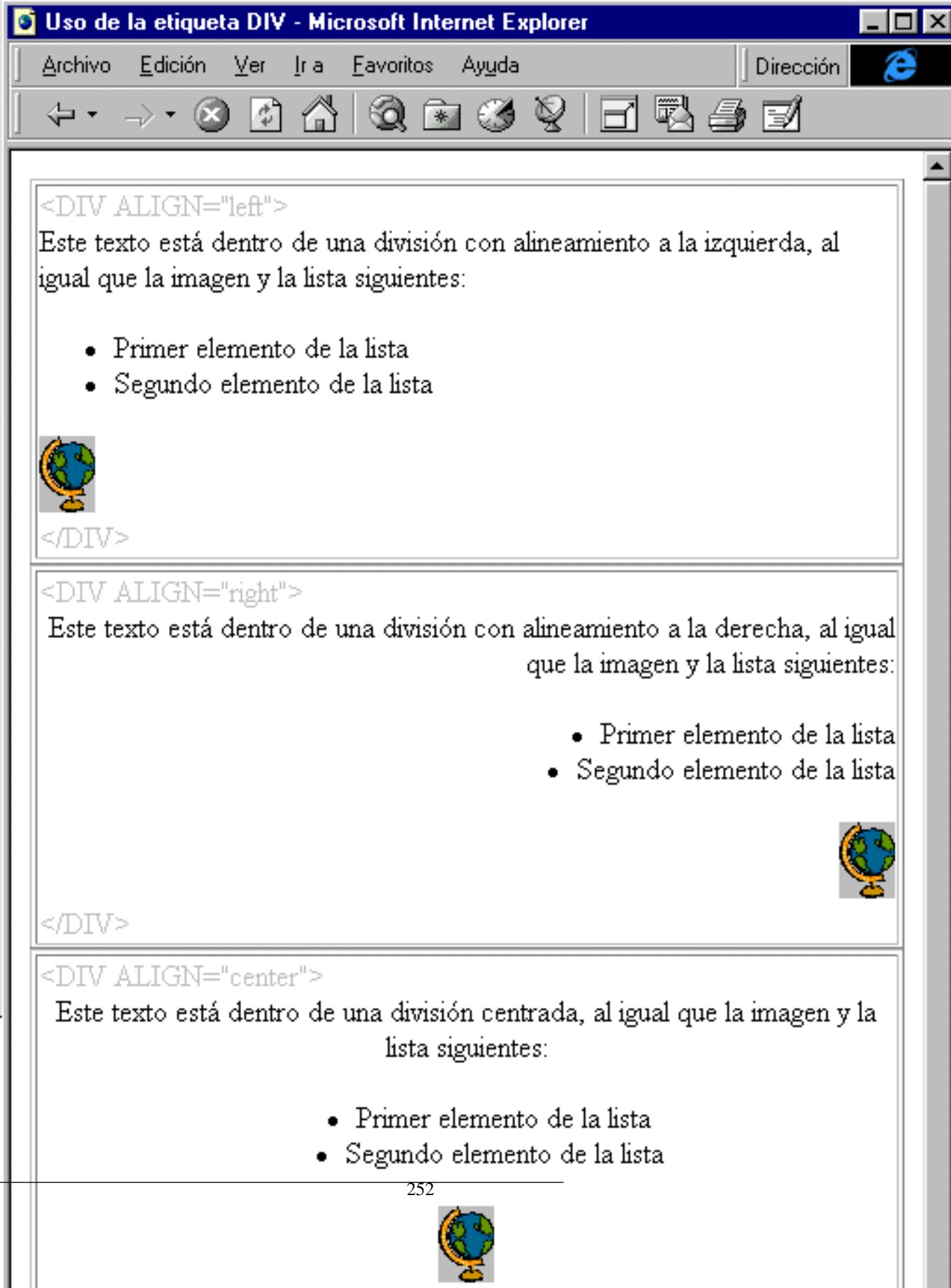
Y...¿hace falta algo más? Quizá lo único que se echa en falta es algo de uniformidad. El atributo ALIGN surgió en algunas etiquetas por la inexistencia de otro medio de alineamiento. La etiqueta CENTER no pertenecía al HTML 2.0 y fue una invención de Netscape TM . A pesar de ello empezó a usarse masivamente y el web Consortium se vio obligado a incluirla en el estándar HTML 3.2.

Pero su enfoque del tema era algo distinto y por ello además de incluir la etiqueta CENTER se creó una nueva: *DIV* . La función de esta etiqueta es realizar divisiones en las páginas de manera que podamos indicar las propiedades de cada una de estas divisiones por separado. Por ahora la única propiedad que podemos controlar es el alineamiento horizontal haciendo uso del conocido atributo ALIGN. Este atributo puede tomar los, ya conocidos, valores: LEFT, CENTER, RIGHT. Es decir, para usar esta etiqueta deberemos escribir el siguiente código:

```
<DIV ALIGN="CENTER">  
... Cualquier etiqueta ...  
</DIV>
```

En este caso hemos usado como ejemplo el valor CENTER, en la figura 7.13 podemos ver un ejemplo con cada uno de los tres valores posibles en los que hemos alineado de una vez un párrafo una lista y una imagen.

Figura 7.13. Con la nueva etiqueta DIV podemos cambiar el alineamiento de todo tipo de elementos HTML (imágenes, listas, párrafos, etc.) de una sola vez.



The screenshot shows a Microsoft Internet Explorer window titled "Uso de la etiqueta DIV - Microsoft Internet Explorer". The browser's address bar is empty, and the menu bar includes "Archivo", "Edición", "Ver", "Ir a", "Favoritos", and "Ayuda". The toolbar contains standard navigation and utility icons. The main content area is divided into three sections, each demonstrating a different alignment for a DIV element:

- Left Alignment:** The first section shows the code `<DIV ALIGN="left">` followed by the text "Este texto está dentro de una división con alineamiento a la izquierda, al igual que la imagen y la lista siguientes:". Below the text is a bulleted list with two items: "Primer elemento de la lista" and "Segundo elemento de la lista". At the bottom left of this section is a small globe icon, followed by the closing tag `</DIV>`.
- Right Alignment:** The second section shows the code `<DIV ALIGN="right">` followed by the text "Este texto está dentro de una división con alineamiento a la derecha, al igual que la imagen y la lista siguientes:". Below the text is a bulleted list with two items: "Primer elemento de la lista" and "Segundo elemento de la lista". At the bottom right of this section is a small globe icon, followed by the closing tag `</DIV>`.
- Center Alignment:** The third section shows the code `<DIV ALIGN="center">` followed by the text "Este texto está dentro de una división centrada, al igual que la imagen y la lista siguientes:". Below the text is a bulleted list with two items: "Primer elemento de la lista" and "Segundo elemento de la lista". At the bottom center of this section is a small globe icon.

At the bottom of the page, the number "252" is centered, with a small globe icon below it.

Nota

En su idea original usar `<DIV ALIGN="center" >` debía ser equivalente a usar `<CENTER >` pero por unas razones o por otras el efecto no es el mismo y por ello la etiqueta DIV no es muy usada.

En el estándar HTML 4.0 la función de la etiqueta DIV sufrirá un cambio fundamental. En este estándar se desaconseja su uso para elegir el tipo de alineamiento, ya que se aconseja que eso se haga con hojas de estilo. La función de DIV será entonces seleccionar el idioma usado u otras características relacionadas con las hojas de estilo que veremos más adelante.

Nota

Internet Explorer™ introdujo como extensión el valor `ALIGN="justify"`. Más tarde ha sido adoptado por Netscape™ y también ha sido incluido en el estándar HTML 4.0 (Cougar). La función de este atributo es justificar los márgenes tanto derecho como izquierdo de la misma manera en que lo están estas columnas de texto de esta publicación.

La etiqueta META

Uno de los mejores métodos para aprender nuevas técnicas en la programación HTML consiste en examinar el código de las páginas que han creado otros. Desde hace ya algún tiempo empezó a proliferar el uso de una etiqueta, de nombre META, que, si bien existe desde los comienzos del lenguaje HTML, era una gran desconocida. Incluso hoy en día muchos programadores de páginas web no conocen el significado de esta etiqueta. A grandes rasgos podríamos decir que la etiqueta META sirve para describir las características de la página web. La forma en que esta información es usada no está siempre determinada, con lo que la mejor forma de aprender a usar la etiqueta META es mediante una serie de ejemplos.

Carga automática de páginas

Esta es, sin duda, la función más útil de todas las que puede desempeñar la etiqueta META y consiste en cargar automáticamente una página desde la actual transcurridos unos segundos sin que el usuario tenga que hacer nada. Para conseguir esto debe incluirse en la cabecera del documento HTML, es decir entre `<HEAD >` y `</HEAD>` la siguiente línea:

```
<META HTTP-EQUIV="Refresh"
CONTENT="5"
URL="http://es.gnome.org">
```

Cuando una página tenga esta línea de código cargará la página `es.gnome.org` [`http://es.gnome.org`] pasados 5 segundos de la carga de dicha página. Por supuesto el número de segundos y la dirección a la que queremos que se salte se pueden cambiar. Ambos datos deben estar entre comillas y separados por punto y coma, tal y como se muestra en el ejemplo anterior.

Información para los buscadores

Todos los navegantes del WWW hemos usado en alguna ocasión alguno de los buscadores como Yahoo (figura 7.14), Altavista, Ole, Ozú, o algún otro.

Figura 7.14. Los buscadores permiten a los navegantes encontrar la información que buscan. Usando la etiqueta META podemos proporcionarles datos para que incluyan nuestras páginas en sus bases de datos.



Estos buscadores tienen una serie de programas llamados arañas, webbots o bots que navegan por la red reclutando páginas para incluir en su buscador. Si incluimos en nuestra página cierta información es muy probable que incluya nuestra página en la base de datos del buscador. Entre la información que puede necesitar el webbot está el autor de la página, una breve descripción de la misma, palabras clave que la definan, etc. Para insertar estos datos haremos uso de los atributos NAME y CONTENT de la etiqueta META. Por ejemplo en la página de Los Alpes será conveniente añadir:

```
<META NAME="author"  
CONTENT="WebMaster de Los Alpes">
```

```
<META NAME="keywords"  
CONTENT="viajes, ofertas, Los Alpes, Orlando, Nueva  
Zelanda">
```

```
<META NAME="description"  
CONTENT="La agencia de viajes Los Alpes tiene un  
gran prestigio en el mundo de las empresas y  
combina una gran calidad con unos excelentes  
precios.">
```

Otros usos

La etiqueta META tiene tres atributos HTTP-EQUIV, NAME y CONTENT. Normalmente se pone una de las dos primeras para indicar el nombre de la información que vamos a incluir y siempre se usa CONTENT para insertar dicha información. El atributo HTTP-EQUIV sirve para insertar información relacionada con el protocolo HTTP. Este es el lenguaje con el que habla el ordenador donde hay un navegador para pedirle páginas a un servidor web y con el que responde este al servir (mandar) dicha página. Un ejemplo de su uso es el que veíamos al principio de solicitar otra página transcurridos unos segundos. Otro posible ejemplo sería indicar una fecha a partir de la cual la página no debe ser servida.

El atributo NAME tiene un significado mucho más general y puede ser usado con multitud de propósitos. Ya hemos visto que puede servir para mandar información a los buscadores y también suelen ser usados por los editores HTML que generan el código HTML automáticamente para incluir información específica.

Otras etiquetas del lenguaje HTML 3.2

Ya casi podemos dar por concluido el aprendizaje del lenguaje HTML 3.2. Conocemos prácticamente todas las etiquetas y todos los atributos que existen en este estándar. En este apartado vamos a explicar lo poco que nos quedan para dar por terminado del todo nuestro aprendizaje de esta versión del estándar. Las etiquetas que vamos a comentar a continuación no tienen demasiada relación entre sí, y no han sido explicadas hasta ahora por que son raramente usadas, porque muchos navegadores no las entienden o porque necesitamos conocer algunas cosas más para entender su funcionamiento.

Descriptor de tipo de documento

Pero antes de empezar vamos a introducir un nuevo concepto que todavía no habíamos mencionado: el *descriptor de tipo de documento*. Aunque suene extraño, no lo es, nos referimos a una declaración que según el estándar todas las páginas web deben tener para ser consideradas como HTML. Esta declaración ocupa una sola línea e incluirá también información sobre la versión de HTML que hemos usado. No es necesario incluir este descriptor en nuestras páginas para que los navegadores la interpreten como HTML, sin embargo incluirlo no causa demasiado trabajo, y estaremos cumpliendo con el estándar. Algunos de estos descriptores más comúnmente usados son:

```
<!DOCTYPE HTML  
PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
```

Todos empiezan por la cadena `<!DOCTYPE ...`, el resto no es necesario entender que significa a excepción de la cadena HTML 3.2 del primero que indica que hemos usado etiquetas del ese estándar y HTML 2.0 del segundo que indica lo mismo esta vez con la versión 2.0 del estándar. El descriptor de tipo de documento debe ser la primera línea de la página, con lo que ahora la estructura básica será:

```
<!DOCTYPE HTML  
PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">  
<HTML>  
<HEAD>...</HEAD>  
<BODY>  
... Código ...  
</BODY>  
</HTML>
```

Otras etiquetas

Existen unas pocas etiquetas del lenguaje HTML 3.2 que no hemos comentado todavía por diversos motivos. Hemos querido enumerarlas y dar una breve descripción para que el lector pueda dar por concluido el aprendizaje de la versión estándar de este lenguaje. En los siguientes capítulo abordará a fondo las extensiones de Netscape™ y Microsoft™.

Las etiquetas de HTML 3.2 que todavía no hemos visto son:

- **APPLET**: Esta es la etiqueta que se usa para insertar applets de Java y será estudiada más adelante.
- **STYLE**: Esta etiqueta está compuesta por una instrucción de inicio, `<STYLE>` y otra de fin `</STYLE>` y debe ser usado en la cabecera de la página. En su interior se introduce el código de las hojas de estilo que trataremos más a fondo en los últimos capítulos dedicados al lenguaje HTML.
- **SCRIPT**: Aunque el estándar HTML 3.2 no soporta los lenguajes de script si admite la existencia de esta etiqueta admitiendo que en un futuro será usada con este fin. Las extensiones de Netscape™ y Explorer™ la usan para insertar JavaScript y VBScript. Estos lenguajes son muy importantes en la programación Internet y por tanto también están incluidos en esta obra.
- **FORM**: Con la etiqueta FORM y una serie de subetiquetas podremos crear formularios interactivos mediante los cuales un usuario podrá mandarnos datos. La creación de estos formularios no se limita a la programación HTML, sino que también hay que tratar de alguna forma la información que se recibe. Por esta razón hemos preferido explicar los formularios un poco más adelante.
- **ISINDEX**: Este es un método antiguo para crear formularios interactivos. Su uso está desaconsejado y por tanto sólo comentaremos su existencia.

Ahora si, ya podemos dar por terminado el aprendizaje del lenguaje HTML estándar. A partir de ahora aprenderemos a crear los efectos más atractivos que podemos encontrar hoy en día en el World Wide web.

Frames, las ventanas llegan al WWW

Jorge Ferrer

Víctor García

Versión 1.0

Copyright Jorge Ferrer y Víctor García. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation. Puede consultar una copia de la licencia en: <http://www.gnu.org/copyleft/fdl.html>

Una de las extensiones usadas más habitualmente en el WWW son los *marcos* o *frames*. A algunos programadores les horroriza la idea de que sus páginas tengan frames, otros no entienden cómo han podido vivir sin ellos hasta ahora. Pero, lo que está claro es que existen páginas web donde se ha demostrado que al menos en ciertos casos un buen uso de frames mejora mucho la experiencia del navegante.

En este capítulo se explica cómo usarlos y usarlos bien. Entre las características avanzadas se verá cómo anidar framesets para crear estructuras complejas, cómo manejar y ocultar los bordes y cómo decidir en qué frame debe mostrarse el resultado de un enlace. Después pasaremos a explicar la quinta práctica del curso, en la que añadiremos frames a la página web de Los Alpes. Como guinda, aprenderemos a usar frames flotantes, un recurso que forma parte del estándar HTML 4.0 pero que no es muy usado aún.

Los frames consisten en una técnica para insertar varios documentos HTML en una sola ventana del navegador. Para ello divide la pantalla en varias zonas o secciones que reciben el nombre de *frames* o *marcos*. En cada uno de estos marcos debemos insertar un contenido, que en general constará de un documento HTML, aunque también puede ser una imagen o algún otro elemento del WWW visualizable por los navegadores. En la figura 8.1 podemos ver un ejemplo de Explorer [™] dividido en 4 frames:

Figura 8.1. Los frames nos permiten dividir la ventana del navegador en varias secciones diferentes. En este caso la ventana a sido dividida en 4 frames.



Una de las ventajas de los frames es que cada uno de los documentos HTML puede ser cambiado sin que ello afecte al resto. Ya en el capítulo anterior (figuras 9.1 y 9.6) vimos algunos ejemplos de uso de frames que nos pueden ayudar a hacernos una idea de su funcionamiento y de los que seremos capaces de hacer tras este capítulo.

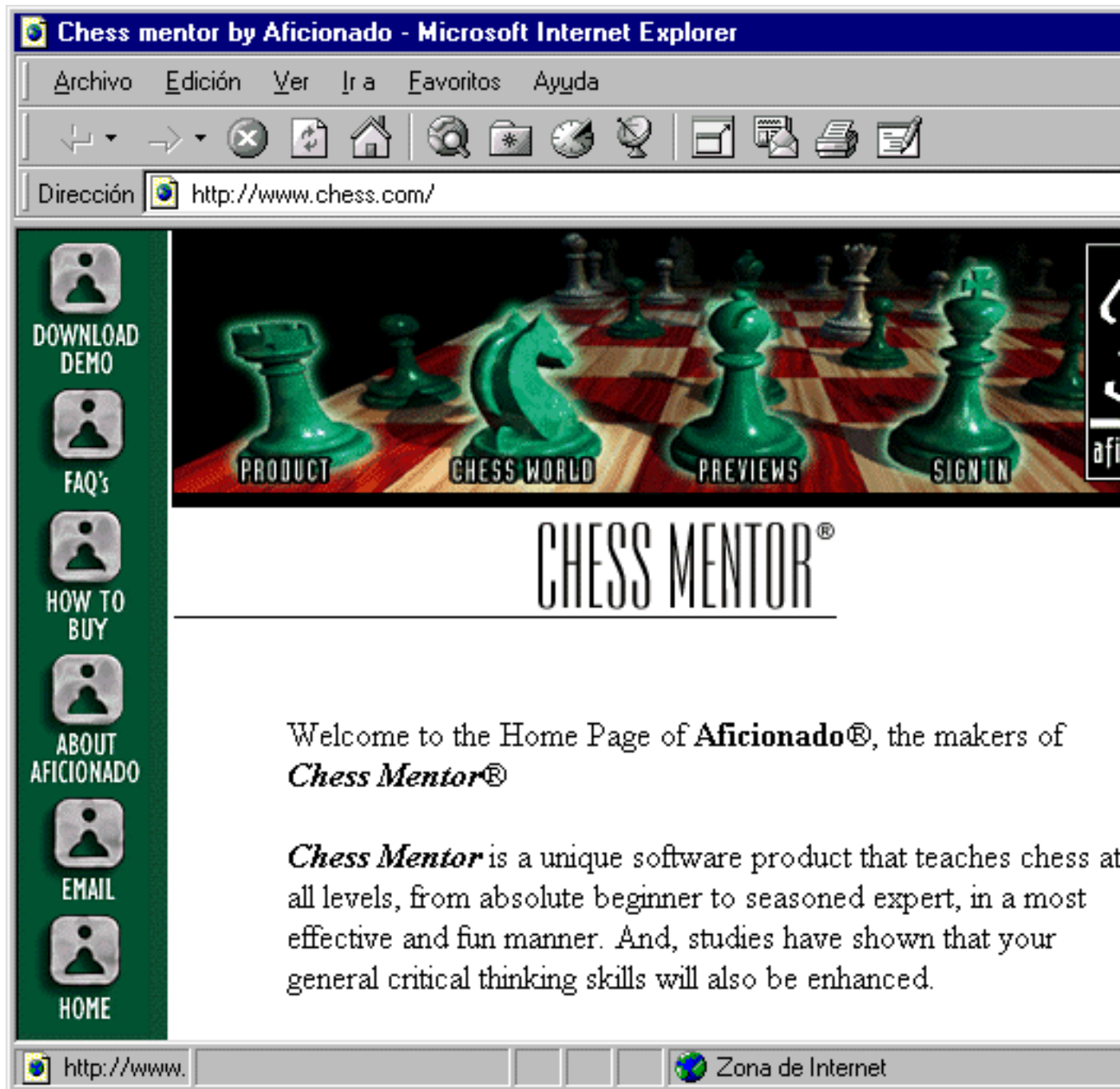
Además de sus ventajas los frames también tienen algunos inconvenientes. Por ello desde su aparición en la versión 1.0 del navegador Netscape Navigator™ han sido el foco de una gran controversia. Uno de los principales problemas de los frames es el hecho de ser una extensión del lenguaje HTML y no formar parte del estándar. Por esta razón no todos los navegadores están preparados para ellos y no podrán visualizar páginas que contengan frames. Pero afortunadamente, hoy en día la mayor parte de los navegadores gráficos los soportan. Entre estos se incluyen Netscape Navigator 1.0™ o superior (por supuesto), Microsoft Internet Explorer 3.0™ o superior y Opera 3.0™. Además existen métodos para solucionar el problema de los navegadores no compatibles.

Nota

Actualmente todos los navegadores gráficos habitualmente usados desde ordenadores personales soportan el uso de frames. Sin embargo en navegadores para dispositivos móviles, navegadores textuales y navegadores que leen la página el uso de frames puede ser perjudicial.

El segundo gran foco de la controversia es mucho menos técnico y radica en la discusión de si los frames son útiles o no. Probablemente el origen de esta discusión surgió con el BOOM sucedido cuando aparecieron los frames. Debido a lo novedoso del tema, una gran parte de los programadores de páginas Web empezaron a usar frames indiscriminadamente, con lo que el WWW se pobló de ellos y en la mayoría de los casos estos sólo eran usados porque eran algo nuevo, pero perjudicaban la navegación. Por esta razón empezaron a surgir detractores de los frames. Pero, ahora, pasado un tiempo ya no existen tantos sitios Webs que usan frames de manera abusiva, con lo que aquellos que discutían empiezan a llegar a un acuerdo. Lo que casi todo el mundo parece admitir es que en ciertas ocasiones, como la que nos encontramos en el Web de la figura 8.2, el uso de frames puede ser conveniente y ayuda a la navegación, que es en definitiva el motivo por el que surgieron en primer lugar.

Figura 8.2. En el Web www.chess.com [<http://www.chess.com>] podemos ver un excelente ejemplo del uso de frames para mantener visibles las barras de navegación.



En este Web existen tres frames. A la izquierda se ha usado uno como barra de navegación donde se encuentran enlaces a las distintas secciones. En la parte superior nos encontramos con un segundo frame que contiene el encabezado gráfico de la página con el logo de los *Aficionado* así como otros links. El tercero de los frames es el que ocupa una mayor área, y en el se encuentra el contenido de la página Web. El objetivo de los frames en este caso consiste en mantener siempre visible el encabezado y barra de navegación mientras el visitante se mueve por el contenido principal usando las barras de desplazamiento, las cuales, como podemos observar, sólo afectan al

tercero de los frames. En definitiva los frames son, sin duda, útiles para mantener visibles barras de navegación (gráficas o simplemente enlaces hipertexto) o encabezados y/o logotipos muy importantes aunque el navegante se desplace por el contenido de la página o navegue por el sitio Web.

Sin perder más tiempo vamos a empezar ya a estudiar las etiquetas y atributos necesarios para crear los frames. Para empezar vamos a ver un atributo, TARGET, y una etiqueta, BASE, que están enormemente relacionados con la creación de páginas con frames. Posteriormente nos adentraremos en las etiquetas necesarias para crear la estructura de frames en las páginas.

El atributo TARGET

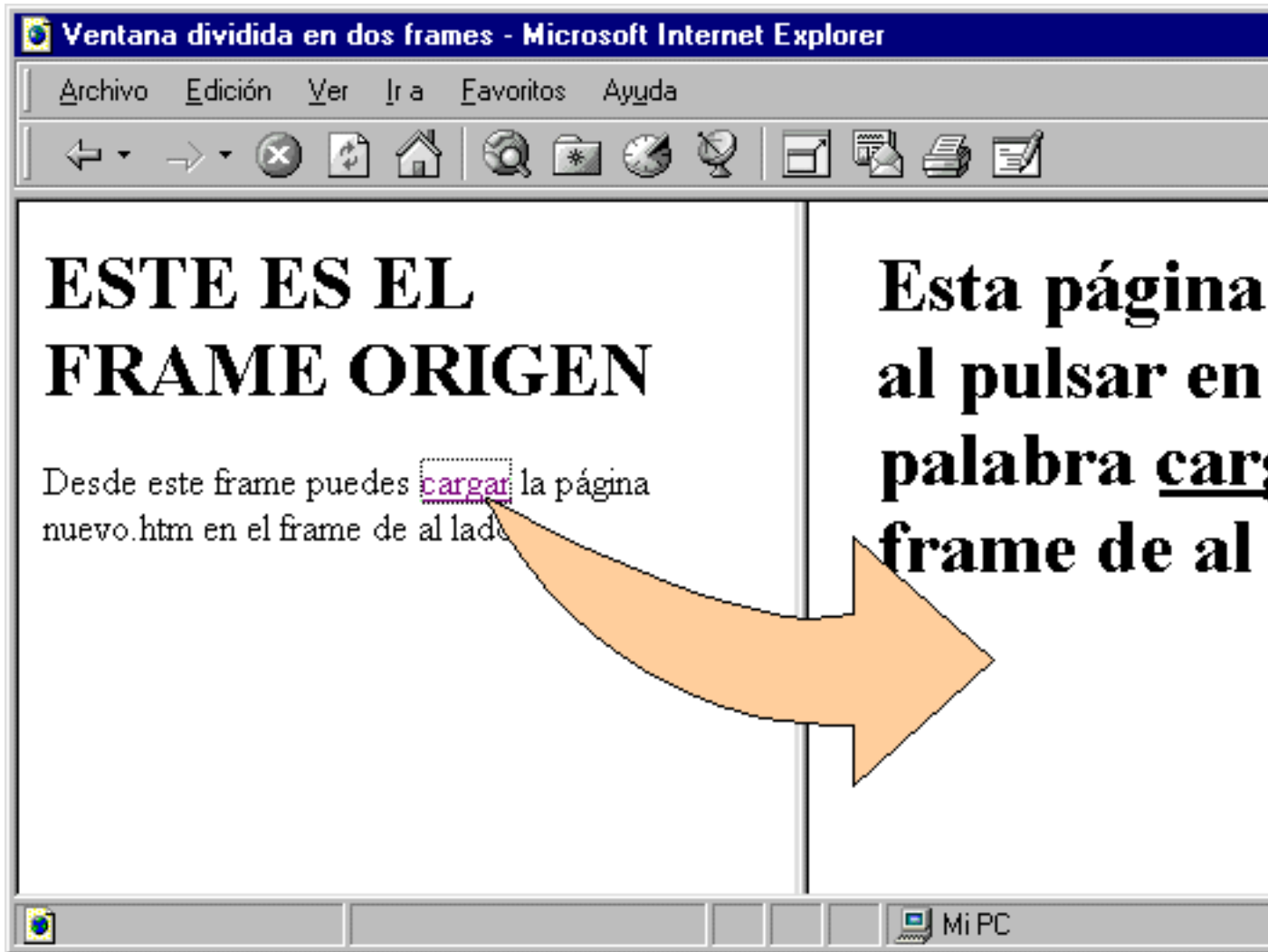
El atributo TARGET es, al igual que los frames, una extensión introducida por Netscape™ y adoptada posteriormente por Explorer™ y Opera™. El principal uso de este atributo es cuando se usa en la etiqueta A para crear enlaces y su función es indicar el objetivo (TARGET significa diana u objetivo) en el que se debe cargar la página a la que saltamos tras seleccionar el texto o imagen de ese enlace. La forma de usar este atributo es:

```
<A HREF="URL" TARGET="nombre_ventana">
```

```
...  
</A>
```

Aunque este atributo fue creado para los frames y es en ellos donde adquiere su verdadera utilidad también puede ser usado aunque estos no existan. Cuando el navegante selecciona un enlace que usa el atributo TARGET busca una ventana o frame (posteriormente veremos como se crean y cómo se les da nombre) cuyo nombre coincida con el que hemos indicado como valor de TARGET. Si existe, muestra la página a la que nos conduce el enlace en esa ventana o frame, pero si no existe abre una nueva ventana del navegador y la da el nombre indicado. En la figura 8.3 vemos una página con dos frames:

Figura 8.3. Usando el atributo TARGET podremos lograr que pulsando el enlace de un frame el resultado se muestre en otro.



En la de la derecha hemos incluido un enlace usando como destino (TARGET) el frame de al lado. La flecha nos indica que al pulsar sobre el enlace aparece una nueva página en el frame de al lado.

Posteriormente tendremos tiempo para practicar con frames, ahora vamos a ver un ejemplo del uso de este atributo en una página sin ellos, como las que hemos visto hasta ahora. Abrimos nuestro editor de texto o HTML favorito y creamos un archivo nuevo al que llamaremos `madre.htm` con este código:

```
<!DOCTYPE HTML
PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>
Ventana Madre
</TITLE>
</HEAD>

<BODY BGCOLOR="#EDB301">
<H1>
Esta es la ventana Madre que generará el resto
</H1>
<P>

<A HREF="ventana1.htm"
TARGET="primera_ventana">
Abrir
</A>
una nueva ventana llamada primera_ventana
<p>

<A HREF="ventana2.htm"
TARGET="segunda_ventana">
Abrir
</A>
una nueva ventana llamada segunda_ventana
<p>

<A HREF="ventana3.htm"
TARGET="primera_ventana">
Poner
</A>
contenido <STRONG>nuevo
</STRONG> en la primera_ventana
</BODY>
</HTML>
```

Posteriormente debemos crear los archivos `ventana1.htm`, `ventana2.htm` y `ventana3.htm`, que tendrán un código muy simple en el que simplemente debemos poner un texto que diga que ventana es esa. Por ejemplo en el caso de `ventana1.htm` podemos escribir el siguiente código.

```
<!DOCTYPE HTML
PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">

<HTML>

<HEAD>
<TITLE>
Página 1
</TITLE>
</HEAD>

<BODY>
<H1>
Esta es la página 1
</H1>
```

```
Esta página debe mostrarse en la ventana
<STRONG>primera_ventana
</STRONG>
</BODY>

</HTML>
```

Creamos el resto de archivos con un código similar a este y abrimos el archivo `madre.htm` con el navegador que preferamos. A continuación seguimos los siguientes pasos:

1. Pulsamos sobre el primero de los enlaces. Observaremos que se crea una nueva ventana con el contenido de `pagina1.htm`.
2. Pulsamos sobre el segundo enlace. Se vuelve a crear una nueva ventana, esta vez con el contenido de `pagina2.htm`.
3. Pulsamos sobre la palabra 'Poner' para activar el tercero de los enlaces. En este caso el navegador busca una ventana de nombre *primera_ventana* y la encuentra, porque fue creada en el paso 1. Una vez encontrada muestra en ella el contenido de `pagina3.htm` quitando el contenido de `pagina1.htm`.

Tras llevar a cabo estos pasos deberíamos obtener el resultado de la figura 8.4, en la que vemos la ventana madre y las otras dos abiertas:

Figura 8.4. Este debe ser el resultado final tras llevar a cabo todos los pasos en el ejemplo de uso del atributo TARGET para crear varias ventanas.



Hemos puesto un color de fondo a `ventana3.htm` para que resalte más.

La etiqueta BASE

Cuando en el capítulo 2 aprendimos a crear enlaces vimos que había dos formas de especificar la dirección de destino. La primera era poniendo la dirección relativa. Por ejemplo, desde un archivo llamado `index.htm`

situado en `freek.futurnet.es` [`http://freek.futurnet.es`] podríamos enlazar una página llamada `pag.htm` y situada en el directorio *varios* usando el código:

```
<A HREF="varios/pag.htm">
...
</A>
```

Una segunda forma para enlazar esta misma página era poniendo la dirección completa (URL) de Internet, siguiendo con el ejemplo anterior debemos usar el siguiente código:

```
<A HREF="http://freek.futurnet.es/varios/pag.htm">
...
</A>
```

Comparando los dos ejemplos anteriores podemos deducir que cuando ponemos una dirección relativa el navegador le añade la dirección completa (exceptuando el nombre del archivo) de la página actual. Es decir, al usar en este caso la dirección relativa, el navegador ve que la página actual está en `freek.futurnet.es` [`http://freek.futurnet.es`] y se lo añade a la dirección relativa para obtener una dirección absoluta. Esta dirección que es añadida a las direcciones relativas para obtener la dirección completa recibe el nombre de *dirección base*, y en todos los casos que hemos visto hasta ahora la dirección base coincidía con la dirección de la página actual.

Pues bien, la etiqueta *BASE* nos permitirá cambiar esa dirección base, y ahora podremos elegir cualquier dirección del WWW como base para todos los enlaces con direcciones relativas de la página. Esta etiqueta debe ser usada en la cabecera del documento, entre `<HEAD>` y `</HEAD>`, dado que afectará a todo este. Para especificar la dirección se usará un atributo llamado *HREF*. Veamos un ejemplo, si queremos poner como dirección base de un documento HTML la dirección `freek.jorgeferrer.com` [`http://freek.jorgeferrer.com`] debemos escribir:

```
<HEAD>
...Más código...

<BASE HREF="http://freek.jorgeferrer.com/">

</HEAD>
```

O bien:

```
<HEAD>
...Más código...
<
BASE HREF="http://freek.jorgeferrer.com/index.htm"
>
```

</HEAD>

En la figura 8.5 podemos ver la diferencia entre usar y no usar BASE y como afecta ello al significado de una dirección relativa.

Figura 8.5. Variando la dirección de base con la etiqueta BASE variará el significado de las direcciones relativas.

Página actual: <http://www.towercom.es/index.htm>

- **No usando BASE**
`... `
Dirección completa del enlace:
www.towercom.es + varios/pag.htm=
<http://www.towercom.es/varios/pag.htm>
- **Usando <BASE HREF="http://www.otro.es">**
`... `
Dirección completa del enlace
:www.otro.es + varios/pag.htm=
<http://www.otro.es/varios/pag.htm>

Aviso

Las direcciones que pongamos como valor del atributo HREF de la etiqueta BASE deben ser siempre absolutas (completas).

Es importante resaltar que aunque en todos los ejemplos que hemos puesto hasta ahora sólo hemos considerado los enlaces hipertexto la dirección base afecta a todas aquellas etiquetas en las que debemos especificar una dirección y podemos hacerlo de forma relativa, como por ejemplo en las imágenes. Por ejemplo si usamos BASE como hemos indicado en el código anterior y posteriormente insertamos una imagen con el código:

```
<IMG SRC="imagen.gif">
```

Esta imagen debe estar en freek.jorgeferrer.com/imagen.gif [http://freek.jorgeferrer.com/imagen.gif] y no en el mismo directorio que la página actual como sucedía hasta ahora cuando no usábamos la etiqueta BASE.

Nota

La etiqueta BASE existe desde la primera versión del estándar HTML: el HTML 2.0. Sin embargo hemos preferido no comentarla hasta ahora porque, como enseguida veremos, es con los frames donde realmente su uso es importante, casi imprescindible.

La pregunta que quizá nos hagamos ahora es ¿para qué puede servirnos cambiar la dirección base?. Puede haber diversas causas, por ejemplo existe la posibilidad de que tengamos nuestra página principal en un servidor, pero el resto de las páginas y enlaces en otro o simplemente que estén todas en otro directorio. En ambos casos podemos ahorrarnos poner el otro servidor o directorio especificándolo como dirección base.

Sin duda esta forma de usar la etiqueta BASE puede llegar a ser útil, pero cuando se convierte en imprescindible es al usar un segundo atributo: *TARGET*. Hemos visto un poco antes que cuando trabajamos con páginas con varios frames se puede incluir el atributo TARGET en los enlaces para indicar el frame en el que debe mostrarse la página enlazada. Pues bien usando este mismo atributo en la etiqueta BASE lo que conseguimos es establecer el frame base donde se mostrará la página enlazada cuando pulsemos uno de los enlaces del frame actual. ¿Porqué es esto tan útil? Lo entenderemos un poco mejor en cuanto empecemos a trabajar con frames, pero podemos adelantar un caso en el que su utilidad resulta obvia. En la figura 8.6 vemos un ejemplo simple pero que ilustra lo que queremos a la perfección:

Figura 8.6. En este Web pulsando sobre los enlaces del frame izquierdo nos movemos por los diferentes temas del frame derecho.



La página contiene una serie de enlaces organizados por temas y está dividida en dos frames. En el frame de la izquierda hay un listado de los diferentes temas y en el de la derecha los enlaces en sí. Pulsando sobre los temas del frame izquierdo se muestran los enlaces de ese tema en el frame derecho. En este caso es claro que todas las páginas a las que apuntan los enlaces deben ser mostradas en el frame derecho y por tanto lo más cómodo (y así es como está hecho en realidad) es usar (suponiendo que el nombre del frame derecho es *derecho*):

```
<BASE TARGET="derecho">
```

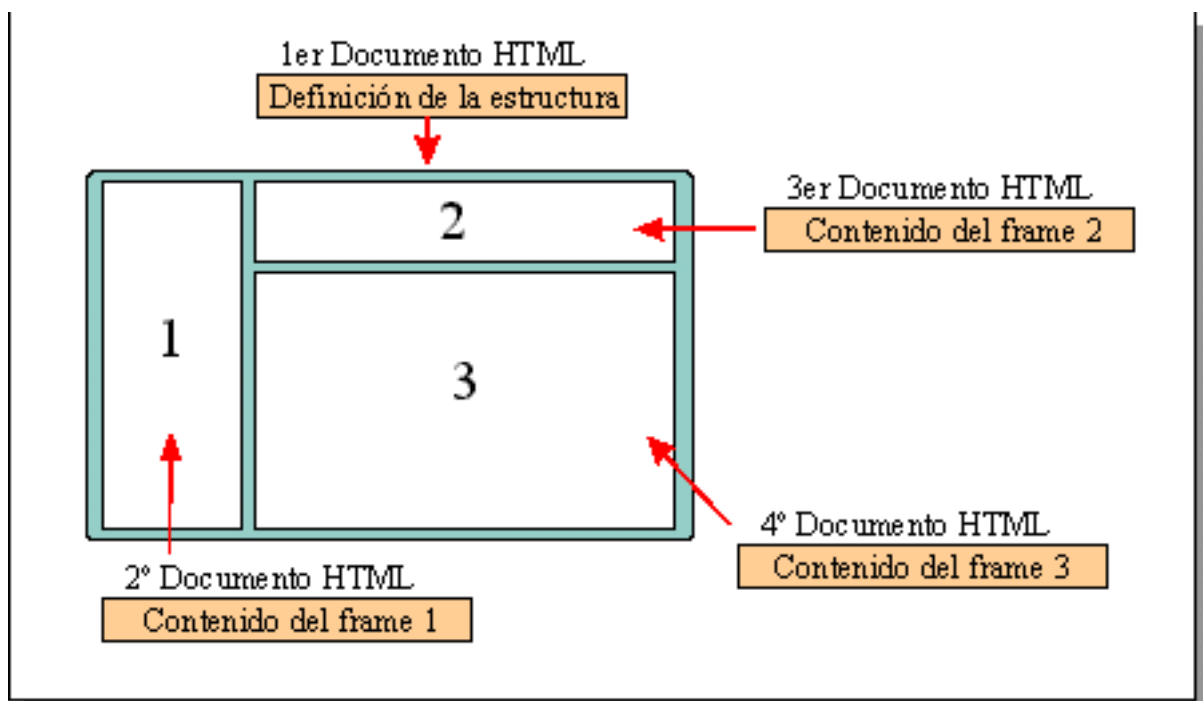
En el código del frame de la izquierda, con lo que nos ahorramos usar el atributo TARGET en cada uno de sus enlaces.

Ya hemos aprendido todo lo necesario para comenzar con la construcción de páginas con frames, así que sin perder más tiempo pasamos ya a estudiar las etiquetas del lenguaje HTML que debemos usar.

Creación de páginas con frames

El hecho de usar frames no sólo cambia por completo la apariencia de nuestra página, sino que también cambia el proceso de su realización. Si hasta ahora los Webs estaban formados por un único documento HTML donde estaba el contenido, al usar frames será necesario un documento HTML para definir la estructura de la página (número de frames a usar y disposición de estos), y posteriormente tantos documentos HTML extra como frames tengamos para insertar su contenido. En la figura 8.7 vemos un ejemplo con una página Web compuesta por 3 frames:

Figura 8.7. Los Webs con frames constarán de varios documentos HTML. En este ejemplo al haber tres frames serán necesarios 4 documentos HTML.

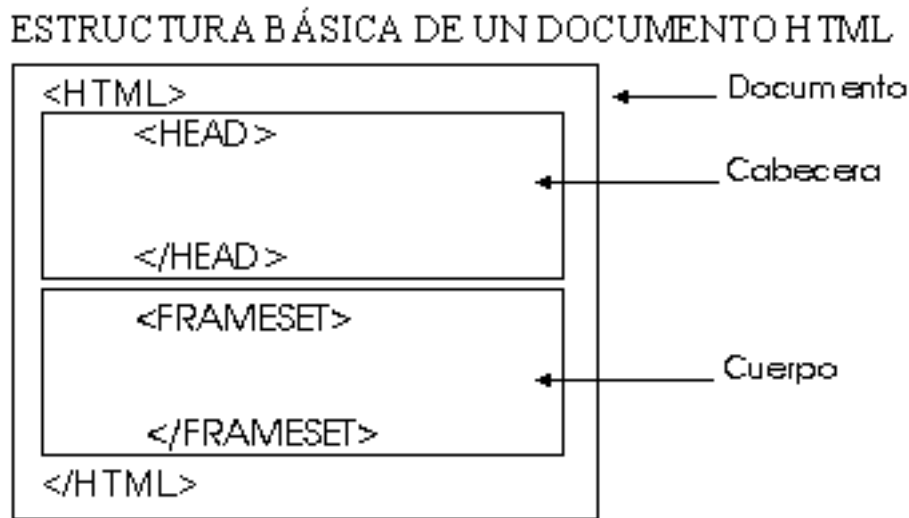


En este caso serán necesarios 4 documentos HTML. El primero simplemente contiene el código que le indica al navegador el número de frames de la página, su tamaño, su posición y el nombre de los archivos donde está el contenido de los 3 frames, a este documento HTML le llamaremos *documento de definición de frames*. Los otros 3 documentos HTML tienen el contenido de cada uno de los tres frames y deben ser creados de la misma manera que hemos venido creando los documentos HTML hasta ahora, eso sí, hay que tener en cuenta que ese contenido será mostrado en un espacio reducido (el del frame correspondiente) y no en toda la ventana del navegador.

Creación del documento de definición de frames

La estructura del documento de definición de frames es parecida a la estructura de las páginas creadas hasta ahora. La principal diferencia será que este documento no tendrá contenido y por tanto no debe usarse la etiqueta BODY. A cambio ha de usarse una nueva etiqueta: *FRAMESET*. Esta será la etiqueta clave en la creación de los frames. La estructura básica del documento de definición será por tanto la que observamos en la figura 8.8:

Figura 8.8. La estructura de las páginas con frames es diferente a las que no tienen. La principal diferencia es la sustitución de BODY por FRAMESET.



En ella se observa claramente como la etiqueta FRAMESET sustituye a BODY. Se podría decir que estas dos etiquetas son exclusivas, cuando esté una no puede estar la otra.

El siguiente paso en la creación de nuestra página con frames será decidir cuantos queremos crear como serán. Aunque posteriormente veremos cómo se pueden crear divisiones tan complejas como queramos, por ahora crearemos páginas en las que la ventana del navegador está dividida en filas o en columnas.

División en columnas

Para crear una página Web formada por varios frames que dividen la ventana del navegador en columnas usamos el atributo *COLS* de la etiqueta FRAMESET. Este atributo toma como valores separados por comas la anchura de los diferentes frames:

```
<FRAMESET COLS="ancho_col1, ancho_col2, ...">
```

Existen diversas formas de indicar el ancho de las columnas, una de ellas es usando porcentajes. Con este método podríamos dividir la ventana en dos columnas usando el siguiente código:

```

<HTML>

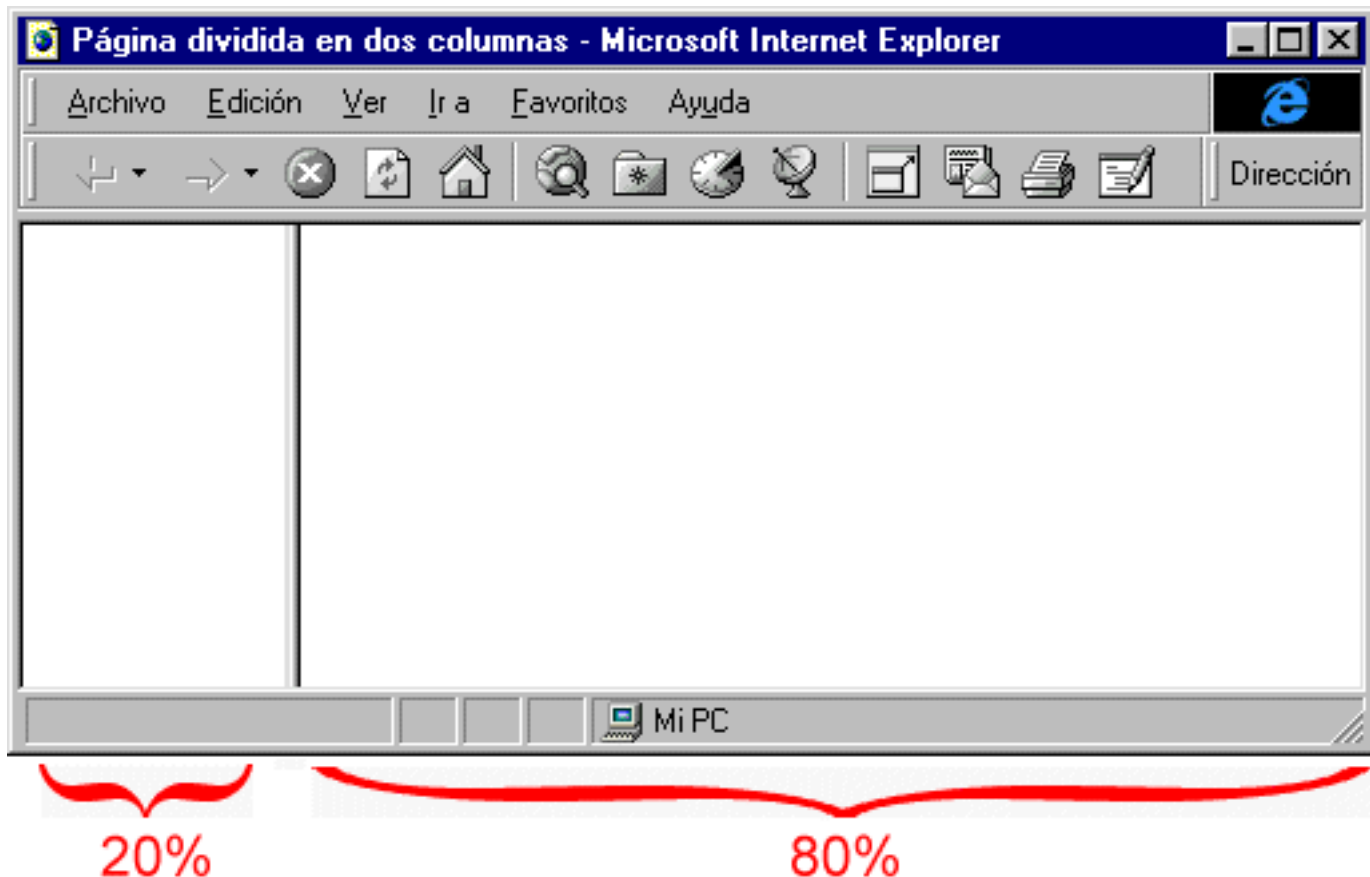
<HEAD>
</HEAD>

<FRAMESET COLS="20%,80%">
</FRAMESET>
  
```

</HTML>

En la que estamos diciendo que la primera columna (la de la izquierda) ocupará el 20% del ancho de la ventana del navegador y la segunda ocupará el 80% (ver figura 8.9).

Figura 8.9. Como vemos la ventana del navegador ha quedado dividida en dos columnas con un ancho del 20% y del 80% del ancho total de la ventana.



Aviso

Si el lector decide probar el ejemplo anterior probablemente no obtenga el resultado deseado en su navegador. Esto es así porque por ahora sólo hemos dicho que queremos que la ventana del navegador se divida en dos columnas pero para indicar que serán frames (lo cual es necesario para obtener el resultado de la figura 8.9) será necesario añadir algo más de código que veremos un poco más adelante.

Si quisiéramos dividir la ventana en 5 columnas simplemente deberíamos especificar cinco valores separados por comas, por ejemplo:

```
<FRAMESET COLS="10%, 50%, 15%, 20%, 5%">
```


Por supuesto la suma de todos los porcentajes debe ser igual a 100.

Nota

El lenguaje HTML no fija ningún límite en el número de filas, columnas o frames en general en que puede dividirse una página. Sin embargo no es recomendable crear más de 4 frames en total, porque si no el espacio dedicado a cada uno de ellos será muy pequeño. Debemos recordar que muchos de los navegantes verán nuestras páginas con un monitor de 14 pulgadas y una resolución de 640x480.

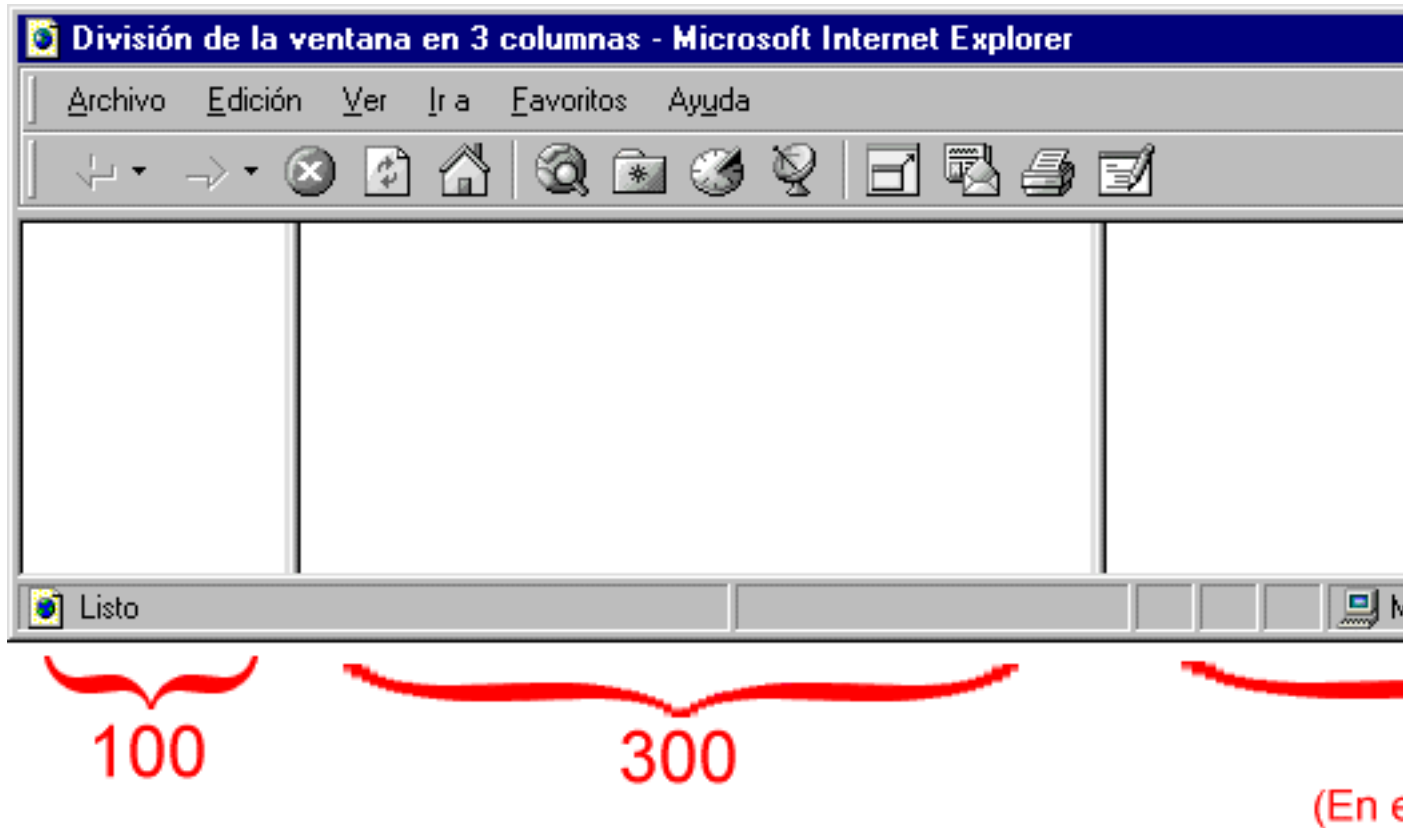
El método de especificar los tamaños de las columnas en porcentajes es sencillo de entender, sin embargo no siempre es útil. Por ello existen otros métodos que es importante comentar.

Uno de ellos consiste en especificar el ancho de las columnas en su tamaño absoluto en pixeles directamente. Sin embargo existe un problema: existen varias resoluciones de pantalla distintas y sería muy extraño que todos los navegantes que visiten nuestras páginas tengan una misma anchura de pantalla. Por esta razón no podemos saber cuál será el ancho total de la ventana del navegador y esto sería imprescindible para poder indicar el ancho de las columnas en pixeles. Para solucionar este problema Netscape [™] inventó el asterisco (*). Cuando se pone un asterisco como ancho de una columna estamos indicando al navegador que use todo el espacio sobrante para ese frame. Por ejemplo si escribimos:

```
<FRAMESET COLS="100,300,*">
```

se crearán tres columnas, la primera de 100 pixeles, la segunda de 300 y la tercera ocupará todo el espacio que reste y que dependerá del tamaño de la ventana del navegador con el que se visualicen las páginas. Si el navegante usa una configuración de 640x480 es de esperar que la ventana de su navegador tenga un ancho entre 600 y 640 pixeles y la tercera columna tendría una anchura de 200-240 pixeles. Si tiene una configuración de 800x600 el ancho será 760-800 pixeles y por tanto la tercera columna ocuparía en este caso 360-400 pixeles de ancho. Esta es la resolución que hemos usado para obtener la figura 8.8.

Figura 8.10. Las dos primeras columnas ocupan un ancho absoluto de 100 y 300 pixeles. La tercera al haber usado el asterisco ocupará el resto, que en este ejemplo son 372 pixeles.



Estas son las dos configuraciones más extendidas y las que por tanto debemos tener en cuenta. Si queremos cuidar mucho estos detalles debemos tener en cuenta las configuraciones de 1024x800 o mayores y el hecho de que los usuarios pueden cambiar el tamaño de la ventana del navegador a su antojo. Sin embargo, por lo general, no es en absoluto necesario llegar hasta esos extremos de cuidado.

El siguiente método para especificar el ancho de las columnas consiste en combinar todos los anteriores. La mejor forma de repasarlos es estudiando unos cuantos ejemplos. Para ellos vamos a suponer un navegador de 800 pixeles de ancho:

1. `<FRAMESET COLS="*, *">` : Al usar dos veces el asterisco se divide *el resto* entre las dos columnas a partes iguales. Es decir, cada una ocupará la mitad de la ventana del navegador (unos 400 pixeles cada una). Este ejemplo es idéntico a usar `< FRAMESET COLS="50%, 50%">`.
2. `<FRAMESET COLS="100, *, *">` : Divide la ventana en tres columnas. La primera de 100 pixeles y entre las otras dos divide a partes iguales el espacio sobrante (700 pixeles) con lo que (suponiendo el ancho total de 800 pixeles) cada una tendrá un ancho de 350 pixeles.

Nota

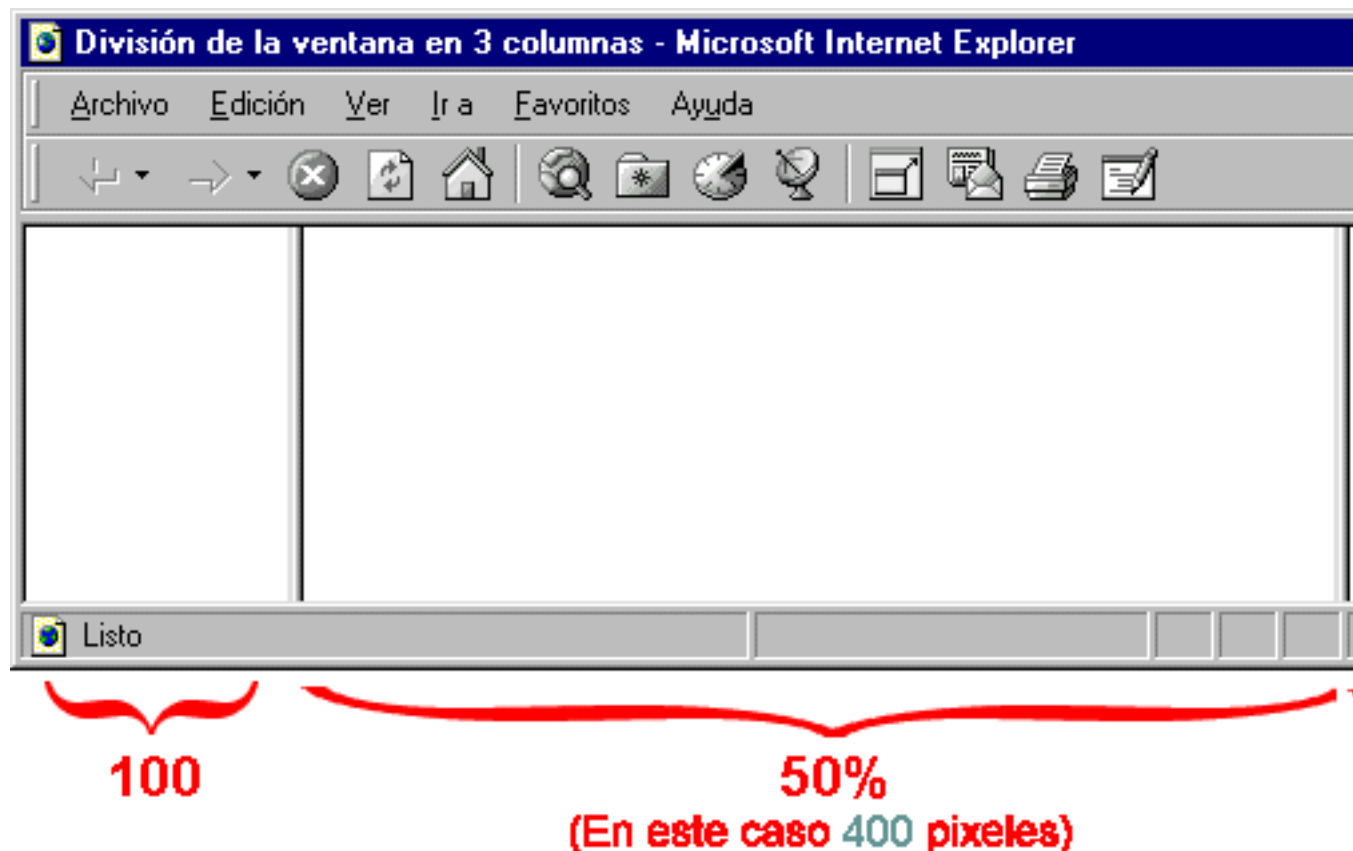
Algunos de estos ejemplos requieren algún tiempo de práctica. En caso de tener alguna dificultad entendiendo alguno de ellos es mejor seguir practicando con frames usando los métodos algo menos complejos. Una vez hayamos adquirido más práctica trabajando con frames podremos volver sobre estos ejemplos y para practicar con ellos.

3. `<FRAMESET COLS="200,2*,*">`: La primera columna ocupará un tamaño absoluto de 200 pixeles. El resto (600 pixeles) es dividido entre las dos siguientes columnas, pero ahora no será a partes iguales, sino que la columna central debe ser el doble que la columna izquierda (este detalle se indica anteponiendo un 2 al asterisco). Por tanto del espacio sobrante, dos tercios (que equivalen a 400 pixeles en este caso), serán para la columna central y un tercio (200 pixeles) será para la tercera.

Si hubiésemos querido que una columna fuese el triple de la otra deberíamos haber usado el código `COLS="200,3*,*"`. Y si quisiéramos una relación 4:5 no tendríamos más que escribir `COLS="200,4*,5*"`.

4. `<FRAMESET COLS="100,50%,*">`: La primera columna ocupa 100 pixeles, la segunda la mitad del ancho de la pantalla (400 pixeles) y la tercera el resto, que en este caso será 300 pixeles. En la figura 8.11 podemos ver un ejemplo en el que se realiza esta división de columnas en un ordenador con configuración 800x600:

Figura 8.11. En esta ocasión vemos un ejemplo del uso de FRAMESET combinando los distintos modos de indicar el tamaño y el ancho resultante de cada uno para una ventana con 800 pixeles de ancho.



Nota

Los tamaños absolutos deben usarse sólo en ocasiones que realmente lo requieran y siempre que los utilicemos debe haber al menos una de las otras columnas con un tamaño definido usando un asterisco (*).

División en filas

Hasta ahora hemos aprendido a dividir la ventana en columnas, pero como es de esperar también se puede dividir en filas. En esta ocasión será necesario usar el atributo *ROWS* (*rows* significa precisamente *filas* en inglés). El funcionamiento del atributo *ROWS* es muy similar al de *COLS*:

```
<FRAMESET ROWS="alto_fila1, alto_fila2, ..." >
```

Aviso

Es importante hacer notar que cuando dividimos la ventana en filas y usamos tamaños relativos, estos se refieren al alto de la ventana, y no tienen en cuenta lo *larga* que puedan ser las páginas o si hay que usar las barras de desplazamiento o no para moverse por ella.

En este caso es más importante, si cabe, tener en cuenta que ni podemos saber el alto de la ventana del navegador de aquellos que visiten nuestras páginas. Por esta razón el uso de tamaños relativos o del asterisco (*) es aún más usual. Veamos algunos ejemplos:

1. `<FRAMESET ROWS="200, *, 100">`: Divide la ventana en tres filas: una superior de 200 píxeles en la que se podría poner un encabezado; una inferior de 100 píxeles en la que puede ponerse una coetilla o una barra de navegación; y una central que ocupa el espacio sobrante y donde se podría situar el contenido principal.
2. `<FRAMESET ROWS="15%, *, 3*">`: Se divide la ventana en una primera fila que ocupa el 15% de la altura; la siguiente fila ocupará un cuarto del resto y la tercera tres cuartos.

No son necesarios más ejemplos para comprender el funcionamiento, así que vamos ya a definir el contenido de cada una de las divisiones que hemos aprendido a crear. Cuando insertemos este contenido, la división se convertirá en lo que realmente se llama *frame*. Mientras tanto ésta estará sujeta a más divisiones como veremos un poco más adelante.

Combinando COLS y ROWS

Los atributos *COLS* y *ROWS* pueden ser usados simultáneamente en una misma etiqueta *FRAMESET*. Si hacemos esto dividiremos la ventana del navegador en cuadrículas de diferentes tamaños. Podríamos interpretar el resultado de usar ambos atributos simultáneamente como que cada una de las filas definidas por *ROWS* es subdividida en el número de columnas que indica *COLS*. La página que mostramos en la figura 8.1 es una muestra del uso de *COLS* y *ROWS* simultáneamente. En ese caso el código usado fue:

```
<FRAMESET COLS="*, *" ROWS="*, *" >
```

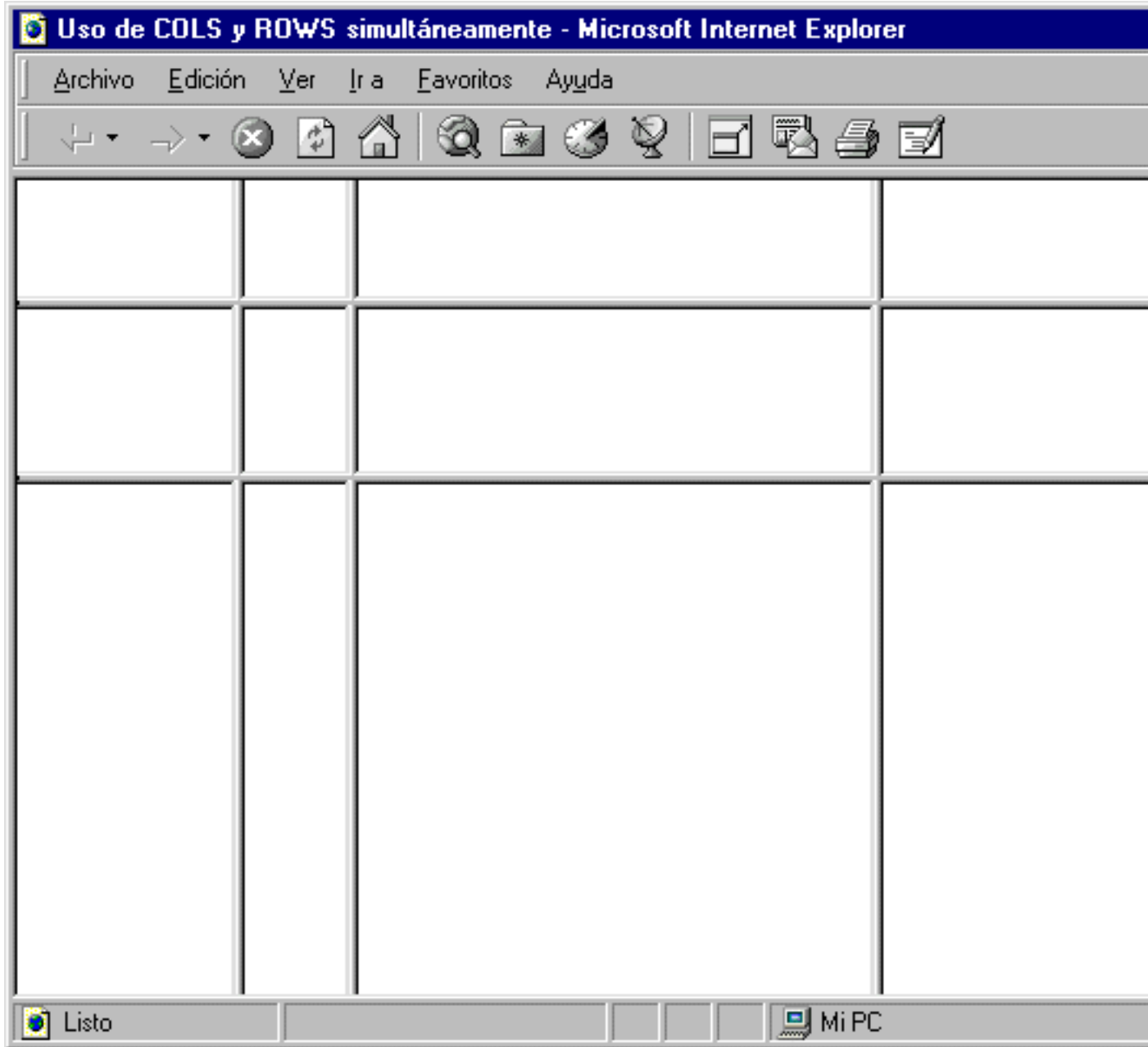
```
...  
</FRAMESET>
```

También se pueden crear combinaciones más complejas como por ejemplo:

```
<FRAMESET COLS="100,50,*,*"
ROWS="15%,*,3*">
... Más código ...
</FRAMESET>
```

Que da lugar a la página de la figura 8.12:

Figura 8.12. Es posible usar los atributos COLS y ROWS simultáneamente para dividir la ventana en varias filas y columnas.



La etiqueta FRAME

La etiqueta usada para insertar el contenido en las filas, columnas o divisiones en general, hechas con la etiqueta FRAMESET, es *FRAME*. Esta etiqueta debe ser usada una vez por cada una de las divisiones que hallamos creado. Cada una de estas veces debemos usar un atributo, *SRC*, que servirá para indicar el documento HTML donde se encuentre el contenido que queremos insertar en esa división:

```
<FRAME SRC="docum.htm">
```

Esta etiqueta consta de una única instrucción, es decir la instrucción de fin `</FRAME>` no existe, y debe ser insertada entre la instrucción de inicio y la instrucción de fin de la etiqueta `FRAMESET`.

Otro de los atributos más importantes de la etiqueta `FRAME` es *NAME*. Con este atributo le daremos un nombre al frame para poder referirnos posteriormente a él con el atributo `TARGET` o con los lenguajes de script. La forma de usar este atributo es:

```
<FRAME SRC="docum.htm"  
NAME="primer_frame">
```

Si posteriormente queremos crear un enlace a este frame desde otro de la misma página no tendremos más que escribir:

```
<A HREF="otro.htm"  
TARGET="primer_frame">
```

Con los conocimientos adquiridos hasta ahora en este capítulo, junto con lo que ya sabemos, estamos en disposición de crear nuestra primera página completa con frames.

Página completa con frames

El primer paso será crear el documento de definición de frames. Abrimos nuestro editor y creamos un nuevo archivo. En este caso lo llamaremos `index.htm` y escribimos en él el siguiente código:

```
<HTML>  
  
<HEAD>  
<TITLE>  
Primera página con frames  
</TITLE>  
</HEAD>  
  
<FRAMESET COLS="20%,80%">  
<FRAME SRC="izquierd.htm">  
<FRAME SRC="derecha.htm">  
</FRAMESET>  
  
</HTML>
```

Con esto tendremos una página que dividirá la ventana del navegador en una columna izquierda y otra derecha que ocupan el 20% y el 80% del ancho respectivamente. El contenido de cada una de estas columnas estará en los archivos `izquierda.htm` y `derecha.htm`. El siguiente paso será la creación de estos archivos. Volvemos a abrir un archivo nuevo en nuestro editor y le llamamos `izquierd.htm`. En este archivo podemos escribir todo el código HTML que queramos siempre teniendo en cuenta sus limitadas dimensiones. Para empezar podemos escribir:

```
<HTML>

<HEAD>
</HEAD>

<BODY BGCOLOR="#99FF99">
<H1>
Frame izquierdo
</H1>
Este frame podría servir de índice:
<UL>
<LI>Tema 1
<LI>Tema 2
<LI>Tema 3
</UL>
</BODY>

</HTML>
```

A continuación abrimos otro archivo nuevo y le llamamos `derecha.htm`. El código a incluir en este tercer documento HTML será:

```
<HTML>

<HEAD>
</HEAD>

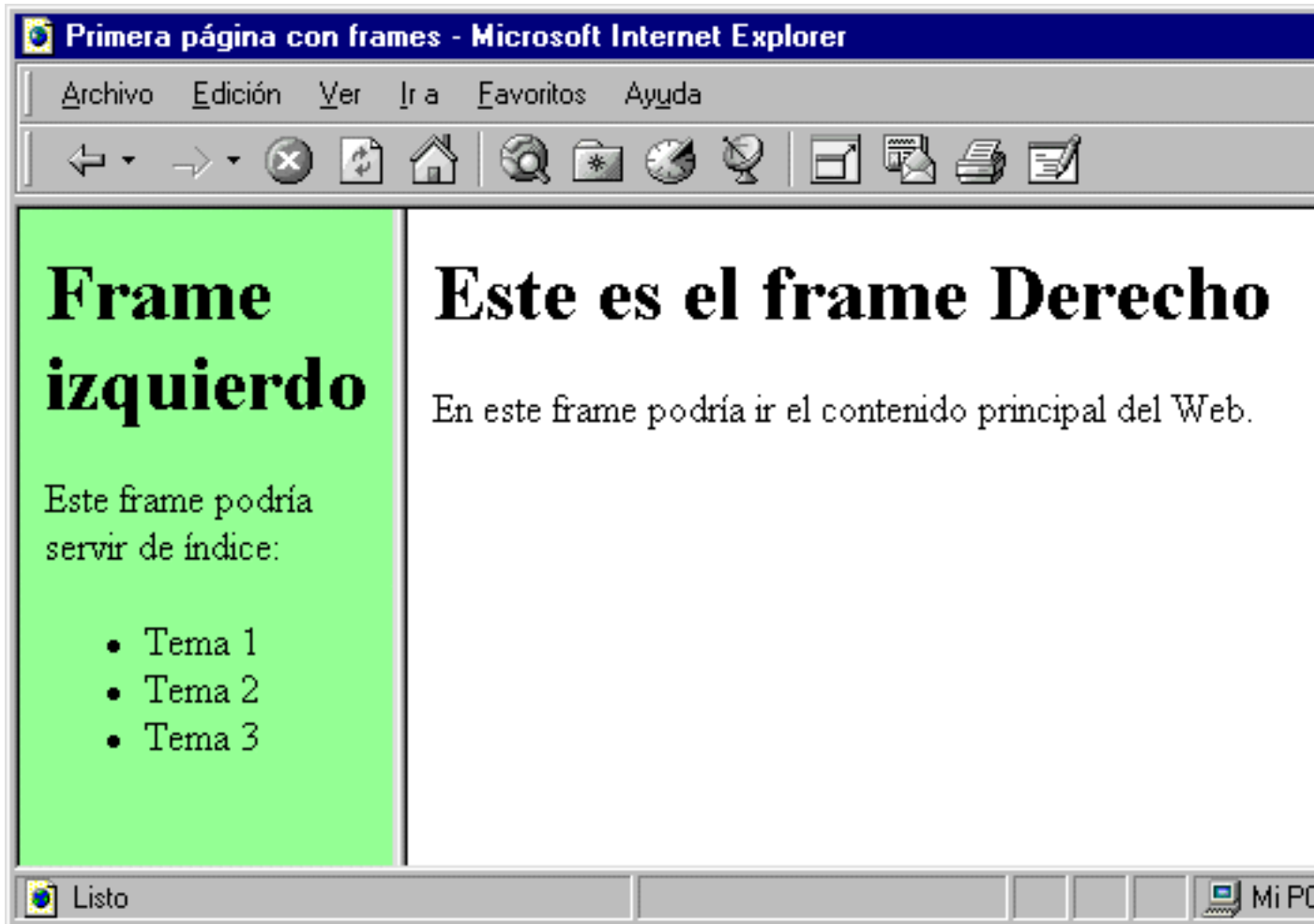
<BODY>
<H1>
Este es el frame Derecho
</H1>
<P>
En este frame podría ir el contenido principal del Web.
</P>
</BODY>

</HTML>
```

El último paso será visualizar el documento de definición de frames, `index.htm`, con el navegador que prefiramos. Por supuesto se podría ver cada uno de los documentos HTML por separado, y de hecho suele ser muy útil hacerlo

mientras escribamos el código para ir comprobando que lo estamos haciendo bien, sin embargo una vez hemos terminado los documentos `derecha.htm` y `izquierd.htm` sólo tienen significado como contenido de frames, y no como páginas Web individuales. Por esta razón el único archivo que debe ser visto *desde fuera* debe ser el documento de definición de frames (en este caso llamado `index.htm`). Una vez abierto `index.htm` con nuestro navegador deberíamos ver una página similar a la que observamos en la figura 8.13.

Figura 8.13. Ejemplo completo de una página con frames formado por un frame izquierdo pequeño que puede servir de índice y otro derecho con el contenido principal.



La creación de páginas con frames es algo más compleja que la creación de las páginas que hemos hecho hasta ahora, y por ello es común cometer errores, sobre todo al principio. Si al intentar ver una página con frames no observamos el resultado obtenido es recomendable comprobar los siguientes detalles:

1. Al igual que con el resto de páginas debemos asegurarnos de haber encerrado todo el código entre las instrucciones `< HTML>` y `</HTML>` y de que hemos guardado todos los archivos con terminación `.htm` o `.html`.

2. Comprobar que los nombres de los archivos coinciden con los nombres indicados como valor del atributo `SRC`. En algunos casos es necesario que los nombres coincidan en mayúsculas y minúsculas. Este será el caso si vamos a publicar nuestras páginas en un servidor UNIX (la mayoría lo son).
3. Es importante que nos aseguremos de no haber usado la etiqueta `BODY` en el archivo `index.htm`. Si lo hemos hecho debemos quitarla para que funcione nuestro ejemplo.
4. Si hemos solucionado todos los problemas anteriores y la página sigue sin visualizarse correctamente probablemente estaremos usando un navegador que no soporte frames.

Atributos de la etiqueta `FRAME`

Además del atributo `SRC` y `NAME`, la etiqueta `FRAME` tiene otros que nos permitirán modificar algunas características de los frames. Todos los atributos que veremos a continuación han sido aceptados en el estándar HTML 4.0.

- ***SCROLLING***: Usando este atributo podemos controlar la aparición o no de barras de desplazamiento. Los valores que puede tomar este atributo son:
 - ***SCROLLING="auto"***: Este es el valor por defecto y provoca que el navegador sólo inserte barras de desplazamiento si el contenido del frame ocupa más del espacio dedicado para él.
 - ***SCROLLING="yes"***: Las barras de desplazamiento aparecerán siempre.
 - ***SCROLLING="no"***: Las barras de desplazamiento no aparecerán nunca. Hay que tener mucho cuidado si usamos esta opción porque si la página ocupa más que el espacio de la ventana dedicado al frame la información que no quepa en un principio no podrá ser vista.
- ***NORESIZE***: Por defecto los usuarios pueden mover los bordes de los frames sin más que situar el ratón sobre ellos y arrastrarlos. Si incluimos este atributo en la etiqueta `FRAME` evitaremos que el borde de ese frame pueda ser movido. Es necesario notar que este atributo no toma ningún valor.
- ***MARGINHEIGHT***: Ajusta el margen superior que debe respetar el contenido de ese frame. El tamaño debe ser dado en píxeles.
- ***MARGINWIDTH***: Ajusta el margen lateral que debe respetar el contenido del frame. Al igual que antes, el margen debe ser dado en píxeles.

Para terminar vamos a ver un ejemplo del uso de todos ellos:

```
<FRAMESET COLS="110,*" ROWS="110,*">

<FRAME NAME="Imagen"
SRC="cuadrado.gif"
MARGINWIDTH="0"
MARGINHEIGHT="0"
SCROLLING="No"
FRAMEBORDER="No"

    <FRAME SRC="f2.htm"

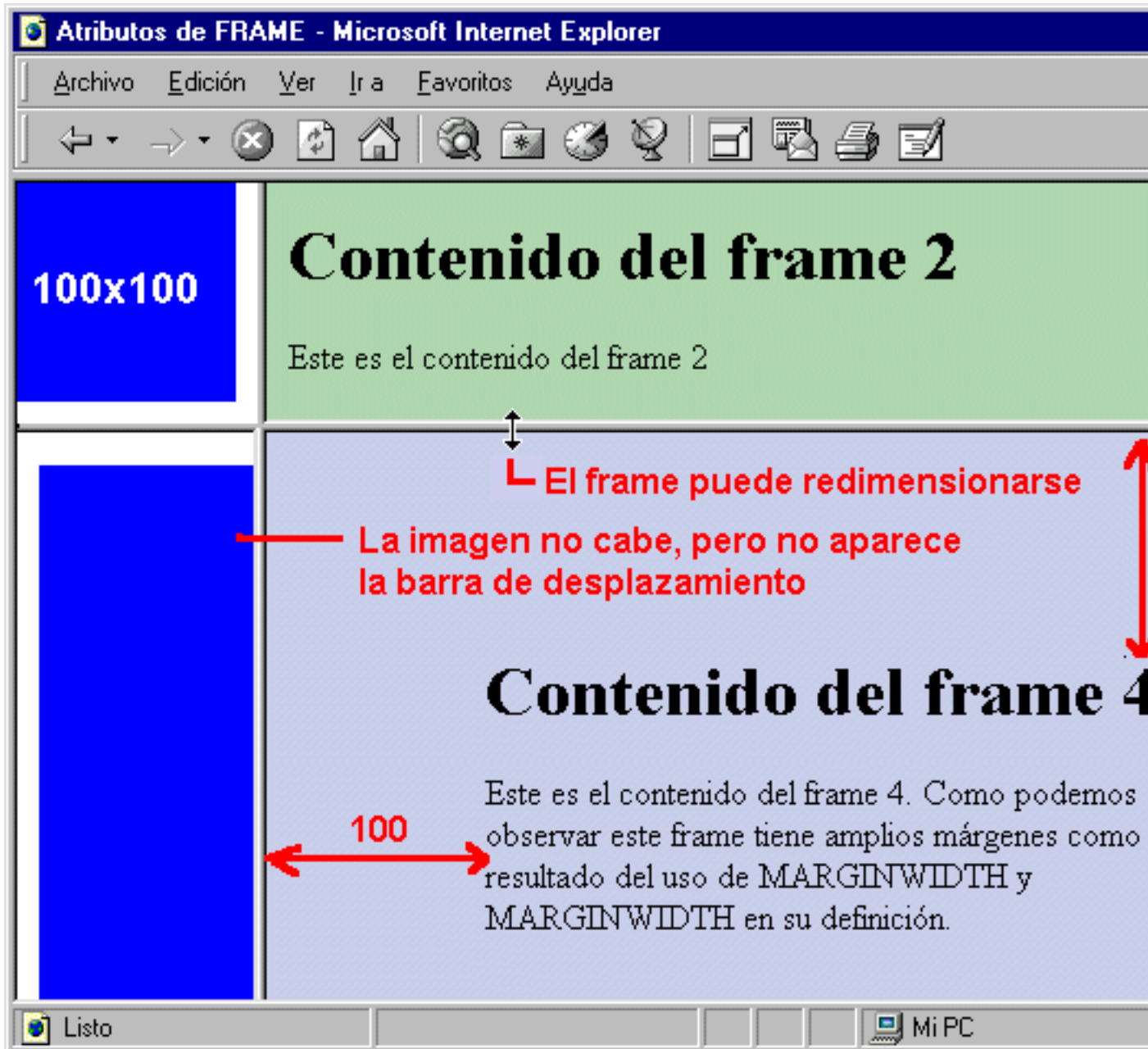
<FRAME SRC="rect.gif" SCROLLING="no">

<FRAME SRC="f4.htm"
MARGINWIDTH="100"
MARGINHEIGHT="100"

</FRAMESET>
```

que puede ser visto en la figura 8.14:

Figura 8.14. Usando los diferentes atributos de la etiqueta **FRAME** podemos tener un mayor control sobre las diferentes características de los frames.



En esta figura observamos, entre otras cosas, que aunque la figura `rect.gif` no cabe en el tercer frame (abajo izquierda) no aparecen las barras de desplazamiento. Otros aspectos interesantes están comentados en rojo sobre la figura.

Nota

En general es muy útil establecer unos márgenes iguales a cero en los frames cuya función sea la de barra de navegación. En los frames con contenido, especialmente si este es texto, se suele poner un margen de 10 píxeles.

Compatibilidad: NOFRAME

Hasta ahora hemos supuesto que los navegantes que vieran nuestras páginas con navegadores con soporte para frames, sin embargo ¿qué ocurre si las páginas son visualizadas con un navegador que no los soporte?. Este es un aspecto clave a considerar porque hoy por hoy los frames no forman parte del estándar y aunque Explorer™, Netscape™ y Opera™ los entienden hay muchos otros que no lo hacen.

Afortunadamente los inventores de los frames tuvieron este aspecto en cuenta y crearon una nueva etiqueta, *NOFRAME*, pensada para este tipo de navegadores. Esta etiqueta debe ser usada entre `<FRAMESET>` y `</FRAMESET>` de la siguiente manera:

```
<FRAMESET ...>
...Código de definición de frames...
<NOFRAME>
...Código alternativo...
</NOFRAME>
</FRAMESET>
```

Los navegadores que tengan soporte para frames simplemente ignorarán todo lo que haya entre la instrucción de inicio, `<NOFRAME >`, y la instrucción de fin, `</NOFRAME>` . Por su parte los navegadores que no entiendan frames, directamente ignorarán todas estas etiquetas (no las entienden) y pasarán a interpretar las etiquetas que siguen a `<NOFRAME >`. Entre ellas debe estar la etiqueta *BODY*, ya que esta etiqueta era imprescindible para las páginas sin frames, y para los navegadores que no las soporten todas las páginas son sin frames. En esta ocasión *BODY* no provocará problemas con *FRAMESET* porque, como hemos dicho, los navegadores que entiendan esta última ignorarán todo el código entre `<NOFRAME>` y `< /NOFRAME>`.

Un uso habitual de *NOFRAME* es:

```
<FRAMESET COLS="200, *">

<FRAME SRC="f1.htm">
<FRAME SRC="f2.htm">

<NOFRAME>

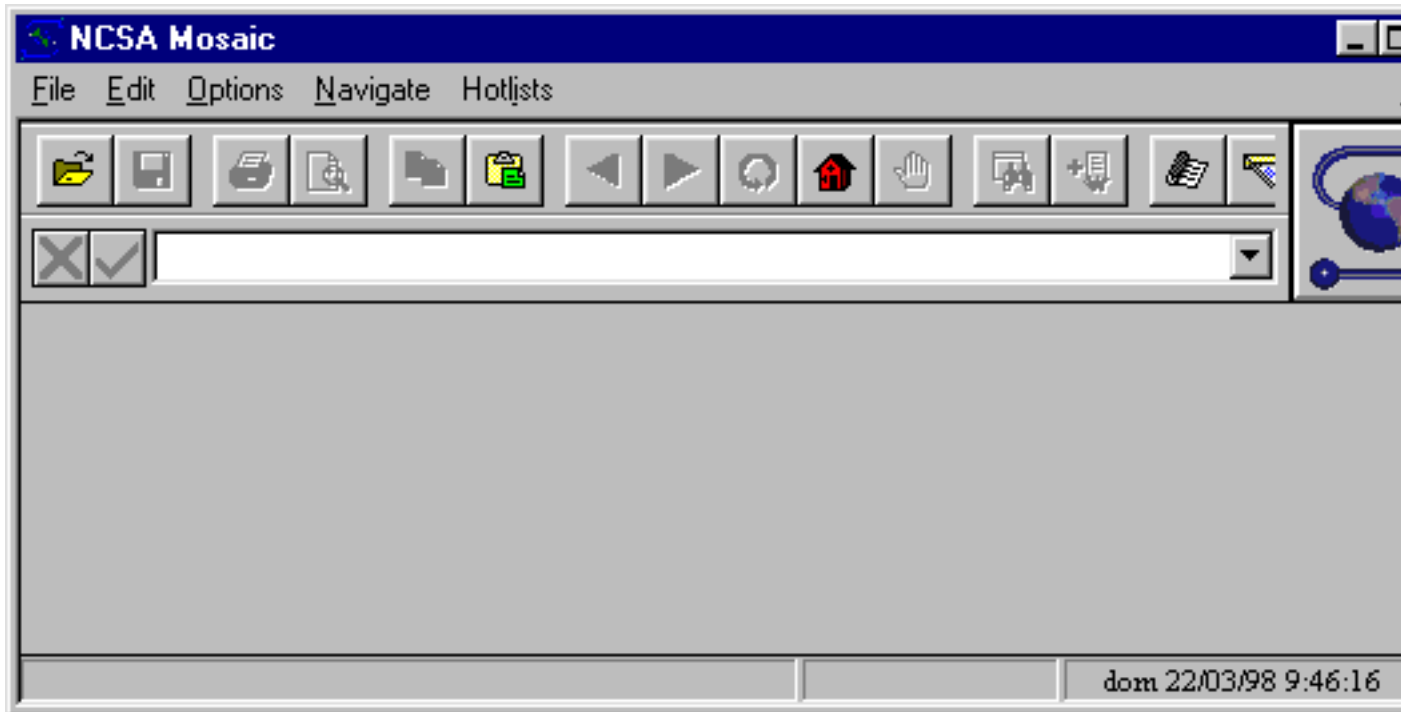
<BODY>
<P>Esta página usa frames y su
navegador no los soporta, por favor vaya a
<A HREF="sinframe.htm">
nuestras páginas sin frames
</A>
</BODY>

</NOFRAME>

</FRAMESET<
```

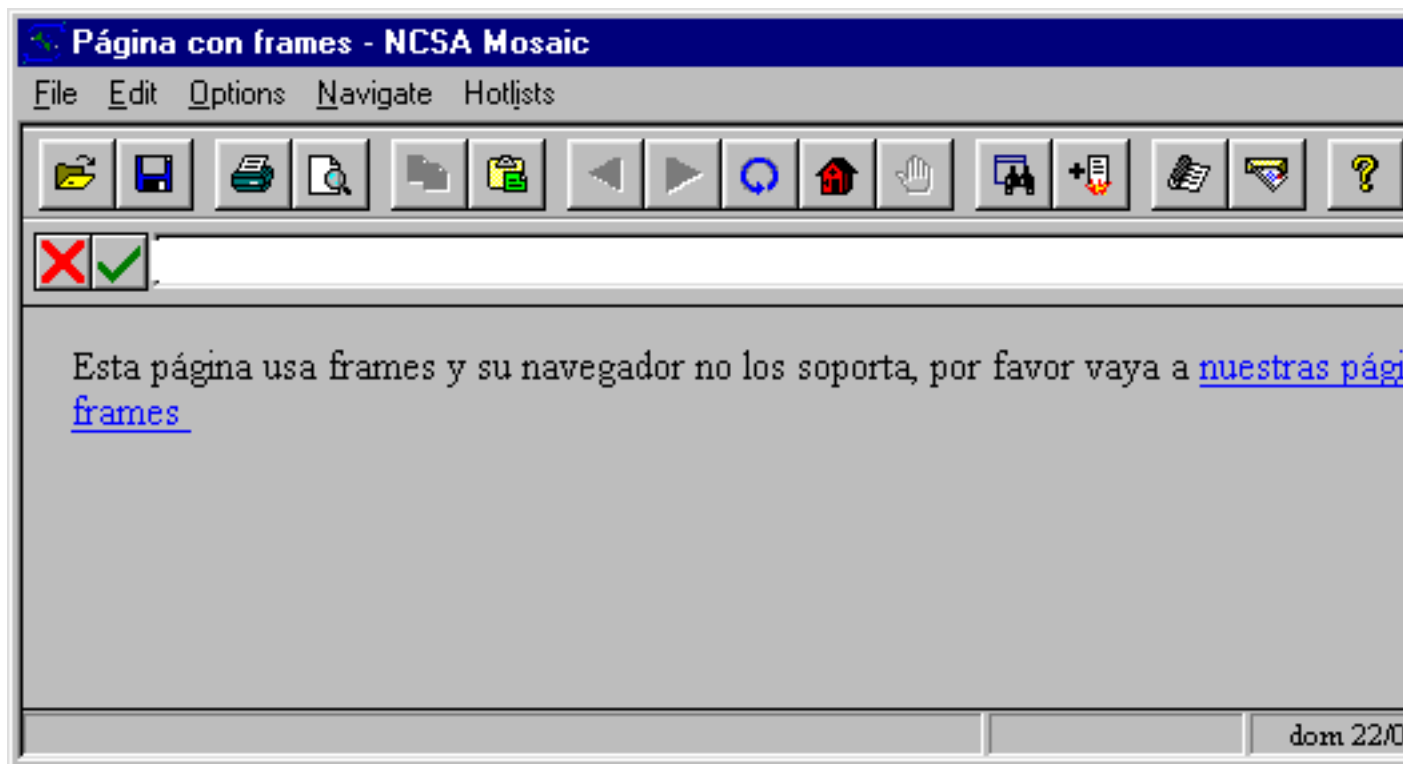
Si no hubiéramos usado la etiqueta *NOFRAME* el resultado de ver la página con este código con un navegador sin soporte para frames sería simplemente una pantalla en blanco como la que vemos en la figura 8.15:

Figura 8.15. El navegador NCSA Mosaic™ no soporta frames, con lo que los usuarios de dicho navegador no verán nada si no usamos la etiqueta NOFRAME.



usándola se obtendrá el resultado mucho más explicativo de la figura 8.16.

Figura 8.16. En este caso sí hemos usado NOFRAMES ofreciendo al navegante un enlace para ver una versión sin frames de las páginas.



Anidamiento de framesets

Hasta ahora hemos aprendido a dividir la ventana del navegador en dos o más filas con el atributo ROW, sin embargo las posibilidades del lenguaje HTML no acaban ahí.

Tanto a lo largo de este capítulo, como en el capítulo anterior hemos ido viendo algunos ejemplos de frames (figuras 7.1, 7.6 y 8.2) que presentaban páginas Web que dividían la pantalla del navegador de una forma más compleja que simplemente dividir en filas o columnas. ¿Cómo lo hacen? La respuesta es simple, anidando FRAMESETS. En la figura 8.17 podemos ver una explicación gráfica de lo que esto significa. Vamos a explicarla paso a paso.

Figura 8.17. Para conseguir estructuras de frames más complejas puede anidarse la etiqueta FRAMESET. En esta figura se indica cómo hacerlo.

FRAMESETS SENCILLOS

CÓDIGO:

```
<FRAMESET COLS="200, *" >
  <FRAME SRC="frame1.htm">
  <FRAME SRC="frame2.htm">
</FRAMESET>
```

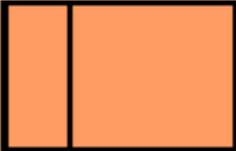
RESULTADO: 

Diagrama de anotación para el código simple:

- 1ª División (punta a la etiqueta <FRAMESET>)
- 2ª División (punta a la etiqueta <FRAME SRC="frame1.htm">)
- 1er NIVEL de anidamiento (punta a la etiqueta <FRAMESET COLS="200, *" >)
- Contenido de la 1ª división (punta a la etiqueta <FRAME SRC="frame1.htm">)
- Contenido de la 2ª división (punta a la etiqueta <FRAME SRC="frame2.htm">)

FRAMESETS ANIDADOS

CÓDIGO:

```
<FRAMESET COLS="200, *" >
  <FRAME SRC="frame1.htm">

  <FRAMESET ROWS="100, *" >
    <FRAME SRC="frame21.htm">
    <FRAME SRC="frame22.htm">
  </FRAMESET>
</FRAMESET>
```

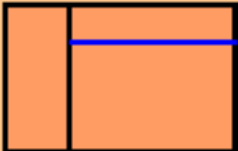
RESULTADO: 

Diagrama de anotación para el código anidado:

- 1ª División (punta a la etiqueta <FRAMESET COLS="200, *" >)
- 2ª División (punta a la etiqueta <FRAME SRC="frame1.htm">)
- 1er NIVEL de anidamiento (punta a la etiqueta <FRAMESET COLS="200, *" >)
- Contenido de la 1ª división (punta a la etiqueta <FRAME SRC="frame1.htm">)
- 2º NIVEL de anidamiento (subdivisión en dos partes de la segunda división) (punta a la etiqueta <FRAMESET ROWS="100, *" >)

Empezamos con el siguiente código:

```
<frameset cols="150,*">
... más código ...
</frameset>
```

Con él estamos dividiendo la ventana del navegador en dos columnas. Como hemos visto un poco antes en este capítulo podemos insertar contenido en cada una de las divisiones creadas usando la etiqueta `FRAME`, por ejemplo:

```
<frameset cols="150,*">
<frame src="frame1.htm"/>
<frame src="frame2.htm"/>
</frameset>
```

Pero existe otra posibilidad. En vez de usar la etiqueta `FRAME` para insertar contenido en la segunda de las divisiones podemos volver a usar la etiqueta `FRAMESET` (anidándola dentro de la anterior aparición de esta misma etiqueta) para subdividir la parte derecha en dos, es decir:

```
<frameset cols="150,*">
<frame src="frame1.htm"/>
<frameset rows="100,*">
... Más código ...
</frameset>
</frameset>
```

Ahora tendremos una página Web formada por tres frames. El primero está situado a la izquierda de la pantalla y tendrá 200 píxeles de ancho. Los otros dos ocupan el espacio sobrante. Uno ocupa los 100 píxeles superiores y el otro el resto. Por supuesto, al haber dos frames más, debemos volver a usar la etiqueta `FRAME` para insertar contenido en ellos. El código total quedaría por tanto.

```
<frameset cols="150,*">
<frame src="frame1.htm"/>
<frameset rows="100,*">
<frame src="frame21.htm"/>
<frame src="frame22.htm"/>
</frameset>
</frameset>
```

Sugerencia

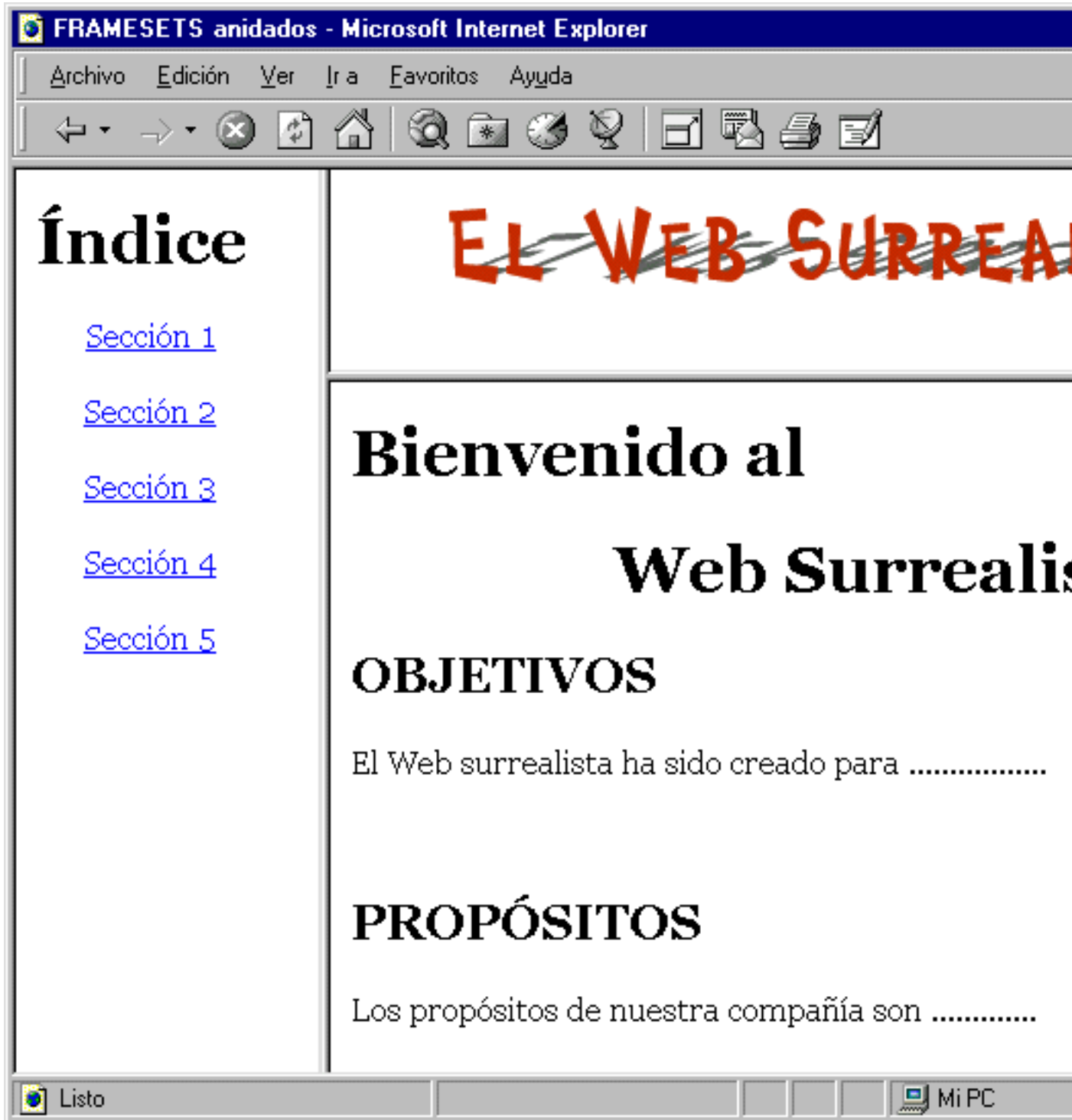
No existe ningún límite en el número de niveles de anidamiento de la etiqueta `FRAMESET` que pueden usarse. La única limitación que debemos tener en cuenta es el tamaño y resolución de pantalla que usarán los navegantes para ver nuestras páginas. Si usamos muchos anidamientos los frames serán muy pequeños y se podrá ver poca información en cada uno de ellos, con lo que se hará necesario usar las molestas barras de desplazamiento.

El hecho de haber elegido esta estructura de tres frames para el ejemplo no es casualidad. Como hemos visto en las capturas de páginas reales mostradas hasta ahora, es una de las más usadas. El frame izquierdo puede usarse como índice y el superior como un espacio para incluir un banner o encabezado gráfico. Estos dos frames pueden mantener su contenido constante mientras en el tercero y más grande de los frames se presenta el contenido principal de la página y es sobre el que se va navegando.

Para tener la página Web completa sólo nos quedaría crear los archivos `extra/ejem18/frame1.htm`, `extra/ejem18/frame21.htm` y `extra/ejem18/frame22.htm`. Nosotros hemos creado unos de ejemplo, incluyendo un encabezado gráfico para el frame superior y hemos obtenido el resultado que vemos en la fig8-18.

Figura 8.18. Nuestra primera página con frames. La ventana está dividida en 3 apartados (los frames) el izquierdo es el índice, el superior el título y el central el que tiene el

contenido.



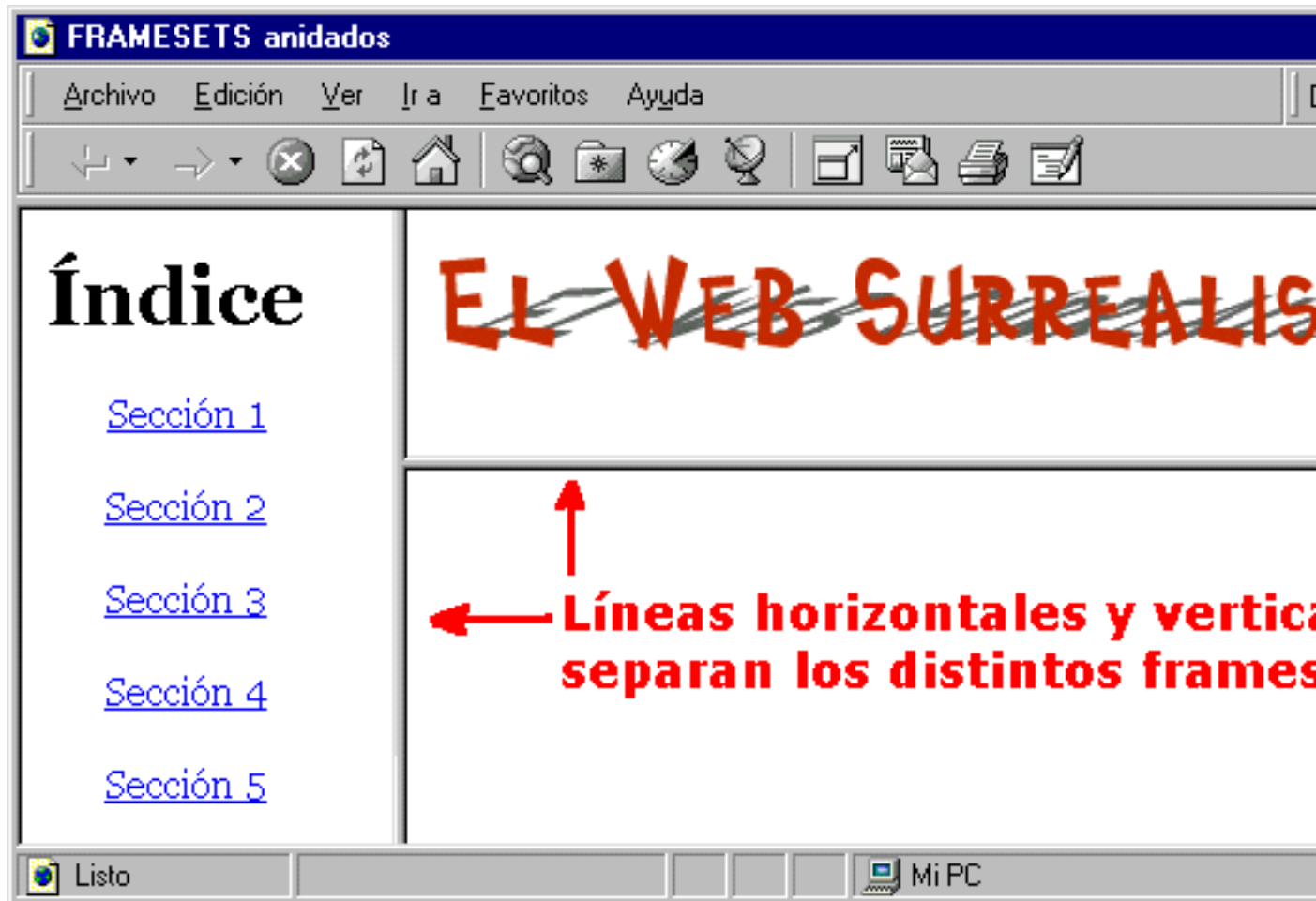
Todos los aspectos relacionados con los frames que hemos visto hasta ahora son iguales para todos los navegadores y han sido adoptados en el estándar HTML 4.0. Esto quiere decir que son entendidos por los navegadores Microsoft Internet Explorer 3.0 o posterior, Netscape Navigator 2.0 o posterior, Opera 3.0 o posterior y los navegadores gráficos que aparezcan de ahora en adelante con soporte completo de HTML 4.0.

Debemos recordar que los frames fueron una invención de Netscape y posteriormente Explorer y otros lo adoptaron. Sin embargo también cambiaron algunas características. Por esta razón en los atributos y etiquetas que veamos a partir de ahora haremos mención explícita de que diferencias existen entre la interpretación del código de los frames por parte de Netscape y Explorer, y lo indicaremos cuando alguna etiqueta, atributo o valor determinado haya sido adoptado por el estándar HTML 4.0.

Bordes de los frames

Cuando creamos páginas con frames estos aparecen divididos por unas líneas horizontales y verticales (ver 8-19) que además el usuario que vea nuestra página podrá mover a su antojo (a no ser que usemos el atributo `NORESIZE`).

Figura 8.19. En general los navegadores dibujan líneas con aspecto 3-D para separar los frames.



Desde la aparición de los frames los programadores de páginas Web han estado reclamando un mayor control sobre la disposición sobre estos bordes y con la aparición de Netscape 3.0 e Internet Explorer 3.0 se introdujeron nuevos métodos para definir sus características.

Estos métodos no son más que la inclusión de tres nuevos atributos para las etiquetas `FRAME` y `FRAMESET`. Estos son `BORDERCOLOR` y `FRAMEBORDER` para `FRAME` y `BORDER` para `FRAMESET`. Aprendamos a usarlos uno por uno.

Cambiando el color del borde

El primero de los atributos, `BORDERCOLOR`, nos permitirá cambiar el color del borde. El color puede indicarse tanto con el número hexadecimal RGB (por ejemplo `#99CC99`) o bien usando uno de los nombres predeterminados que vimos en el capítulo 5. Por ejemplo para poner el borde de color rojo podríamos usar:

```
<FRAME BORDERCOLOR="red">
```

O bien:

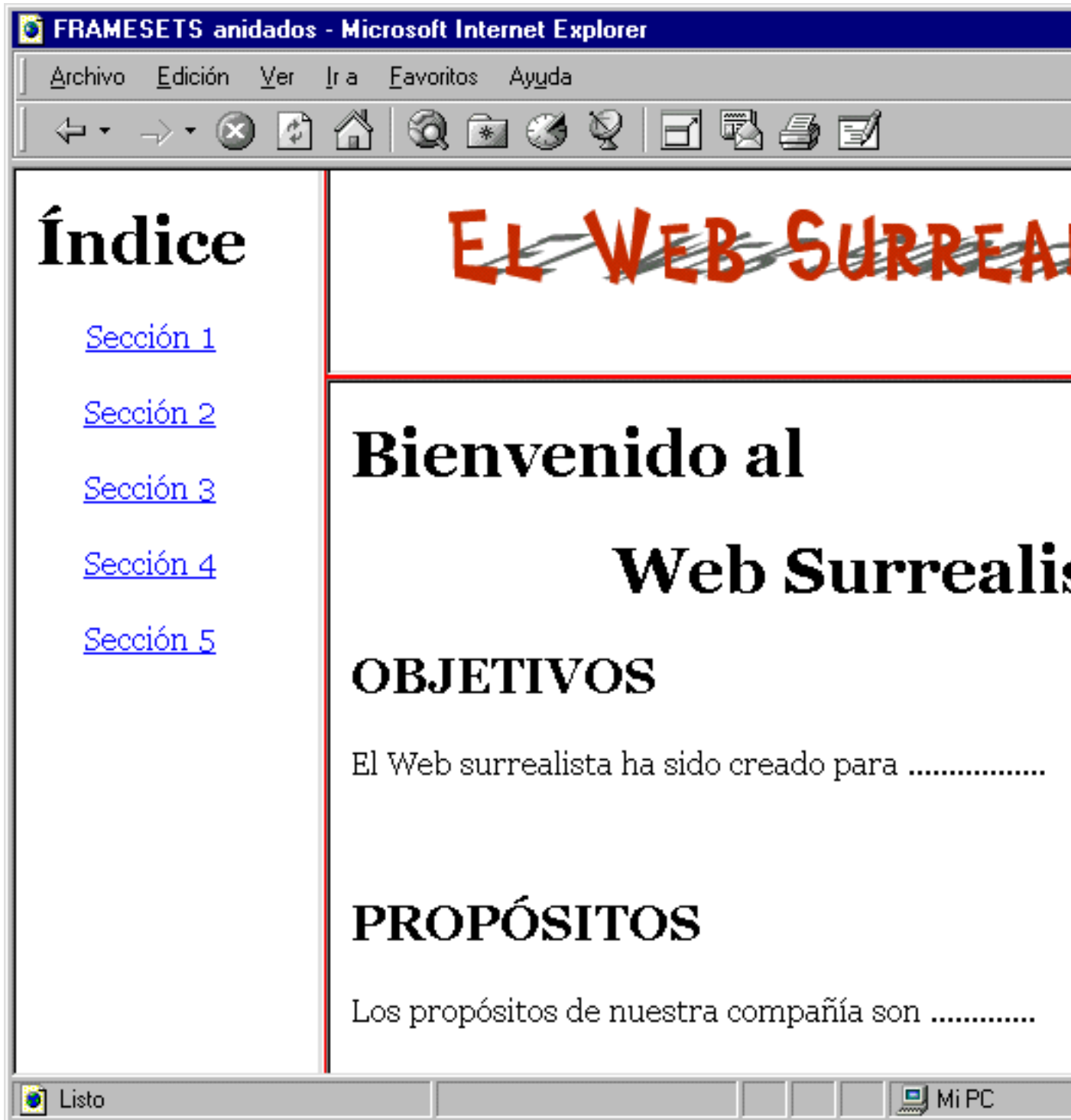
```
<FRAME BORDERCOLOR="#FF0000">
```

Dado que los bordes son compartidos por varios frames, en el caso de que cada uno de los bordes indique un color diferente el resultado es impredecible y depende del navegador. Por esta razón no podemos fiarnos de que si se da esta situación lo que nosotros con nuestro navegador será lo mismo que vean los navegantes y por tanto debemos evitarla en lo posible.

Si en cada uno de los frames de las páginas creadas en la sección anterior usamos este atributo para indicar bordes rojos obtendremos el resultado de la figura 8.20. El código usado es:

```
<FRAMESET COLS="150,*">
<FRAME SRC="frame1.htm" BORDERCOLOR="#FF0000">
<FRAMESET ROWS="100,*">
<FRAME SRC="frame21.htm" BORDERCOLOR="#FF0000">
<FRAME SRC="frame22.htm" BORDERCOLOR="#FF0000">
</FRAMESET>
</FRAMESET>
```

Figura 170. Con el atributo BORDERCOLOR podemos elegir el color que queremos para los bordes de los frames.



Hay que admitir que el cambio en la apariencia de la página no es muy importante, sin embargo puede ser muy útil en ciertos Webs en los que los colores están muy cuidados.

El navegador Netscape permite también usar el atributo `BORDERCOLOR` en la etiqueta `FRAMESET`, en este caso el valor que demos a este atributo afectará a todos los frames dentro de ese `frameset`. De esta manera podríamos habernos evitado escribir `BORDERCOLOR="#FF0000"` tres veces en el ejemplo anterior. Sin embargo debemos tener en cuenta que esta posibilidad sólo la ofrece Netscape Navigator y no Explorer ni otro navegador. Además tampoco ha sido aceptada por el estándar HTML 4.0.

Como quitar el borde

Muchos diseñadores de Webs que habían sido detractores de los frames cambiaron su opinión sobre ellos una vez se ofreció la posibilidad de quitar los bordes de estos. De esta forma se puede crear la ilusión de que existe un sólo frame (o una página sin frames), mientras aprovechamos las posibilidades para mantener siempre visibles ciertas partes de la página.

La etiqueta `FRAME` tiene un atributo que puede ser usado para quitar los bordes de los frames: `FRAMEBORDER`. Este atributo puede tomar dos valores: "1" y "0" que significan mostrar los bordes y ocultarlos respectivamente. El valor por defecto es mostrar los bordes (`FRAMEBORDER="1"`) y si queremos que estos desaparezcan debemos escribir explícitamente:

```
<FRAME FRAMEBORDER="0">
```

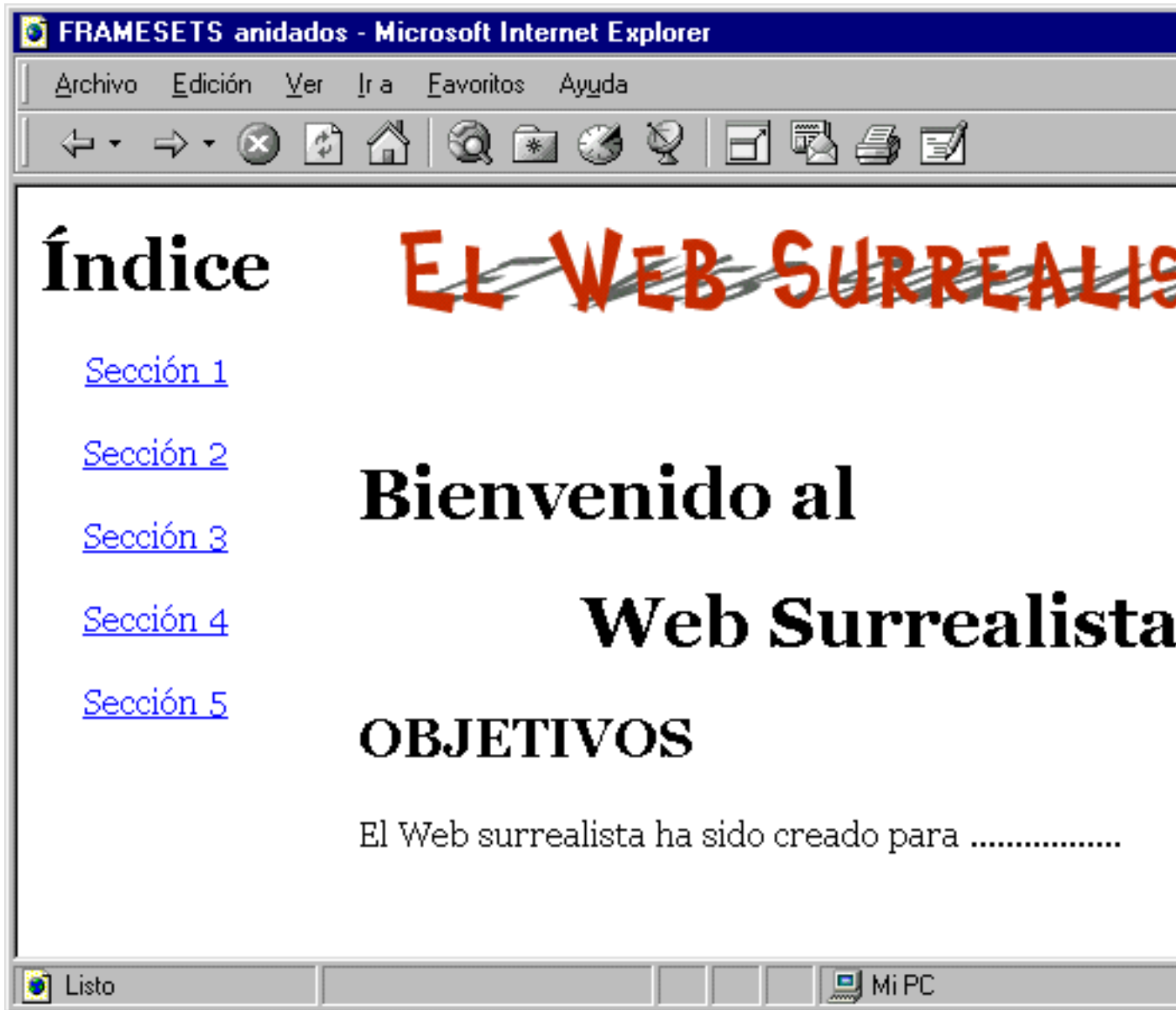
Al igual que antes se pueden producir conflictos si un borde es compartido por dos frames uno de los cuales indica `FRAMEBORDER="0"` y el otro `FRAMEBORDER="1"`. La decisión que tome el navegador en esta situación es impredecible y por tanto debe ser evitada en la medida de lo posible.

Aunque no ha sido aceptado por el estándar HTML 4.0, es importante hacer notar que tanto Netscape como Explorer admiten el uso del atributo `FRAMEBORDER` en la etiqueta `FRAMESET`. De esta forma podemos conseguir quitar todos los bordes de una vez poniendo `FRAMEBORDER="0"` en la primera etiqueta `FRAMESET` en lugar de repetirla para cada uno de las apariciones de `FRAME`.

Atención

El atributo `FRAMEBORDER` surgió como extensión al estándar de HTML. Inicialmente los valores válidos eran *yes* y *no*. Sin embargo HTML 4.0 introdujo el soporte de frames en el estándar y los valores aceptados de `FRAMEBORDER` son 1 y 0. La conclusión es que aunque en las páginas existentes podamos encontrarnos cualquier combinación de los valores anteriores, en las páginas nuevas deberíamos usar sólo los valores numéricos estándar.

Figura 171. Usando `FRAMEBORDER="0"` ocultamos los bordes de los frames con lo que se consigue una gran mejora en la apariencia.



En la mayoría de las ocasiones quitando el borde de los frames se consigue una apariencia muy mejorada de nuestras páginas. Para conseguir este efecto en las páginas creadas en el ejemplo anterior sólo tenemos que cambiar ligeramente el código para incluir el atributo `FRAMEBORDER="0"`:

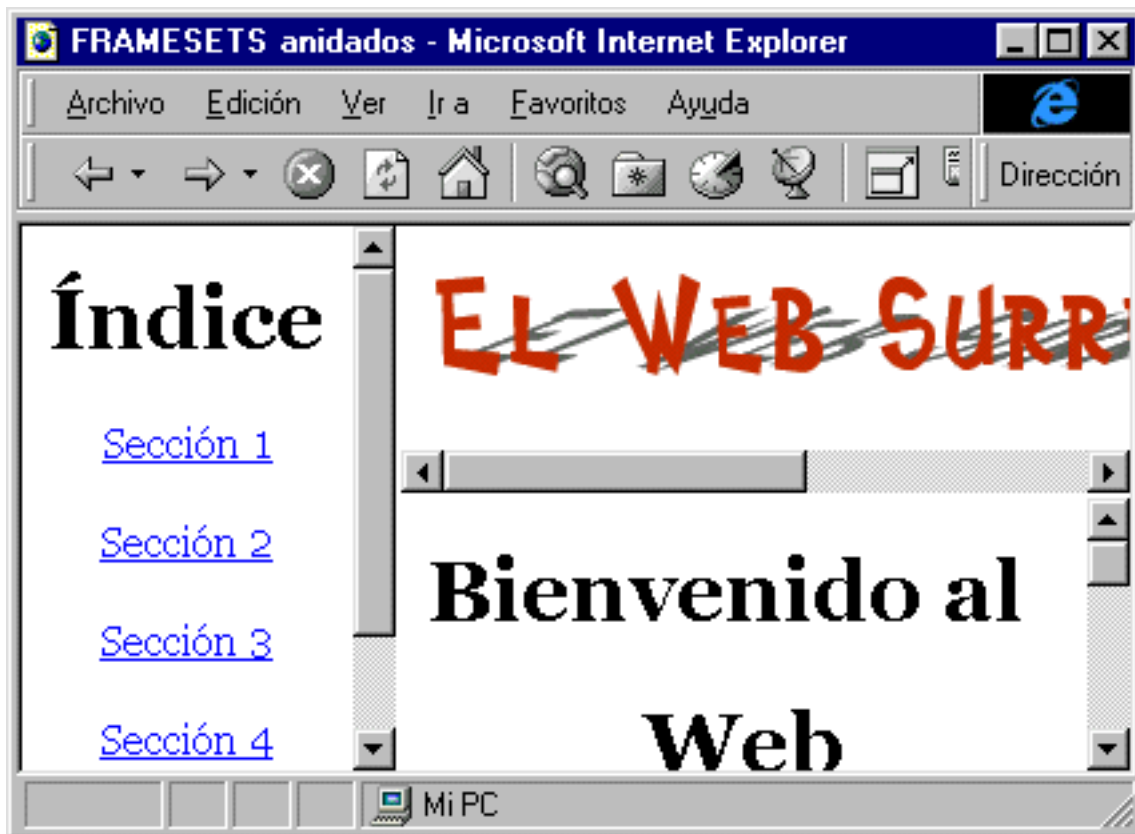
```
<FRAMESET COLS="150, *">
<FRAME SRC="frame1.htm" FRAMEBORDER="0">
<FRAMESET ROWS="100, *">
<FRAME SRC="frame21.htm" FRAMEBORDER="0">
<FRAME SRC="frame22.htm" FRAMEBORDER="0">
</FRAMESET>
</FRAMESET>
```



```
</FRAMESET>
```

Con este código obtendríamos el resultado de la figura 8.21 en la que no se distinguen los bordes de los frames y por lo tanto no se pueden redimensionar. Sin embargo si hacemos la ventana del navegador más pequeña aparecerán en cada uno de los frames unas barras de desplazamiento que delatarán su presencia, como vemos en la figura 8.22.

Figura 8.22. Aún usando el atributo `FRAMEBORDER` si hacemos la ventana del navegador pequeña aparecerán barras de desplazamiento en cada uno de los frames.



Grosor del borde

Una de las últimas extensiones de Netscape Navigator relacionadas con los frames es el atributo `BORDER`. Con él podemos indicar el grosor del borde que separa los frames asignándole un valor en píxeles. La forma de usar este atributo es:

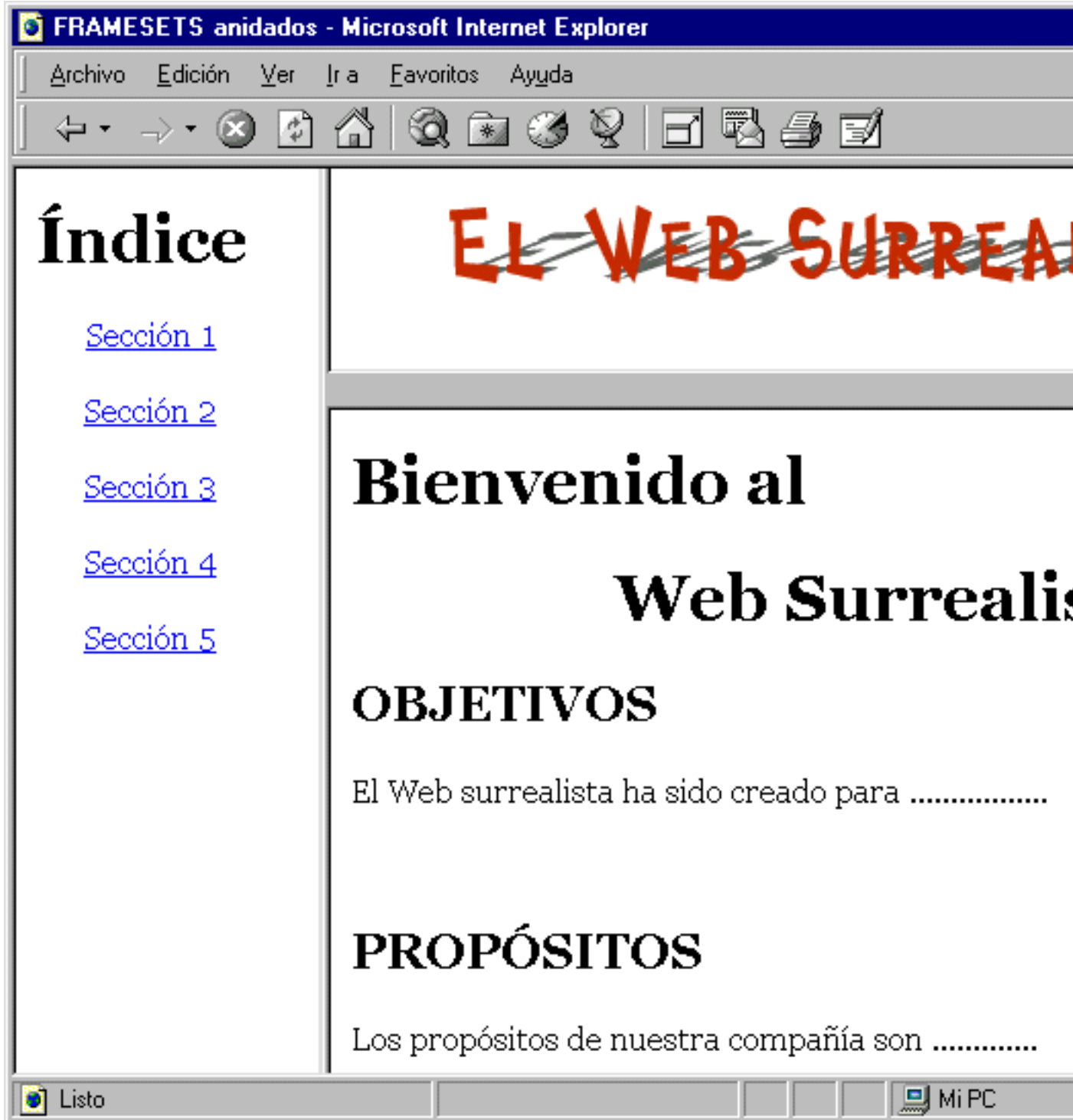
```
<FRAMEBORDER BORDER="TamañoBordeEnPíxeles">
...
</FRAMEBORDER>
```

Si existen varias etiquetas `FRAMEBORDER` en una misma página (es decir si las hemos anidado) podemos usar el atributo `BORDER` en cualquiera de ellas, pero es conveniente usarla una única vez. Vamos a usarla con nuestro ejemplo para crear un borde grueso entre las dos divisiones del segundo de los `FRAMESETS`. Para ello usamos el código:

```
<FRAMESET COLS="150,*">  
<FRAME SRC="frame1.htm">  
<FRAMESET ROWS="100,*" BORDER="20">  
<FRAME SRC="frame21.htm">  
<FRAME SRC="frame22.htm">  
</FRAMESET>  
</FRAMESET>
```

Con lo que obtenemos el resultado de la figura 8.23.

Figura 8.23. Con el atributo `BORDER` podemos cambiar el grosor de los bordes. En este caso el borde horizontal tiene un grosor de 20 píxeles.



Valores especiales del atributo `TARGET`

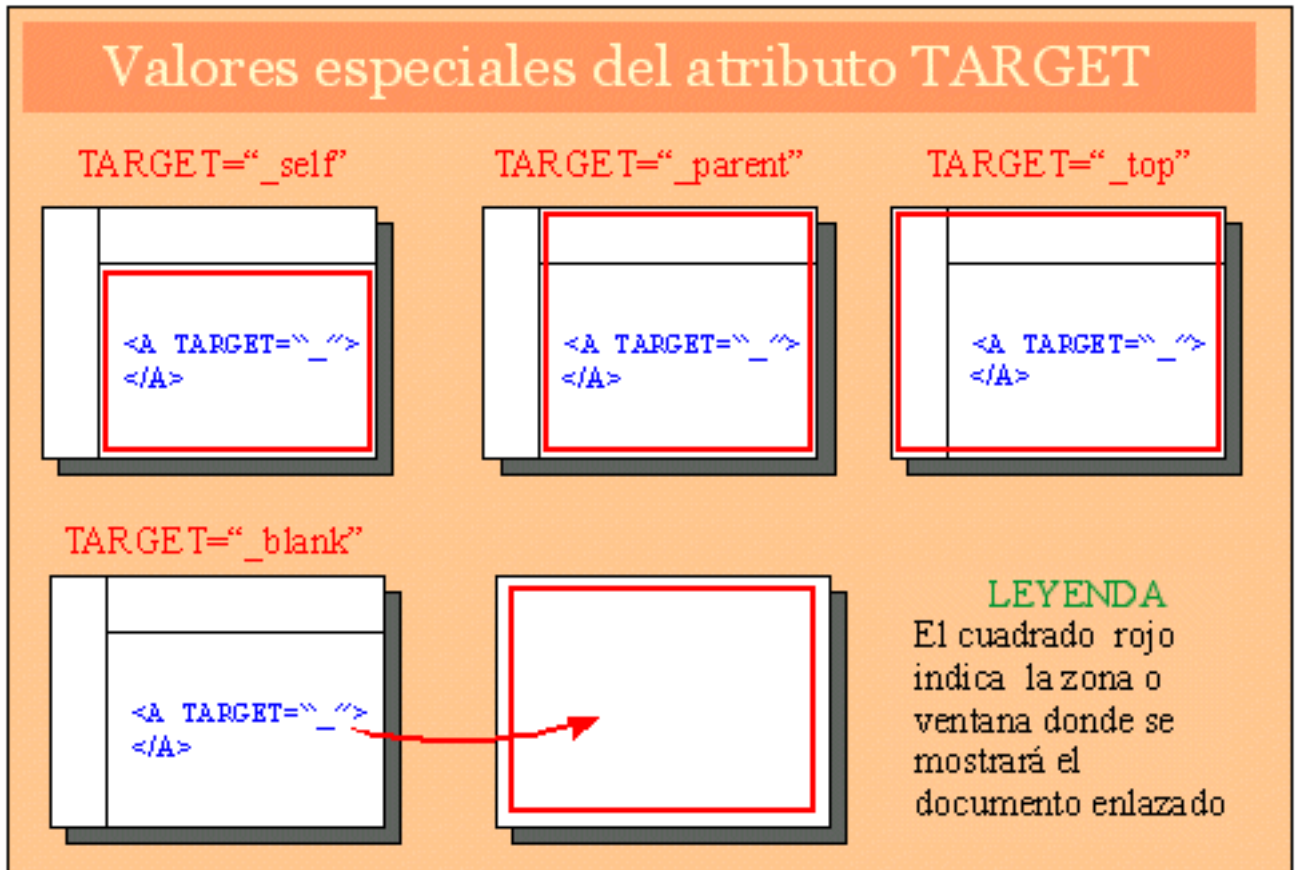
Ya conocemos todo lo que necesitamos sobre los frames, pero antes de empezar la siguiente práctica es importante conocer una serie de valores de `TARGET` con significado especial.

Como hemos visto, el atributo `TARGET` nos permite mostrar el resultado de un enlace en el frame que queramos indicando el nombre de dicho frame. Existen, además, ciertos valores que puede tomar este atributo que tienen un significado especial:

<code>TARGET="_blank"</code>	Fuerza que el documento referenciado por el enlace sea mostrado en una nueva ventana del navegador.
<code>TARGET="_self"</code>	Usando este valor el documento enlazado será mostrado en el mismo frame o ventana donde está el enlace. Este valor es especialmente útil cuando se ha usado la etiqueta <code>BASE</code> para especificar un frame destino por defecto distinto del actual.
<code>TARGET="_parent"</code>	Este valor provoca que el documento sea mostrado en el <code>FRAMESET</code> padre del frame actual.
<code>TARGET="_top"</code>	Fuerza a que el enlace sea mostrado usando todo el espacio de la ventana del navegador destruyendo toda estructura de frames. Este valor debe ser usado siempre que creamos un enlace a una página externa a nuestro sitio web.

En la figura 8.24 podemos ver una representación del significado de cada uno de estos valores que nos ayudará a resolver las dudas que tengamos sobre ellos.

Figura 8.24. Representación de los cuatro valores especiales del atributo TARGET: `_self`, `_parent`, `_top`, `_blank`.



Ejercicio con TARGET

A continuación es aconsejable realizar algunas prácticas para habituarse al uso de estos valores. Cree una página sencilla de prueba:

```
<HTML>
<HEAD></HEAD>
<BODY>
<H1>Página de prueba</H1>
</BODY>
</HTML>
```

Guárdela con el nombre `prueba.htm`. A continuación usaremos el ejemplo de página con 3 frames creado anteriormente. A partir del archivo `extra/ejem18/frame22.htm` [extra/ejem18/frame22.htm] edítelo, incluya el siguiente código en él y guárdelo en el mismo directorio que `prueba.htm`:

```
<A HREF="prueba.htm" TARGET="_self">
Enlace a la página de prueba
```

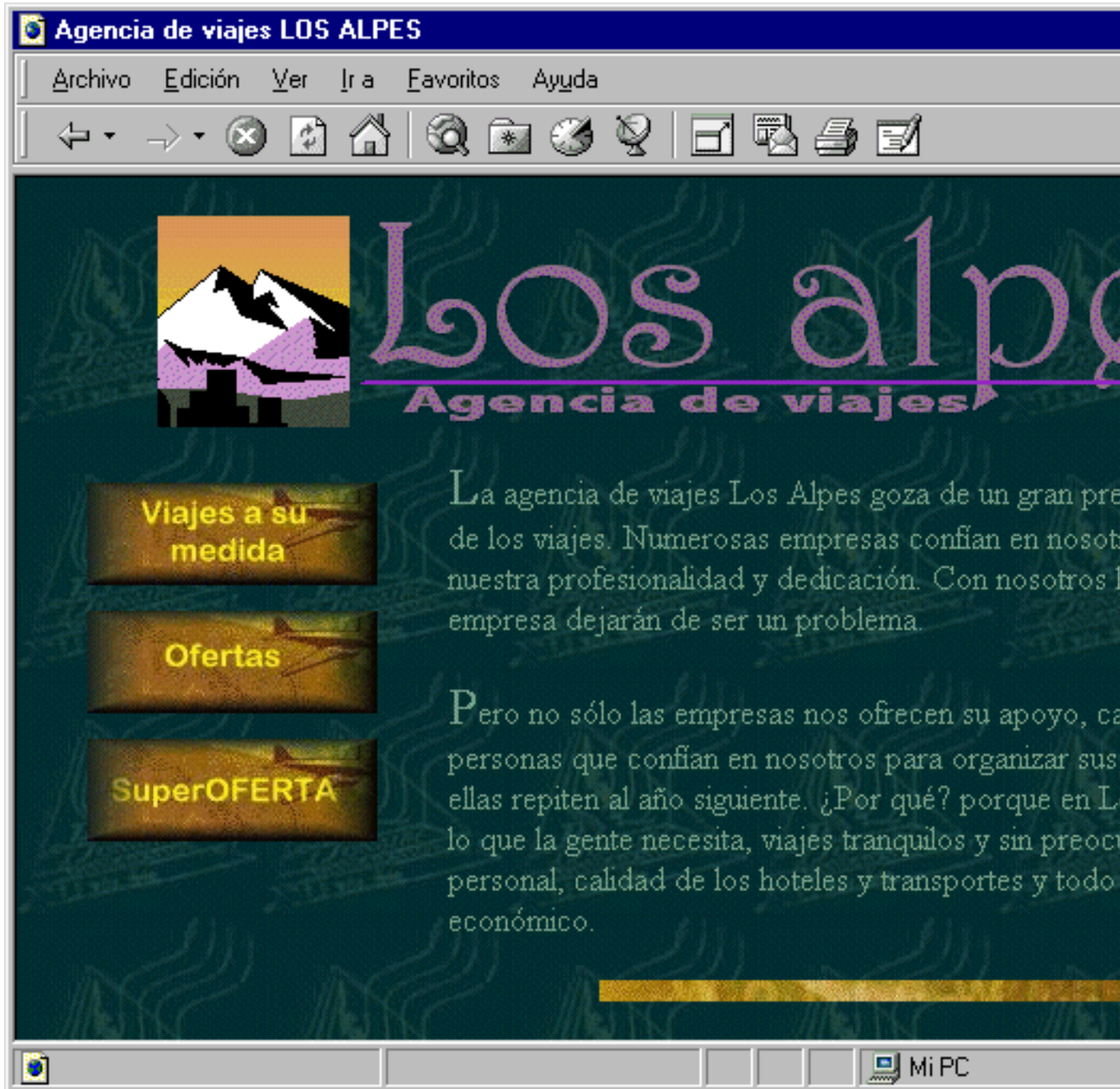

Para comprobar el efecto que se ha conseguido hay que abrir con el navegador la página `index.htm`, que era la que contenía los framesets y pulsar sobre el enlace. La página de prueba sustituirá al frame donde estaba el enlace.

A continuación editamos de nuevo el archivo sustituyendo `TARGET="_self"` por `TARGET="_parent"` y volvemos a probar el enlace en el navegador. Seguidamente probamos los otros dos valores especiales. Este es un buen momento también para practicar usando los nombres de los frames como valor del atributo `TARGET`.

Práctica 5: Web de Los Alpes con frames

El objetivo de nuestra quinta práctica consistirá en realizar una versión con frames del sitio Web de la agencia de viajes Los Alpes que creamos en la práctica 4 (figura 8.25). Como es nuestro primer gran proyecto con frames nos limitaremos a la división de la ventana del navegador en dos frames fila evitando así las dificultades del anidamiento de frames. Nuestra recomendación para el lector avanzado es que realice la práctica tal y como aquí se expone y una vez completada con éxito intente mejorarla introduciendo un tercer frame. Este tercer frame puede servir por ejemplo para mantener siempre visible los botones de las páginas `mes.htm` y `medida.htm`. Para ello aconsejamos crear una división en dos columnas reservando la izquierda para los botones y usando la derecha para mostrar el contenido principal.

Figura 175. Página Web de Los Alpes creada en la práctica 4 de este curso. Ahora la modificaremos para crear una versión con frames.



Documento de definición de frames

Volvamos a la versión de dos frames. El frame superior (banner) lo usaremos para mantener siempre visible el logo de Los Alpes junto con los botones que nos llevan a las distintas secciones de la página. En el frame inferior (principal) iremos mostrando cada una de las páginas de las que se compone el sitio Web. Estas deberán

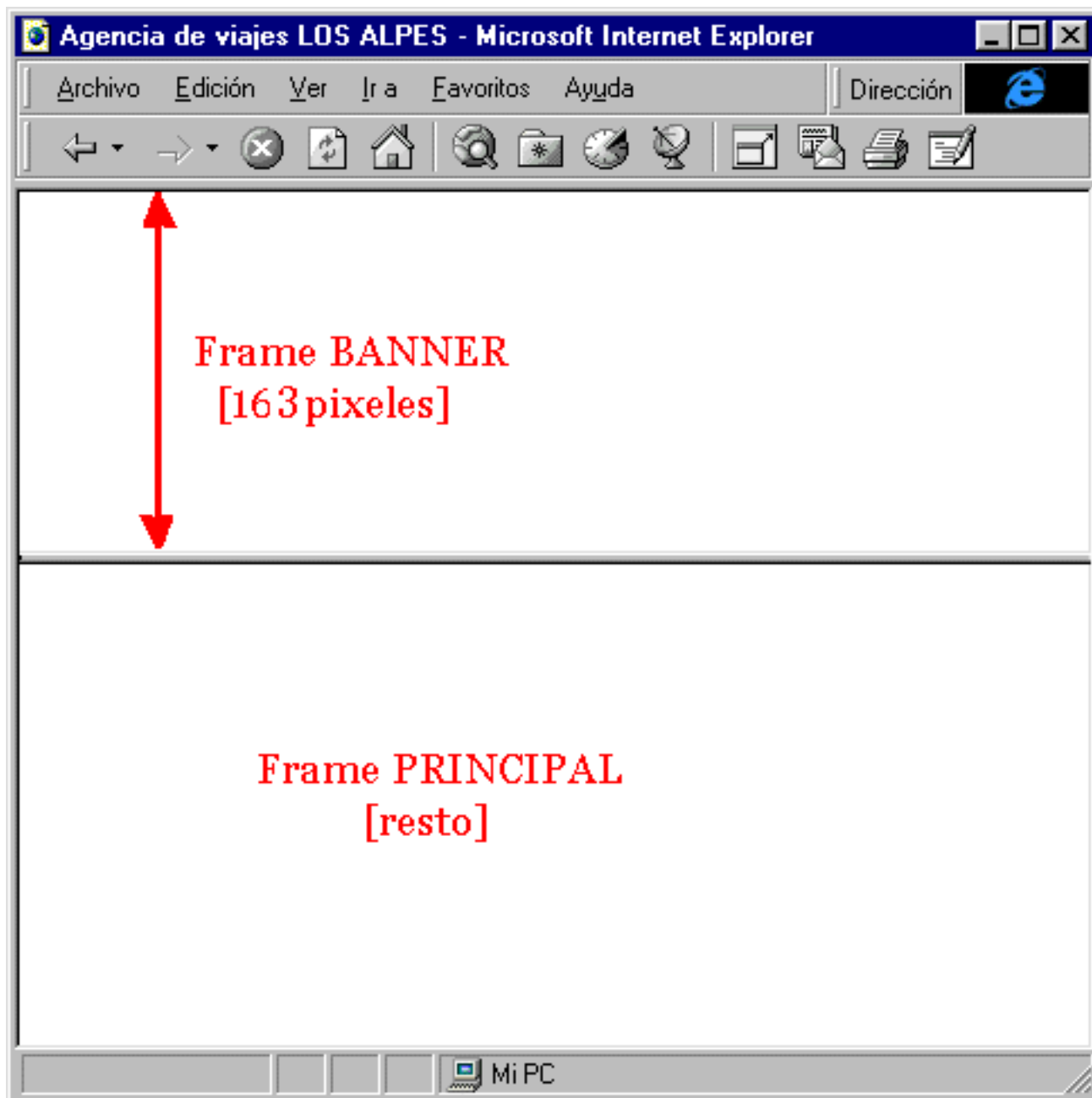
ser modificadas ya que el logo y los botones ya no deben mostrarse. En su momento veremos que cambios son necesarios. Por ahora empezaremos con el documento de definición de frames. Este será ahora el archivo inicial de nuestro sitio Web por lo que debe llamarse `index.htm`. El documento HTML con el contenido principal deberá ser renombrado. En esta ocasión hemos elegido darle el nombre `main.htm`.

Debemos dividir la ventana en dos filas. La fila superior debe tener una altura mínima de 163 pixeles para que quepan las imágenes que queremos incluir. La fila inferior ocupará el resto del espacio. El código que define esta estructura es por tanto:

```
<FRAMESET ROWS="163,*">  
... contenido frames ...  
</FRAMESET>
```

Que nos divide la ventana en dos tal y como vemos en la figura 8.26.

Figura 176. El frame superior será de 163 pixeles, mientras que el inferior ocupará el resto del espacio de la ventana del navegador.



El contenido del primero de los frames será el archivo banner.htm. El contenido del segundo será el archivo main.htm.

Además de insertar el contenido de los frames debemos incluir un código alternativo para los navegadores que no soporten frames en el que mostraremos un texto acompañado de un enlace a una versión de las páginas sin frames. Para crear esta segunda versión creamos un directorio de nombre noframes y copiamos a él todos los documentos de la práctica 4. Con todo esto el código completo queda:

```
<HTML>
<HEAD>
<TITLE>Agencia de viajes LOS ALPES</TITLE>
</HEAD>
<FRAMESET ROWS="163,*">
<FRAME SRC="banner.htm" NAME="banner" MARGINWIDTH="0"
MARGINHEIGHT="0">
<FRAME SRC="main.htm" NAME="principal">
</FRAMESET>
<NOFRAMES>
<BODY>
Esta página tiene frames y su navegador no los
soporta. Por favor, visite nuestras páginas especialmente
preparadas para navegadores
<A HREF="noframes/index.htm">sin frames</A>.
</BODY>
</NOFRAMES>
</HTML>
```

Un aspecto que es necesario comentar de este código es que hemos usado los atributos `MARGINWIDTH` y `MARGINHEIGHT` dándoles el valor cero en el frame superior. Esta es una práctica muy habitual en los frames que contengan imágenes ya al eliminar los márgenes podremos reducir el tamaño del frame.

Sugerencia

En un capítulo anterior aprendimos a usar la etiqueta `META`, para cargar automáticamente otra página (o refrescar la misma) pasados unos segundos. Para ello debíamos incluir en la cabecera del documento HTML el siguiente código:

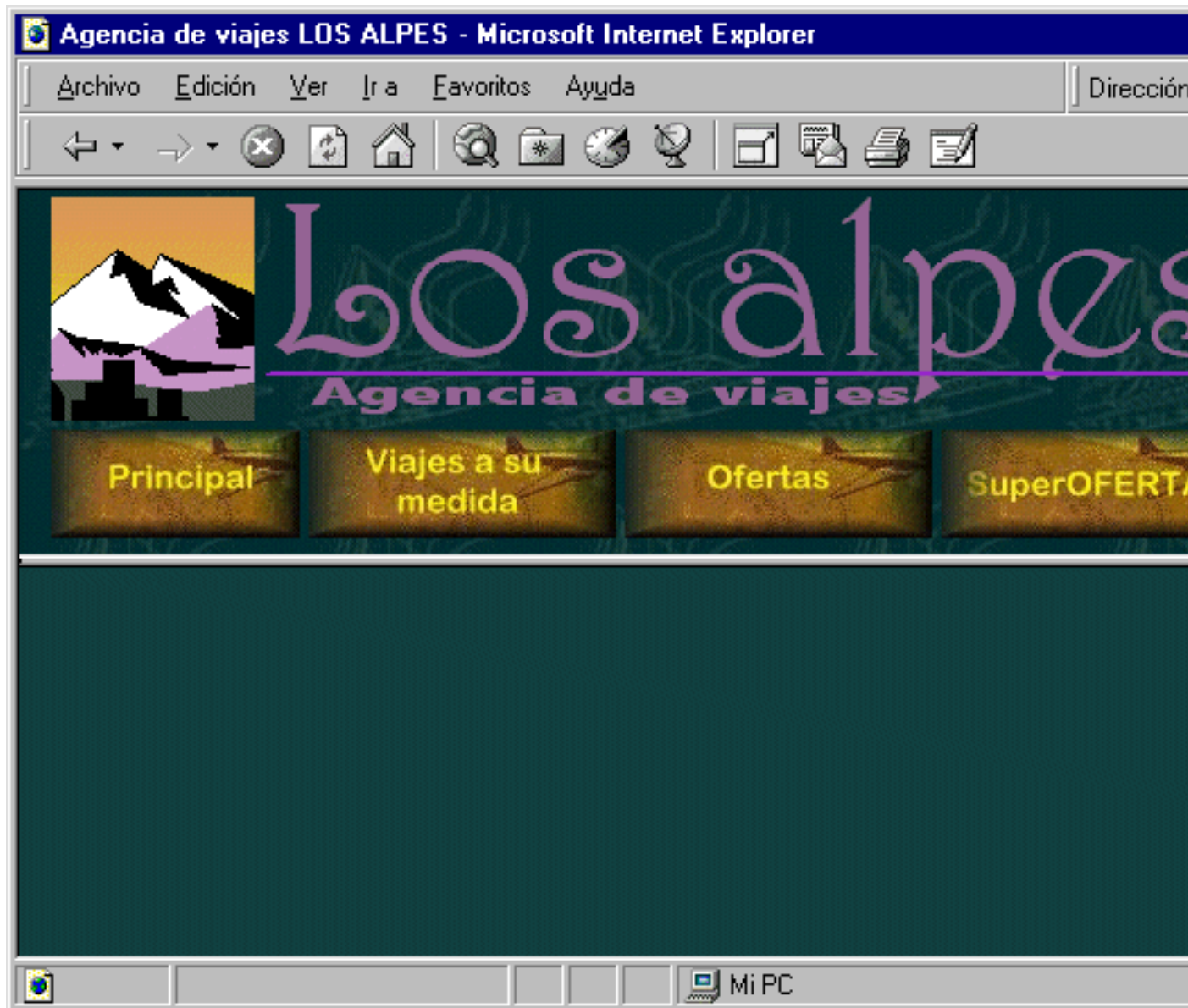
```
<META HTTP-EQUIV="refresh" CONTENT="5;URL=otrapagina.htm">
```

Podemos usar esta técnica para redirigir a los usuarios de navegadores sin soporte para frames a otras páginas que no los tengan pasados 5 segundos.

El frame Banner

El contenido de este frame debe estar en un archivo que hemos decidido llamar `banner.htm`, y el siguiente paso en nuestro proyecto será escribir su código. Para ello no tenemos más que copiar las primeras líneas del archivo `mes.htm` (en realidad vale cualquiera menos `main.htm`). En estas primeras líneas se incluía el código para insertar el logo de Los Alpes y los botones que nos dirigían a las distintas secciones del sitio Web. Nuestro objetivo será que todo ello sea mostrado en este frame para que permanezca siempre visible y se muestre como en la figura 8.27 independientemente del contenido del frame inferior.

Figura 177. El contenido de banner.htm, el logo y encabezado de Los Alpes así como los botones de navegación, serán mostrados en el frame superior.



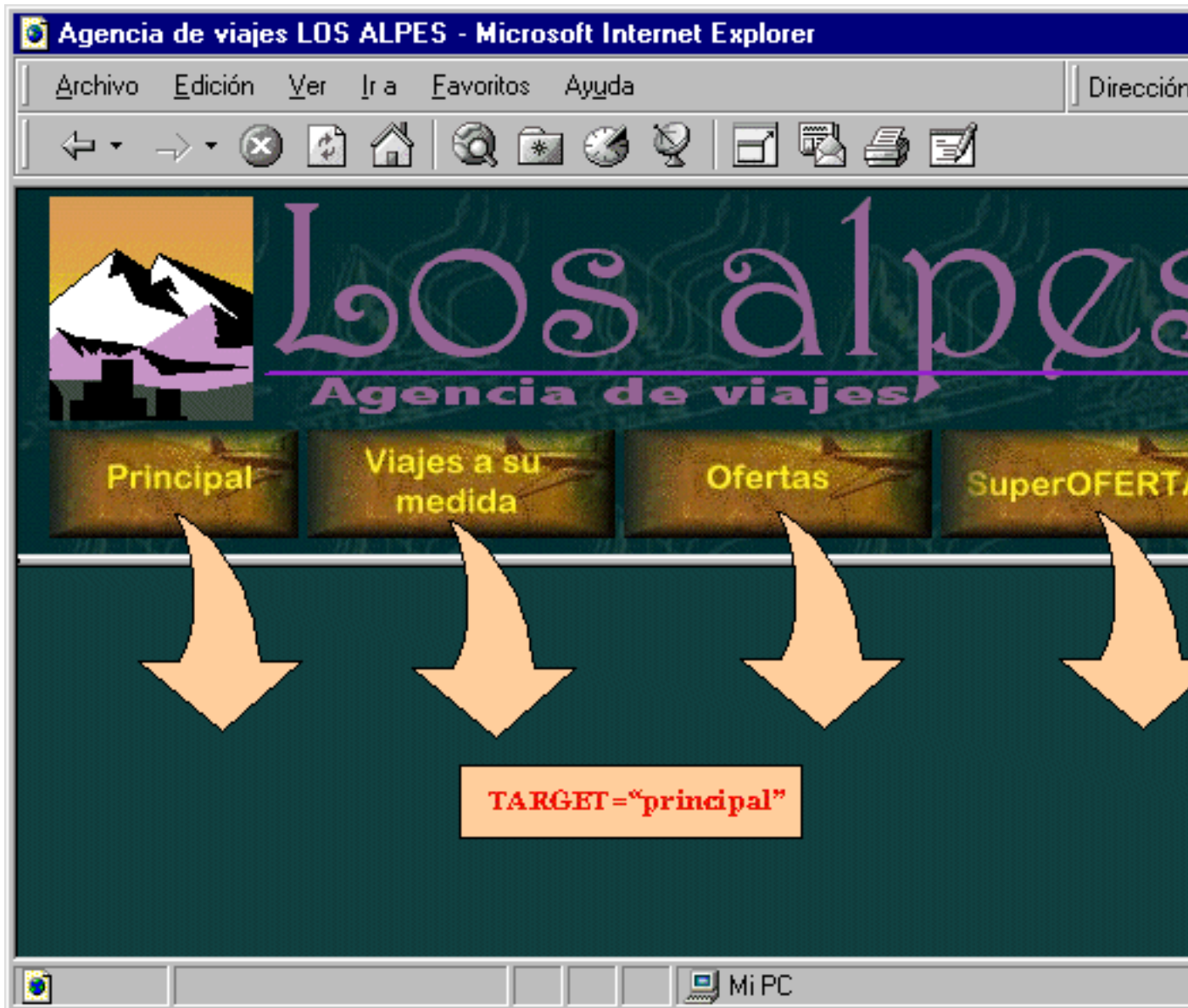
Este es el momento de tener en cuenta un detalle importante de la creación de toda página con frames. En este documento hay una serie de enlaces (los botones) y debemos elegir cual será el destino donde se mostrará las páginas a las que nos conducen dichos enlaces. Si no especificamos ningún destino cuando pulsemos, por ejemplo, sobre el botón OFERTAS, el documento mes.htm sustituirá a banner.htm en el frame superior. Esto no es lo que queremos ya que precisamente el objetivo de este frame era mantenerse siempre visible. Queda claro por tanto que el mes.htm debe mostrarse en el frame principal. Ya sabemos como hacerlo, no hay más que usar el atributo TARGET en la etiqueta que definía el enlace:

```
<A HREF="mes.htm" TARGET="principal">
<IMG SRC="img/boton2.jpg" WIDTH=136 HEIGHT=48
BORDER=0 ALT="Ofertas del Mes">
```


El siguiente paso sería añadir el atributo `TARGET` en el resto de enlaces de `banner.htm` ya que todos ellos deben mostrar el resultado en el frame principal (figura 8.28). Sin embargo, si recordamos, en estos casos podíamos usar la etiqueta `BASE` en la cabecera del documento para indicar cual es el frame destino de todos los enlaces del documento. El código que debe emplearse es:

```
<BASE TARGET=principal">
```

Figura 178. Todos los botones tienen como destino el frame principal. Por tanto es conveniente usar la etiqueta `BASE` con el atributo `TARGET=principal`.



Es importante recordar que la etiqueta `BASE` debía incluirse en la cabecera del documento, entre las instrucciones `<<HEAD>>` y `</>`.

El código completo del archivo `banner.htm` es:

```
<HTML>

<HEAD>
<TITLE>LOS ALPES - Encabezado</TITLE>
<BASE TARGET="principal">
</HEAD>

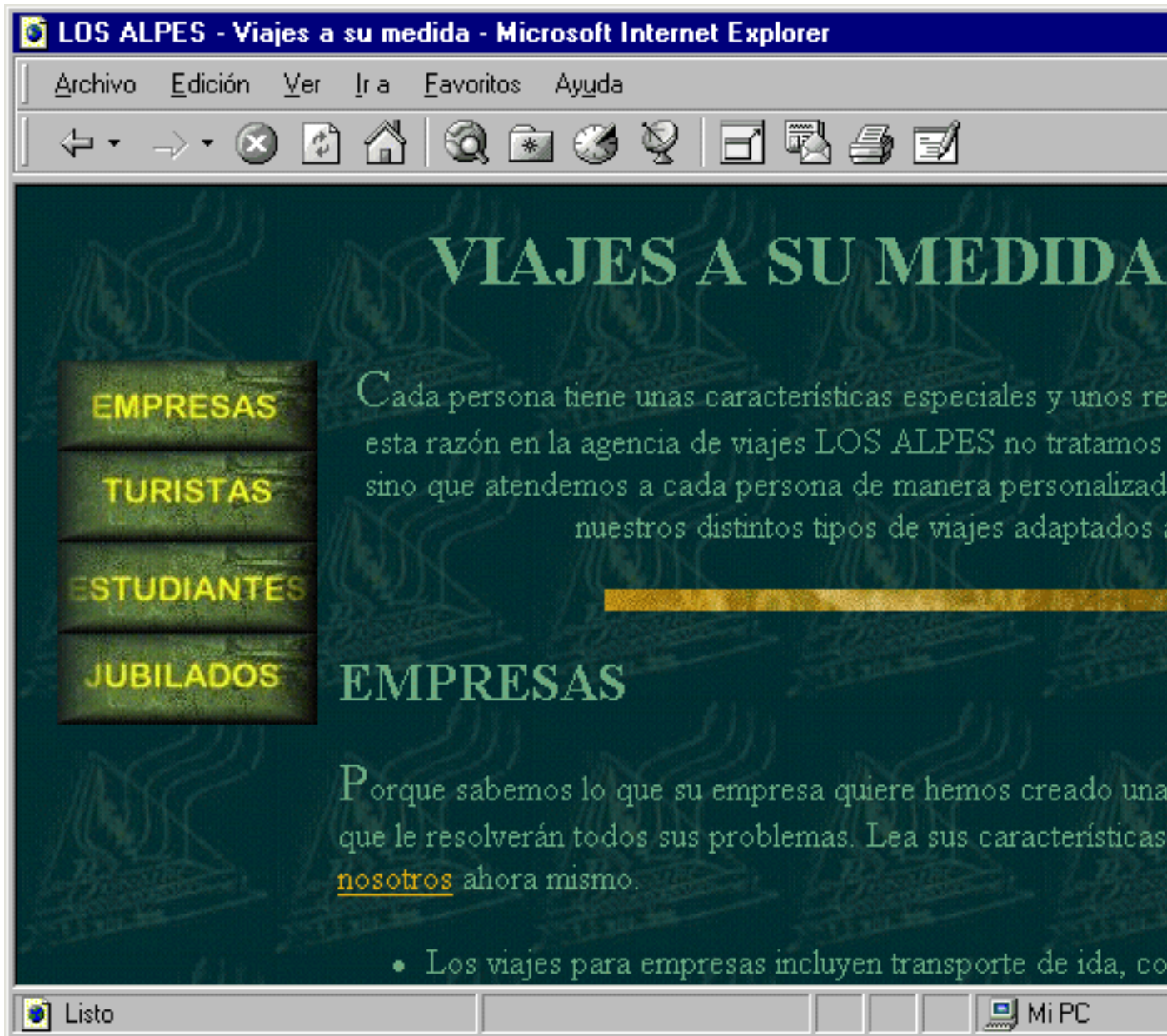
<BODY LINK="#FED354" VLINK="#EDB301" ALINK="#FFFF00"
BGCOLOR="#335142" TEXT="#71AC84"
BACKGROUND="img/tratilea.jpg">
<A NAME="arriba"></A>
<CENTER>
<TABLE ALIGN="center" border="0">
<TR>
<TD><IMG SRC="img/logo.gif" WIDTH=90 HEIGHT=99 BORDER=0
ALT=" [LOGO] "></TD>
<TD><IMG SRC="img/losalpes.gif" WIDTH=440
HEIGHT=99 BORDER=0 ALT="LOS ALPES - Agencia de
Viajes"></TD>
</TR>
<TR>
<TD COLSPAN="2">
<A HREF="main.htm"><IMG SRC="img/boton0.jpg"
WIDTH=110 HEIGHT=48 BORDER=0 ALT="Página
principal"></A>
<A HREF="medida.htm"><IMG SRC="img/boton1.jpg"
WIDTH=136 HEIGHT=48 BORDER=0 ALT="Viajes a su
medida"></A>
<A HREF="mes.htm"><IMG SRC="img/boton2.jpg"
WIDTH=136 HEIGHT=48 BORDER=0 ALT="Ofertas del Mes"></A>
<A HREF="superofe.htm"><IMG SRC="img/boton3.jpg"
WIDTH=136 HEIGHT=48 BORDER=0 ALT="La
superOFERTA"></A></TD>
</TR>
</TABLE>
</CENTER>
</BODY>
</HTML>
```

Frame principal

El frame inferior será nuestro frame principal, su contenido irá cambiando cuando el visitante navegue por nuestras páginas, mientras que el frame con el banner se mantendrá siempre igual. Este contenido será el de los documentos `mes.htm`, `superofe.htm`, `medida.htm` y `main.htm`. La manera de pasar de unos a otros será pulsar sobre los botones del frame banner.

Estas páginas ya existen y están creadas, sin embargo todas ellas contienen una cabecera con el título y la navegación. Estos elementos ya no son necesarios, dado que se encuentran en el frame superior. Por ello será necesario editar los archivos y quitar esa parte de cada uno. En la figura 8.29 se muestra el documento `mes.htm` una vez se le ha quitado la cabecera.

Figura 179. Página sin cabecera, lista para ser insertada en un frame.



Últimos retoques

El último paso en la creación del sitio Web de los Alpes en su versión con frames consistirá en dar los últimos retoques. Suele ser aconsejable asegurarnos que todo lo anterior funciona antes de realizarlos.

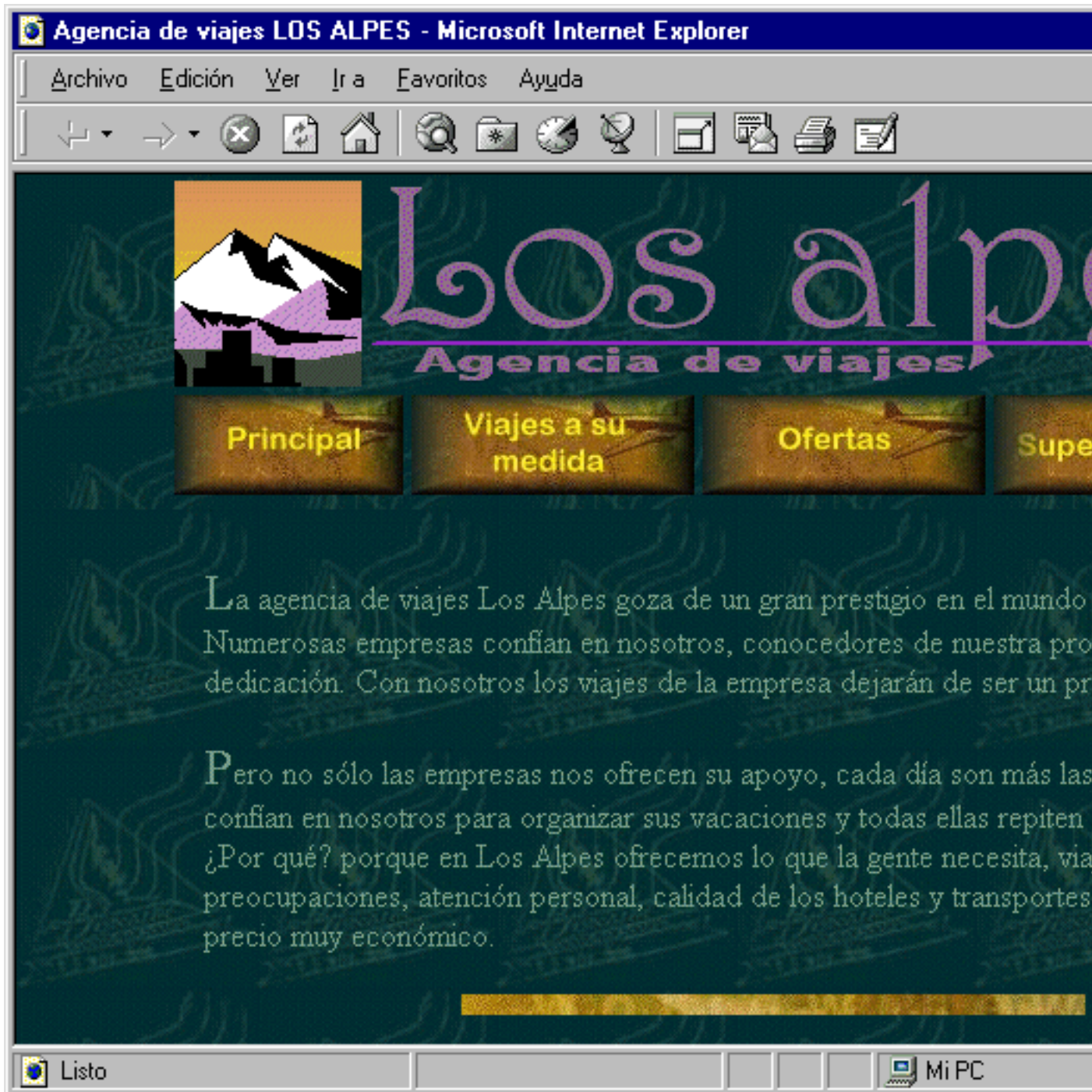
Los retoques abarcan redimensionar los tamaños de los frames para adaptarlo al contenido final que se ha incluido en ellos, cambiar el fondo de algunos de ellos para que contrasten mejor, incluir márgenes para los párrafos, eliminar los bordes, etc. Nosotros nos centraremos en estas dos últimas modificaciones.

Para eliminar el borde debemos editar de nuevo el archivo `index.htm` e incluir los atributos necesarios para hacer los bordes invisibles. En un principio, y para cumplir el estándar incluimos `FRAMEBORDER="0"` en las dos etiquetas `FRAME` que tenía el documento. Posteriormente, para asegurar compatibilidad con todos los navegadores incluimos además: `FRAMEBORDER="0"`, `BORDER="0"` y `FRAMESPACING="0"` en la etiqueta `FRAMESET`.

Por otro lado editaremos el archivo `main.htm` que contenía el texto de bienvenida al sitio Web y daremos formato a dicho párrafo incluyendo márgenes a la izquierda y a la derecha suyo. Para crear los márgenes usaremos la técnica comentada en el capítulo dedicado a las tablas. Crearemos una de tres columnas. La izquierda estará vacía (contendrá un espacio vacío, ` `) y constituirá un margen con un ancho igual al 10% de la pantalla. La columna derecha, igualmente vacía, tendrá un ancho del 5% y constituye el margen derecho. En la columna central irá, por supuesto el texto, y ocupará el resto del espacio.

La práctica ha terminado. El resultado final se distribuye junto con este curso en un archivo comprimido: `practica5.tgz` [`extra/practica5.tgz`]. El resultado final se puede apreciar en la figura 8.30.

Figura 180. Estructura de frames mostrando la página main.htm en el frame central.



Frames flotantes

Los frames flotantes son muy parecidos a los que hemos visto hasta ahora, siendo su principal diferencia que pueden ser insertados en cualquier lugar de una página web, y su función ya no será dividir la ventana del

navegador en varias partes, sino insertar en medio de un documento HTML el contenido de otro en un área del tamaño que indiquemos.

Nota

Los frames flotantes fueron originalmente una extensión de Internet Explorer™ que posteriormente pasó a formar parte de el estándar HTML 4.0. Casi todos los navegadores usados actualmente lo soportan con la excepción de Netscape Navigator 4.x™.

Para crear frames flotantes debemos usar la etiqueta `IFRAME` que significa *Inline Frame* por su similitud con las imágenes inline que se insertan en cualquier lugar de la página con la etiqueta `IMG`. La etiqueta `IFRAME` es un contenedor, esto es, está formada por una instrucción de inicio, `<<IFRAME>>`, donde se especifican todos las características del frame por medio de atributos, y una instrucción de fin, `</>/IFRAME>>`. Entre estas dos instrucciones se puede incluir código alternativo para aquellos navegadores que no soporten los frames flotantes. Los que si lo hagan ignorarán dicho código. Dicho esto ya podemos escribir el código (de manera general) necesario para insertar un frame flotante:

```
<IFRAME ...>
...Código alternativo...
</IFRAME>
```

Este código puede ser insertado en cualquier parte del cuerpo de una página. Esto quiere decir que para insertar un frame flotante no es necesaria ser una página especial para poder insertar frames. También es importante destacar que podemos insertar tantos frames flotantes como queramos.

Atributos de los frames flotantes

En el código anterior incluimos tres puntos suspensivos en el interior de la instrucción `<FRAME>`, estos representan los atributos que debemos usar para indicar el documento HTML donde se encuentra el contenido que queremos insertar en el frame flotante, el tamaño de éste, el alineamiento y otras características variadas.

El atributo `SRC` es de uso obligatorio y toma como valor la dirección URL, absoluta o relativa, del documento HTML que servirá como contenido. Conocido este atributo ya podemos escribir un primer ejemplo de una página web con un frame flotante:

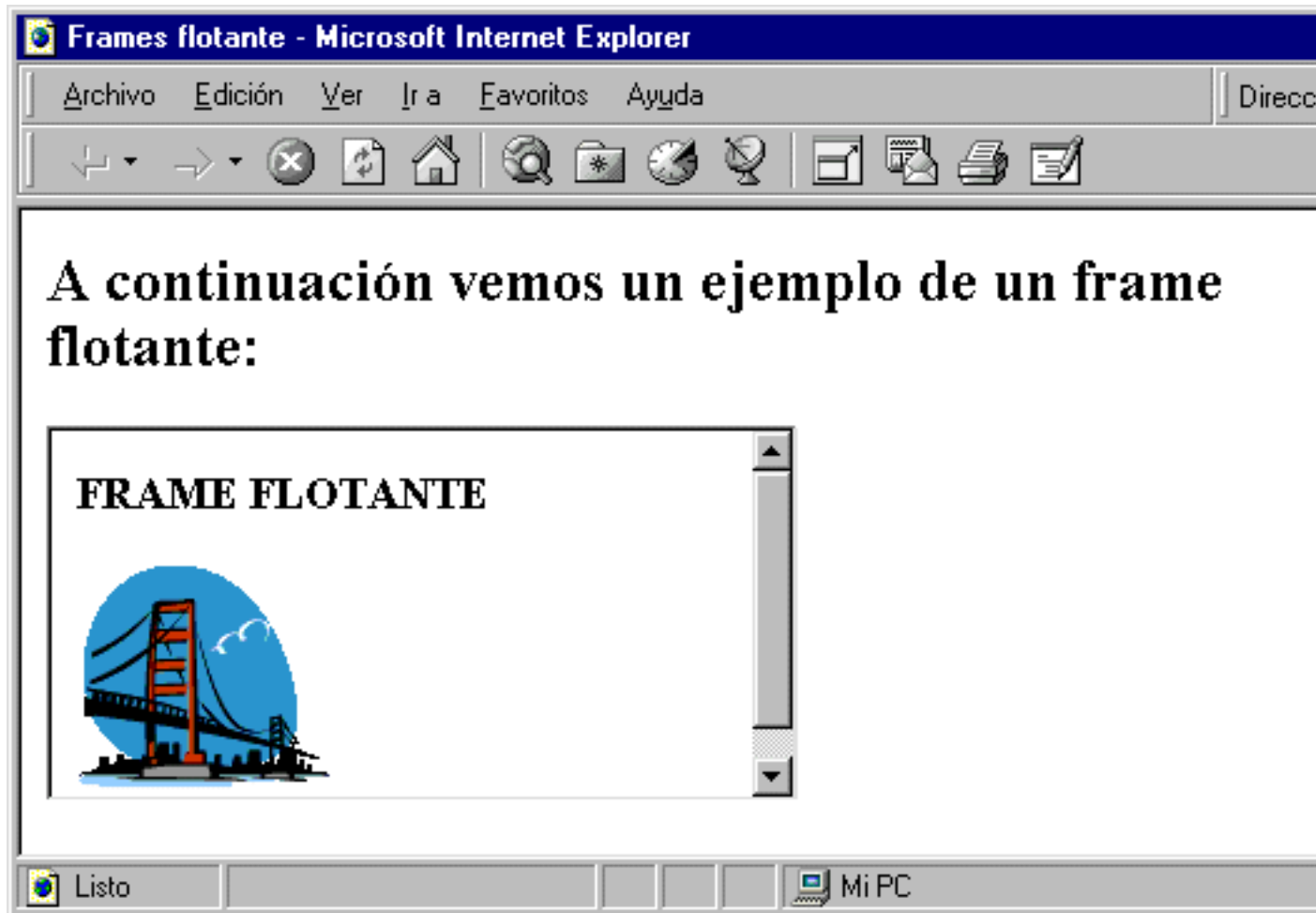
```
<HTML>
<HEAD>
<TITLE>Frames flotante</TITLE>
</HEAD>

<BODY>
<H2>A continuación vemos un ejemplo de un
frame flotante:</H2>
<IFRAME SRC="doc.htm">
Tu navegador no soporta frames flotantes, para ver
el contenido ve a <A HREF="doc.htm">doc.htm</A>
</IFRAME>
</BODY>

</HTML>
```

A continuación guardamos este archivo con el nombre `iframe.htm`, por ejemplo, y creamos el archivo `doc.htm` con un código HTML sencillo (nosotros hemos incluido un poco de texto, una imagen y un fondo azul) y visualizamos `iframe.htm` con Explorer obtendremos el resultado de la figura 8.31.

Figura 181. Los frames flotantes nos permiten insertar un documento HTML completo en una página Web.



Unas barras de desplazamiento delatan la presencia de un frame. Un aspecto a destacar de esta figura es que Explorer asigna un tamaño por defecto a dicho frame en el que mostrará todo el contenido del documento HTML que le hayamos indicado independientemente de lo largo que este sea.

Si esta misma página es visualizada por un navegador sin soporte para frames flotantes, por ejemplo Netscape Navigator 4.0, se mostraría el código que hemos introducido entre la instrucción de inicio y la de fin, como vemos en la figura 8.32.

Figura 182. Los navegadores que no soporten frames flotantes mostrarán el código alternativo incluido entre las instrucciones `<<IFRAME>>` y `</<IFRAME>>`



Además del atributo SRC la etiqueta IFRAME tiene los siguientes:

NAME	Este atributo nos permitirá asignarle un nombre al frame de manera que podamos referirnos a él por ejemplo para usarle como destino de un enlace usando el atributo TARGET.
WIDTH y HEIGHT	En la figura 8.30 hemos visto que Internet Explorer asignaba un tamaño por defecto al frame flotante. Usando estos dos atributos podremos indicar la anchura y altura respectivamente que no queramos. El valor dado será interpretado como un número en pixeles. Si en el código usado para obtener dicha figura añadimos un tamaño lo suficientemente grande desaparecerá la barra de desplazamiento.
FRAMEBORDER	Este atributo funciona de igual manera que en los frames normales: sirve para quitar y poner los bordes. Escribiendo FRAMEBORDER="0" en el ejemplo realizado anteriormente obtenemos el resultado de la figura 8.33, en la que vemos que el borde del frame ha desaparecido. Sin embargo no hemos ampliado el tamaño del frame con los atributos WIDTH y HEIGHT con lo que el documento es más grande que el espacio asignado y la barra de desplazamiento sigue ahí.
SCROLLING	Este atributo nos permitirá quitar o mostrar la barra de desplazamiento. Puede tomar tres valores: value Es el valor por defecto e indica que la barra de desplazamiento sólo será mostrada si es necesario (es decir si el documento HTML completo no cabe en el espacio reservado al frame)

yes significa que las barras de desplazamiento deben mostrarse siempre
no indica que no se mostrarán nunca.

HSPACE y MARGINWIDTH Al igual que ocurría con las imágenes se puede especificar un margen alrededor del frame flotante. Con estos atributos concretos indicaremos la distancia mínima en píxeles que debe haber entre el frame y cualquier otro elemento que esté a su derecha o izquierda (margen horizontal). Es decir, sirve para especificar la anchura del margen. La única diferencia entre ambos atributos es que MARGINWIDTH es el nombre aceptado por el estándar y HSPACE está considerado como una extensión de Explorer.

VSPACE y MARGINHEIGHT Con estos atributos indicaremos la altura en píxeles del margen del frame. Al igual que antes MARGINHEIGHT es el nombre aceptado por el estándar y VSPACE está considerado como una extensión de Explorer.

ALIGN Permite alinear el frame respecto a la ventana del navegador y respecto al resto de elementos de la página. De nuevo, este atributo funciona exactamente igual que con las imágenes y puede tomar los mismos valores: left, middle, right, top y bottom.

En la figura 8.34 podemos ver un ejemplo con dos frames flotantes uno alineado a la izquierda (ALIGN="left") y el otro a la derecha (ALIGN="right"). En el frame izquierdo hemos incluido un margen de 100 píxeles (MARGINWIDTH=100) y en el derecho hemos quitado la barra de desplazamiento (SCROLLING=no).

BORDER Este atributo es una extensión de Explorer y tiene la misma función que el atributo del mismo nombre de la etiqueta FRAME, indicar el grosor en píxeles del borde que delimita el frame.

BORDERCOLOR Con este atributo podremos cambiar el color del borde. Es, al igual que el anterior, una extensión de Explorer no aceptada por el estándar HTML 4.0.

Figura 183. Usando `FRAMTBORDER=0` podemos eliminar el borde del frame. Sin embargo si el contenido no cabe en frame la barra de desplazamiento sí se mostrará.

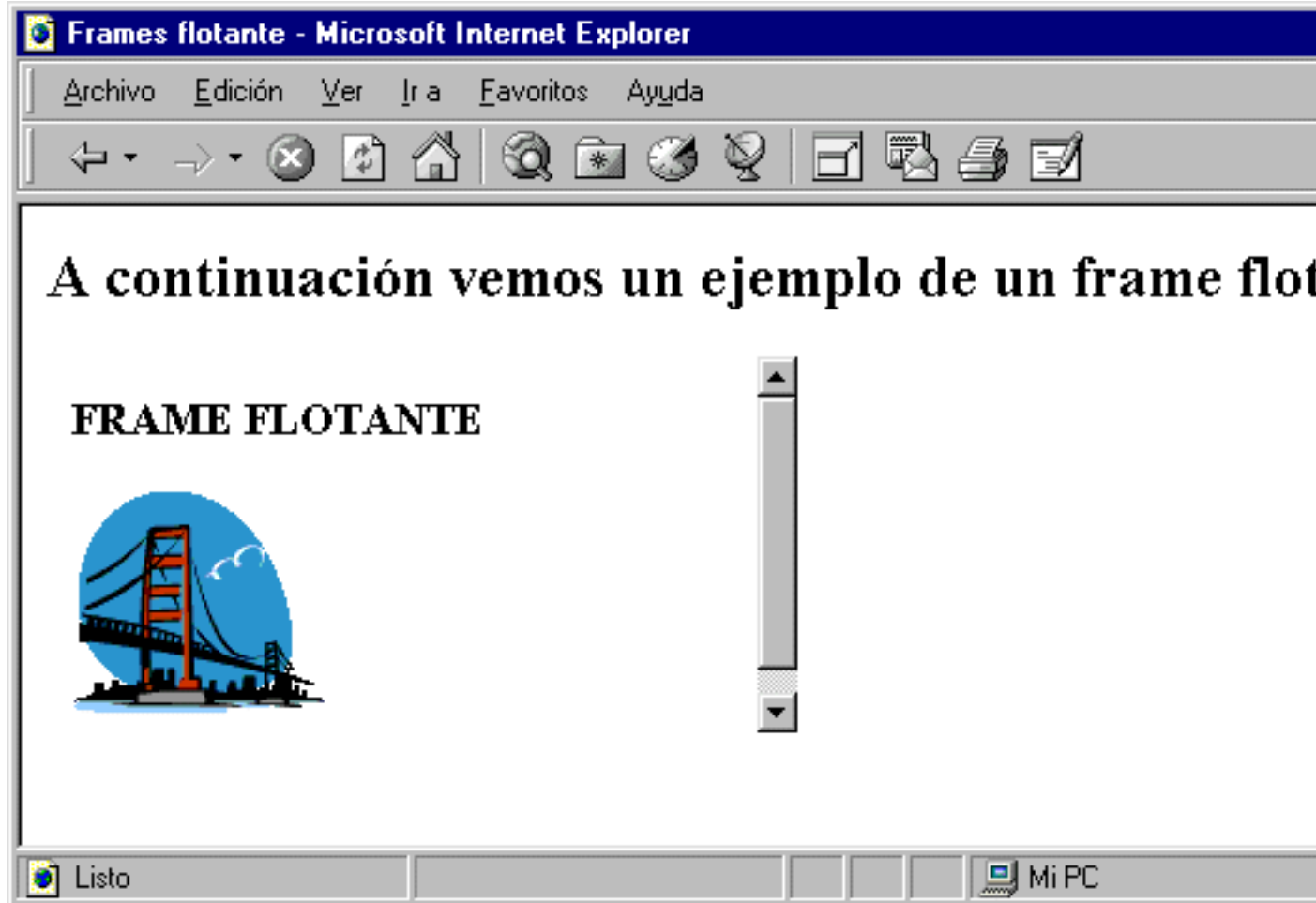
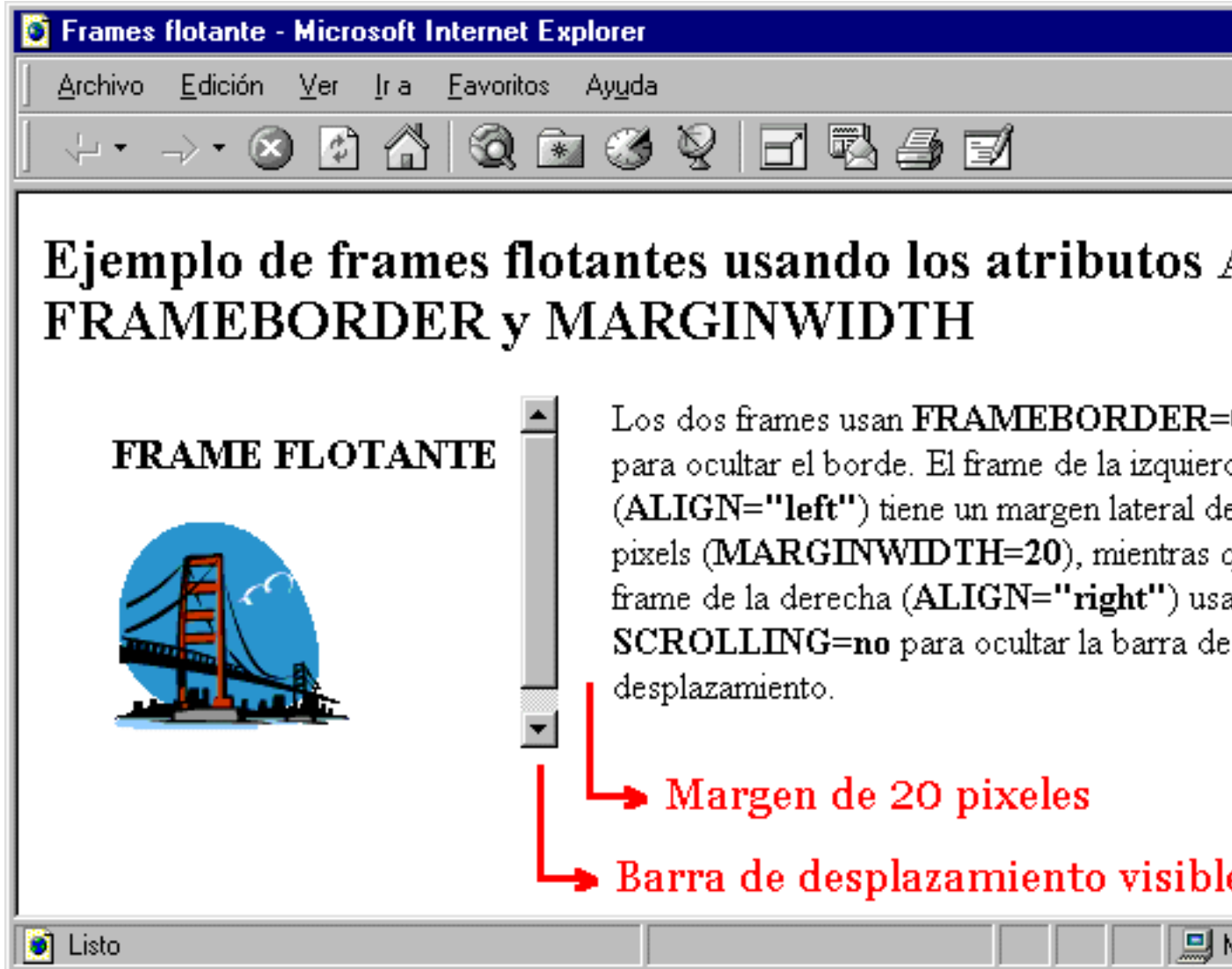


Figura 184. Con los atributos `ALIGN`, `MARGINWIDTH` y `SCROLL` podremos controlar las diversas características de los frames flotantes.



Con esto terminamos el aprendizaje del uso de frames en páginas web. En el próximo capítulo del curso nos adentraremos en el mundo de los sonidos, vídeos y otros contenidos multimedia en el World Wide Web.

Vídeos, Sonido y Multimedia

Jorge Ferrer

Víctor García

Versión 1.0

Copyright Jorge Ferrer y Victor García. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation. Puede consultar una copia de la licencia en: <http://www.gnu.org/copyleft/fdl.html>

Para más detalles consultenos a través de la dirección jferrer@acm.org

Animaciones, vídeos, música, sonidos, y mucho más son los nuevos contenidos que encontramos cada vez más a menudo en el World Wide Web. Todo con los objetivos de hacer un web más atractivo o emplearlo como nuevo medio de distribución de arte. Sin embargo es importante tener cuidado para no hacer páginas web que sólo puedan ser vistas en determinados navegadores.

En los últimos años la informática ha conocido lo que se ha convenido a llamar la revolución multimedia, cada vez es más fácil encontrar CD-ROMs con multitud de imágenes, vídeos, animaciones tridimensionales, juegos con una calidad gráfica y sonora impactante y mucho más. Esta revolución también ha salpicado a Internet y en concreto al World Wide Web. Sin embargo los contenidos multimedia dejan de ser divertidos cuando tenemos en cuenta las velocidades actuales de la Red de Redes y los problemas de incompatibilidad entre distintas plataformas (distintos ordenadores y/o sistemas operativos). El gran potencial del contenido multimedia en el WWW consiste en el gran interés que provoca, pero por en los que a Internet se refiere debe limitarse a pequeños ficheros de sonido o vídeo. Afortunadamente con un poco de ingenio esto no será problema para crear efectos muy atractivos.

Cuando hablamos de multimedia en páginas Web debemos diferenciar entre los contenidos multimedia externos y los contenidos denominados *inline*. Esta diferenciación es muy similar a la que hicimos con las imágenes, con el término *inline* nos referimos a todos los elementos que se incluyen en la misma página Web, junto con el texto, listas, encabezados, etc. Las imágenes o ficheros multimedia externos se enlazarán a esta página de manera que serán bajados cuando los navegantes seleccionen el enlace apropiado. La gran pregunta es ¿cuál debo usar?. Para responder a esta pregunta veamos las principales ventajas de cada uno:

Multimedia inline	Su gran ventaja es que aparece directamente junto con el resto de elementos de la página y esto es mucho más cómodo para los navegantes.
Multimedia externa	Nos permitirá incluir muchos más formatos de sonido, vídeo y cualquier otro fichero multimedia, ya que si el navegador no lo entiende llamará a alguna aplicación que le ayude. Estas aplicaciones auxiliares pueden ser configuradas por el usuario del navegador.

La conclusión puede ser que en general podemos incluir contenido multimedia *inline* siempre y cuando esté en un formato entendido por los navegadores y no ocupe demasiado tiempo de carga (es decir que su tamaño en bytes sea pequeño). En caso de ser un archivo grande o de un formato no soportado por los navegadores debemos usar multimedia externa.

Nota

Es una norma de ética entre los creadores de páginas del WWW incluir tanto el formato de los archivos como su tamaño en el texto que sirve como enlace o en un texto, por ejemplo: Vídeo QuickTime [.mov] de 200 Kb de la escena del elefante rosa de la película Dumbo. De esta manera el navegante podrá decidir si le merece o no la espera de cargar el archivo.

Pasemos a estudiar cada uno de estos métodos más en detalle.

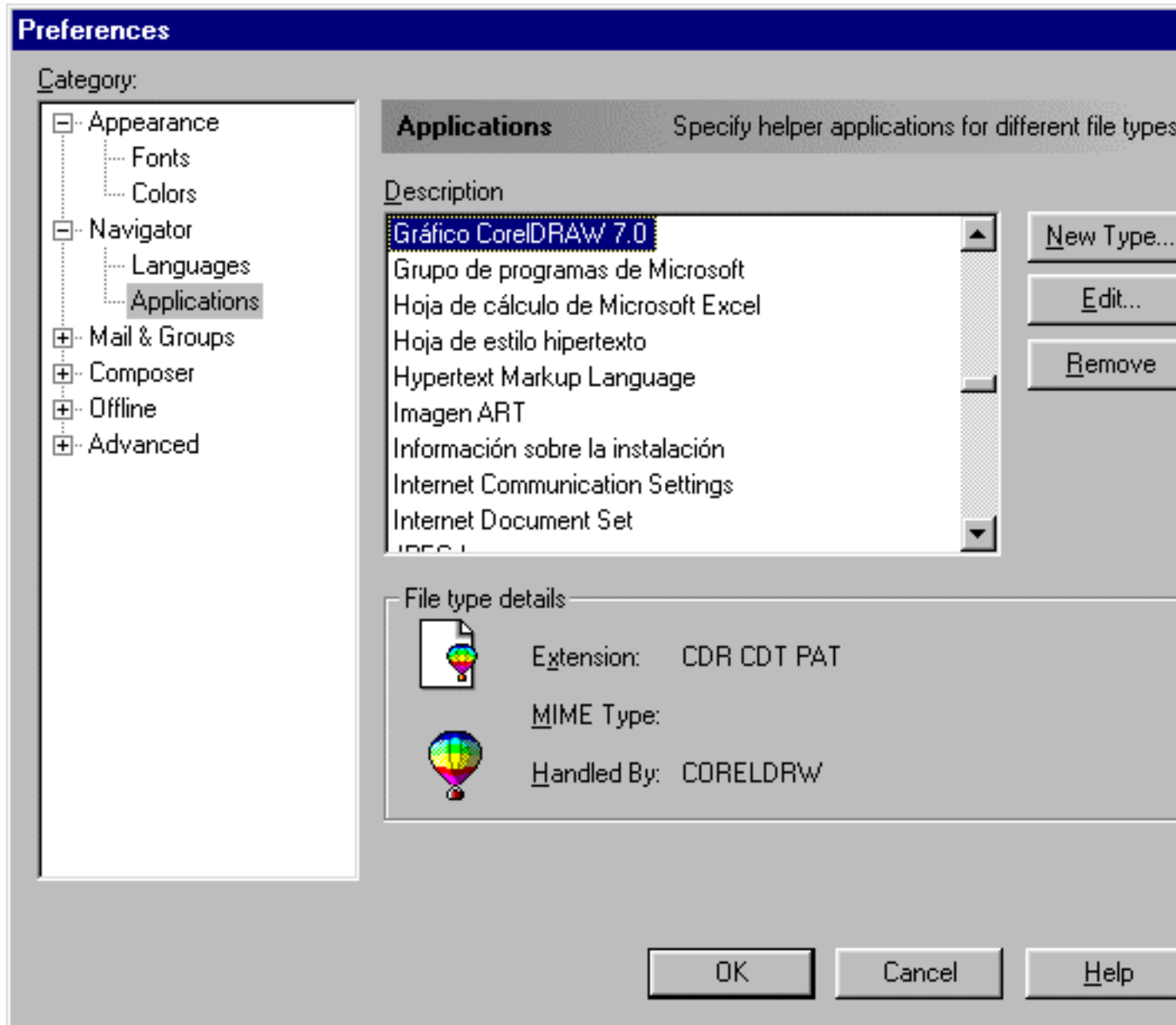
Sonido vídeo y otros ficheros externos

Para enlazar un fichero multimedia externo a una página Web no hay más que crear un enlace a ese documento:

```
<A HREF="fichero_multimedia">  
Un fichero multimedia  
</A>
```

El siguiente paso vendrá por parte del navegante que debe seleccionar dicho enlace (esto significa pulsar sobre el texto en Navegadores gráficos, pero no olvidemos a los demás). Y ¿qué ocurre entonces?. Para algunos tipos de ficheros, como pueden ser imágenes, archivos de texto o, por supuesto, documentos HTML, el propio navegador puede ser capaz de mostrar el contenido en su misma ventana. En muchos otros casos, sin embargo, el navegador bajará el fichero al ordenador del navegante y una vez allí se lo pasará a otro programa que sea capaz de entenderlo. Estos programas reciben el nombre de aplicaciones auxiliares o, en inglés, *helper applications*, y pueden ser configurados por el usuario del navegador para elegir que aplicaciones deben ser usadas para cada uno de los tipos de archivos. En la figura 10.2 podemos ver el cuadro de diálogo de configuración de Netscape Navigator 4.0™, por su parte las últimas versiones de Microsoft Internet Explorer™ usan directamente la configuración de Windows para averiguar que programa debe llamar en cada ocasión:

Figura 10.2. Netscape Navigator™ permite configurar en este cuadro de diálogo las aplicaciones externas asociadas a cada navegador.



Si un navegador no es capaz de averiguar cual es el formato de un fichero mostrará un mensaje preguntando al usuario qué debe hacer: guardar el fichero en el disco duro, elegir un programa que entienda el fichero o alguna otra opción.

Una pregunta que nos podemos hacer es ¿cómo es capaz el navegador de reconocer que tipo de fichero es el que hemos enlazado?. Sin duda debe hacerlo de alguna forma, porque si no, no podría decidir a que aplicación llamar o si debe mostrarlo el mismo. De hecho no existe un solo método, sino dos. Por un lado la *extensión* del fichero: Las páginas HTML deben tener la extensión *.html* o *.htm*, las imágenes GIF deben tener la extensión *.gif*, los ficheros PKZIP deben tener terminación *.zip*, etc. Cuando el navegador lee ficheros directamente del disco duro usa la extensión para determinar de que tipo de fichero se trata.

Cuando el navegador obtiene una página de un servidor Web entra en funcionamiento el segundo de los métodos, que consiste en averiguar el *tipo de fichero MIME* asociado al archivo o *content-type*. Este es un código que usan los servidores Web para indicar que tipo de fichero están mandando. Algunos ejemplos de *content-type* son: text/plain (texto normal), text/html (documento HTML), image/gif (imagen GIF), video/mpeg (vídeo con formato MPEG) o application/msword (Documento de Microsoft Word™). Tanto el navegador como el servidor Web tienen tablas que relacionan las extensiones con el tipo de fichero (content-type) asociado. Y el navegador tiene un dato adicional: la aplicación que debe usar con ese tipo de fichero.

Ficheros de sonido

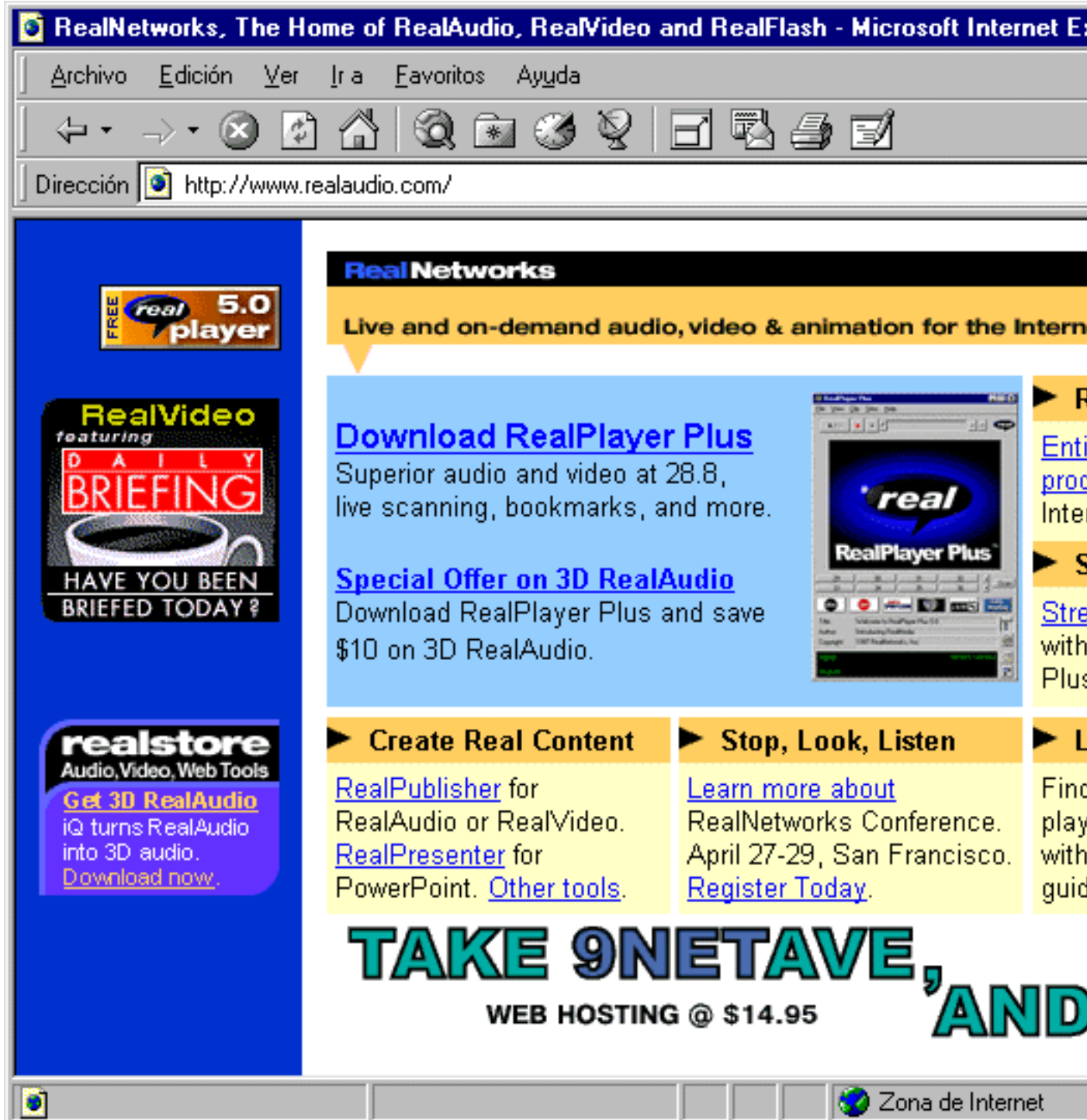
Para incluir un enlace a un fichero de sonido desde una de nuestras páginas Web dicho fichero debe estar un formato apropiado para contener sonido, tal y como ocurría con las imágenes. Al igual que con estas existen multitud de formatos distintos y será nuestra labor elegir el más apropiado. A continuación haremos un repaso de los formatos más importantes.

El formato *AU* creado por Sun Microsystems™ es el único que es soportado por todas las diferentes plataformas de Internet, entre ellas Windows™, UNIX™ y Macintosh™. Este formato permite varios tipos de codificación de sonido, aunque el más popular es la codificación de 8 bits conocida como *u-law* por lo que en ocasiones el formato AU se llama directamente formato *u-law*. Los ficheros AU tienen una calidad aceptable, pero sin ser demasiado buena debido a la codificación de 8 bits que provoca un efecto parecido al del sonido telefónico.

Para conseguir mayor calidad de sonido pueden usarse los formatos *WAVE* (.wav) para Windows, *AIFF* para Macintosh™ o *MPEG sound*. El principal problema de los dos primeros es que sólo funcionan en sus respectivas plataformas y en ninguna otra. El formato MPEG sin estar tan extendido como AU si está presente en multitud de plataformas y puede usarse con mayor libertad.

Por último nos encontramos con el formato *Real Audio* que ha sido desarrollado específicamente para reproducir ficheros de sonido en Internet y en el World Wide Web. En la figura 10.3 podemos ver la página Web creada por RealAudio que ofrece gran cantidad de información así como el software necesario para crear y escuchar este tipo de sonido:

Figura 10.3. RealAudio es un formato de sonido específicamente creado para la red el software necesario para su reproducción puede ser obtenido a través de su Web.



La gran ventaja que ofrecen frente al resto de formatos es que, mientras que en estos es necesario esperar a que llegue todo el fichero para poder escucharlo, Real Audio soporta una técnica conocida como *streaming* que nos permitirá escuchar el sonido según va llegando y sólo una pequeña pausa ocurre al principio antes de empezar a

oírse. El único inconveniente que tiene este formato consiste en que es necesario instalar un servidor especial para poder enviar este tipo de ficheros y enlazarlos a la página requiere una técnica ligeramente distinta que veremos un poco más adelante.

Para que un navegador reconozca el tipo de sonido es necesario que el fichero tenga la extensión adecuada. En la tabla 10.1 podemos ver la extensión asociada a cada uno de los distintos formatos vistos hasta ahora.

Figura tabla 10.1.

Formato de Sonido	Extensión asociada
AU/ μ -law	.au
AIFF/AIFC	.aiff, .aif
WAVE/WAV	.wav
MPEG Audio	.mp2

Una vez tenemos el fichero de sonido con el formato adecuado no tenemos más que escribir el código necesario para incluirlo en la página Web. Por ejemplo:

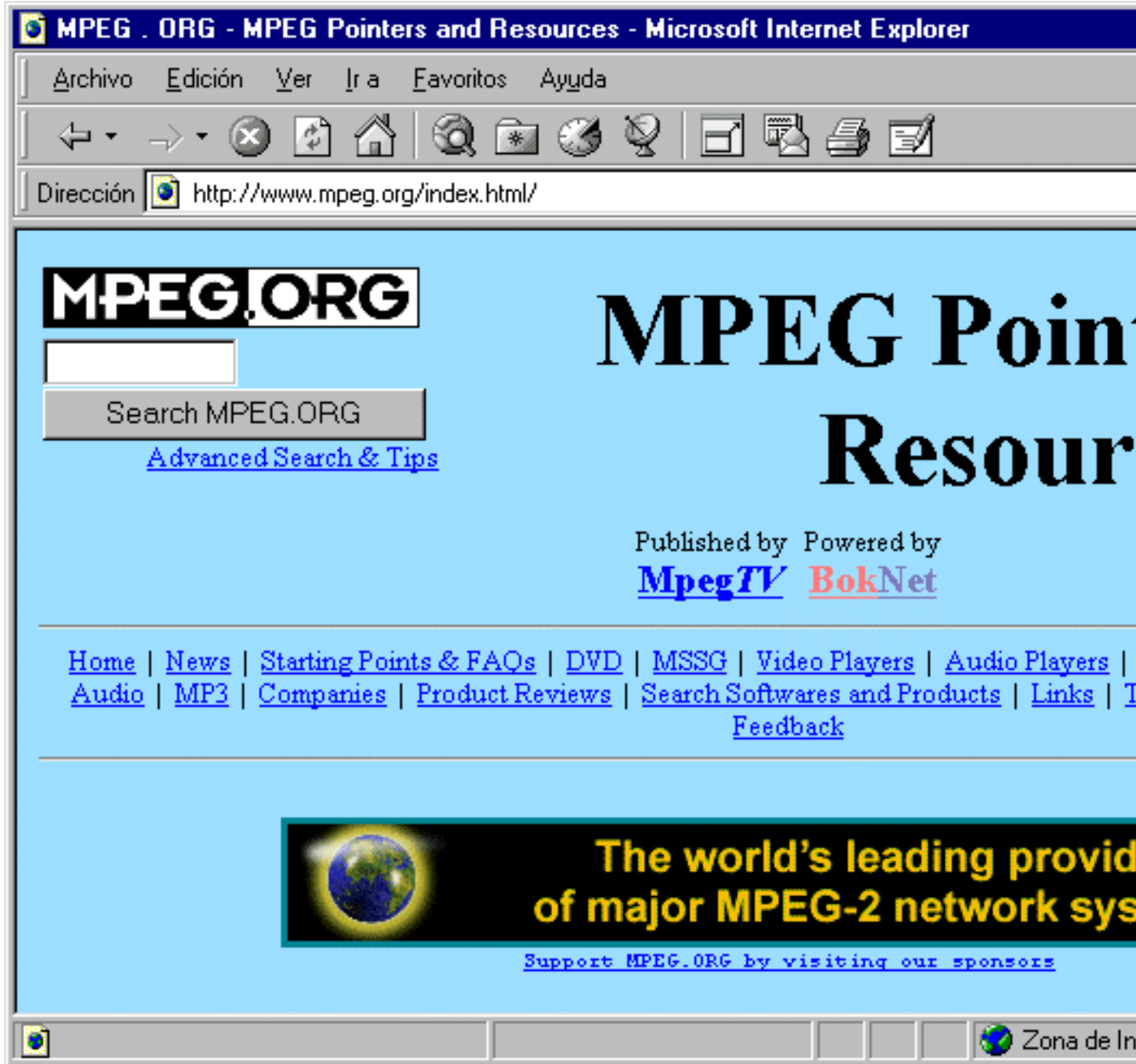
```
<A HREF="sinfonia_5.au">
quinta sinfonía
</A>
```

En este caso hemos usado el formato AU por ser el más extendido. Si queremos mayor calidad debemos usar alguno de los otros tres formatos, pero en ese caso es recomendable ofrecer no sólo un formato sino varios, a través de diferentes enlaces para asegurarnos que todo el mundo tiene acceso al sonido si así lo desea.

Ficheros de vídeo

Al igual que los ficheros de sonido, existen multitud de formatos para codificar vídeo. El estándar actual en el World Wide Web es *MPEG vídeo*, que será entendido en multitud de plataformas. Este formato, al igual que MPEG audio, ha sido creado por una organización de expertos en vídeo y sonido que tienen presencia en Internet en la dirección www.mpeg.org [<http://www.mpeg.org>] (ver figura 10.4).

Figura 10.4. La organización MPEG es la creadora de este estándar que permite la codificación de audio y vídeo de gran calidad en archivos de pequeño tamaño.



Otras opciones son el formato AVI creado por Microsoft™ para su sistema Windows™ y que necesita de la aplicación Vídeo For Windows™ para ser visto o QuickTime™, creado por Apple™ y que últimamente se ha estado extendiendo por el Web gracias a la expansión de diversos visualizadores en diversas plataformas. En la tabla 10.2 vemos las extensiones asociadas a cada uno de estos tres archivos:

Figura tabla 10.2.

Formatos de vídeo	Extensión asociada
MPEG	.mpeg, .mpg
QuickTime	.mov
AVI	.avi

Al igual que ocurría con las imágenes el último paso consistirá en insertar el código HTML:

```
Recién llegado de EEUU
<A HREF="show1.mpeg">
unas escenas
</A>
de la última película de George Lucas.
```

Un truco usado en ocasiones y con el que se consigue un interesante efecto consiste en convertir algún fotograma del vídeo a imagen y hacer que esta misma sea un enlace al vídeo, en este caso no debemos olvidar incluir en una descripción del vídeo en la etiqueta ALT:

```
<A HREF="show1.mpeg">
<IMG SRC="foto02.gif"
ALT="Escenas de la última película de George
Lucas">
</A>
```

Otros tipos de archivos

Los enlaces a archivos externos no están limitados a imágenes y vídeo. Actualmente el World Wide Web está plagada de multitud de otros formatos entre los que destacan los archivos comprimidos ZIP (.zip o .gz [UNIX]), los ejecutables (.exe) y archivos PDF (Portable Document Format) que pueden ser vistos por Acrobat Reader™ que se distribuye gratuitamente. También se encuentran a menudo imágenes en multitud de formatos como Corel Draw™ (.cdr), Adobe Freehand™ o Adobe Photoshop™ (.psd) entre otros.

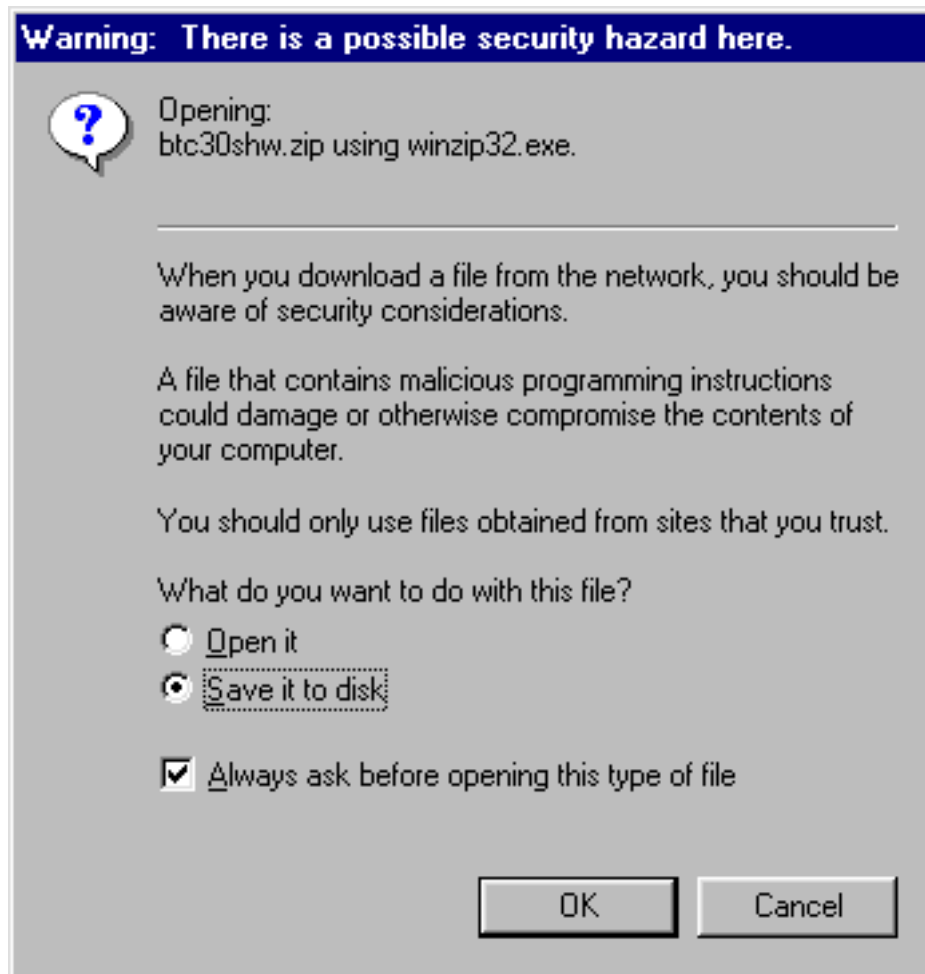
Cuando un enlace a uno de estos ficheros sea pulsado el navegador nos permitirá guardarlo en el disco duro o bien abrirlo usando una aplicación externa si es un archivo de tipo conocido. En la figura 10.5 vemos un caso en el que hemos pulsado sobre un enlace a un archivo con formato ZIP e Internet Explorer™ nos pregunta si deseamos abrirlo o guardarlo en el disco duro:

Figura 10.5. Al pulsar sobre un enlace a un archivo que no puede mostrar él mismo, Internet Explorer™ muestra este cuadro de diálogo preguntando al usuario que quiere hacer.



Netscape Navigator™ actúa de forma prácticamente idéntica, en la figura 10.6 vemos el cuadro de diálogo que muestra la versión 4.0 en inglés:

Figura 10.6. Cuadro de diálogo para archivos externos mostrado por Netscape Navigator™. Si elegimos *open* la aplicación externa asociada será ejecutada.



Multimedia Inline

Hasta hace muy poco el único sistema disponible en el WWW para incluir contenido multimedia era usar archivos externos como acabamos de ver. Sin embargo gracias a las iniciativas de Microsoft™ y Netscape™ están apareciendo nuevas etiquetas y atributos del lenguaje HTML junto con nuevas técnicas que permiten al diseñador de páginas Web incluir dicho contenido en una página.

Vídeo *inline*

Uno de los primeros mecanismos fue introducido en Microsoft Internet Explorer™. Este navegador incluye extensiones a la etiqueta IMG que permiten insertar vídeo con formato AVI. Estas extensiones han sido adoptadas también por el navegador Opera™, pero por ningún otro, y tampoco por el estándar HTML 4.0 pero al ser ignorado por los navegadores que no lo soportan no impide la legibilidad de la página para los usuarios de estos.

La extensión en la que se basa la inclusión de vídeo *inline* es el atributo DYN SRC. Este atributo se usa de manera muy similar al ya conocido SRC, pero ahora el valor deberá ser la dirección de un vídeo en formato AVI en lugar de una imagen. Una de las grandes ventajas de este sistema para insertar vídeo consiste en la posibilidad (de hecho es obligatorio) de usar simultáneamente los dos atributos mencionados, SRC y DYN SRC, de manera que si

el navegador soporta la inclusión de vídeo en la página mostrará el vídeo y en caso contrario mostrará la imagen. Veamos un ejemplo:

```
<IMG DYN SRC="baile.avi"
  SRC="baile.gif"
  ALT=" [Un baile] ">
```

También es posible usar el resto de atributos de la etiqueta `IMG` como `ALIGN`, `VSPACE`, `HSPACE`, `WIDTH`, `HEIGHT`, etc. Pero además de estos, Microsoft™ introdujo otros nuevos como acompañamiento del atributo `DYN SRC` y que nos permitirán tener un mayor control sobre el vídeo, estos son:

- **CONTROLS:** Este es un atributo que no toma ningún valor. Si es incluido, muestra el vídeo AVI junto con una serie de controles para que el usuario pueda parar y reproducir el vídeo a su antojo.
- **LOOP:** Toma como valor un número que determina el número de veces que el vídeo será mostrado. Por ejemplo con `LOOP="5"` el vídeo volverá a reproducirse desde el principio 5 veces y luego parará. Con un valor `LOOP="-1"` el vídeo será repetido infinitas veces.
- **START:** Determina cuando debe empezar a reproducirse el vídeo. Puede tomar dos valores: con `START="FILEOPEN"` (valor por defecto) el vídeo se reproducirá tan pronto como la página y el fichero de vídeo lleguen al ordenador; con `START="MOUSEOVER"` el vídeo no empezará a reproducirse hasta que el usuario sitúe el cursor sobre él.
- **LOOPDELAY:** [Solo Internet Explorer 4.0™] Con este atributo podemos especificar el retardo en milisegundos que el navegador esperará tras acabar antes de empezar a reproducir el vídeo de nuevo.

El atributo `LOWSRC`

Este atributo es una extensión de Netscape™ a la etiqueta `IMG`. Su función es ofrecer la posibilidad al programador de especificar una segunda imagen, junto con la indicada con `SRC`, que sirva de previsualización de la imagen final. Esta previsualización suele ser la misma imagen pero con peor calidad, por ejemplo en formato JPEG con alto grado de compresión, y de mucho menor tamaño. De esta forma cargará mucho antes y el navegante podrá hacerse una idea de cómo será la imagen completa antes de que esta llegue:

```
<IMG SRC="alta_calidad.gif"
  LOWSRC="baja_calidad.gif">
```

Los navegadores que no entiendan esta etiqueta simplemente la ignorarán, y no mostrarán nada hasta que la imagen de alta calidad llegue, exactamente igual que si no hubiéramos usado `LOWSRC`. Por tanto podemos concluir que este atributo no causa ningún perjuicio para los usuarios de dichos navegadores. `LOWSRC` no ha sido aceptado por el estándar HTML 4.0.

Animaciones usando `LOWSRC`

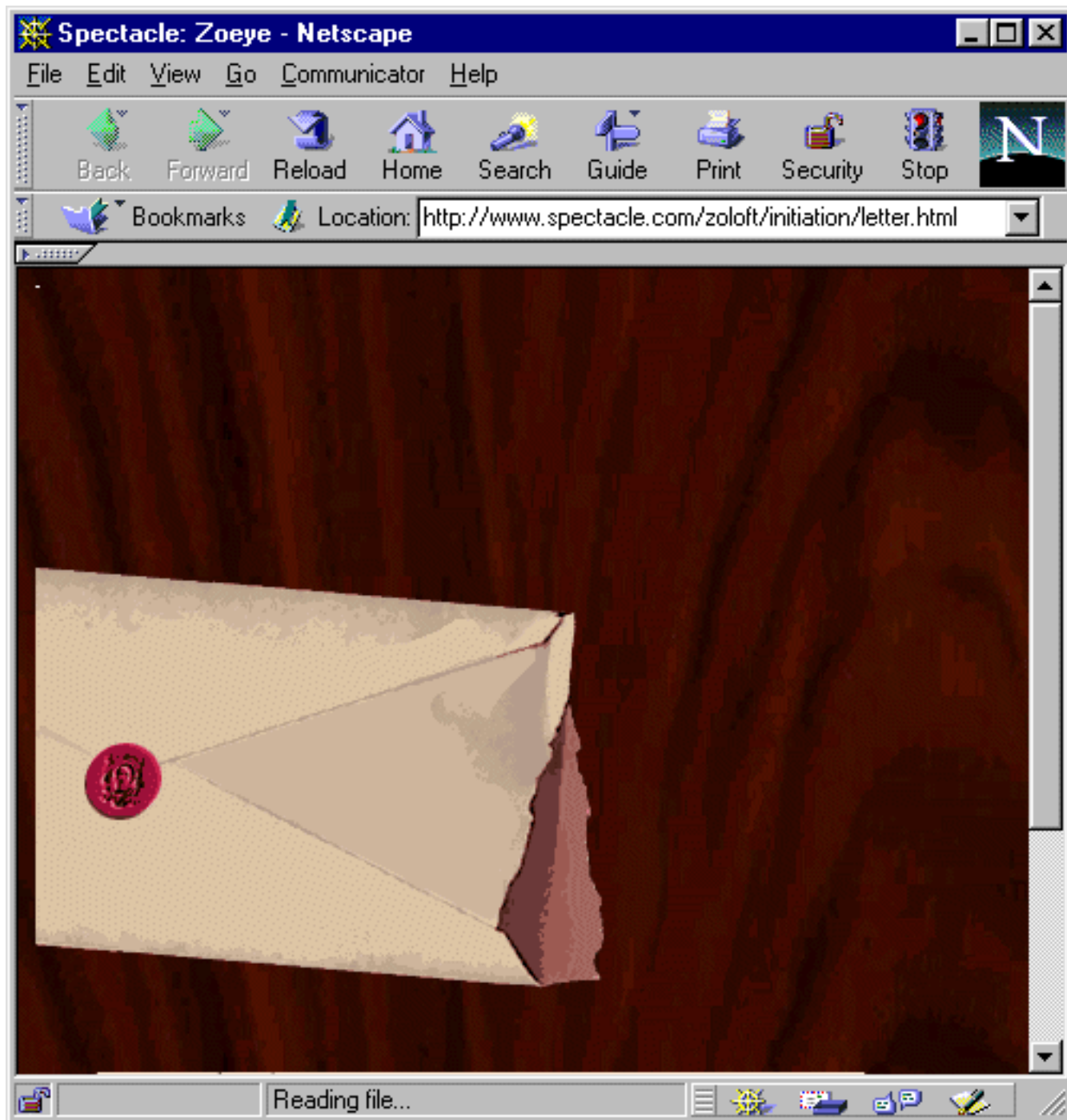
Pronto los diseñadores de páginas Web se dieron cuenta que podían usar el atributo `LOWSRC` para algo más que para mostrar una previsualización de la imagen. Existen varios Webs que destacan por su excepcional diseño que

se sirven de este atributo para crear una sencilla animación formada únicamente por dos imágenes. Un excelente ejemplo de este efecto puede ser visto en la página creada por Josh Feldman que encontramos en la dirección:

- www.spectacle.com/zoloft/initiation/letter.html [<http://www.spectacle.com/zoloft/initiation/letter.html>]

En esta página se usa el atributo LOWSRC para mostrar una imagen de un sobre cerrado y SRC para mostrar ese mismo sobre de fondo pero en primera instancia la carta ya abierta. En la figura 10.7 podemos ver la página tal y como aparece inicialmente con el sobre cerrado (imagen LOWSRC) y la carta final abierta que es mostrada pocos segundos después.

Figura 10.7. Usando el atributo LOWSRC se pueden crear animaciones sencillas pero impactantes como es el caso del web de *Spectacle*.



Sonidos Inline

Microsoft Internet Explorer™ también ha introducido una nueva etiqueta para posibilitar la inclusión de sonido en las páginas Web. Este sonido será cargado junto con la página y empezará a ser tocado sin requerir ninguna acción

por parte del usuario. Para incluir un sonido de fondo como éste debemos usar la etiqueta `BGSOUND`. Veamos un ejemplo:

```
<BGSOUND SRC="sinfonia_5.au">
```

Esta etiqueta no producirá ningún efecto visual en la página, simplemente si empezará a reproducirse el sonido de fondo. Los usuarios de otros navegadores no serán perjudicados por la inclusión de este atributo, simplemente no podrán oír el sonido.

El resto de atributos son:

- **LOOP:** Al igual que con el vídeo este atributo permite controlar el número de veces que el fichero de sonido será reproducido. Si usamos `LOOP="-1"` se reproducirá repetidamente hasta que el navegante abandone la página.
- **BALANCE:** (Sólo Internet Explorer 4.0™ o posterior) Este atributo permitirá al programador Web cambiar el balance de sonidos estéreo. Puede tomar un valor entre -10000 y 10000. El efecto de cada uno de los dos extremos dependerá de la configuración del navegante, pero en cualquier caso ambos significarán que el sonido se oírá únicamente por uno de los dos altavoces. Con los valores intermedios controlamos el balance en uno u otro.
- **TITLE:** (Sólo Internet Explorer 4.0™ o posterior). Con este atributo damos un título al sonido. Este título es solo informativo y no se mostrará ningún indicativo visual.
- **VOLUME:** (Sólo Internet Explorer 4.0™ o posterior). El volumen con el que se reproducirá el sonido puede ser controlado gracias a este atributo que puede tomar un valor entre -10000 y 0. Si usamos `VOLUME="0"` (máximo volumen) el sonido se reproducirá al 100% de la configuración actual del usuario. Usando un valor menor el volumen bajará, no hay ninguna forma usando HTML de aumentar el volumen por encima de la configuración elegida en su ordenador por cada persona.

Nota

Es muy importante no poner un sonido que pueda resultar molesto a los visitantes de nuestras páginas, ya que con ello conseguiremos que las abandonen rápidamente para dejar de oírlo.

Microsoft Internet Explorer™ soporta tres formatos de sonido *inline*:

- **AU:** El formato de Sun™ que usa la codificación u-law.
- **WAV:** Muestras de sonido específicas para Windows™.
- **MIDI:** Estos ficheros deben tener la extensión *.mid*.

Hay que tener mucho cuidado a la hora de incluir sonidos en nuestras páginas, ya que en general este tipo de ficheros son de gran tamaño y pueden incrementar enormemente el tiempo de carga. Por otro lado no es recomendable que sea reproducido más de una vez, ya que la repetición cansará a los visitantes.

Nota

En el directorio *Sonido* hemos incluido un ejemplo (*sonido1.htm*) que usa la etiqueta `BGSOUND` para reproducir como sonido de fondo un archivo en formato MIDI.

Al igual que ocurría con las extensiones para vídeo de Microsoft™ la etiqueta `BGSOUND` no es soportada por ningún otro navegador y tampoco ha sido aceptado por el estándar HTML 4.0. Sin embargo existe otro método que veremos un poco más adelante mucho más extendido basado en PLUG-INS y en una etiqueta llamada `EMBED`.

Marquesinas animadas

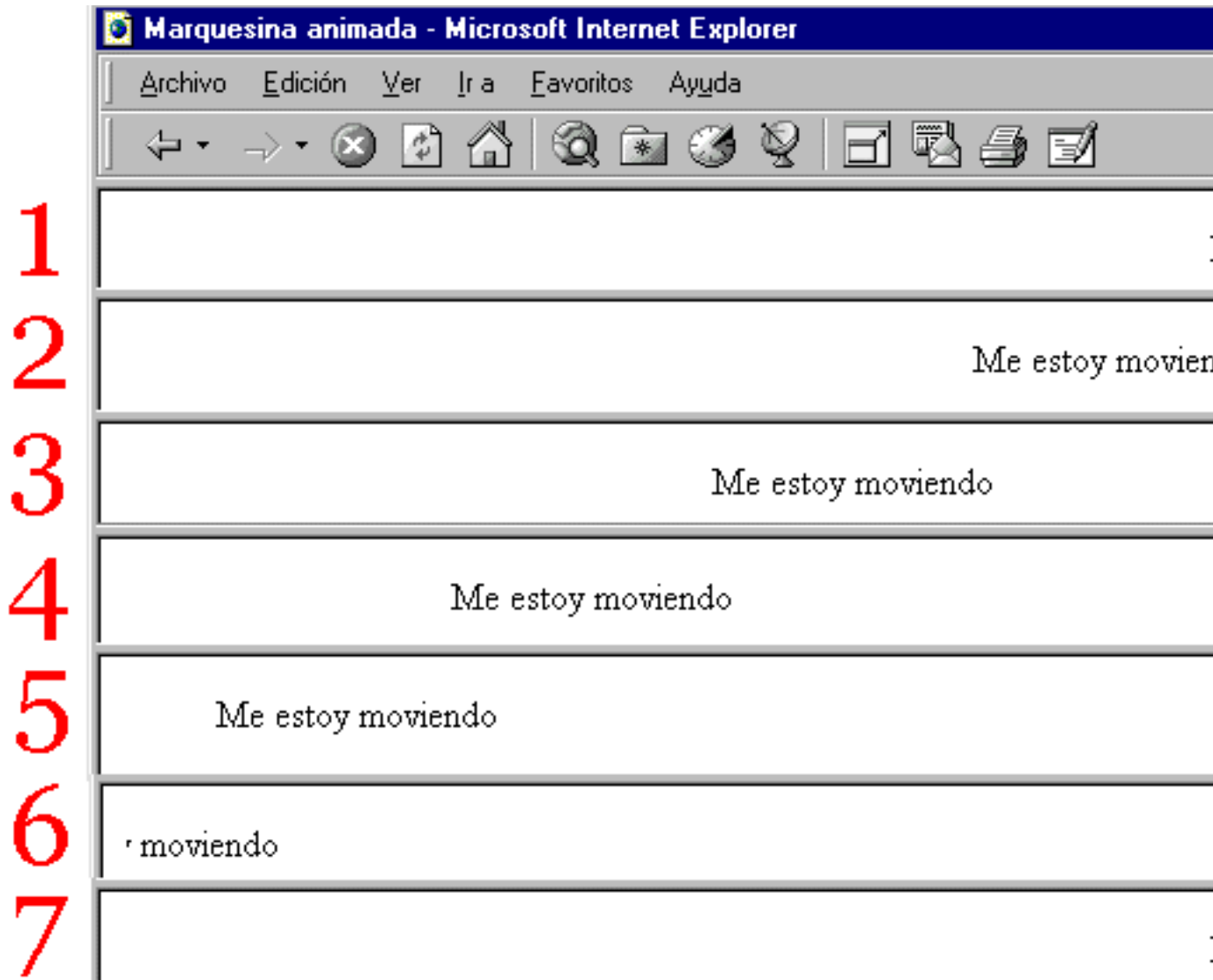
Cuando de HTML se trata, una marquesina animada consiste en una línea de texto que se desplaza por la pantalla de izquierda a derecha, de derecha a izquierda o en ambas direcciones alternativamente. La etiqueta específica de Internet Explorer™ `MARQUEE` nos permitirá crear este tipo de marquesina con gran facilidad y rapidez. No serán necesarias imágenes ni vídeos. Tal y como ocurría con las dos extensiones que acabamos de ver, la etiqueta `MARQUEE` sigue siendo específica de Explorer™ y no es soportada por ningún otro navegador ni ha sido aceptada por el estándar HTML 4.0

El funcionamiento de esta etiqueta es muy sencillo, consta de una instrucción de inicio y otra de fin, entre ellas debemos escribir el texto que queremos que se desplace. Por ejemplo:

```
<MARQUEE>  
Me estoy moviendo  
</MARQUEE>
```

Si visualizamos este código con Internet Explorer™ veremos aparecer el texto “Me estoy moviendo” por la derecha de la pantalla y desplazarse hacia la izquierda hasta llegar al borde izquierdo de la pantalla. Entonces volvería a aparecer de nuevo por la derecha y así indefinidamente. Es difícil plasmar con una figura el movimiento de la marquesina, por esta razón hemos incluido todos los ejemplos de esta sección en el archivo *marquee.htm* [extra/marquee.htm]. Además en la figura 10.8 hemos capturado una página con este código en siete instantes de tiempo consecutivos para que el lector pueda hacerse una idea del funcionamiento de la marquesina animada:

Figura 10.8. La marquesina animada muestra un texto que va moviéndose de derecha a izquierda en la ventana del navegador. En esta figura vemos 7 momentos consecutivos.



El texto de una marquesina se sitúa siempre en una nueva línea, y el es mostrado con las mismas características que sería cualquier texto normal de la página, por ello podemos usar las etiquetas para formatear texto que conocemos para cambiar el formato del texto de una marquesina. Por ejemplo para conseguir un encabezado de tamaño 1 moviéndose podríamos usar:

```
<H1>
<MARQUEE>
Me estoy moviendo
</MARQUEE>
</H1>
```

Igualmente podemos cambiar el color o el tipo de fuente del texto encerrando la marquesina con la etiqueta FONT:

```
<FONT COLOR="#FF0000"  
  FACE="arial"  
  SIZE="4">  
<MARQUEE>  
Me estoy moviendo  
</MARQUEE>  
</FONT>
```

Nota

Internet Explorer 3.0™ ignora todo el código HTML existente entre la instrucción de inicio, <MARQUEE>, y la instrucción de fin, </MARQUEE>. Sin embargo en Internet Explorer 4.0™ esta política ha cambiado y es posible usar las etiquetas formateadoras de texto o saltos de línea en su interior. Por esta razón debemos tener en cuenta que si usamos estas nuevas posibilidades los usuarios de Internet Explorer 3.0™ (que son muchos) no podrán apreciar el formato del texto.

Cómo cambiar la apariencia

Además de estos métodos la propia etiqueta MARQUEE consta de una serie de atributos que nos permitirán cambiar su apariencia:

- **BGCOLOR**: Especificando un color en código RGB (#RRGGBB) o por nombre (blue, red, etc.) cambiaremos el color de fondo de la marquesina.
- **HEIGHT** y **WIDTH**: Determinan la altura y anchura respectivamente de la marquesina. Ambos atributos pueden tomar un valor en píxeles o bien como porcentaje de la ventana del navegador. Por ejemplo `HEIGHT="50%"` provocará que la marquesina tenga una altura mitad de la de la ventana del navegador. Estos atributos no afectan al tamaño del texto.
- **HSPACE** y **VSPACE**: Determinan los márgenes entre los bordes de la marquesina y el texto u otros elementos de la página. **HSPACE** determina el margen a cualquiera de los lados y **VSPACE** los márgenes superior e inferior.
- **ALIGN**: Este atributo controla el alineamiento vertical del texto que bordea la marquesina. Puede tomar cinco valores diferentes: `ALIGN="top"` (arriba), `ALIGN="middle"` (centro), `ALIGN="bottom"` (abajo), `ALIGN="left"` (izquierda) `ALIGN="right"` (derecha) que tienen un comportamiento idéntico al que tenían con las imágenes. Este atributo no modifica la posición del texto en el interior de la marquesina, ya que este esta siempre en la parte superior.

Una vez vistos estos atributos nos queda por hacer un ejemplo que los use. Nuestro objetivo será crear una marquesina con el fondo rojo, que ocupe una anchura mitad de la pantalla y esté alineado a la derecha. El código que debemos usar es:

```
<H1>
<MARQUEE ALIGN="RIGHT"
  BGCOLOR="#FF0000"
  HEIGHT="20%"
  WIDTH="50%">
Gran oferta del MES: Un Mercedes a precio de coste
</MARQUEE>
</H1>
```

En la figura 10.9 podemos ver el resultado obtenido:

Figura 10.9. La etiqueta **MARQUEE** está acompañada de numerosos atributos que nos permiten controlar la apariencia de la marquesina.



Para una mejor apreciación es recomendable visualizarlo con Internet Explorer™ por lo que este ejemplo está incluido en el directorio *Ejemplo4* (ejemplo 4).

Cómo cambiar el comportamiento

Como hemos comentado antes, cuando creamos una marquesina el texto se desplaza de derecha a izquierda desapareciendo totalmente antes de volver a mostrarse por la derecha. Este efecto continúa indefinidamente lo

suficientemente despacio para posibilitar su lectura. Este comportamiento, dirección, número de repeticiones y velocidad de movimiento puede ser modificado con 5 atributos de la etiqueta `MARQUEE`:

- **BEHAVIOR:** Este atributo nos permite cambiar el comportamiento del texto y puede tomar tres valores. `BEHAVIOR="scroll"` es el valor por defecto y consiste en el efecto de desaparecer por un lado y aparecer por el otro. Si usamos `BEHAVIOR="slide"` el texto tiene un comportamiento similar pero para en cuanto llega al borde derecho. La última opción es `BEHAVIOR="alternate"`, en este caso el texto aparecerá igualmente por la derecha, pero una vez llegue al borde izquierdo rebotará y volverá hacia la derecha donde volverá a rebotar repitiendo este comportamiento indefinidamente. Los ejemplos 5 y 6 de la página de ejemplos `marquee.htm` [`extra/marquee.htm`] muestran estas dos últimas posibilidades.
- **DIRECTION:** Con este atributo podemos controlar la dirección de desplazamiento del texto cuando este es de tipo `SCROLL`. Puede tomar dos valores: `LEFT` (izquierda) y `RIGHT` (derecha) siendo el primero el valor por defecto (movimiento de derecha a izquierda). En el ejemplo 7 del archivo antes mencionado podemos ver un ejemplo de este atributo.
- **LOOP:** Determina cuantas veces se repetirá el desplazamiento del texto por la pantalla. Por ejemplo con `SCROLL="4"` el texto se desplazará por la pantalla únicamente 4 veces, mientras que si usamos `SCROLL="-1"` o bien `SCROLL="infinity"` seguirá desplazándose indefinidamente.
- **SCROLLAMOUNT:** Aunque haya sensación de movimiento el texto se desplaza a pequeños saltos. Este atributo determina el número de píxeles que debe desplazarse el texto en cada uno de ellos. Junto con el siguiente atributo podemos controlar la velocidad del movimiento. Si ponemos un valor alto el texto se moverá más rápidamente pero con saltos más bruscos (ejemplos 8 y 9).
- **SCROLLDELAY:** Determina el tiempo de espera entre cada uno de los saltos. Este tiempo debe estar dado en milisegundos. Valores mayores también provocan mayor rapidez y brusquedad. Experimentando con `SCROLLAMOUNT` y `SCROLLDELAY` encontraremos la combinación con la que satisfacemos nuestras necesidades de velocidad y suavidad de desplazamiento (ejemplos 10 y 11).

Texto que parpadea

Junto con el navegador Navigator 1.0™, Netscape™ introdujo una extensión al lenguaje HTML que permite insertar texto que parpadea. Esta extensión consiste en una única etiqueta: `BLINK`. Su funcionamiento es simple, todo el texto que introduzcamos entre la instrucción de inicio y la instrucción de fin parpadeará. Por ejemplo:

```
<BLINK>  
Este texto parpadea  
</BLINK>
```

En el WWW encontramos en numerosas ocasiones con esta etiqueta siendo usada en general para llamar la atención. El problema de `BLINK` es que resalta demasiado. Como hace que el texto parpadeante atraiga la atención y el parpadeo es continuo (no para en ningún momento) puede provocar distracción y provocar que los navegantes no se fijen en nada más de nuestra página.

La gran mayoría de diseñadores Web desaconsejan encarecidamente el uso esta etiqueta porque muchos navegantes lo consideran muy molesto. De hecho en el WWW se dice habitualmente que la etiqueta `BLINK` es el equivalente informático a rallar una pizarra con las uñas.

Si lo que queremos es resaltar una frase es recomendable usar las etiquetas de HTML pensadas para ello, aparte, si queremos podemos usar `BLINK`, pero así nos estaremos asegurando que la frase aparecerá resaltada (aunque no parpadee) en todos los navegadores.

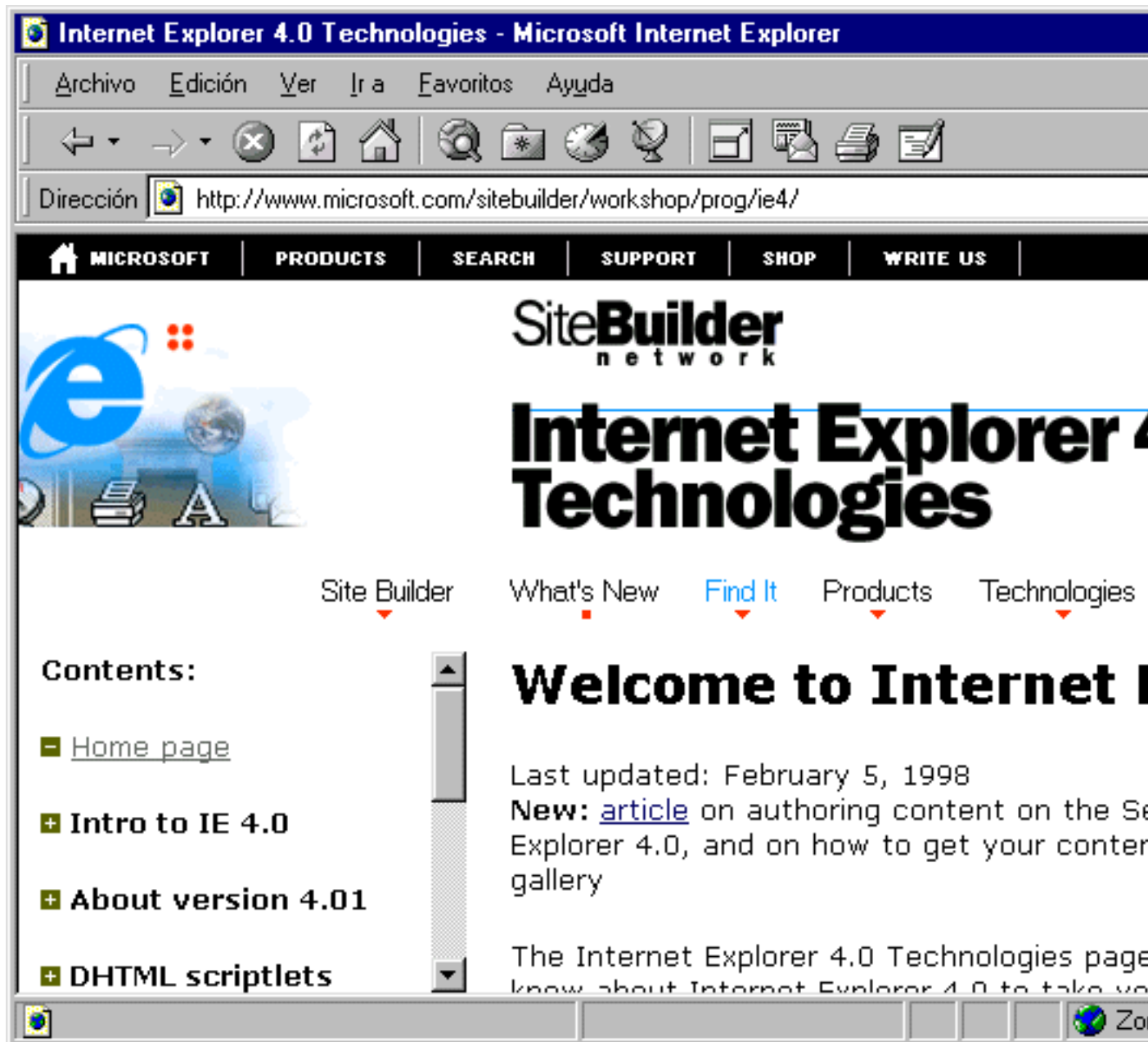
Controles multimedia

Con Internet Explorer 4.0™, Microsoft™ ha introducido un conjunto de controles multimedia que permiten insertar gráficos y efectos multimedia a las página Web. Entre estos controles destacan:

- *Behaviors*: Comportamientos especiales de los controles y otros elementos de la página.
- *Effects*: Aplica un filtro gráfico a cualquier elemento de una página.
- *Hot Spot*: Permite especificar acciones a realizar al pulsar en diferentes zonas de la ventana del navegador.
- *Mixer*: Mezcla varios ficheros de sonido WAV.
- *Path*: Permite crear movimiento de objetos por un camino establecido.
- *Sequencer*: Facilita el control del tiempo en los eventos de nuestra página.
- *Sprite*: Sirve para crear animaciones.
- *Sprite Buttons*: Para crear botones animados.
- *Structured Graphics*: Permite incluir en la página gráficos que pueden ser cambiados de tamaño y rotados por el navegante.

El uso de estos controles puede ser algo complicado y se sale fuera de los objetivos de este curso. Microsoft™ ha creado una página dedicada a las nuevas tecnologías de Internet Explorer 4.0™, de la cual podemos ver una muestra en la figura 10.10:

Figura 10.10. Internet Explorer 4.0™ incluye una serie de nuevas tecnologías entre las que destacan los nuevos controles multimedia.



en la dirección:

- www.microsoft.com/sitebuilder/workshop/prog/ie4/ [http://www.microsoft.com/sitebuilder/workshop/prog/ie4/]

dónde podemos encontrar amplia información. Para un tutorial concreto sobre estos controles podemos acceder a:

- www.microsoft.com/workshop/prog/ie4/directani/directani.htm [<http://www.microsoft.com/workshop/prog/ie4/directani/directani.htm>]

Plug-ins y objetos incrustados

Uno de los avances que han aparecido recientemente en el World Wide Web que se han asentado con mayor fuerza son los llamados PLUG-INS. Estos son una especie de aplicaciones auxiliares que trabajan junto con el navegador ofreciéndole nuevas características. Por ejemplo un plug-in de Excel™ podría permitir que hojas de cálculo de esta aplicación pudieran ser mostradas directamente en una página Web. Otro de vídeo permitiría incluir vídeo inline de diferentes formatos a los vistos.

No debemos confundir los plug-ins con las aplicaciones auxiliares externas de las que hemos hablado anteriormente. Su funcionamiento es parecido, pero estas se ejecutan de manera totalmente independiente al navegador, mientras que los plug-ins se ejecutan con él y le añaden nuevas capacidades *inline*.

Netscape™ introdujo el concepto de plug-ins con Navigator 2.0™. Actualmente existen numerosos plug-ins de audio y vídeo disponibles y la nueva versión de Netscape Navigator™ incluye muchos de ellos que le añaden soporte de AU, AIFF, WAV, MIDI y QuickTime™.

El problema de los plug-ins es que si usas sus capacidades en una página Web todos los visitantes necesitan tener un navegador que soporta plug-ins (como Navigator™ o Internet Explorer™) además del plug-in correspondiente, para visualizarla correctamente. Actualmente la gran mayoría de plug-ins han sido desarrollados para Netscape Navigator™ 3.0 y 4.0 y para Internet Explorer™ 3.0 y 4.0. Los plug-ins más conocidos y usados son:

- *Acrobat Reader*: Permite mostrar ficheros de tipo PDF (Portable Format Document, Documento de formato portable). Más información en:

- www.adobe.com/prodindex/acrobat/main.html [<http://www.adobe.com/prodindex/acrobat/main.html>]

- *Shockwave*: Shockwave es un plug-in creado para permitir la reproducción de Vídeos creados con Macromedia Director™ de manera *inline* en una página Web. Macromedia Director™ es una herramienta enormemente popular entre los desarrolladores profesionales de contenido multimedia para crear presentaciones que incluyan vídeo y sonido. Para obtener más información podemos acceder a la página dedicado a ello de la cual se ofrece una muestra en la figura 10.11:

Figura 10.11. Macromedia™ distribuye a través de su Web el plug-in necesario para visualizar en un Web animaciones creadas por su herramienta Director™.



La dirección de la página es:

- www.macromedia.com/shockwave [<http://www.macromedia.com/shockwave>]
- *RealAudio* y *RealVideo*: Añaden al navegador de reproducir audio y vídeo en estos dos formatos especialmente diseñados para el WWW. Para más información podemos acceder a la página:
 - www.realaudio.com [<http://www.realaudio.com>]

Junto con estos también es posible encontrar numerosos plug-ins dedicados a mostrar contenido multimedia y mundos virtuales VRML (Virtual Reality Modeling Language, Lenguaje de modelado de realidad virtual).

El primer paso para usar un plug-in es crear el objeto que queremos incluir en nuestra página. El proceso de creación depende del tipo de objeto. Para archivos de sonido y vídeo serán necesarios programas de edición y retoque de estos. Para objetos más avanzados como Acrobat™ o Shockwave™ será necesario adquirir las herramientas necesarias para su creación.

Como usar los plug-ins

El siguiente paso será insertar en nuestra página, usando código HTML, el objeto que queremos que sea usado por el plug-in. El propio navegador al darse cuenta que no es capaz de mostrar ese objeto el mismo buscará entre la lista de plug-ins que tiene instalados y llamará a aquel que es capaz de tratar con el objeto. Desgraciadamente si no lo encuentra mostrará un mensaje o un icono indicativo que puede resultar molesto para el navegante.

El termino elegido para denominar a estos objetos es *embedded objects*, es decir, objetos incrustados. La etiqueta HTML con la que insertamos este tipo de objetos es `EMBED`. El atributo `SRC` nos permitirá indicar el nombre y dirección del archivo tal y como hacíamos con las imágenes. Otros atributos usados en multitud de ocasiones son `WIDTH` y `HEIGHT` con los que podemos controlar la anchura y altura del objeto respectivamente. Un ejemplo típico de inserción de un objeto para plug-in podría ser:

```
<EMBED SRC="pelicula.mov"  
WIDTH="120"  
HEIGHT="180">
```

Este objeto aparecerá en su propia línea y centrado en la ventana del navegador. Además de los tres vistos la etiqueta `EMBED` tiene una serie de atributos opcionales que varían enormemente entre los diferentes plug-ins. Todos ellos constan del nombre de un parámetro y su valor. Por ejemplo para el plug-in de vídeo QuickTime™ nos encontramos, con el parámetro `LOOP="TRUE"`. Para averiguar los atributos concretos deberemos leer la documentación concreta de cada plug-in.

Alternativas para navegadores sin soporte para PLUG-INS

Al usar PLUG-INS estamos imponiendo muchos requisitos a los navegantes para ver correctamente nuestras páginas. Como ya hemos mencionado antes, deben tener un navegador con soporte para plug-ins y además deben tener instalado el plug-in adecuado. Si no los cumplen recibirán mensajes de error o cuadros de diálogo que permiten obtener el plug-in necesario (el atributo `PLUGINSOURCE="url"` nos permite indicar la dirección). En otros casos simplemente verán un espacio vacío en la página con un icono indicativo de error como el que podemos ver en la figura 10.12, con la consiguiente frustración que ello conlleva para el navegante.

Figura 10.12. Si el navegador tiene algún problema con el plug-in muestra un icono y un espacio en blanco que puede estropear la apariencia de nuestra página.



Los navegadores que no tengan soporte para plug-ins directamente ignorarán la etiqueta `EMBED`. Para ofrecer una alternativa al objeto incrustado se ha creado la etiqueta `NOEMBED`, que consta de una instrucción de inicio y otra de fin entre las que pueden insertarse cualquier elemento de HTML. Esta etiqueta puede ser usada con diferentes propósitos:

- Mostrar una imagen que represente el significado del objeto.
- Incluir un enlace al fichero del objeto, es decir, incluirlo como fichero externo. De esta forma el navegante podrá verlo con el programa que quiera.
- Poner cualquier código HTML, applet de Java o algún otro elemento que ayude a que la presentación de la página tenga sentido también para navegadores sin soporte de `EMBED`.

Pongamos un ejemplo en el que hemos creado una animación en la que se van mostrando pisadas como si alguien fuese pisando por encima de nuestra página. Podríamos ofrecer como alternativa a la animación una imagen que incluya todas las pisadas y como alternativa a la imagen una serie de asteriscos. Es decir:

```
<EMBED SRC="huellas.cdr"
  WIDTH="640"
  HEIGHT="30"
  ALT="* * * * * * * * * *">
<NOEMBED>
<IMG SRC="huellas.gif"
  WIDTH="640"
```

```
HEIGHT="30">
</NOEMBED>
```

Todo el código entre la instrucción de inicio, `<NOEMBED >`, y la instrucción de fin, `</NOEMBED>` , será ignorado por los navegadores que entiendan la etiqueta `EMBED`.

Sonido *inline* con la etiqueta `EMBED`

En una sección anterior aprendimos a insertar sonido inline para el navegador Internet Explorer™ usando la etiqueta `BGSOUND`. El principal inconveniente de esta etiqueta radicaba en que sólo los usuarios de ese navegador serían capaz de oirlo. Con la etiqueta `EMBED` se nos ofrece la posibilidad de insertar sonido que pueda ser oido también por los usuarios de Netscape Navigator™ gracias a un plug-in que se incluye en la instalación estándar. Veamos un ejemplo:

```
<EMBED SRC="reggae.mid"
WIDTH="0"
HEIGHT="0"
AUTOSTART="true">
```

Con los atributos `WIDTH` y `HEIGHT` puestos a cero conseguimos que los controles de sonido no sean mostrados y usando `AUTOSTART=" true"` hacemos que el sonido empiece a reproducirse inmediatamente. Esto es necesario, ya que al esconder los controles no hay otra forma para comenzar la reproducción.

Pero ahora viene lo más interesante del tema. Podemos añadir el código HTML para insertar sonido para Internet Explorer™ entre las instrucciones `< NOEMBED>` y `</NOEMBED>`, con lo que conseguimos compatibilidad con los dos navegadores principales. El código completo queda:

```
<EMBED SRC="reggae.mid"
WIDTH="0"
HEIGHT="0"
AUTOSTART="true">
<NOEMBED>
<BGSOUND SRC="reggae.mid">
</NOEMBED>
```

Si hubieramos usado `WIDTH="150"` y `HEIGTH="70"` se hubieran mostrado los controles de sonido. En la figura 10.13 vemos dichos controles en Internet Explorer 4.0™ (que soporta la etiqueta `EMBED` al contrario que la versión 3.0):

Figura 10.13. Usando los atributos `WIDTH` y `HEIGHT` de la etiqueta `EMBED` podemos lograr que los controles de sonido aparezcan. Internet Explorer™ usa los controles de Windows™.



y en la figura 10.14 vemos el resultado del mismo código en Netscape Navigator 4.0™.

Figura 10.14. El plug-in usado por Netscape navigator [™] ha creado sus propios controles que tienen la capacidad añadida de poder controlar el volumen.



Applets de Java™ y Controles

ActiveX™

Jorge Ferrer

Rodrigo García

Versión 1.0

Copyright Jorge Ferrer y Rodrigo García. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation. Puede consultar una copia de la licencia en: <http://www.gnu.org/copyleft/fdl.html>

Si en el capítulo anterior vimos el cambio radical que sufría el WWW con la inserción de contenido multimedia la revolución llevada a cabo por la tecnología Java™ y los controles ActiveX™ no se queda atrás. Estos dos nuevos elementos permiten, no sólo insertar contenidos activos, sino además llegar a interactuar con el navegante de una manera antes impensable.

Java™ en las Páginas Web.

Hasta ahora hemos visto una serie de métodos que nos permiten incluir sonido, vídeo y otros contenidos multimedia así como multitud de archivos de diferentes formatos en una página Web. Al insertar alguno de estos objetos conseguimos incluso tener cierta interactividad con el visitante de las páginas y gracias a ellos quedan cubiertas gran parte de nuestras necesidades, pero seguimos teniendo algunas limitaciones. Gracias al lenguaje Java™ podremos insertar contenido multimedia con mucha mayor libertad y, lo que es más importante, conseguiremos interactividad completa con los navegantes.

Java™ es un lenguaje de programación desarrollado por la empresa Sun Microsystems. En la actualidad esta misma empresa se encarga, junto con otras empresas asociadas, de su desarrollo. En la página Java™.sun.com (figura 11.1) podemos encontrar gran cantidad de recursos relacionados con Java™, incluyendo tutoriales, libros o código algunos de los cuales se distribuye gratuitamente. Centrándonos en el lenguaje, éste es muy similar a los usados para realizar las aplicaciones que usamos habitualmente en nuestros ordenadores (Word™, Netscape, Explorer™ o el propio Windows™). La gran diferencia de Java™ con respecto a otros lenguajes consiste en la posibilidad de hacer pequeños programas y ejecutarlos desde una página Web. Estos programas reciben el nombre de *applets*. Cada día hay más páginas en Internet que los usan consiguiendo lo que hasta hace poco era imposible. En la página de *Gamelan*(www.Gamelan.com) podemos encontrar gran cantidad relacionada con la tecnología Java™, en la figura 11.2 podemos ver una muestra de esta página. En una de sus páginas de su nueva ubicación: [www.developer.com/directories/pages/dir.Java™.sites.html](http://www.developer.com/directories/pages/dir.Java%20sites.html) nos ofrecen un listado de las mejores Webs que podemos encontrar hoy en día que usen la tecnología Java™, es muy recomendable visitar alguna de ellas para hacernos una idea real de lo que puede llegar a conseguirse usando este excelente lenguaje.

Figura 11.1. En la página Web de Sun Microsystems, creadora del lenguaje Java™, podemos encontrar gran cantidad de documentación y ejemplos de sus posibilidades.



Figura 11.2. Gamela es una excelente fuente de recursos Java™ conocida desde hace tiempo. En la actualidad su dirección está en proceso de cambio a: www.developer.com.



El conocimiento del lenguaje Java™ se muestra cada vez más importante para todos aquellos programadores de Internet. Más adelante en esta obra abordaremos ampliamente su estudio y aprenderemos a crear nuestras propias aplicaciones y applets. Sin embargo no es necesario conocer este lenguaje para aprovechar sus características en nuestras páginas Web. Existen gran cantidad de applets gratuitos que se distribuyen a través del WWW y que

pueden ser usados libremente en nuestras páginas. En esta sección aprenderemos a insertarlos en nuestras páginas y a usarlos adecuadamente para aprovechar las posibilidades de esta nueva tecnología.

Nota

La propia compañía Sun Microsystems ha creado una página en la que se muestran ejemplos y se distribuyen gratuitamente gran cantidad de applets. La dirección es: Java™.sun.com/applets/index.html. Otra forma de buscar applets gratuitos en el WWW consiste en dirigirse al índice Yahoo y buscar la cadena "Java™ applets".

Inserción de Applets Java™ .

De ahora en adelante consideraremos que ya tenemos un applet, bien porque lo hemos programado nosotros mismos, bien porque estamos usando un applet de distribución gratuita, es hora de aprender a insertarlo en nuestra página Web. Para ello usaremos una nueva etiqueta: `APPLET`. Esta etiqueta está formada por una instrucción de inicio, `<APPLET>`, y una instrucción de fin, `</APPLET>`.

Nota

La etiqueta `APPLET` fue introducida por Netscape en la versión 2.0 de Navigator™ fruto de un acuerdo con Sun para soportar la tecnología Java™. Todos sus navegadores posteriores también la soportan. El otro gran navegador, Microsoft Internet Explorer™, soporta la tecnología Java™ desde la versión 3.0. En cuanto al estándar se refiere, esta etiqueta se incluye en la especificación HTML 3.2, aunque no podemos asegurar que los navegadores posteriores que respetan la norma HTML 3.2 soporten la tecnología Java™, simplemente entienden la etiqueta, pero no son capaces de ejecutar un applet.

La inserción de un applet en una página es similar a la inserción de una imagen, a lo cual ya estamos muy acostumbrados. Deberemos especificar el archivo donde se encuentra el applet y las dimensiones (anchura y altura) que este debe ocupar en la página. Al igual que ocurría con las imágenes el applet se introducirá en el lugar exacto donde indique su código, si es necesario se insertará entre el texto, pero no pasará a una nueva línea, como sucedía con las marquesinas, si no lo indicamos específicamente (con la etiqueta `
`). Para indicar el archivo en el que se encuentra el applet usaremos el atributo `CODE` y para especificar la anchura y altura en píxeles usaremos `WIDTH` y `HEIGHT` respectivamente. En resumen, para insertar un applet debemos usar el siguiente código en cualquier lugar de la página:

```
<APPLET CODE="nombre_applet.class" WIDTH="ancho" HEIGHT="alto">
</APPLET>
```

Los tres atributos que hemos usado deben incluirse *obligatoriamente* siempre que insertemos un applet. Esto parece obvio con el atributo `CODE`, ya que no se podría insertar el applet sin indicar donde está, pero resulta más extraño con los atributos `WIDTH` y `HEIGHT`. Aun así, si no especificamos el tamaño que debe ocupar el applet en la página no será mostrado ni ejecutado por el navegador. Debemos estar muy atentos a este detalle ya que hasta ahora estos dos atributos eran totalmente optativos y por ello es un error común olvidarlos y no ser capaz de descubrir porqué no funciona el applet.

Y aquí vemos como insertar un applet de ejemplo:

```
<HTML>
<HEAD>
<TITLE>Applet sonoro</TITLE>
</HEAD>
<BODY>
```

```
<APPLET CODE="SoundExample.class" WIDTH=450 HEIGHT=50></APPLET>  
</BODY>  
</HTML>
```

Aviso

Los applets de Java™ son sensibles a las mayúsculas y a las minúsculas, si no ponemos el nombre del archivo exactamente igual, incluyendo mayúsculas o minúsculas, no hubiera funcionado. Por ejemplo, si hubiéramos escrito `CODE="soundexample.class"` el código anterior no hubiera funcionado. Este es un ejemplo muy común por lo que debemos tener cuidado.

Si visualizamos este archivo con un navegador que soporte la tecnología Java™ debemos obtener el resultado de la figura 11.3.

Figura 11.3. Este es el aspecto que muestra nuestro primer applet insertado en una página Web. En un navegador con soporte Java™ podemos pulsar sobre los botones para obtener diferentes sonidos.



Nota

Los applets Java™ tienen la terminación `.class`, como ya hemos comentado antes, esto exige que para poder usar un applet debemos utilizar un sistema operativo que soporte nombres largos. Desgraciadamente los usuarios de Windows™ 3.1 no podrán probar esta tecnología localmente en su ordenador. Afortunadamente esto no significa que no puedan disfrutar los applets que estén publicados en el WWW.

Pasemos a explicar que ocurre cuando el navegador se encuentra con el código anterior. En primer lugar busca el archivo `SoundExample.class` en la misma dirección URL y en el mismo directorio donde se encuentre la página actual y lo baja a nuestro ordenador (este es un ejemplo en el que ambos archivos ya están en nuestro ordenador, con lo que este paso no sería necesario). Mientras reserva un rectángulo, con las dimensiones especificadas por los atributos `WIDTH` y `HEIGHT`, donde será mostrado el applet. Una vez ha llegado el archivo anterior el navegador llama a lo que se conoce como *máquina virtual Java™* (también llamada JVM, *Java™ Virtual Machine*) que pasará a ejecutar el applet. A partir de ese momento el applet se ejecutará como cualquier otro programa de nuestro ordenador, aunque lo hará en el interior de la página Web. De esta forma cuando nuestro curso este dentro del rectángulo de 450x50 creado el control pasará al applet y si pulsamos los botones el propio applet será el encargado de realizar las acciones oportunas. La Máquina Virtual Java™ implementa además ciertas medidas de seguridad para que el applet no pueda, por ejemplo, borrar nuestro disco duro. Una vez conocemos el funcionamiento de los applets podemos adentrarnos más a fondo en las diferentes características de estos que pueden ser modificadas con el lenguaje HTML.

Especificando la dirección del applet.

Dejando a un lado el ejemplo y volviendo al código general es obvio que el código es, todavía, muy sencillo. Para empezar el archivo con el applet (que siempre tiene la extensión `.class`) se encuentra en el mismo directorio que el documento HTML de la página actual. Lo habitual, sin embargo, es agrupar todos los applets Java™ en un directorio dedicado de la misma manera que hacíamos con las imágenes. Podríamos pensar que para reflejar este cambio no tendríamos más que cambiar ligeramente el código y escribir:

```
<APPLET CODE="applets/nombre_applet.class" WIDTH="ancho" HEIGHT="alto">
</APPLET>
```

Por desgracia, este código es *incorrecto*. El atributo `CODE` permite únicamente especificar el nombre del archivo donde se encuentra el applet, pero no podemos incluir un directorio ni, por supuesto, una dirección de Internet. ¿Significa esto que es imposible? Afortunadamente no, los diseñadores de esta etiqueta pensaron en ello e introdujeron un nuevo atributo, `CODEBASE`, que nos permitirá indicar la dirección URL o ruta de directorios donde se encuentre el archivo especificado en el atributo `CODE`. Usando `CODEBASE` podemos reescribir el código anterior de manera correcta:

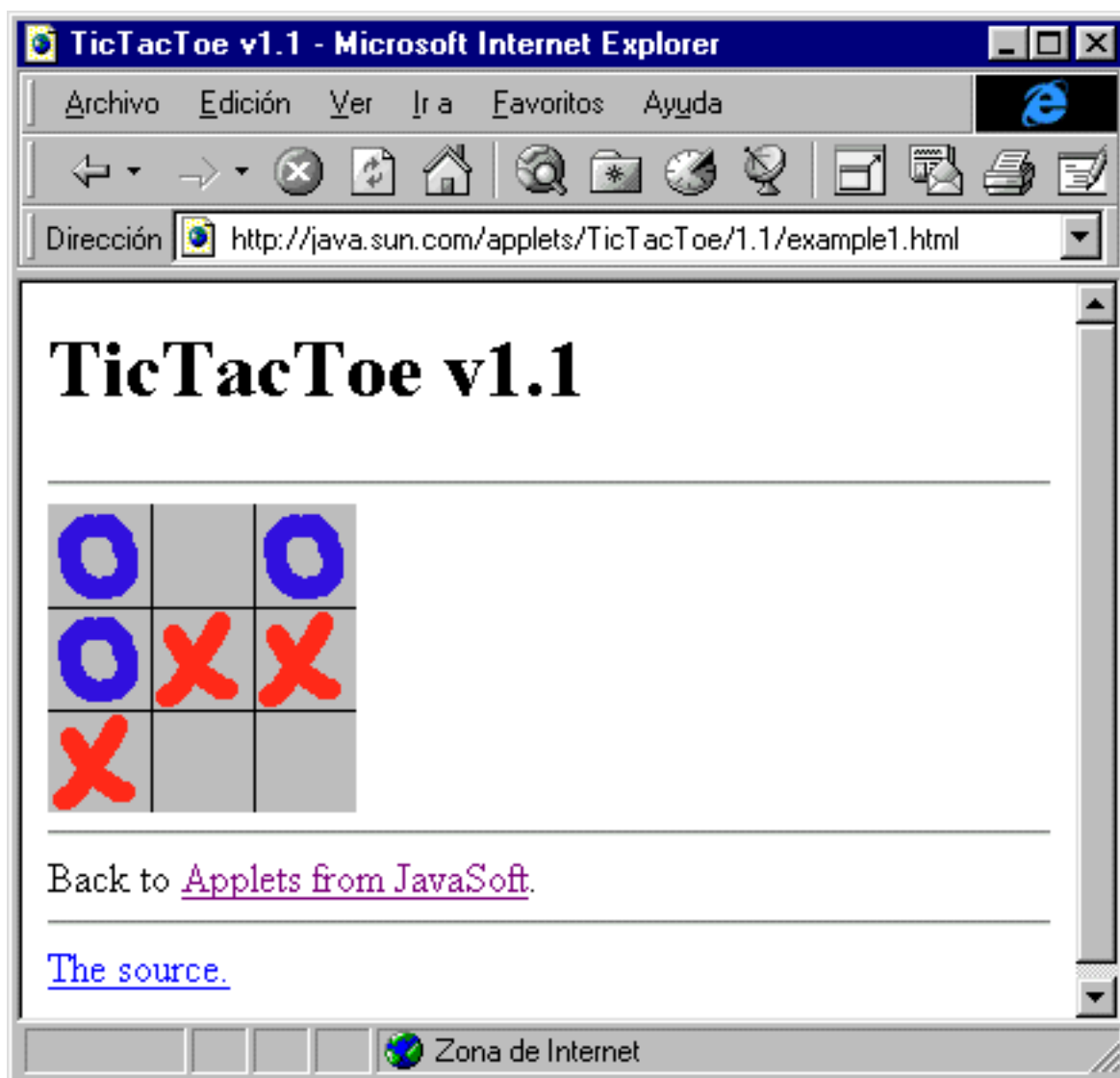
```
<APPLET CODEBASE="applets/" CODE="nombre_applet.class" WIDTH="ancho" HEIGHT="alto">
</APPLET>
```

Sin embargo podemos ir más allá. Podemos usar en nuestra página un applet que se encuentre en cualquier otro lugar del WWW sin tener que copiarlo a nuestro servidor. Para ello no tendremos más que usar el atributo `CODEBASE` para especificar la dirección base. Por ejemplo:

```
<APPLET code=TicTacToe.class width=120 height=120
CODEBASE="http://Java™.sun.com/applets/TicTacToe/1.1/">
</APPLET>
```


Esta dirección es real, si incluimos este código en una página habremos insertado un juego de las tres en raya como el que vemos en la figura 11.4 en el que cualquier visitante de nuestra página podrá jugar (contra el ordenador). Si vemos en cualquier página un applet que nos gusta y sus autores permiten que sea utilizado en páginas ajenas (esto es muy importante) no debemos dudarle y usarlo. La carga del applet no será más lenta que si estuviera en nuestro mismo servidor y además tendremos a nuestra disposición una mayor cantidad de applets, ya que muchos autores permiten el uso de applets siempre y cuando estos no sean copiados de su servidor sino usados directamente desde él.

Figura 11.4. Usando applets Java™ podemos insertar juegos interactivos como en este caso. Usando el atributo CODEBASE no es necesario que el applet esté en nuestro servidor.



Otros atributos de la etiqueta `APPLET`.

Además de los cuatro atributos que hemos vistos hasta ahora, que suelen usarse prácticamente siempre, existen otros que pueden resultarnos muy útiles. Todos ellos se encuentran incluidos en el estándar HTML 3.2:

NAME="nombre del applet": Este atributo es usado para identificar el applet entre el resto de elementos u otros applets de la página. Poner el nombre del applet no es necesario, pero los buenos programadores aconsejan hacerlo por una cuestión de estilo. Por otro lado tener el applet identificado es tremendamente útil o incluso obligatorio si posteriormente queremos que este se comunique, por ejemplo, con código Java™ Script.

ALIGN: Con esta etiqueta podremos controlar el alineamiento horizontal de la página. Puede tomar tres valores: *left*, *right* y *center* para conseguir que el applet esté alineado a la izquierda, a la derecha o esté centrado respectivamente. El funcionamiento de estos tres valores es similar al de las imágenes y si especificamos un alineamiento a cualquiera de los dos lados el texto circundante envolverá al applet.

VSPACE="pixeles": Con este atributo establecemos un margen vertical del tamaño en pixeles especificado. El margen vertical se refiere a la distancia entre la parte superior del applet y los elementos de la página que estén sobre él y entre la parte inferior y los elementos situados debajo.

HSPACE="pixeles": Este atributo nos permite establecer el ancho del margen horizontal, es decir el espacio entre el applet y los elementos que estén situados a su derecha y a su izquierda.

ALT="texto alternativo": Esta etiqueta es usada por aquellos navegadores que *sí entienden* la etiqueta `APPLET`, pero no soportan la tecnología Java™. Todos los navegadores que cumplan la norma HTML 3.2 entienden esta etiqueta, pero eso no quiere decir que sean capaces de ejecutar un applet. Esto ocurre, por ejemplo, con los navegadores de sólo texto, ya que al ser los applets gráficos en su mayoría no son capaces de mostrarlos en pantalla. Este texto también es mostrado si en un navegador con soporte de tecnología Java™ tiene algún problema y no es capaz de ejecutar el applet.

Continuando con el ejemplo anteriormente mostrado podemos usar estos atributos para obtener un código más completo. Para empezar es muy recomendable usar siempre los atributos `NAME` y `ALT`, con lo que quedaría (mostramos únicamente el código correspondiente a la inserción del applet):

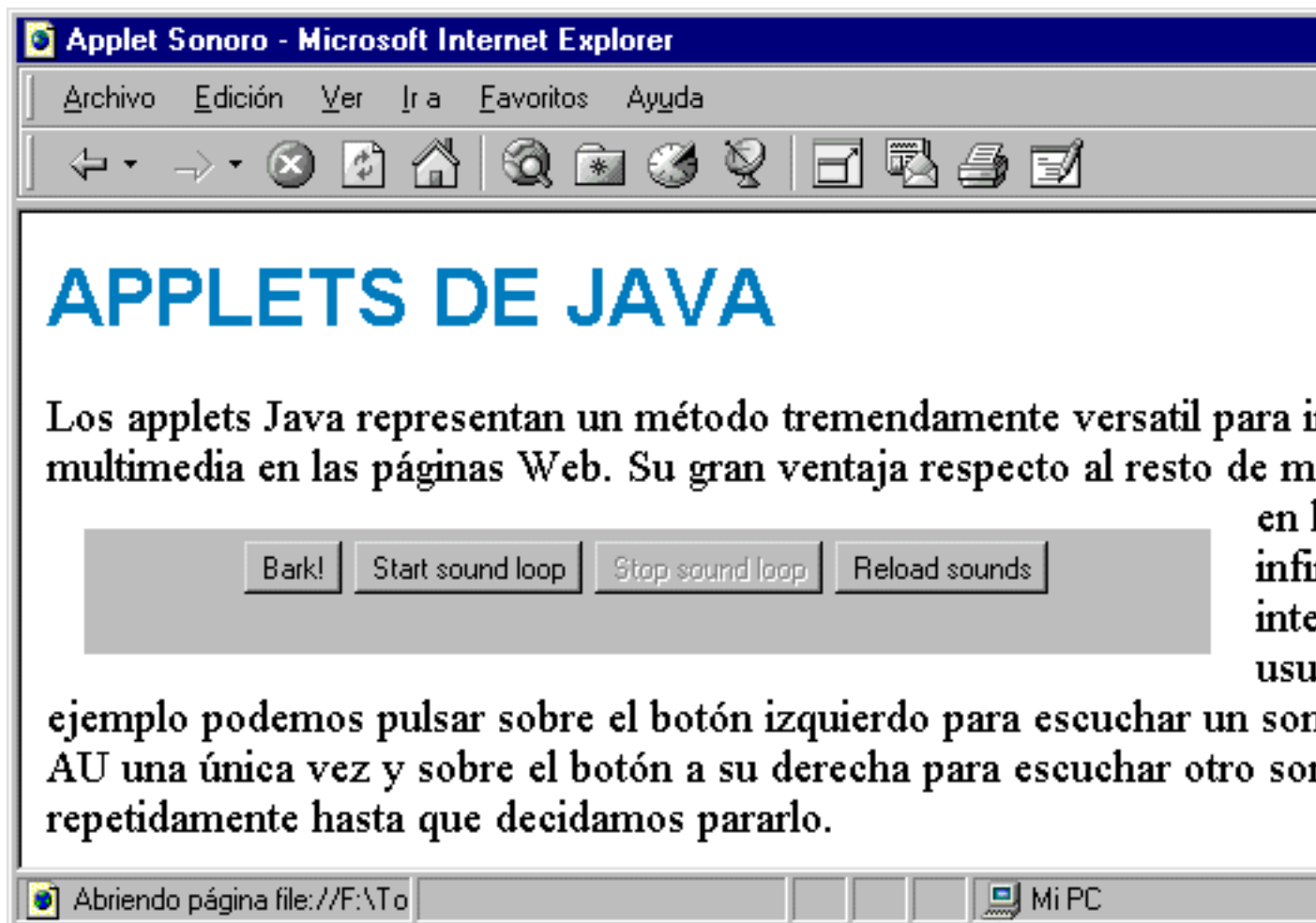
```
<APPLET CODE="SoundExample.class" WIDTH=450 HEIGHT=50 ALT="Su navegador soporta la tecnología Java™, pero por algún error no es capaz de mostrar este applet"
NAME="Ejemplo Sonoro">
</APPLET>
```

Por otro lado podemos usar el atributo `ALIGN` para que el texto bordee el applet. En este caso es recomendable usar también `VSPACE` y `HSPACE` para crear un margen:

```
<APPLET CODEBASE="ej1/" CODE="SoundExample.class" WIDTH=450 HEIGHT=50 ALIGN="left"
HSPACE=15 VSPACE=15>
</APPLET>
```

Escribiendo texto antes y después de esta etiqueta obtenemos el resultado de la figura 11.5.

Figura 11.5. Usando los atributos `ALIGN`, `VSPACE` y `HSPACE` podemos insertar un applet rodeado de texto.



Los atributos anteriores son los que aparecieron inicialmente junto con la etiqueta `APPLET` y que fueron adoptados por el estándar HTML 3.2 pero además de estas, tanto Microsoft como Netscape han ido incorporando otros nuevos en sus navegadores:

`ARCHIVE="archivo_comprimido":` Permite especificar un archivo auxiliar al applet de Java™ donde pueden ser insertados todas las imágenes, sonidos y cualquier otro archivo auxiliar que pueda ser necesario para la ejecución del applet. La existencia de este atributo ha sido pensada para comprimir todos los archivos auxiliares en uno sólo de menor tamaño para que el tiempo de carga sea menor. Cualquier otro fichero que necesite el applet que no este incluido en este archivo comprimido será buscado por los métodos habituales. Este atributo es exclusivo del navegador Netscape Navigator™.

`MAYSCRIPT:` Este atributo se usa por si sólo, sin ningún valor, y su presencia significa que el applet Java™ puede ser accedido usando código Java™ Script.

`TITLE="texto"`: Este atributo exclusivo de Internet Explorer™ 4.0 permite indicar un texto que será mostrado en un bocadillo cuando el navegante pase el ratón por encima del applet. Aunque el contenido de este atributo sólo es entendido por Explorer™ puede ser usado con total libertad, ya que no perjudica nada a los usuarios del resto de navegadores.

Estos son todos los atributos que pueden ser usados en la etiqueta `APPLET`, pero existe otro método adicional para especificar parámetros particulares de cada uno de estos programas escritos en Java™: la etiqueta `PARAM`.

La etiqueta `PARAM`.

Esta etiqueta consta de una única instrucción, `<PARAM>`, que debe ser insertada en el interior de la etiqueta `APPLET` (entre la instrucción de inicio y la instrucción de fin) y cuya función será definir parámetros para el applet. Los parámetros son propiedades específicas de cada applet que permiten modificar sus características. Los parámetros constan de un nombre y un valor que son indicados usando los atributos `NAME` y `VALUE` de la siguiente manera:

```
<APPLET CODE="nombre_applet.class" WIDTH="ancho" HEIGHT="alto">
<PARAM NAME="nombre_parámetro" VALUE="valor_parámetro">
</APPLET>
```

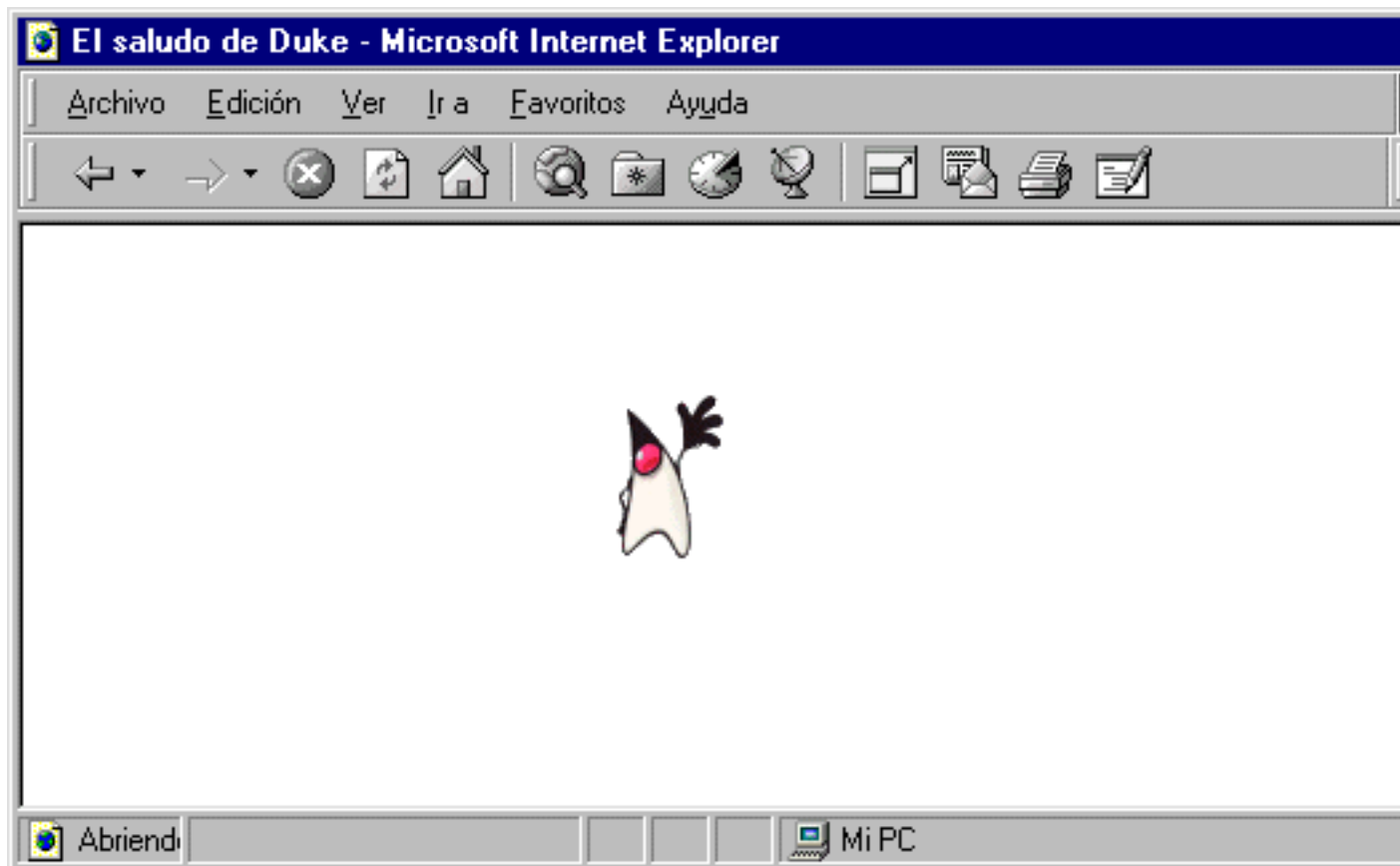
La etiqueta `PARAM` puede ser usada tantas veces sea necesario para un mismo applet siendo necesario usarla una vez para cada uno de los parámetros. En general los applets que se distribuyen gratuitamente constan de gran número de parámetros, para poder adaptarse a las necesidades de los distintos usuarios que vayan a utilizarlo. Por esta razón siempre van acompañados de documentación. Quizá el applet más conocido es `Animator.class`, este applet toma un conjunto de imágenes y las va mostrando ordenadamente para crear un efecto de animación. Este era el único método existente para crear animaciones antes de la aparición de los GIFS animados. Tras la aparición de estos la utilización de este applet ha decrecido, pero sigue siendo útil en ciertas ocasiones y nos servirá como excelente ejemplo para mostrar el uso de la etiqueta `PARAM`. Para insertar el applet debemos copiar el contenido de este directorio a otra ubicación y crear un documento HTML con el siguiente código:

```
<HTML>
<HEAD>
<TITLE>El saludo de Duke</TITLE>
</HEAD>
<BODY>
<APPLET CODE="Animator.class" WIDTH=55 HEIGHT=68>
  <PARAM NAME="imageSource" VALUE=".">
  <PARAM NAME="endImage" VALUE=10>
  <PARAM NAME="PAUSE" VALUE=100>
  <PARAM NAME="PAUSES" VALUE="2500|100|100|100|100|100|100|100|100|100">
</APPLET>
</BODY>
</HTML>
```

En este caso hemos usado 4 parámetros del applet, aunque en realidad tiene muchos más. Los explicaremos brevemente. El parámetro `imageSource` nos permite indicar el directorio donde se encuentran las imágenes, un punto significa el directorio actual. Estas imágenes deben llamarse obligatoriamente `T1.gif`, `T2.gif`, `T3.gif`, etc. El parámetro `endImage` indica el número de imágenes de los que consta la animación y en esta ocasión serán 10, desde `T1.gif` hasta `T2.gif`. Los parámetros `pause` y `pauses` nos permiten controlar las pausas entre las imágenes, con el último controlamos de manera independiente las pausas que se realizan entre cada una de las imágenes de manera individual. Al visualizar este código con un navegador con soporte Java™ veremos la mascota de Java™, Duke, saludando. En la figura 11.6 podemos apreciar una captura del resultado. Una gran

ventaja de este método para crear animaciones es que simplemente cambiando las imágenes `T1.gif`, `T2.gif`, etc. por unas propias podemos crear otra animación distinta sin necesitar ninguna aplicación como era el caso de los GIFs animados.

Figura 11.6. El applet `Animator.class` es uno de los más conocidos y usados en las páginas Web por la facilidad que ofrece para crear animaciones.



Nota

Para obtener más información sobre el applet `Animator.class` o sobre muchos otros applets de distribución gratuita podemos visitar la página antes mencionada Java™.sun.com.

Alternativas a los applets Java™.

Dado que la tecnología Java™ es relativamente nueva y requiere grandes recursos por parte del navegador, sólo algunos como Netscape Navigator™ 2.0 o superior y Internet Explorer™ 3.0 o superior la soportan. Dado que no sabemos que navegadores usarán los visitantes de nuestras páginas es conveniente añadir alternativas para aquellos que no sean capaces de mostrar el applet. Anteriormente hemos visto la existencia del atributo `ALT` que permite especificar un texto alternativo. Sin embargo tiene dos inconvenientes:

- Este atributo sólo será entendido por aquellos navegadores que entiendan la etiqueta `APPLET`, con lo que seguimos sin tener una alternativa para el resto de navegadores.

- Con ALT únicamente podemos insertar un texto, ni imágenes ni ningún otro elemento HTML, con lo que nuestras posibilidades para ofrecer alternativas quedan muy menguadas.

Afortunadamente existe otro método. Como hemos indicado al principio de esta sección, y como ya hemos comprobado por los ejemplos, la etiqueta `APPLET` consta de una instrucción de inicio y de una instrucción de fin. Vimos que entre ellas podíamos insertar una o varias etiquetas `PARAM` para especificar los parámetros específicos de cada applet. Pero además podemos insertar otras etiquetas HTML que serán ignoradas por aquellos navegadores con soporte Java™, pero serán mostradas por todos aquellos que no soporten esta tecnología, con lo que nos servirá como contenido alternativo al applet:

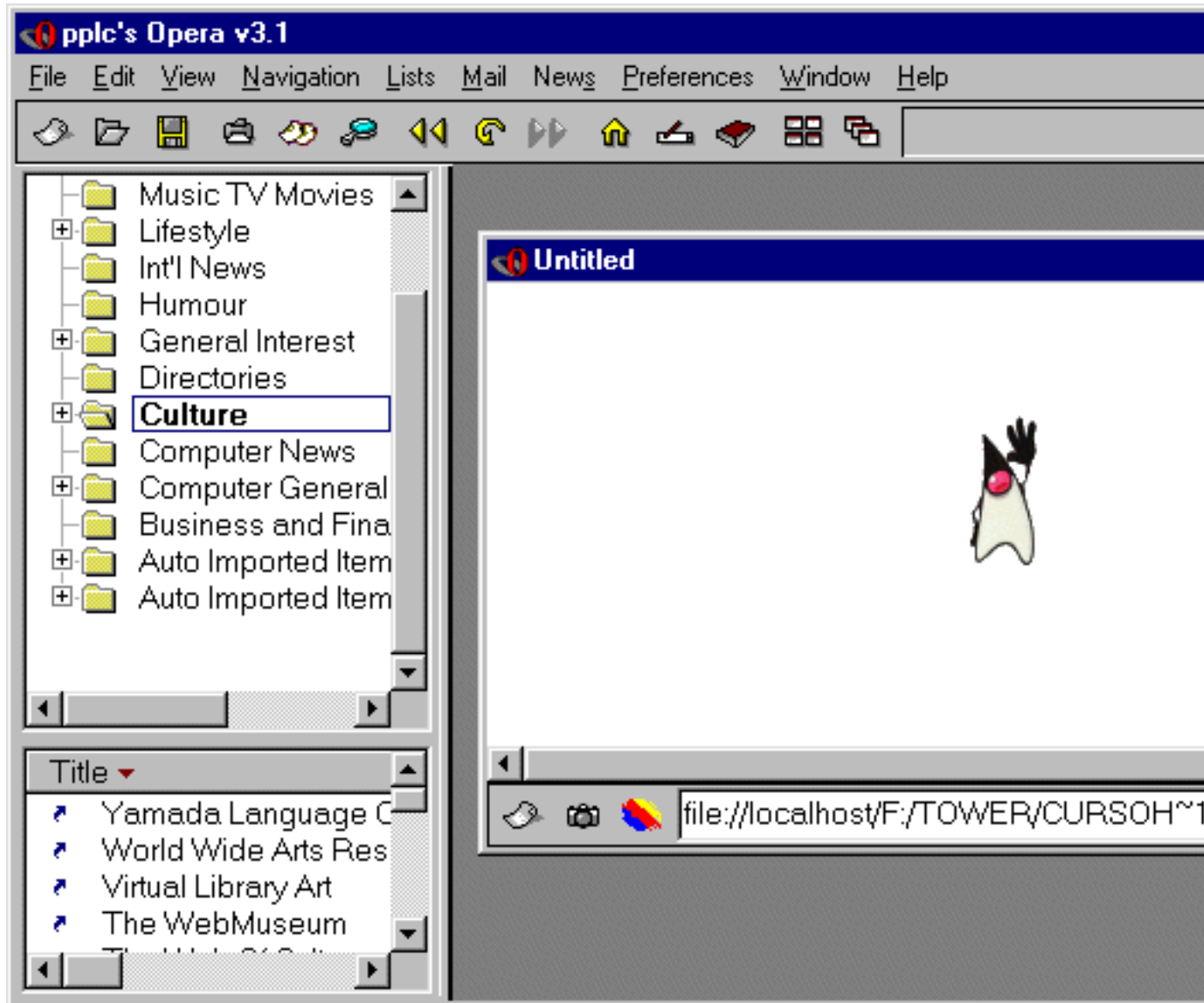
```
<APPLET CODE="nombre_applet.class" WIDTH="ancho" HEIGHT="alto">
<PARAM NAME="nombre_parámetro" VALUE="valor_parámetro">
... Código HTML alternativo ...
</APPLET>
```

Siguiendo con el ejemplo de la mascota de Java™, si queremos que nuestra página este preparada para todos los navegadores debemos añadir el atributo `ALT` así como contenido alternativo antes de la instrucción `</APPLET>`. He aquí un ejemplo:

```
<APPLET CODE="Animator.class" WIDTH=55 HEIGHT=68 ALT="Lo siento no puede ver el
applet">
  <PARAM NAME="imagesource" VALUE=".">
  <PARAM NAME=ENDIMAGE VALUE=10>
  <PARAM NAME=PAUSE VALUE=100>
  <PARAM NAME=PAUSES VALUE="2500|100|100|100|100|100|100|100|100|100">
  <IMG SRC="T5.gif" WIDTH=55 HEIGHT=68 ALT="Duke te saluda">
</APPLET>
```

En este caso los usuarios de navegadores sin soporte Java™ verán una imagen de Duke saludando (aunque sin movimiento) del mismo tamaño que el applet. Si en el texto de la página hemos hecho referencia a que los navegantes van a ver un applet en acción debemos insertar también junto con la imagen un texto que advierta que el applet no puede ser visualizado con ese navegador y que a cambio se muestra una imagen. En la figura 11.7 vemos el ejemplo anterior visto con el navegador Opera™, que no soporta la tecnología Java™. En un medio estático como este no se aprecia la diferencia, pero esta existe, ya que en este caso la imagen no tiene movimiento.

Figura 11.7. El navegador Opera™ no soporta la tecnología Java™, por lo que mostrará el código alternativo, en este caso uno de los GIFs de la animación.



Con este ejemplo damos por concluido el aprendizaje de la etiqueta `APPLET`. En la figura 11.8 podemos ver un ejemplo.

Figura 11.8. Resumen de todos los atributos de la etiqueta **APPLET** y de su subetiqueta **PARAM**.

Curso WEBMASTER - Resumen de la etiqueta APPLET - Microsoft Internet Explorer	
Archivo Edición Ver Ir a Favoritos Ayuda	
< APPLET	Etiqueta usada para i
CODE = <i>applet</i>	Indica el nombre del
HEIGHT = <i>alto</i>	Especifica la altura en
WIDTH = <i>ancho</i>	Especifica la anchura
[CODEBASE = <i>URL</i>]	Dirección URL o ruta archivo indicado en C
[ALT = <i>'Texto alternativo'</i>]	Texto alternativo
[NAME = <i>'Nombre'</i>]	Nombre del applet (f identificarlo)
[ALIGN = <i>alineamiento</i>]	Permite alinear el app center
[VSPACE = <i>pixeles</i>]	Margen en pixeles ar
[HSPACE = <i>pixeles</i>]	Margen en pixeles a
[ARCHIVE = <i>archivo_comprimido</i>]	(Sólo Netscape Navi comprimido que cont necesita el applet.
[MAYSCRIPT]	(Extensión) Indica qu lenguaje de Script
[TITLE= <i>"título"</i>]	(Sólo Internet Explor que será mostrado en encima.
[< PARAM NAME = <i>Parametro1</i> VALUE = <i>valor1</i> >] [< PARAM NAME = <i>Parametro2</i> VALUE = <i>valor1</i> >] ...	Parámetros extras es
[<i>Código HTML alternativo</i>]	Código alternativo pa etiqueta APPLET
</APPLET>	

El nuevo estándar: la etiqueta OBJECT.

Hasta ahora hemos aprendido a insertar diferentes tipos de objetos en las páginas Web, para insertar imágenes usábamos la etiqueta `IMG`, para insertar objetos para plug-ins usábamos `EMBED`, para insertar frames flotantes `IFRAME`, para insertar applets Java™ `APPLET`, etc. Observando esta diversidad y viendo que cada desarrollador de navegadores creaba sus propios métodos para incluir objetos en las páginas el consorcio W3C creó un grupo de trabajo para buscar una solución que englobará y unificará todos los métodos actualmente existentes. La solución fue la creación de una nueva etiqueta para el lenguaje HTML: `OBJECT`, que fue introducida en el estándar HTML 3.2, aunque es a partir de la versión *HTML 4.0* cuando se está empezando a darle la importancia que merece.

Usando esta etiqueta es posible incluir varios tipos de ficheros multimedia (vídeo y sonido MPEG, ficheros Shockwave™/Director™, etc.), applets de Java™, controles ActiveX™ (que veremos en la siguiente sección), documentos HTML o prácticamente cualquier otro tipo de objeto. Por ejemplo para incluir un vídeo AVI debemos usar el siguiente código:

```
<OBJECT DATA="flores.avi" TYPE="video/avi" WIDTH=100 HEIGHT=100>
<IMG SRC="flores.gif" WIDTH=100 HEIGHT=100 ALT="[FLORES]">
</OBJECT>
```

En este caso el funcionamiento es muy similar al de la etiqueta `EMBED` y ahora el código alternativo deberá ser mostrado entre `<OBJECT>` y `</OBJECT>`. Este código será ignorado por los navegadores que entiendan la etiqueta `OBJECT`.

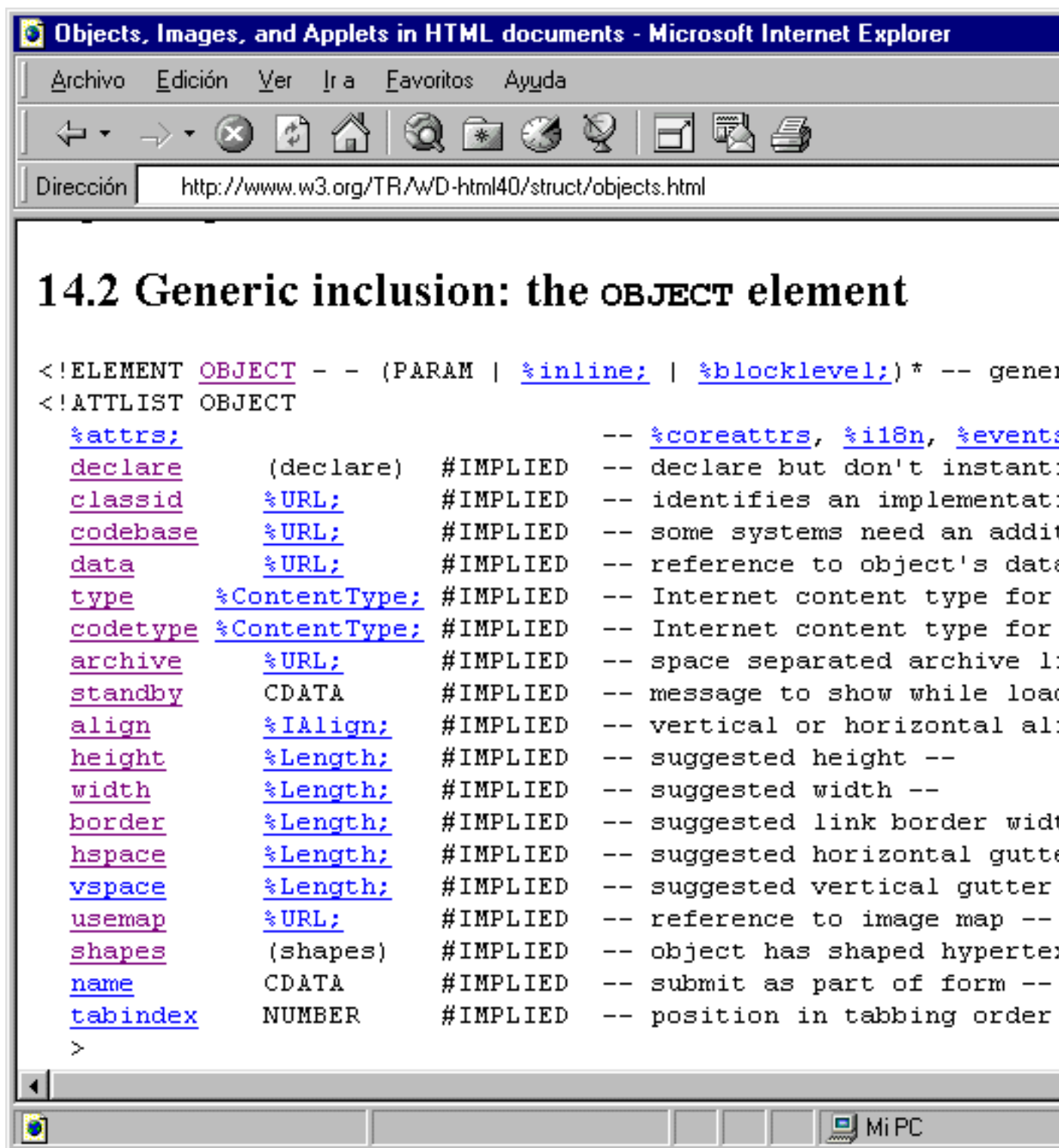
¿Qué navegadores soportan este nuevo estándar? Las versiones 4.0 y posteriores de Internet Explorer™ y Netscape Navigator™ entienden la etiqueta, aunque todavía no soportan todas las posibilidades que ofrece. Sin embargo el Web Consortium le está dando un fuerte impulso, con lo que se supone que las futuras versiones de estos navegadores sí lo harán y que poco a poco se tenderá hacia un mayor uso de la etiqueta `OBJECT`.

Esta sección es sólo una introducción a este nuevo estándar. Hemos considerado que un estudio detallado no era apropiado porque esta etiqueta no puede ser usada todavía de manera fiable (excepto para insertar controles ActiveX™ que enseguida veremos). El lector interesado puede adquirir más información a través del Web Consortium en las siguientes direcciones:

- <http://www.w3.org/pub/www/TR/WD-object.html>
- <http://www.w3.org/TR/WD-html40/struct/objects.html#h-14.3>

En la figura 11.9 hemos incluido de modo informativo la página del Web Consortium que explica el funcionamiento del atributo `OBJECT` y todos sus atributos.

Figura 11.9. La etiqueta OBJECT ha sido creada por los expertos del Web Consortium para englobar la inclusión de todo tipo de objetos en las páginas Web.



Una sección que si hemos considerado importante explicar es la inserción de los applets Java™ recién vistos usando la etiqueta applet, ya que aunque actualmente este método sea operativo, se irá imponiendo con el tiempo.

Inserción de un applet Java™.

El objetivo actual del Web Consortium es que la etiqueta `OBJECT` conviva por un tiempo con la etiqueta `APPLET` para sustituirla más adelante. Hoy en día el uso de la etiqueta `OBJECT` no está muy extendido con lo que la forma más segura de insertar un applet sigue siendo usando la etiqueta `APPLET`, sin embargo la siguiente generación de navegadores permitirá también el uso de `OBJECT` que acabará siendo el único método aceptado y como buenos programadores nosotros debemos estar preparados.

Para poder llevar a cabo todas las funciones de la etiqueta `APPLET` también es posible usar la etiqueta `<PARAM>` para especificar parámetros para los applets u otros objetos que los necesiten. Aunque esta etiqueta debe ser insertada entre `<OBJECT>` y `</OBJECT>` no será ignorado por los navegadores que entiendan esta etiqueta.

Además la etiqueta `PARAM` tiene dos nuevos atributos al ser usada en el interior de `OBJECT`, estos son:

`VALUETYPE="..."`: Este atributo especifica que tipo de valor recibirá el parámetro. Hay tres posibles valores:

Data: El valor especificado será pasado al objeto como una cadena de caracteres. Este es el valor por defecto y antes de la existencia del atributo `VALUETYPE` era la única posibilidad.

Ref: El valor especificado es una dirección URL que indica donde están almacenados los valores para dicho parámetro. La dirección debe ser pasada tal cual al objeto.

Object: El valor es el nombre de otro objeto del mismo documento precedido del símbolo '#'.

`TYPE="tipo_mime"`: Este atributo especifica el tipo mime del valor asignado a este atributo con `VALUE`.

Exceptuando estas diferencias, el funcionamiento es idéntico. Por ejemplo, para insertar el applet del segundo de nuestros ejemplos ('El saludo de Duke') usando la etiqueta `OBJECT` debemos escribir:

```
<OBJECT codetype="application/octet-stream"
  classid="Java™:Animator.class"
  width=55 height=68
  ALT="Lo siento, no puede ver el applet">
<PARAM NAME="imagesource" VALUE=".">
<PARAM NAME=ENDIMAGE VALUE=10>
<PARAM NAME=PAUSE VALUE=100>
<PARAM NAME=PAUSES VALUE="2500|100|100|100|100|100|100|100|100|100">
<IMG SRC="T5.gif" WIDTH=55 HEIGHT=68 ALT="Duke te saluda">
</OBJECT>
```

Como vemos la diferencia con la etiqueta `APPLET` es pequeña, con la única complejidad añadida de indicar en el atributo `CODETYPE` el tipo `MIME` del applet, que es el indicado en este ejemplo siempre. Esto es necesario para indicar al navegador que se trata de un applet de modo que si no soporta esta tecnología este pueda ser capaz de decidir no bajar el archivo.

ActiveX™, la tecnología del futuro.

La, denominada, tecnología ActiveX™ desarrollada por Microsoft hizo su aparición en Internet con el navegador Internet Explorer™ 3.0. Su objetivo es similar al de los plug-ins, insertar objetos de diferente tipo en una página Web, aunque va mucho más allá al añadir mayores posibilidades de interacción y comunicación con programas externos. Existen páginas en Internet que basan toda su presentación en controles ActiveX™ para crear Webs realmente impactantes. En la figura 11.10 vemos la página de ForecastX que ofrece el parte meteorológico en tiempo real usando esta novedosa tecnología.

Figura 11.10. ForecastX ofrece un servicio meteorológico en tiempo real gracias a la tecnología ActiveX™.



ActiveX™ esta compuesto por dos tipos de objetos: *controles Activos (Active controls)* y *documentos Activos (Active documents)*. Estos últimos permiten insertar documentos con formato PDF™, DOC™, etc. La tecnología ActiveX™ funciona de una manera similar al mecanismo de Microsoft OLE™ (Object Linking and Embedding,

Enlazado e incrustación de objetos) que usa su sistema operativo Windows™, por lo que los usuarios de éste les resultará familiar, lo realmente novedoso es la aplicación de esta tecnología al WWW.

Los controles ActiveX™ guardan parecidos con los objetos para plug-ins y con los applets Java™, aunque presenta algunas mejoras con respecto a ambos. Como ocurría con los plug-ins, los controles y documentos de ActiveX™ pueden ser insertados en un página Web, sin embargo no requieren un pequeño programilla para cada tipo de objeto ActiveX™, esta nueva tecnología ha sido denominada como *auto-contenida* porque cada objeto tiene suficiente información para ejecutarse él mismo sin ayuda de ninguna aplicación.

Tal y como ocurre con los applets podemos crear nuestros propios controles o usar los creados por otros programadores que nos los venden o distribuyen gratuitamente. Si optamos por programarlos nosotros mismos existen herramientas que lo convierten prácticamente en un juego de niños, entre ellas destaca las creadas para este fin por Microsoft y que distribuye a través de su Web (www.microsoft.com/ie/).

Hoy por hoy el único inconveniente de la tecnología ActiveX™ consiste en que sólo puede ser visto desde es sistema operativo Windows™ y con el Navegador Internet Explorer™, aunque afortunadamente parece que esto cambiará pronto.

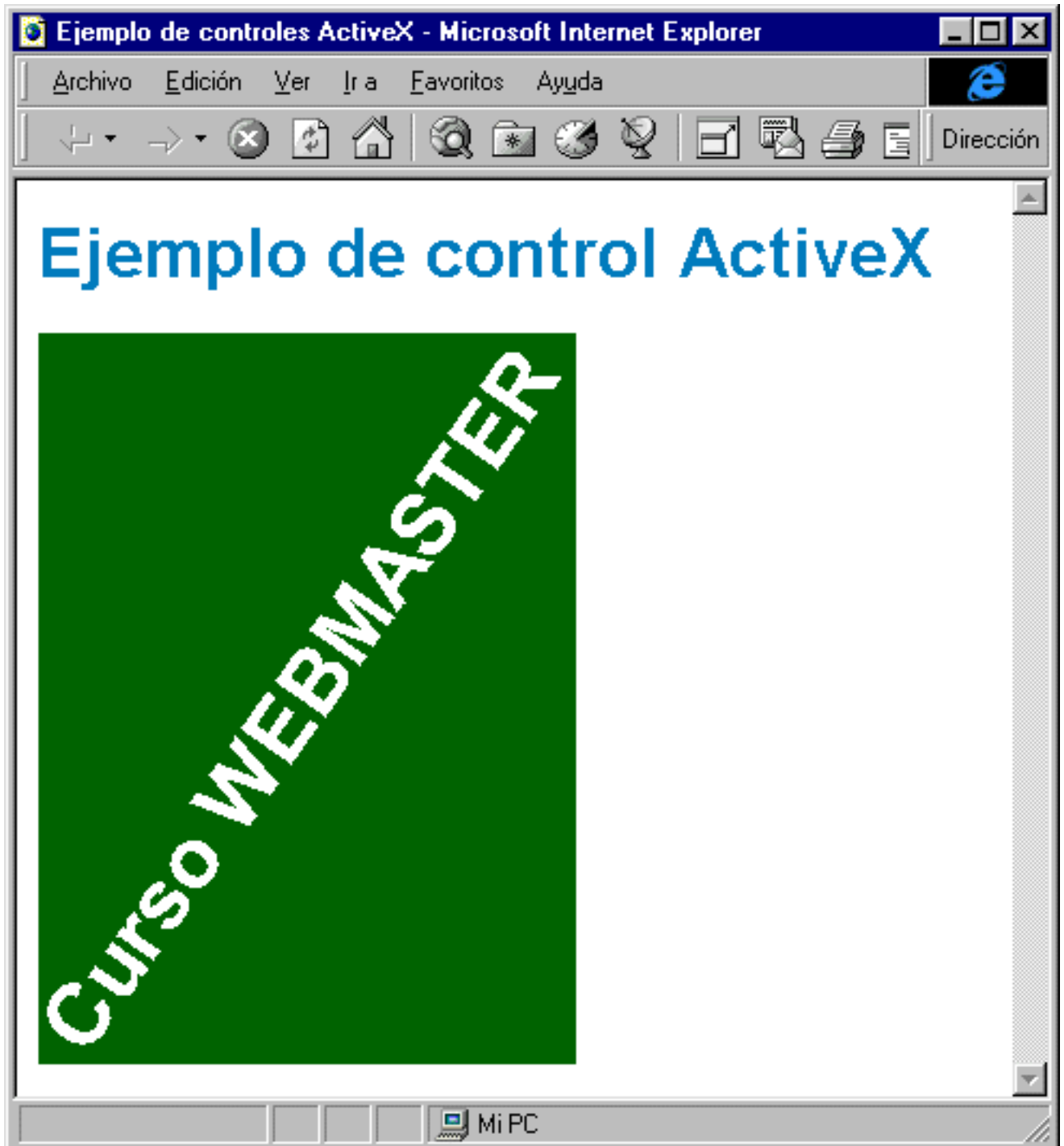
Inserción de un control ActiveX™.

El código HTML necesario para insertar un control o documento ActiveX™ en una página Web fue desarrollado por Microsoft en colaboración con el Web Consortium, con lo que se acordó el uso del estándar OBJECT que hemos visto en la sección anterior. A continuación veremos un sencillo ejemplo que no requiere ningún archivo especial, el único requisito para poder visualizarlo consistirá en tener instalado el navegador Internet Explorer™ 3.0 o superior. El código usado para insertar el control ActiveX™ será:

```
<OBJECT WIDTH=250 HEIGHT=340
CLASSID="clsid:99B42120-6EC7-11CF-A6C7-00AA00A47DD2">
<PARAM NAME="angle" VALUE="55">
<PARAM NAME="alignment" VALUE="2">
<PARAM NAME="BackStyle" VALUE="3">
<PARAM NAME="caption" VALUE="Curso WEBMASTER">
<PARAM NAME="FontName" VALUE="Arial">
<PARAM NAME="FontSize" VALUE="30">
<PARAM NAME="FontBold" VALUE="1">
<PARAM NAME="BackColor" VALUE="#006600">
<PARAM NAME="ForeColor" VALUE="#FFFFFF">
Para visualizar el control ActiveX™ necesita Explorer™ 3.0 o posterior.
</OBJECT>
```

En la figura 11.11 podemos ver el resultado obtenido tal y como se ve con Internet Explorer™ 4.0. Pasemos a explicar el este código: El valor del atributo CLASSID es el que identifica el tipo de control ActiveX™ que estamos insertando. Este código es complejo y se sale de los objetivos de esta obra.

Figura 11.11. Usando este control ActiveX™ podemos crear efectos variados con texto, con lo que pueden sustituir el uso de imágenes con la consiguiente disminución tiempo de carga de nuestras páginas.

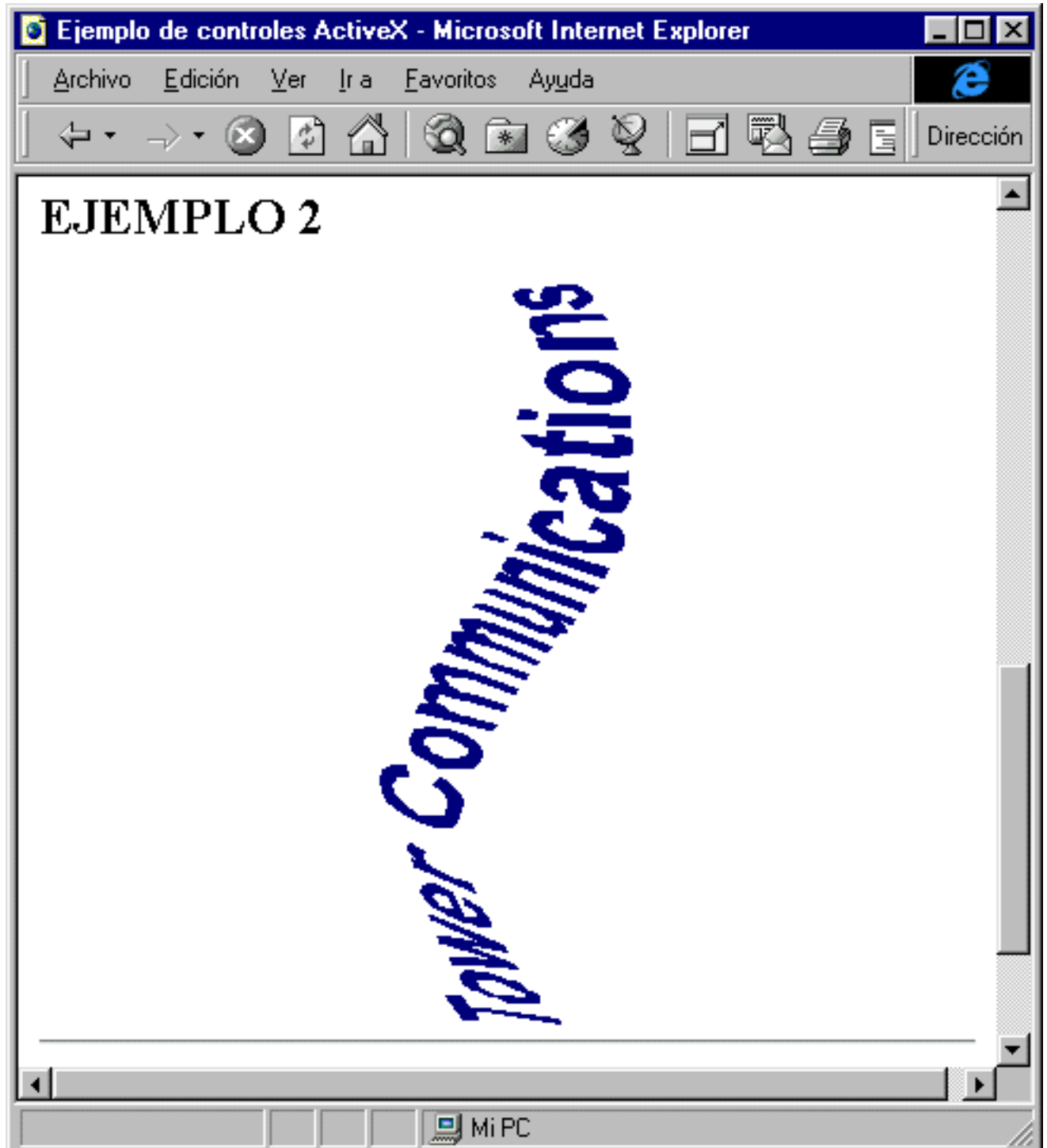


Volviendo a nuestro ejemplo, entre los principales parámetros usados en la inserción de este control ActiveX™ encontramos:

<i>Caption:</i>	Es el texto que debe insertarse con las características que se indican con el resto de parámetros.
<i>Angle:</i>	Es el ángulo del texto con la horizontal en grados. Si le damos el valor 90 el control mostrará el texto en vertical.
<i>FontSize:</i>	Es el tamaño del texto.
<i>FontName:</i>	Es el tipo de letra.
<i>BackColor</i> y <i>ForeColor:</i>	Son el color del fondo y el color del texto respectivamente.

Invitamos al lector a hacer pruebas modificando los valores de estos y el resto atributos y cambiando el texto para adaptarlo a nuestras necesidades y poder incorporarlo en sus propias páginas. Vemos un ejemplo en la figura 11.12.

Figura 11.12. Usando el mismo control ActiveX™ que en la figura 11.25 pero con un código algo más complejo podemos crear este efecto tan atractivo.



Por último nos queda resaltar que existe un plug-in para Netscape Navigator™, *Ncompass*, que añade a este navegador la posibilidad de entender controles y documentos ActiveX™. Este plug-in puede ser obtenido en la página Web de Ncompass en el WWW: www.ncompasslabas.com. Si pensamos incorporar la tecnología

ActiveX™ a nuestras páginas es muy recomendable incluir un enlace a esta página para que todos los usuarios de Netscape que las visiten puedan adquirirlo y disfrutar, también, de los controles que hayamos usado. También es importante destacar que Microsoft ha llegado a acuerdos con diversas compañías para exportar la tecnología ActiveX™ a Mac™ y UNIX, con lo que es de esperara que en poco tiempo se convierta en un método estándar para insertar objetos multi-plataforma aunque hoy por hoy no se puede decir que la sea, al contrario que Java™.

Formularios.

Jorge Ferrer
Rodrigo García

Versión 1.0

Copyright Jorge Ferrer y Rodrigo García. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation. Puede consultar una copia de la licencia en: <http://www.gnu.org/copyleft/fdl.html>

Para más detalles consultenos a través de la dirección jferrer@acm.org

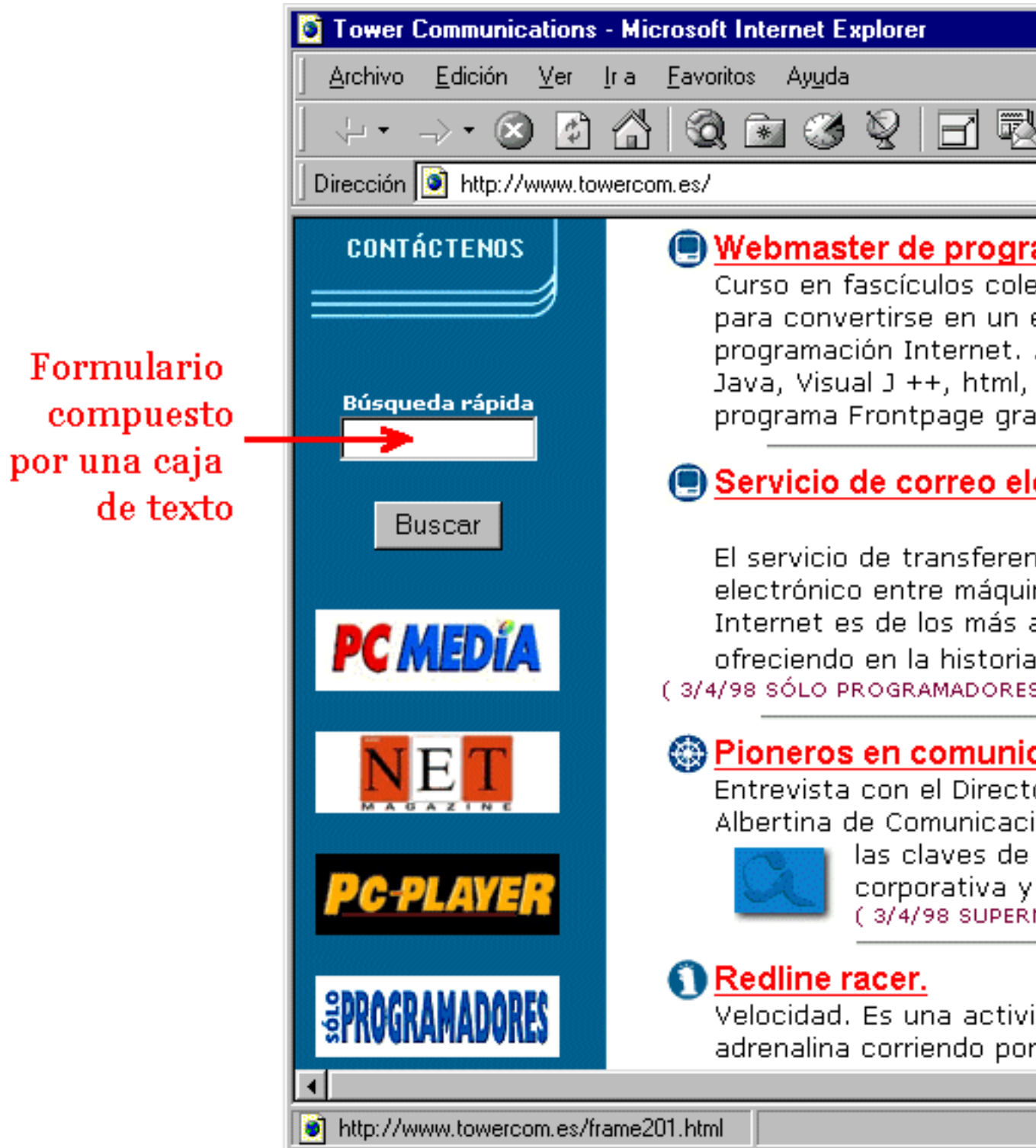
La habilidad de interactuar con los usuarios es una de las principales características de los ordenadores y las redes informáticas. La posibilidad de realizar preguntas y recibir respuestas es una de las formas fundamentales de conseguir esta interactividad y el lenguaje HTML proporciona la habilidad de crear formularios e insertar cajas de texto, botones de opción y otros controles que nos permitirán aprovechar esa posibilidad de interactividad en nuestras páginas Web.

Formularios en HTML.

No es necesario navegar mucho tiempo por Internet para encontrar páginas Web en las que se piden datos a los navegantes. Quizá las primeras páginas que hicieron uso de esta posibilidad fueron las de los sistemas de búsqueda. Los formularios usados en estos sistemas son, en general, bastantes sencillos, únicamente es necesario un elemento, conocido como caja de texto, donde sea posible introducir la palabra o frase que deseamos buscar.

En la figura 12.1 vemos un buen ejemplo de este tipo de uso. En ella vemos la página principal de Tower Communications se ofrece una caja de texto donde el usuario debe introducir una cadena de texto que desea buscar. Después debe pulsar sobre el botón ENVIAR, tras lo cual se mandará la información al servidor Web que la procesará y responderá enviando la información encontrada. Más adelante en la sección 'Como usar los datos de un formulario' veremos como es posible recoger y procesar esta información, pero por ahora nos centraremos en la creación y en el diseño de los formularios.

Figura 12.1. En la página principal de Tower Communications vemos un formulario que es usado para permitir a los visitantes introducir una cadena de texto que deseen buscar.



Un caso más complejo de formularios es usado en las páginas en las que se ofrece a los navegantes la posibilidad de realizar sugerencias, ofrecer comentarios o mandar cualquier otro tipo de información. En general estos

formularios ocupan una página entera, y están formados por diversos elementos. En la página de Tower podemos encontrar una página de sugerencias que ilustra la utilidad de esta forma de usar los formularios. En la figura 12.2 podemos apreciar esta página, que como vemos permite introducir datos en varios campos, tras rellenarlos deberemos pulsar sobre el botón 'Enviar formulario' para mandar la información.

Figura 12.2. En otra sección de la página de Tower Communications vemos un formulario más complicado formado por numerosas cajas de texto y otros elementos.

The image shows a screenshot of a Microsoft Internet Explorer browser window. The title bar reads "Tower Communications - Microsoft Internet Explorer". The address bar shows the URL "http://www.towercom.es/frame15.html". The page content includes a blue header with the "TOWER" logo and navigation links: "noticias", "Revistas", "Libros", "Multimedia", "Cursos", "Towerlabs", and "Turismo". The main heading is "Contacte con Tower Communicat". Below it, a blue text prompt says: "Para enviar sus sugerencias, opiniones o comentarios, COMPLETE EL SIGUIENTE FORMULARIO. Gracias." The form itself consists of several fields: a dropdown menu for "Seleccione destinatario de su mensaje" (currently set to "Servicio técnico"), text boxes for "Nombre y apellidos", "Domicilio", "Ciudad", "Cód. postal", "Provincia", "Teléfono", "E-mail", and "Fecha de nacimiento". The browser's status bar at the bottom shows "Listo".

Estos dos ejemplos son probablemente los más comunes entre las páginas del WWW, pero no los únicos. Los

formularios son muy versátiles, como enseguida veremos, y la posibilidad de interactuar con los usuarios de las páginas será útil en multitud de ocasiones. El único límite será nuestra creatividad.

Si queremos preguntar a los visitantes de nuestras páginas su nombre, su dirección de correo electrónico, crear un sistema de búsqueda, hacer un libro de visitas o simplemente pedir sugerencias necesitaremos crear formularios. Un formulario no es más que un conjunto de elementos que permiten introducir datos a los visitantes de nuestras páginas de una manera sencilla. Estos datos son en general texto o una serie de opciones entre las que se debe elegir una o varias. En este capítulo aprenderemos todas las posibilidades que tenemos y como usarlas adecuadamente.

El lenguaje HTML consta de una serie de etiquetas que permitirán crear de forma rápida y sencilla numerosos elementos de entrada de datos. Estos elementos, que reciben el nombre de controles, serán gráficos en la mayoría de navegadores, pero también existen métodos para que los navegadores que sólo pueden mostrar texto, como Lynx™, puedan mostrarlos. Un formulario no es más que un conjunto de estos controles cuya información será enviada conjuntamente cuando el usuario pulse sobre el botón de envío.

Para crear un formulario el lenguaje HTML proporciona la etiqueta *FORM*. Al contrario que la mayoría de etiquetas que hemos visto en los últimos capítulos ésta existe desde la especificación HTML 2.0 y por tanto es entendida prácticamente por cualquier navegador existente actualmente, por lo que puede ser usada con total libertad.

Esta etiqueta consta de una instrucción de inicio, `<FORM>`, y una instrucción de fin, `</FORM>`, entre las cuales podremos insertar todos los controles que deseemos. Dicho esto podemos crear nuestra primera página con un formulario. En ella adelantaremos uno de los controles más usados y sobre el que ya hemos hablado antes: la caja de texto. Un poco más adelante la trataremos más en detalle, pero la usaremos en este ejemplo para hacernos una idea de cómo se crea un formulario en una página Web. Para empezar, como siempre que creamos una página nueva, creamos un archivo nuevo en nuestro editor de texto o nuestro editor HTML. En él debemos insertar el siguiente código:

```
<HTML>
<HEAD>
<TITLE>Formulario de ejemplo</TITLE>
</HEAD>
<BODY>
<H1>FORMULARIO DE EJEMPLO</H1>
<FORM>
Introduzca su nombre: <INPUT TYPE="Text">
</FORM>
</BODY>
</HTML>
```

Guardamos el archivo con un nombre de nuestra elección y lo abrimos con el navegador. El resultado será muy similar, si no igual, al que apreciamos en la figura 12.3. La caja de texto es el elemento que sigue al texto *'Introduzca su nombre:'*. En este caso estamos usando para visualizarla el navegador Internet Explorer™ para Windows™ 95, por lo que la caja de texto tiene el aspecto habitual en este sistema operativo. Si visualizamos este mismo código desde un navegador de UNIX™ o Macintosh™ la caja de texto sería mostrada con la apariencia habitual en estos entornos.

Figura 12.3. Nuestro primer formulario está compuesto por una caja de texto. Para introducirla hemos usado la etiqueta `INPUT` con `TYPE="text"`.



La gran novedad de la caja de texto respecto a todos los elementos de las página Web que hemos visto hasta ahora es que permite a los usuarios introducir texto en ella. Para ello no tenemos más que pulsar con el ratón sobre ella (o seleccionarla en el caso de navegadores de texto) y escribir como si estuviésemos en un editor de texto. El texto aparecerá según lo vamos escribiendo. Una vez hemos escrito nuestro nombre (ver figura 12.4) enviaremos los datos pulsando la tecla ENTER.

Figura 12.4. Pulsando sobre la caja de texto nos aparecerá un cursor parpadeante que nos indica que podemos empezar a escribir texto.



Nota

En este ejemplo todavía no hemos indicado al navegador dónde debe enviar los datos. Esto se hace con el atributo *ACTION* de la etiqueta *FORM*, que será visto un poco más adelante. Al no indicar la dirección de destino el navegador no mandará los datos al pulsar la tecla ENTER (o pulsar el botón de envío), aunque por ahora nos olvidaremos de este detalle.

Pero ¿no existía un botón de envío? Si, en los ejemplos de la figura 12.1 y 12.2 veíamos que ambos formularios constaban de un botón de envío sobre el que había que pulsar para mandar los datos. En los casos en los que el formulario conste de más de un control, es decir, de más de un campo de entrada de datos, será necesario incluir el botón. Si el formulario simplemente tiene uno, como ocurre en este primer ejemplo que hemos realizado nosotros, el navegador es capaz de entender que al pulsar la tecla ENTER el usuario quiere mandar los datos de ese formulario, y por tanto el uso de botón de envío es opcional.

Nota

Los botones de envío son mostrados como enlaces normales en los navegadores de sólo texto, y por tanto sería más correcto usar también el termino 'enlace para enviar el formulario' . Sin embargo, por simplicidad, seguiremos usando únicamente el término botón.

Bien, pero ¿cómo se crea un botón de envío? El lenguaje HTML consta de una etiqueta, *INPUT*, que permite insertar varios tipos de controles, entre los que se encuentra el botón de envío. Para indicar el tipo de control que queremos insertar se usa el atributo *TYPE*. Para crear el botón de envío debemos usar *TYPE="submit"*. Por tanto si en el ejemplo anterior queremos insertar uno, debemos añadir esta etiqueta (de ahora en adelante omitiremos las etiquetas *HTML*, *HEAD* y *BODY*, pero por supuesto debe seguir usándose):

```
<H1>FORMULARIO DE EJEMPLO</H1>
<FORM>
<P>Introduzca su nombre: <INPUT TYPE="Text">
<P><INPUT TYPE="Submit">
</FORM>
```

Si visualizamos este código con un navegador, veremos que la caja de texto está acompañada ahora de un botón con un texto que invita a pulsarlo para mandar los datos. En la figura 12.5 vemos que al visualizar este ejemplo con Internet Explorer™ 4.0 aparece el texto *'Enviar consulta'* en el botón, este texto ha sido puesto por defecto por este navegador, pero si visualizamos este código con otro puede cambiar. Si el navegador usado es una versión inglesa, el texto del botón estará en inglés. Esta situación es poco deseable en la mayoría de las ocasiones, por los que se creó un atributo, *VALUE*, que nos permitirá especificar nosotros mismos el texto que queremos que sea mostrado en el botón. Así en podemos reescribir el código para insertar el botón como:

Figura 12.5. Usando el código `<INPUT TYPE="submit">` podemos insertar un botón de envío de manera que al pulsar sobre él se envían los datos del formulario.



```
<P><INPUT TYPE="Submit" VALUE="Envíeme">
```

Con lo que el resultado obtenido sería el que apreciamos en la figura 12.6. En este caso en el botón el mensaje es *'Envíeme'* y será éste siempre, independientemente del navegador que usemos para visualizar la página. En esta misma figura podemos apreciar que el tamaño del botón se adapta a la longitud del texto que pongamos en él.

Figura 12.6. Con el atributo `VALUE` podemos cambiar el texto por defecto del botón de envío por uno de nuestra elección. El ancho del botón se adaptará a la longitud del texto.



Aunque lo habitual es incluir un único botón de envío el lenguaje HTML permite la posibilidad de incluir varios. En este caso debemos usar un nuevo atributo, `NAME`, para dar un nombre a cada uno que permita al servidor diferenciar cual se ha pulsado. Este nombre será enviado junto con los datos cuando se pulse sobre el botón. De esta forma es posible realizar acciones diferentes dependiendo de sí se pulsa uno u otro. Por ejemplo podían incluirse botones que simulasen distintas direcciones a las que acceder tras introducir los datos:

```
<INPUT TYPE="Submit" VALUE="Arriba" NAME="arriba">
<INPUT TYPE="Submit" VALUE="Izquierda" NAME="izquierda">
<INPUT TYPE="Submit" VALUE="Derecha" NAME="derecha">
<INPUT TYPE="Submit" VALUE="Abajo" NAME="abajo">
```

Es importante indicar que el atributo `NAME` no debe usarse únicamente para el botón de envío. A continuación aprenderemos a insertar los distintos tipos de controles existentes y crearemos formularios más complejos que hasta ahora, por esta razón deberemos usar `NAME` en cada uno de los controles. Cuando se pulse el botón de envío se enviará los datos de cada control junto con el nombre especificado con este atributo. De esta forma el servidor podrá saber el control en que se ha insertado cada dato. Recordemos brevemente todos los atributos de la etiqueta `INPUT` cuando la usamos para insertar el botón de envío:

```
<INPUT TYPE="Submit" NAME="Nombre" VALUE="Texto del botón">
```

Pasemos, sin esperar más tiempo, a estudiar cada uno de los controles existentes en el lenguaje HTML.

Controles de Formularios.

Todos los controles que pueden usarse a la hora de crear un formulario permitirán al navegante insertar información. Unas veces esta información consistirá en la inserción de un texto, que podrá ser de una sola línea o varias y que podrá estar oculto o no. En otras ocasiones se ofrecerá entre una serie de opciones entre las que simplemente hay que elegir una o varias. Todo ello se podrá realizar de diferentes formas que ahora mismo comenzamos a estudiar.

Para empezar vamos a ver una serie de controles todos los cuales se crean con la etiqueta `INPUT`. Para diferenciar entre unos controles y otros se hará uso del atributo `TYPE` que puede tomar los siguientes valores: `text`, `password`, `radio`, `checkbox`, `submit`, `image`, `reset`, `file` y `hidden`. Además del atributo `TYPE`, esta etiqueta consta de algunos más cuya función variará del tipo de control estemos insertando, es decir, en función del valor del atributo `TYPE`. Uno común a todos es `NAME`, ya mencionado, con el que damos un nombre al control, y que es recomendable usarlo siempre. Otro también común a todos será `ALIGN`, que explicaremos con las cajas de texto.

Una vez hecho este comentario, podemos empezar a estudiar cada uno de los tipos de control uno a uno.

Cajas de texto.

Ya hemos introducido antes las cajas de texto. Como resumen recordaremos que para insertar este tipo de control en una página Web se usa la etiqueta `INPUT` (esta misma etiqueta será la utilizada para insertar la mayoría de controles), con `TYPE="text"`. Es decir:

```
<FORM>
<INPUT TYPE="text">
</FORM>
```

Si nuestra intención es insertar más de un control será conveniente darle un nombre a la caja de texto. Para ello usaremos, tal y como hacíamos con el botón de envío, el atributo `NAME`:

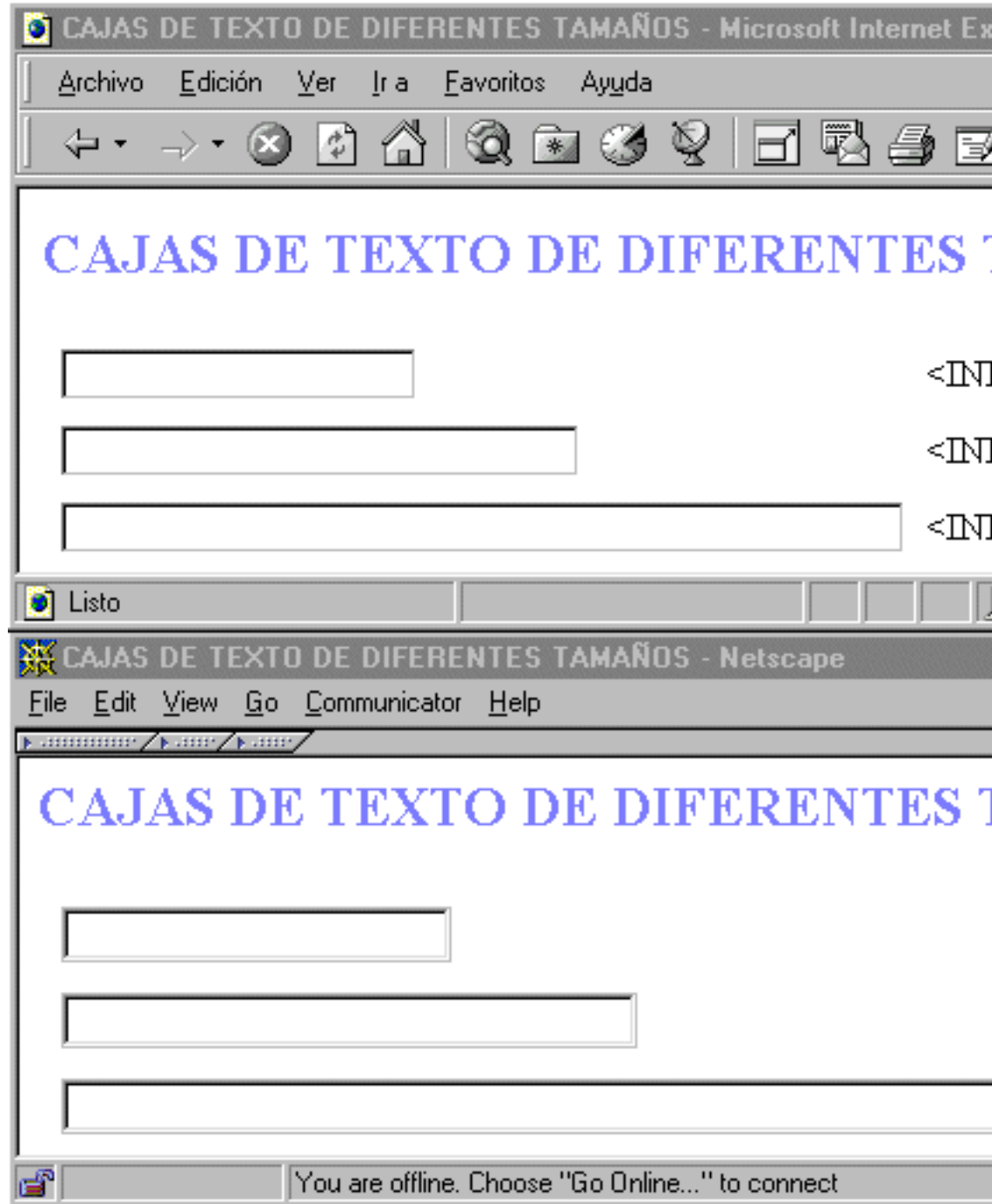
```
<FORM>
<INPUT TYPE="text" NAME="mitexto">
</FORM>
```

Como podemos apreciar la etiqueta `INPUT` consta de una única instrucción. Ya hemos visto los atributos `TYPE` y `NAME` que existe siempre en la etiqueta `INPUT`. Cuando insertamos una caja de texto, es decir, cuando ponemos `TYPE="text"` existen otros dos:

SIZE:

Determina la anchura de la caja de texto. El valor por defecto de este atributo es 20 caracteres, esta es la longitud de los ejemplos que veíamos en las figuras 12.3 a 12.6. En ocasiones convendrá decrementar o incrementar este valor por defecto, pero en todo caso siempre será conveniente mantener un valor menor de 50 caracteres de manera que la caja de texto quepa en la mayoría de pantallas. Un problema adicional que dificulta una correcta elección del tamaño de la caja es que este tamaño será considerado de distinta forma por los distintos navegadores y más aún si estos son de distintos sistemas operativos. En la figura 12.7 podemos ver una serie de cajas de texto de diferentes tamaños y las diferencias entre Internet Explorer™ y Netscape™ Navigator™ al mostrar el mismo código.

Figura 12.7. El atributo **SIZE** nos permite variar la longitud de la caja de texto. Debemos tener en cuenta que en diferentes navegadores un mismo valor no se corresponde a una misma longitud.



MAXLENGTH:

Con este atributo limitamos el número máximo de caracteres que pueden ser escritos en una caja de texto. El valor de este atributo puede ser mayor o menor que el especificado en *SIZE*, y que es totalmente independiente. Si es mayor cuando lleguemos al final de la caja de texto los nuevos caracteres que insertemos irán desplazando hacia la izquierda a los primeros, que dejarán de estar a la vista (OJO, esto no quiere decir que sean borrados). Para entender el funcionamiento es mejor comprobarlo *in situ* con unos ejemplos. Recomendamos al lector que pruebe a visualizar los siguientes códigos, y que intente escribir una frase algo larga en cada uno de ellos:

```
<INPUT TYPE="text" SIZE="20" MAXLENGTH="20">
<INPUT TYPE="text" SIZE="20" MAXLENGTH="5">
<INPUT TYPE="text" SIZE="20" MAXLENGTH="40">
```

Por último queda decir que si no usamos el atributo *MAXLENGTH* el número de caracteres que pueden introducirse en la caja de texto no tendrá límite.

VALUE:

Sirve para especificar un texto que debe aparecer por defecto en la caja de texto, antes de que el usuario escriba nada. Este texto suele ser, en general, o bien instrucciones o bien la respuesta más probable. Veamos un ejemplo que ilustra estos dos casos:

```
<FORM>
<P>Nombre:<INPUT TYPE="text" VALUE="Introduzca aquí su nombre">
<P>¿Le gusta nuestro Web? <INPUT TYPE="text" VALUE="Si, muchísimo">
</FORM>
```

Invitamos al lector a que pruebe este ejemplo en su navegador y compruebe los resultados.

Alineamiento de controles.

Tal y como hemos indicado antes, todos los controles que insertamos con la etiqueta *INPUT* tienen un atributo, llamado *ALIGN*, que nos permitirá seleccionar entre varios tipos de alineamiento. Este atributo apareció en el estándar *HTML 3.2*, al contrario que el resto de etiquetas y atributos que hemos visto hasta ahora en este capítulo, que existen desde la versión anterior del estándar (*HTML 2.0*).

El atributo *ALIGN* puede tomar los siguientes valores:

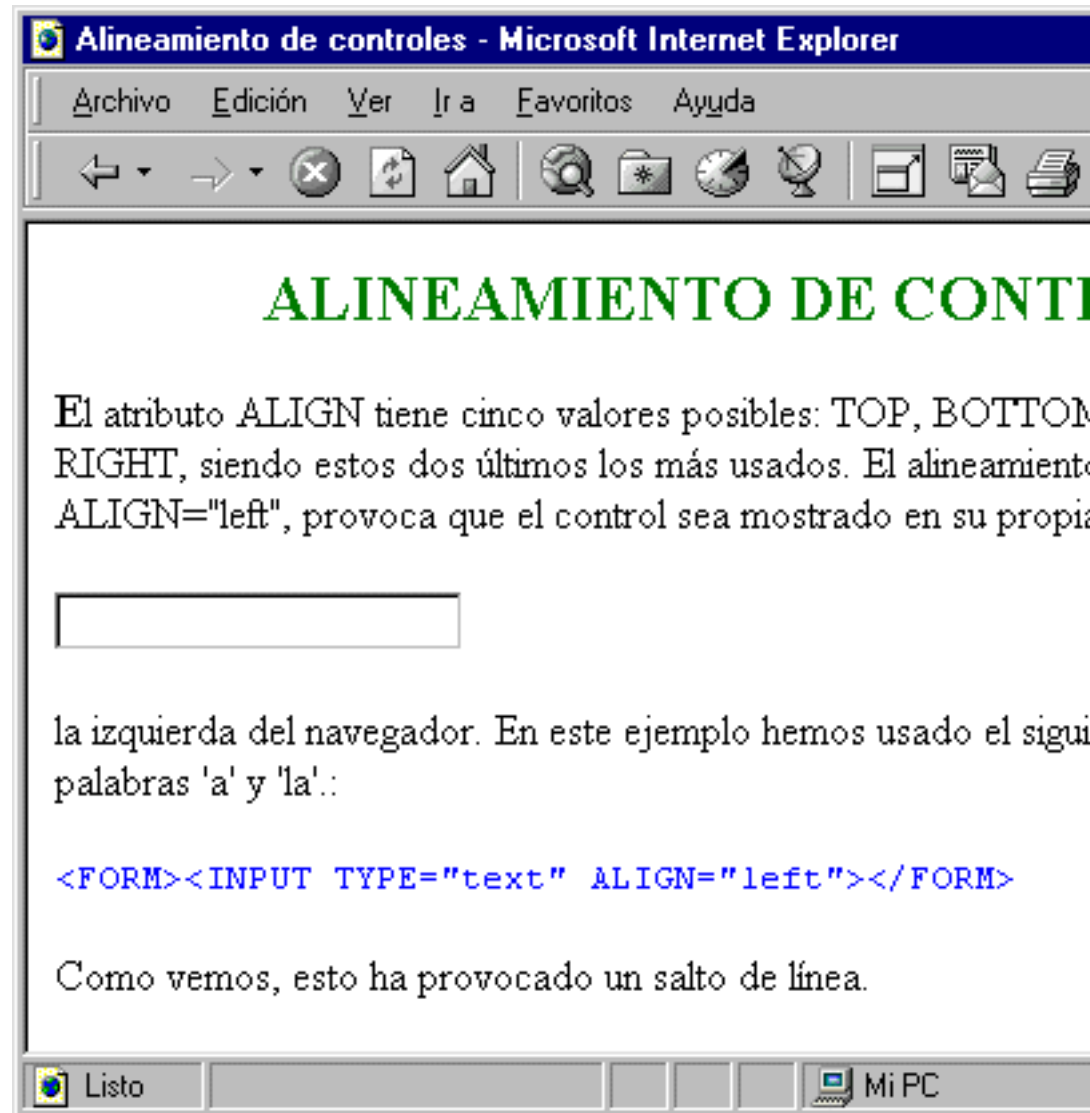
ALIGN="top": Alinea verticalmente el control con la parte superior de la línea en que es insertado.

ALIGN="bottom": Alinea verticalmente el control con la parte inferior de la línea.

ALIGN="middle": Sitúa el control a una altura media entre el resto de elementos de la línea.

ALIGN="left": En este caso estamos alineando el control horizontalmente a la izquierda. Al contrario de lo que ocurría con las imágenes y con las tablas, el texto no bordeará el control por su derecha, situándose éste en una línea propia. En la figura 12.8 podemos ver un ejemplo de este tipo de alineamiento y del siguiente:

Figura 12.8. El atributo `ALIGN` de la etiqueta `INPUT` nos permite alinear los controles. En este caso vemos una caja de texto alineada a la izquierda (`ALIGN="left"`).



`ALIGN="right"`:

Este valor es idéntico al anterior en funcionamiento, sólo que ahora el control se situará a la derecha de la ventana del navegador.

Todos el atributo `ALIGN` y todos estos valores existen para todos los controles que veamos a partir de ahora y que usen la etiqueta `INPUT`.

Para terminar con las cajas de texto veamos un resumen de todos sus atributos y sus funciones:

```
<INPUT TYPE="text" NAME="nombre" VALUE="valor por defecto" SIZE="tamaño"
MAXLENGTH="longitud_máxima" ALIGN="alineamiento">
```

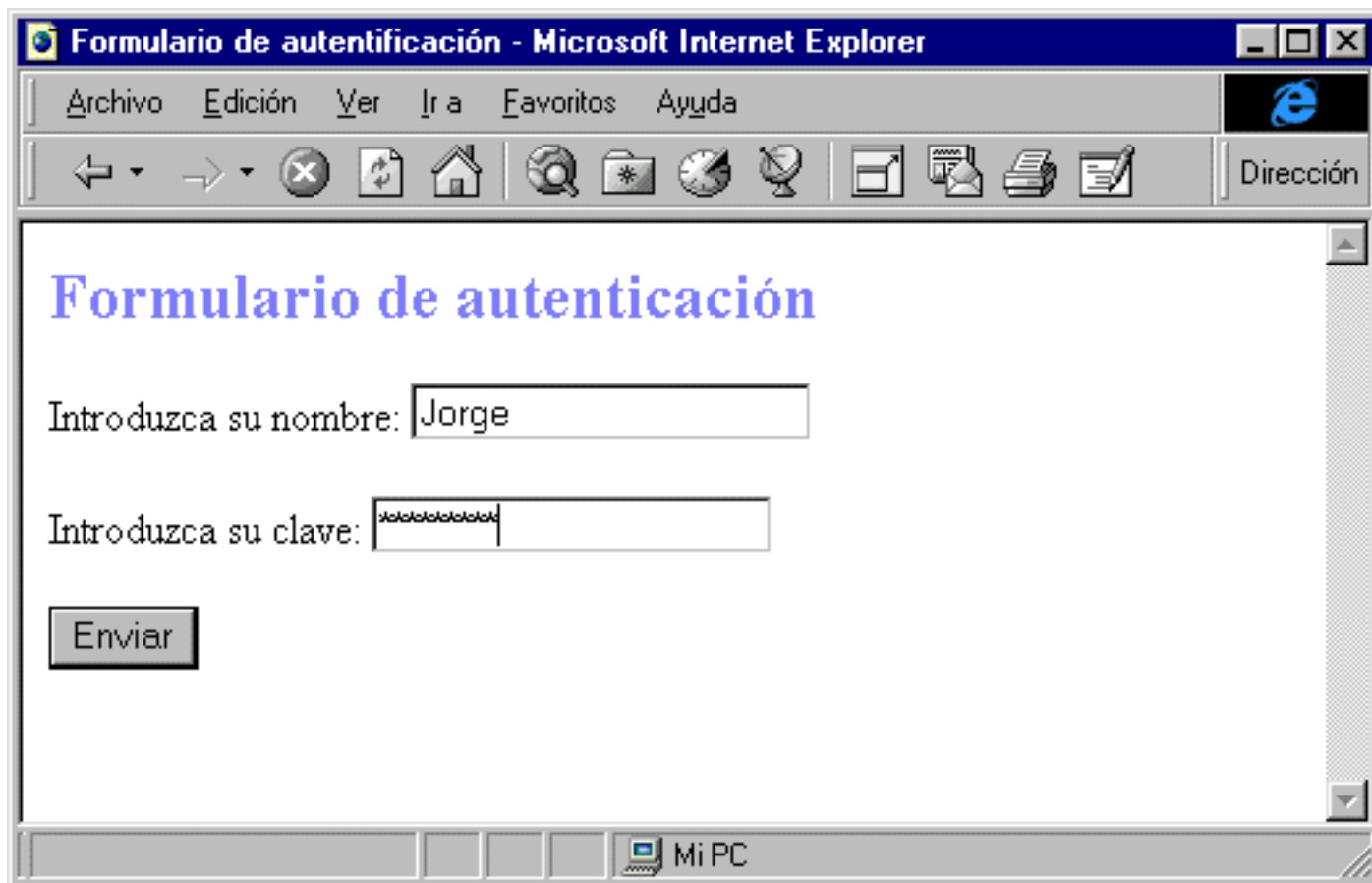
Cajas de texto para claves.

En ocasiones puede ser conveniente pedir al usuario algún tipo de información confidencial como puede ser una clave de acceso o *password*. Debemos tener en cuenta que algunos usuarios pueden estar en algún lugar público o, al menos, con gente alrededor que pueden leer lo que escriba en la pantalla y por tanto no le interesará que lo que escribe sea mostrado en ésta. El lenguaje HTML permite utilizar, en estos casos, una caja de texto modificada en la que al escribir se mostrarán únicamente asteriscos. Para ello debemos usar `TYPE="password"` en la etiqueta `INPUT`. Un ejemplo típico de uso de `password` es al pedir una identificación, por ejemplo para entrar en algunas secciones de una página. Para crear este formulario podemos usar el siguiente código:

```
<H2><FONT COLOR="#8080FF">Formulario de autenticación</FONT></H2>
<FORM>
<P>Introduzca su nombre: <INPUT TYPE="text" NAME="nombre">
<P>Introduzca su clave: <INPUT TYPE="password" NAME="clave">
<P><INPUT TYPE="submit" VALUE="Enviar">
</FORM>
```

Visualizando este código con Explorer™ obtenemos el resultado de la figura 12.9. En ella vemos como el texto que hemos escrito en el segundo campo, el de la caja de texto para passwords, no ha sido mostrado siendo sustituido por asteriscos.

Figura 12.9. Las cajas de texto para claves muestran asteriscos en lugar del texto que escribimos para aumentar la confidencialidad.



Aviso

La caja de texto para claves o passwords simplemente impide que el texto que se escribe sea mostrado en pantalla, sin embargo este texto no es codificado al ser mandado al servidor Web, y por tanto no es seguro mandar datos confidenciales.

Las cajas de texto para claves también tienen los atributos `SIZE`, `MAXLENGTH`, `VALUE` y `ALIGN` que vimos en las cajas de texto. Al igual que hicimos con estas, veamos un resumen de todos los atributos de las cajas de texto para claves:

```
<INPUT TYPE="password" NAME="clave" SIZE="tamaño" MAXLENGTH="longitud_máxima"
VALUE="clave por defecto" ALIGN="alineamiento">
```

Botones de elección.

Estos controles reciben también el nombre de botones de radio, como traducción directa de su denominación inglesa *radio buttons* y porque en general los navegadores suelen darles una forma circular. Para insertar un botón de elección usaremos de nuevo la etiqueta `INPUT`, pero esta vez el valor del atributo `TYPE` será `radio`:

```
<FORM>
<INPUT TYPE="radio">
</FORM>
```

Este tipo de controles tiene dos estados o posiciones: seleccionado (`ON`) o no seleccionado (`OFF`), estando inicialmente todos en la posición `OFF`. También es posible especificar que un botón determinado esté seleccionado inicialmente insertando el atributo `CHECKED` en la etiqueta `INPUT` de la siguiente manera:

```
<INPUT TYPE="radio" CHECKED>
```

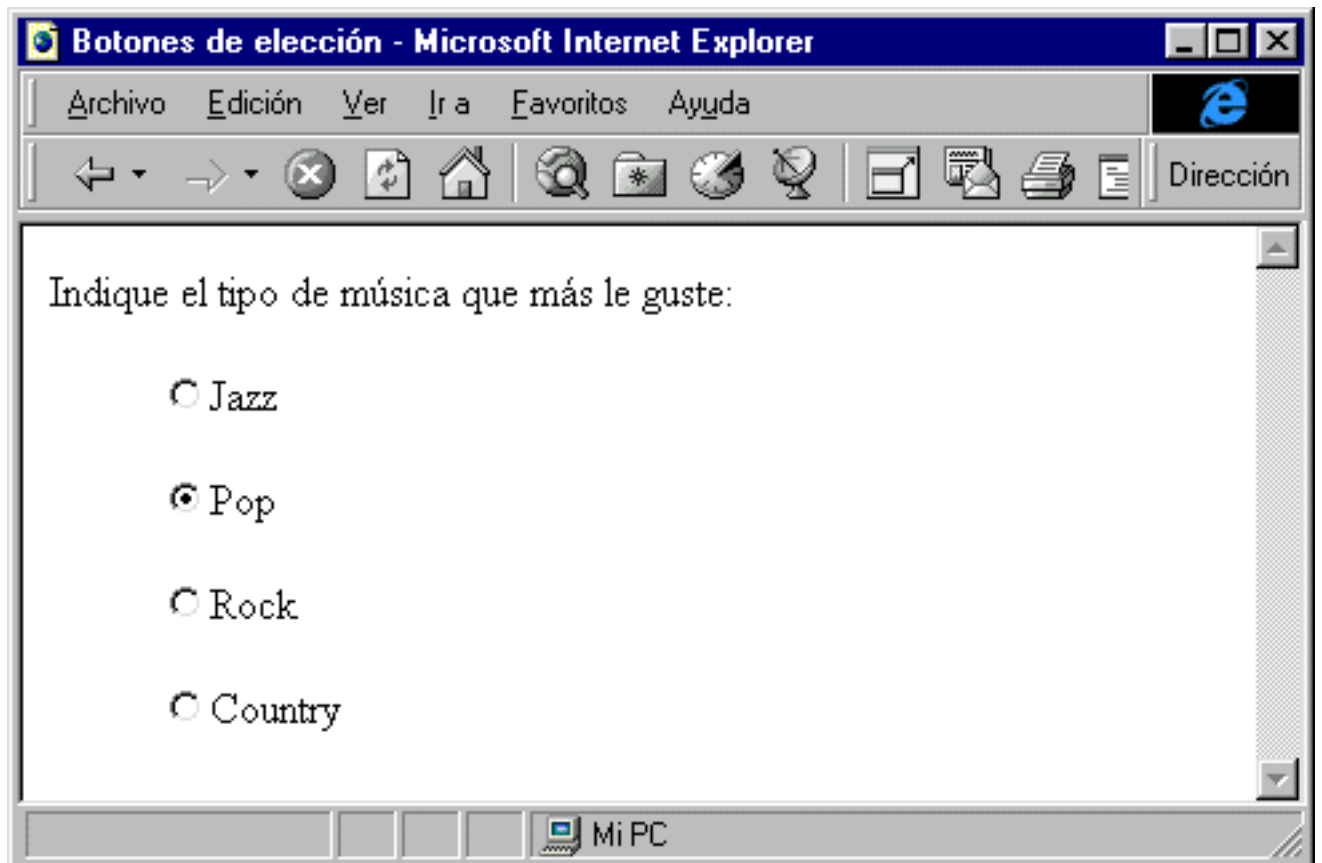
Los botones de elección suelen ser insertados en los formularios en grupos, dando al usuario la posibilidad de elegir entre una serie de opciones. Cuando pulsamos sobre un botón de radio le pasamos a la posición `ON` y permanecerá en ese estado hasta que pulsemos en otra opción del mismo grupo. Esto es así porque entre los botones de radio de un mismo grupo sólo uno de ellos puede estar seleccionado, por tanto cuando seleccionamos uno, aquel que estuviese seleccionado previamente dejará de estarlo.

Para indicar que una serie de botones de elección pertenecen a un mismo grupo debemos incluir el mismo valor en el atributo `NAME` en todos ellos. Además debemos usar el atributo `VALUE` para dar un nombre distinto a cada uno de los botones. Veamos un ejemplo:

```
Indique el tipo de música que más le guste:
<FORM>
<P><INPUT TYPE="radio" NAME="musica" VALUE="Jazz">Jazz
<P><INPUT TYPE="radio" NAME="musica" VALUE="Pop">Pop
<P><INPUT TYPE="radio" NAME="musica" VALUE="Rock">Rock
<P><INPUT TYPE="radio" NAME="musica" VALUE="Country">Country
</FORM>
```

Con este código obtenemos el resultado que apreciamos en la figura 12.10. En ella vemos que únicamente el botón de radio correspondiente a 'Pop' está seleccionado. Invitamos al lector a que pruebe este ejemplo y observe cómo al seleccionar otro de los botones de elección el anterior deja de estar seleccionado.

Figura 12.10. Los botones de elección permiten elegir entre un conjunto de opciones, de las cuales podemos elegir sólo una.

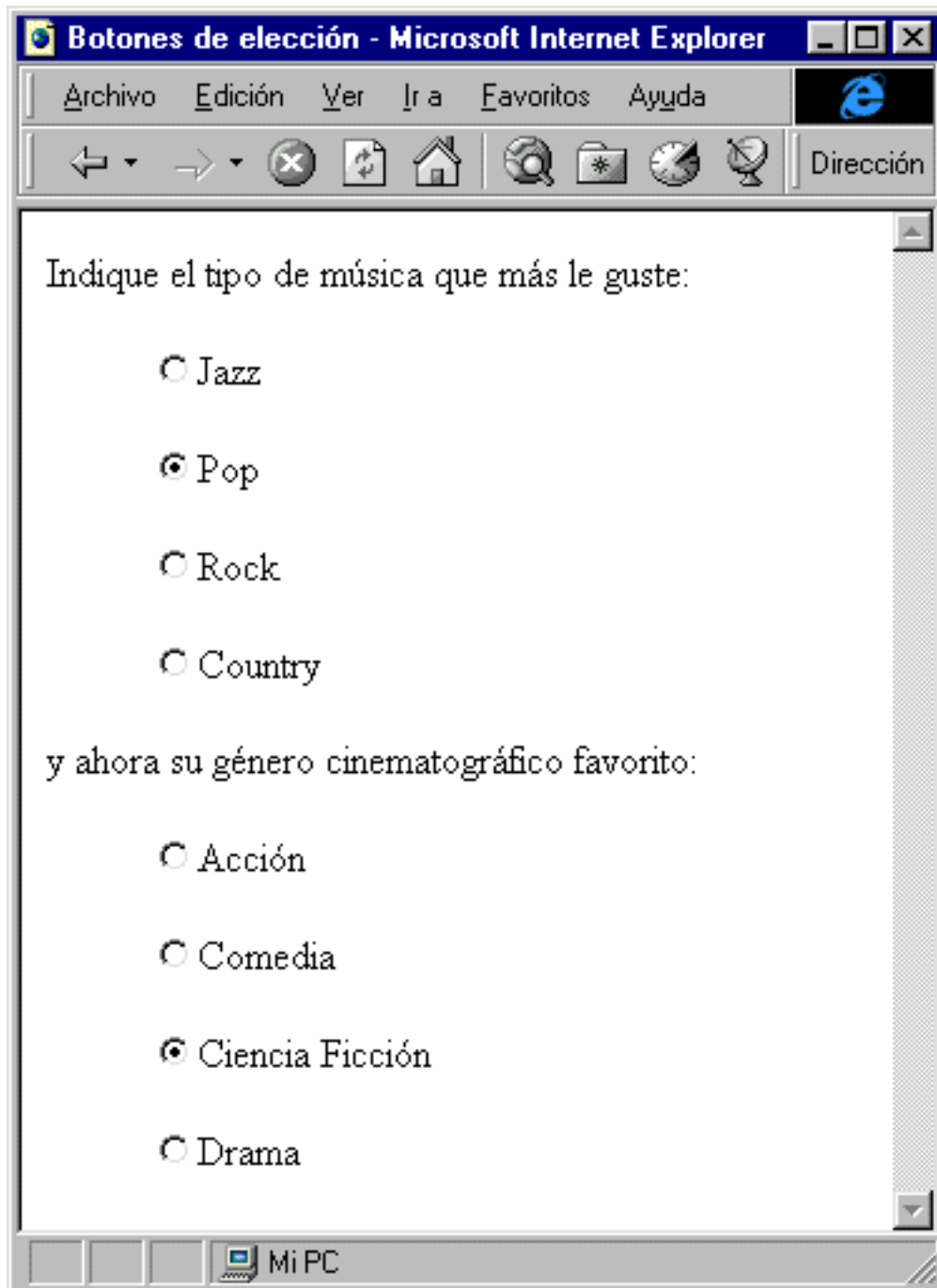


Por supuesto, es posible incluir varios grupos de botones de selección dentro de un mismo formulario. Por ejemplo al código anterior podríamos añadirle (antes de </FORM>) este otro:

```
<P><INPUT TYPE="radio" NAME="cine" VALUE="Accion">Acci&oacute;n;
<P><INPUT TYPE="radio" NAME="cine" VALUE="Comedia">Comedia
<P><INPUT TYPE="radio" NAME="cine" VALUE="CienciaFiccion">Ciencia-Ficci&oacute;n;
<P><INPUT TYPE="radio" NAME="cine" VALUE="Drama">Drama
```

De esta manera tendríamos el formulario de la figura 12.11 en la que vemos que se ha podido seleccionar 'Pop' como tipo de música favorito y 'Ciencia-ficción' como género cinematográfico preferido, al pertenecer los botones a distintos grupos.

Figura 12.11. En un mismo formulario podemos incluir varios grupos de botones de elección. Los visitantes podrán seleccionar una opción en cada uno.



Quizá el lector haya tenido ya la siguiente duda. Si en las cajas de texto se manda al servidor el texto que introduce el usuario, ¿qué se envía cuando usamos botones de elección? La respuesta es simple. Por un lado es necesario enviar el nombre del grupo (el valor del atributo `NAME`) y por otro la opción elegida (el valor del atributo `VALUE`). En el ejemplo anterior en el que se han seleccionado 'Pop' y 'Ciencia- Ficción' se enviará el siguiente mensaje:

```
musica=Pop
```

```
cine=CienciaFiccion
```

Nota

En realidad, el mensaje que se envía no es exactamente el anterior, ya que éste es codificado. Aún así las diferencias no son muchas, y en este caso el mensaje, ya codificado, que se enviaría constaría de una única línea: *musica=Pop&cine=CienciaFiccion*.

Veamos un resumen de los atributos de este control:

```
<INPUT TYPE="radio" NAME="Nombre_del_grupo" VALUE="Nombre_de_esta_opción"
ALIGN="alineamiento">
```

Cajas de selección.

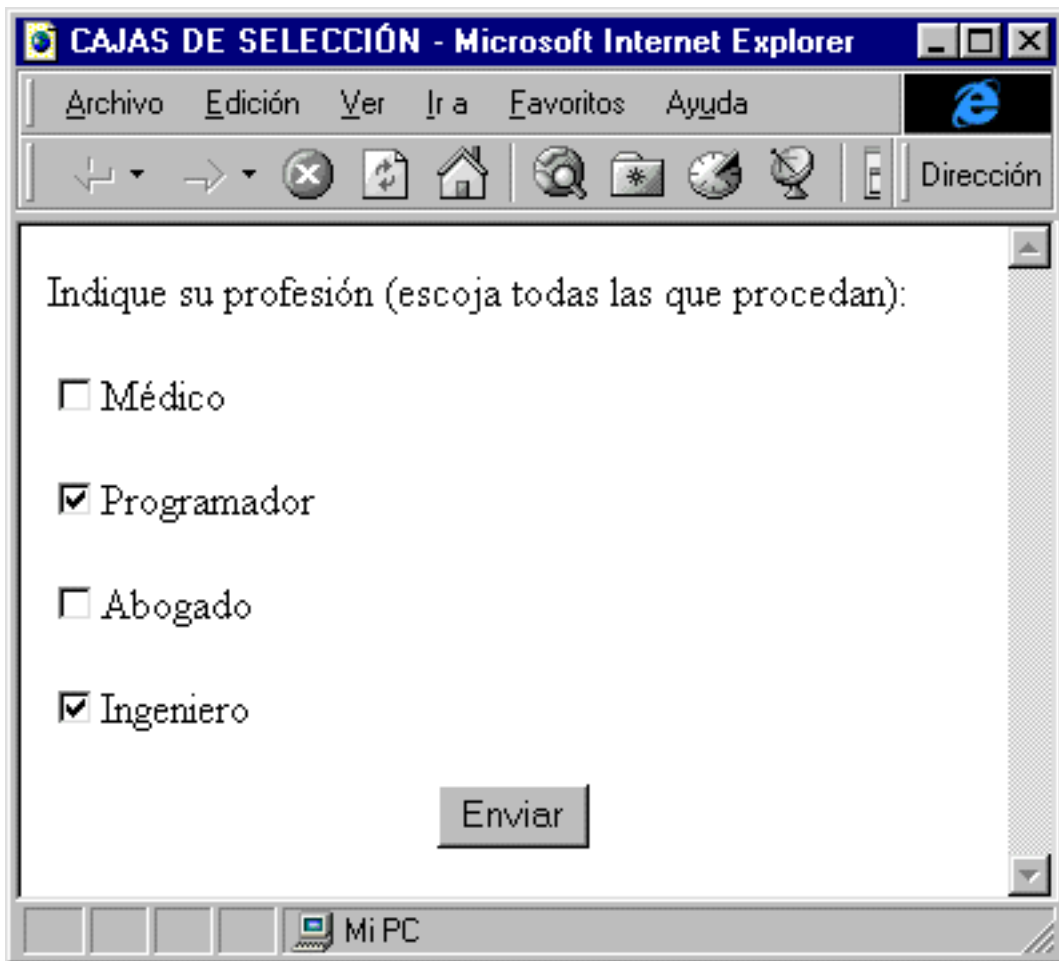
Las cajas de selección guardan ciertos parecidos con los botones de radio, pero además permitirán seleccionar varias opciones en una lista. Al igual que dichos botones tienen dos posiciones, seleccionados o no seleccionados, estando en esta última posición inicialmente, a no ser que hayamos usado el atributo `CHECKED`. Cada caja de selección es independiente del resto, y por tanto el valor del atributo `NAME` debe ser diferente en cada una.

Para insertar una caja de selección debemos usar de nuevo la etiqueta `INPUT`, pero esta vez con `TYPE="checkbox"`. Veamos un ejemplo de uso de este tipo de controles:

```
Indique su profesión (escoja todas las que procedan) :
<FORM>
<P><INPUT TYPE="checkbox" NAME="medico">Médico
<P><INPUT TYPE="checkbox" NAME="programador" CHECKED>Programador
<P><INPUT TYPE="checkbox" NAME="abogado">Abogado
<P><INPUT TYPE="checkbox" NAME="ingeniero">Ingeniero
<CENTER><INPUT TYPE="submit" VALUE="Enviar"></CENTER>
</FORM>
```

Pulsando sobre cada caja de selección se cambia de seleccionado a no seleccionado, independientemente de la posición del resto. En el código podemos apreciar que es posible indicar que una caja de selección esté seleccionada por defecto. En la figura 12.12 vemos este ejemplo, y podemos apreciar como es posible seleccionar varias opciones simultáneamente.

Figura 12.12. Las cajas de selección son independientes unas de otras, por los que es posible tener varias seleccionadas.



Con las cajas de selección también podemos usar el atributo `VALUE`, en este caso no será imprescindible, ya que el nombre dado con `NAME` es diferente en cada caso, pero a veces puede ser conveniente. En el caso de la figura se mandaría al servidor el siguiente mensaje:

```
programador=ON&ingeniero=ON
```

Si hubiésemos usado el atributo `VALUE` se sustituiría la palabra `ON` (que es la que se pone por defecto) por la que nosotros hubiésemos indicado en las cajas de selección con las opciones 'programador' e 'ingeniero'. Por ejemplo si en el código anterior hubiésemos escrito lo siguiente en el código de la caja del ingeniero, dejando el resto igual:

```
<P><INPUT TYPE="checkbox" NAME="ingeniero" VALUE="SI">Ingeniero
```

El navegador hubiera enviado el mensaje:

```
programador=SI&ingeniero=ON
```

Todos los atributos de este control quedan resumidos en el siguiente código:

```
<INPUT TYPE="checkbox" NAME="Nombre" ALIGN="alineamiento" VALUE="valor_que_debe Enviarse">
```

Imágenes.

El lenguaje HTML permite una manera alternativa para insertar un botón de envío personalizado, al poder usar en lugar del botón normal que hemos visto anteriormente una imagen creada por nosotros mismos. Para insertar este tipo de control debemos usar `TYPE="image"`, mientras que la dirección de la imagen se indica con el atributo `SRC` al igual que hacíamos con la etiqueta `IMG`. Aprovecharemos el ejemplo de este tipo de control para realizar un formulario que englobe lo visto hasta ahora.

Ejemplo 12.1.

Hemos elegido como motivo del formulario un sistema de reserva de habitaciones de un hotel. El usuario deberá introducir ciertos datos y enviar posteriormente la reserva. Comenzamos el formulario creando un archivo nuevo y escribiendo la estructura básica de la página:

```
<HTML>
<HEAD>
<TITLE>Botón de envío personalizado</TITLE>
</HEAD>
<BODY BGCOLOR="#B6D6B6">

</BODY>
</HTML>
```

A continuación insertamos un encabezado y la etiqueta `FORM`, donde iremos insertando los distintos controles de los que consta el formulario (todo el código que mostremos de ahora en adelante en este ejemplo debe introducirse entre las instrucciones `<BODY ...>` y `</BODY>`):

```
<CENTER><H2><FONT COLOR="#8080FF">Hoteles El Paraíso</FONT></H2></CENTER>
<H3>Sistema de reserva de habitaciones:</H3>
<FORM>

</FORM>
```

Los colores que hemos elegido ahora para el encabezado como antes para el color de fondo pueden ser cambiados por el lector a su gusto.

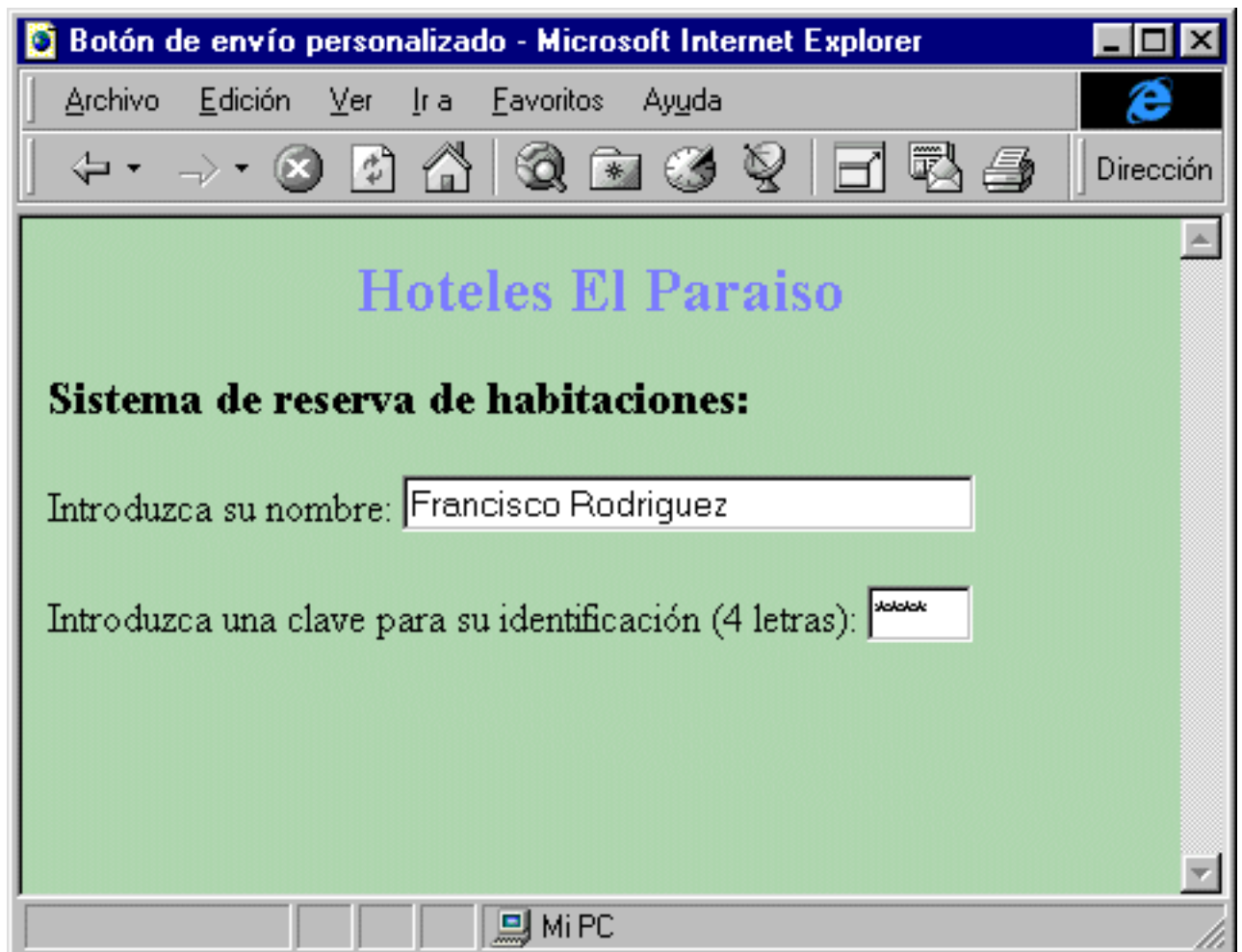
Es hora de insertar los controles. En primer lugar debemos introducir un campo para preguntar el nombre. Para ello lo más indicado es usar una caja de texto, elegimos un tamaño de 30 caracteres con el atributo `SIZE` y no usamos `MAXLENGTH` para no limitar la longitud del nombre que se pueda insertar. Además introducimos un segundo campo donde debe insertar una clave secreta, que debe elegir el mismo usuario, y que le servirá para identificarse cuando llegue al hotel junto con su nombre. En este caso usaremos una caja de texto para claves.

Dado que la clave sólo puede constar de 4 caracteres limitamos el tamaño de la caja (`SIZE`) y la longitud del texto que se puede escribir (`MAXLENGTH`) a este valor:

```
<FORM>
<P>Introduzca su nombre: <INPUT TYPE="text" NAME="nombre" SIZE="30"></P>
<P>Introduzca una clave para su identificación (4 letras): <INPUT TYPE="password"
NAME="clave" SIZE="4" MAXLENGTH="4"></P>
<!-- aquí iremos insertando el resto del código -->
</FORM>
```

Veamos lo que llevamos. Visualizando el código que hemos escrito hasta ahora obtenemos el resultado de la figura 12.13.

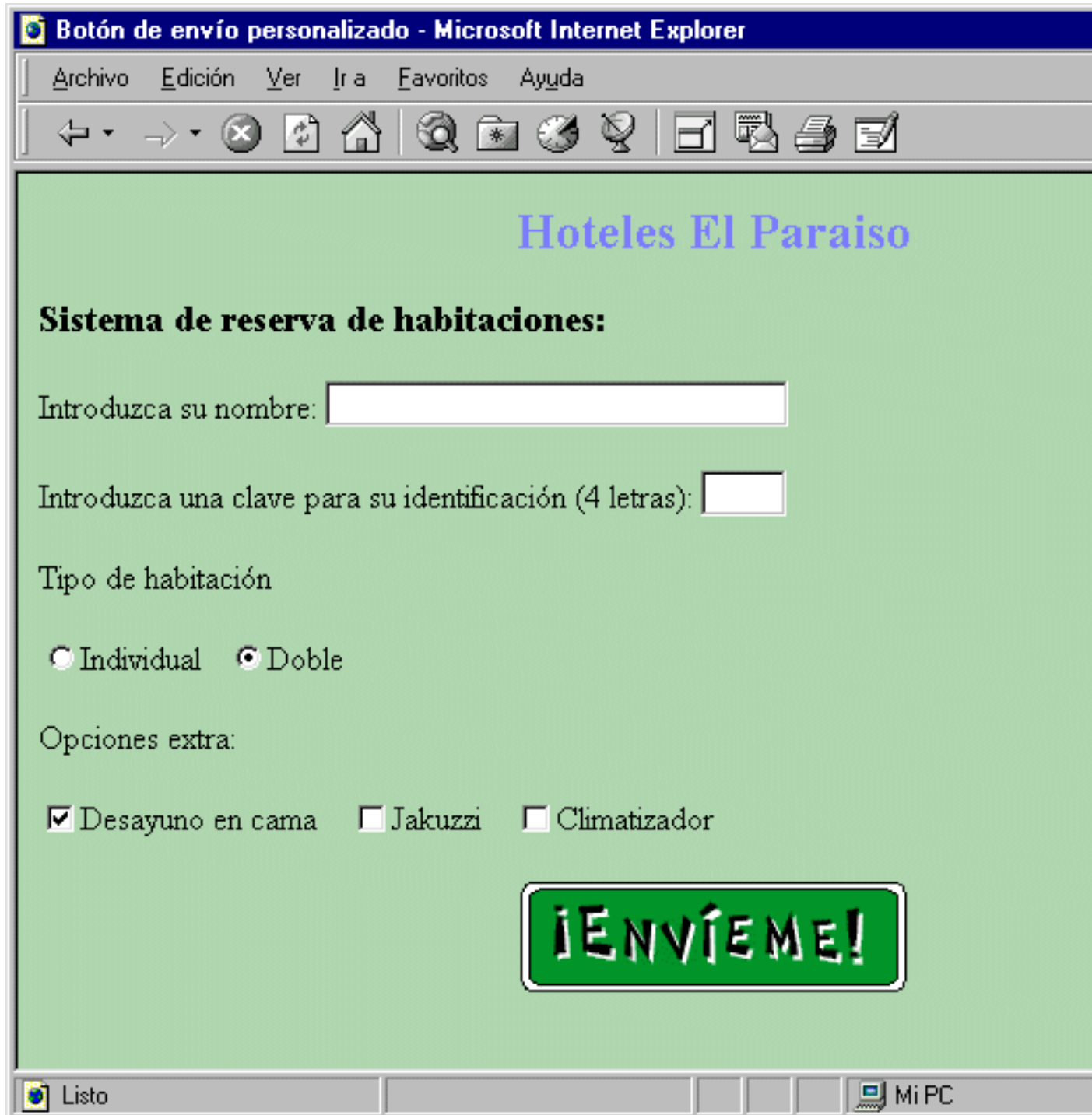
Figura 12.13. Este es el aspecto que tiene el código que hemos escrito hasta ahora. En el formulario de la clave sólo se podrán insertar 4 caracteres.



</FORM>

Con esto que podemos dar por concluido el formulario. En la figura 12.14 podemos apreciar el resultado de visualizar este formulario con el navegador Internet Explorer™ 4.0. Este es el aspecto inicial, con las opciones 'Doble' y 'Desayuno en cama' preseleccionados al usar el atributo CHECKED.

Figura 12.14. Este es el resultado final del formulario del ejercicio 12.1 en el que destaca el uso de una imagen personalizada como botón de envío.



Nota

El navegador Explorer™ permite el uso de los atributos `WIDTH` y `HEIGHT` para indicar el tamaño de

las imágenes de formularios. También permite usar otros atributos de la etiqueta `IMG` como `VSPACE` y `HSPACE` para controlar los márgenes verticales y horizontales alrededor de la imagen y `BORDER` para controlar el grosor del borde.

En este ejemplo hemos usado una imagen creada por nosotros mismos como sustituto del botón de envío estándar. Sin embargo las imágenes de formulario tienen una función extra, ya que el navegador envía, cuando el usuario pulsa una imagen, las coordenadas 'x' e 'y' sobre las que se ha pulsado, en relación a la esquina superior izquierda, como si fueran parte de la información del formulario. De esta forma el servidor Web puede realizar diferentes acciones en función del lugar de la imagen donde se ha pulsado. De hecho este es un método alternativo para crear una imagen mapa. Para ello no tenemos más que usar el código:

```
<FORM>
<INPUT TYPE="image" SRC="imagenmapa.gif">
</FORM>
```

Posteriormente debemos crear un programa en el servidor que sea capaz de leer los datos (es decir, las coordenadas donde se ha pulsado) de este formulario y devolver la respuesta oportuna. Más sobre esto en la sección 'Como usar los datos de un formulario' que veremos más adelante.

En el capítulo 6 vimos la diferencia entre las imágenes mapa en el servidor y las imágenes mapa en el cliente (las que podíamos crear con el lenguaje HTML). Este nuevo tipo se corresponde con las procesadas con el servidor, y por tanto tiene el inconveniente de necesitar un programa externo que se ejecute en el servidor, por lo que no nos basta con el lenguaje HTML.

Para terminar con las imágenes de formulario veremos, como hemos venido haciendo hasta ahora, un resumen con sus atributos:

```
<INPUT TYPE="image" SRC="URL" NAME="Nombre">
```

Botón de borrado

La mayoría de los controles anteriores permitían al usuario introducir una serie de datos o elegir entre una serie de opciones. Inicialmente las cajas de texto aparecían vacías, a no ser que hubiésemos usado el atributo `VALUE` para asignarles por defecto. Igualmente los botones de elección y cajas de selección aparecían en posición de no seleccionados a no ser que hubiésemos indicado lo contrario usando el atributo `CHECKED`. Una vez a cargado la página con el formulario en blanco o con unas opciones por defecto el navegante puede actuar sobre los controles y modificar sus valores (en esto se basa la interactividad). Pero quizás en algún momento decida que quiere dar marcha atrás y recuperar los datos que aparecieron al principio. Con este fin existe en el lenguaje HTML un control que permite borrar los datos actuales de todos los campos del formulario y restablecer los valores por defecto, si es que los había. Este control recibe el nombre de botón de borrado o botón de reseteado (*reset button*). La creación de este control es muy sencilla. De nuevo usaremos la etiqueta `INPUT`, esta vez con `TYPE="reset"`:

```
<INPUT TYPE="reset">
```

Este código provocará la aparición de un botón como el botón de envío con un texto por defecto que indica su función. Por ejemplo Internet Explorer 4.0™ muestra el mensaje 'Restablecer'. Este texto dependerá del navegador que usen los visitantes de nuestra página, y dado que esto no siempre es deseable podemos usar el

atributo *VALUE*, al igual que hacíamos con el botón de envío, para especificar el texto que prefiramos. Veamos un ejemplo algo más completo:

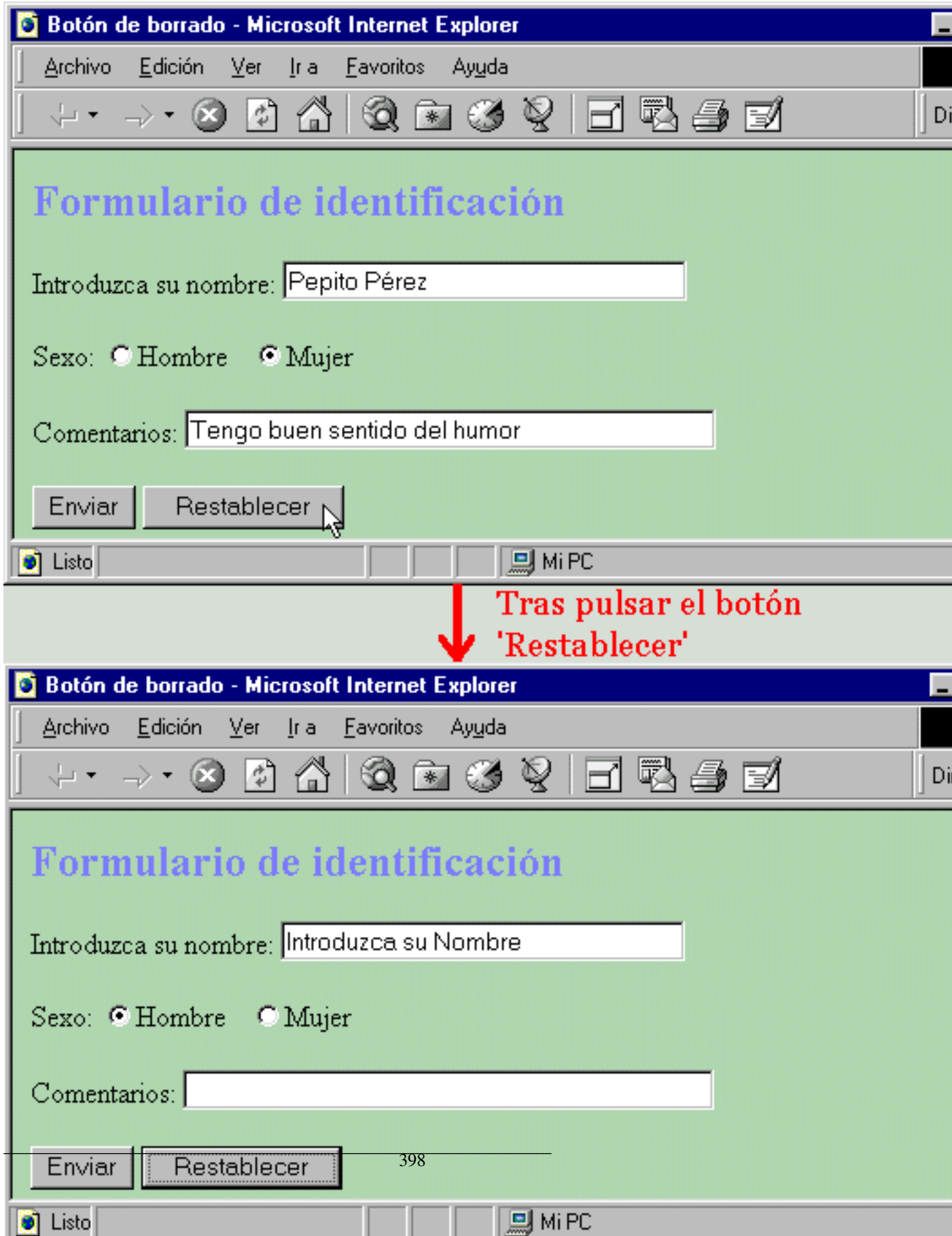
```
<H2>
<FONT COLOR="#8080FF">
  Formulario de identificación
</FONT>
</H2>

<FORM>

<P>Introduzca su nombre:
<INPUT TYPE="text"
  NAME="nombre"
  VALUE="Introduzca su Nombre"
  SIZE="30">
<P>Sexo:
<INPUT TYPE="RADIO"
  NAME="sexo"
  VALUE="hombre"
  CHECKED>Hombre
<INPUT TYPE="RADIO"
  NAME="sexo"
  VALUE="mujer">Mujer
<P>Comentarios:
<INPUT TYPE="text"
  NAME="comentarios"
  SIZE="40">
<P>
<INPUT TYPE="submit"
  VALUE="Enviar">
<INPUT TYPE="reset"
  VALUE="Restablecer">
</FORM>
```

En la figura 12.15 podemos ver este formulario con unos datos introducidos por un usuario (arriba) y tras pulsar el botón de borrado (abajo). Si no hubiésemos puesto unos valores por defecto en nuestro código HTML los datos campos simplemente se hubieran quedado vacíos.

Figura 12.15. El botón de borrado (`<INPUT TYPE="reset" >`) permite al usuario borrar los datos introducidos y recuperar los iniciales.



Al igual que hemos hecho con los controles anteriores concluiremos con un resumen del código del botón de borrado. En éste hemos incluido el atributo NAME, que, como comentamos al principio, está presente en todos los controles:

```
<INPUT TYPE="reset"
NAME="nombre"
VALUE="Texto del botón">
```

Botón genérico

Para completar la colección de botones de los formularios, nos encontramos con la posibilidad de insertar un botón genérico, es decir, que no tiene ninguna función definida, sino la que nosotros queramos darle. Para insertar un botón genérico usaremos la etiqueta INPUT con *TYPE="button"*:

```
<INPUT TYPE="button"
VALUE="Pulseme">
```

Las acciones que debe llevar a cabo este botón al ser pulsado sólo pueden indicarse con lenguajes de script como JavaScript o VBScript y no con HTML estándar. Para insertar el código de estos lenguajes se usa el atributo ONCLICK:

```
<INPUT TYPE="button"
VALUE="Texto del botón"
ONCLICK="código de script">
```

No todos los navegadores soportan los lenguajes de script, y por tanto el botón genérico debe ser considerado como una extensión. Si lo usamos debemos ser conscientes que únicamente los navegadores que soporten el script usado podrán hacer uso suyo.

Por ahora no nos adentraremos en ningún lenguaje de script, ya que más adelante en esta obra, éstos serán tratados más en detalle, pero mostraremos dos ejemplos sencillos que ilustran el funcionamiento de este tipo de botones. Estos ejemplos han sido probados con los navegadores Internet Explorer™ y Netscape Navigator™ en sus versiones 3.0 y posteriores, pero pueden no funcionar con otros navegadores.

El primero consistirá en un botón cuya función será *cerrar una ventana del navegador*. El código necesario será:

```
<FORM>
```

```
<INPUT TYPE="button"
```

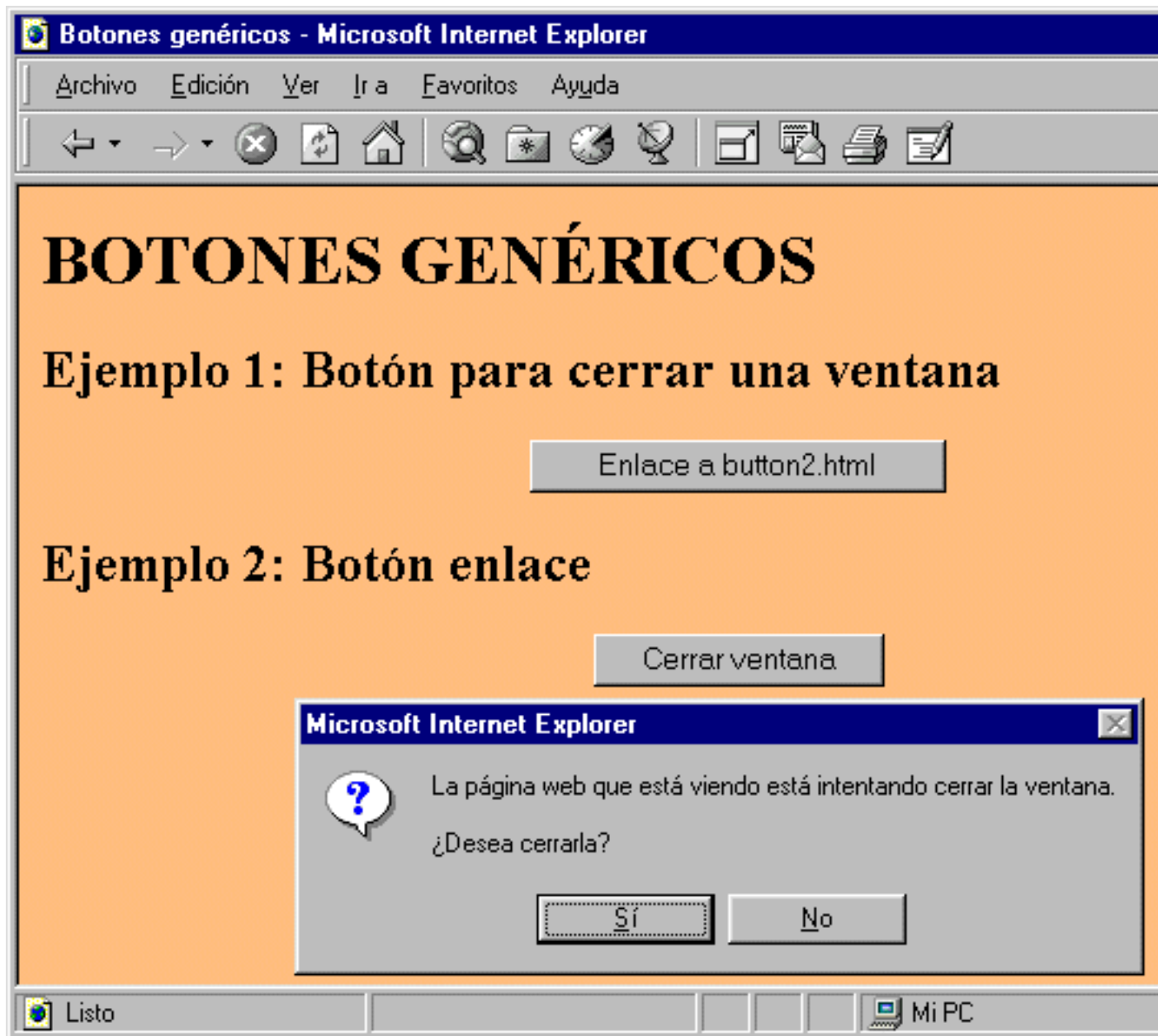
```
VALUE="Cerrar ventana"  
onClick="window.close();">
```

```
</FORM>
```

Es probable que al pulsar el botón nos aparezca un mensaje de aviso del navegador pidiéndonos confirmación para cerrar la ventana (ver figura 12.16).

Figura 12.16. Los botones genéricos tienen multitud de usos, entre los más usados nos encontramos con el botón enlace y el botón para cerrar una ventana, que, como todos,

requieren hacer uso de un lenguaje de script.



El segundo ejemplo es más usado aún y consiste en usar el botón con la misma función que un *enlace hipertexto*. El resultado es más atractivo, pero debemos usarlo con moderación, ya que no todos los visitantes tendrán navegadores que entiendan este código:

<FORM>

```
<INPUT TYPE="button"  
VALUE="Cargar otra ventana">
```

```
onClick="window.location.replace  
( 'button2.html' ); ">
```

```
</FORM>
```

En el directorio *botones* podemos encontrar un archivo con ambos códigos para que el lector pueda probarlos. En la figura 12.16 vemos una muestra de este documento junto con el cuadro de diálogo con el que Internet Explorer 4.0™ pide confirmación para cerrar la ventana tras pulsar el botón del primer ejemplo.

Nota

El nuevo estándar HTML 4.0 introduce una nueva etiqueta para insertar botones en nuestra página: *BUTTON*. Esta etiqueta tiene un atributo, llamado *TYPE*, que nos permitirá elegir entre los tres tipos de botones disponibles: envío (*TYPE=submit*), borrado (*TYPE=reset*) o genérico (*TYPE=button*). La gran innovación de esta etiqueta es que podemos insertar en el botón imágenes o texto formateado con código HTML. Veamos un ejemplo de cómo funcionará *BUTTON*:

```
<BUTTON name="reset "  
  type="restablecer">  
Restablecer<IMG src="pics/imagen.gif"  
  alt="">  
</BUTTON>
```

El texto e imagen entre la instrucción de inicio y la de fin se insertarán en el botón.

Campos ocultos

Los campos ocultos son un tipo de control que únicamente sirve para incluir información adicional entre los datos del formulario. Este tipo de control es invisible, y por tanto el navegador no muestra ningún tipo de gráfico para advertir de su existencia.

¿Para qué sirve entonces un control oculto?. Si no aparece en la página y el navegante no puede hacer nada con él ¿Cuál es su función?. En general, los campos ocultos se usan para enviar información específica al programa del servidor Web que procesará los datos enviados, por ejemplo si este programa es capaz de realizar varias funciones distintas con estos datos, podemos usar un campo oculto para pasarle la información de que debe hacer con los datos de este formulario concreto (debemos tener en cuenta que este mismo programa puede procesar la información de otros formularios también). Aún así esta problemática queda fuera de lo que pretendemos abarcar en esta sección dedicada al lenguaje HTML.

Los campos ocultos se crean usando la etiqueta *INPUT* con *TYPE="hidden"*. Además debemos usar los atributos *VALUE* y *NAME*, para indicar los datos y su nombre:

```
<INPUT TYPE="hidden"  
  NAME="nombre"  
  VALUE="datos">
```

Por ejemplo:

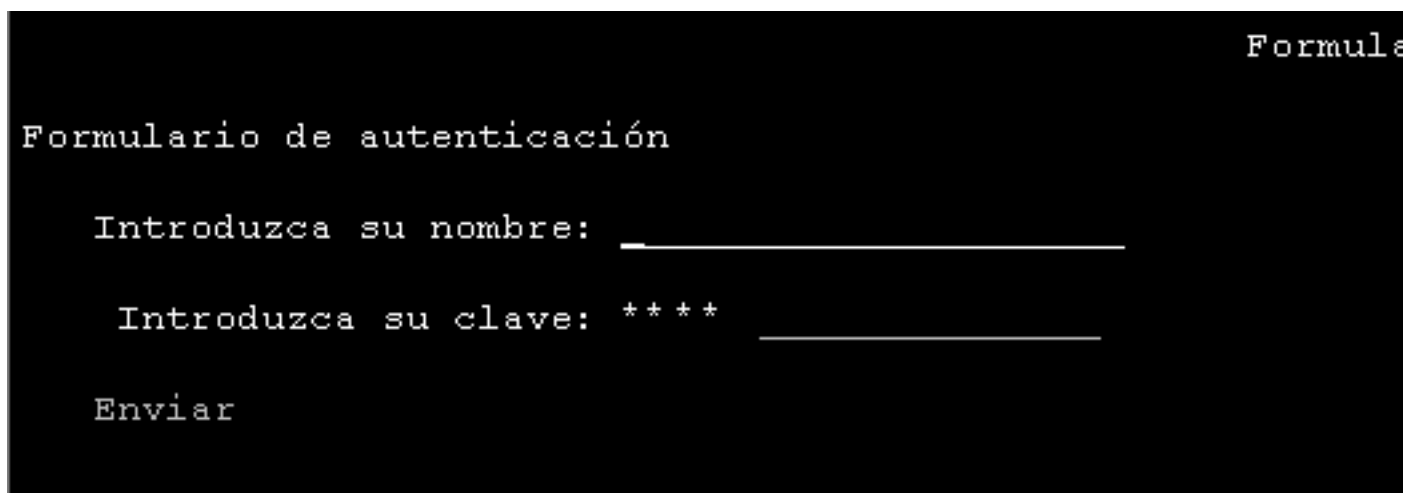
```
<INPUT TYPE="hidden"
  NAME="funcion"
  VALUE="enviar_mail">
```

Los formularios vistos desde Lynx

Hasta ahora nos hemos limitado a mostrar como se veían los controles con navegadores gráficos, usando como ejemplo a Microsoft Internet Explorer 4.0™. Pero los formularios pertenecen al estándar HTML 2.0 y por tanto funcionan con todo tipo de navegadores, incluidos los que trabajan en modo texto como Lynx. Es interesante para los programadores de páginas Web conocer como son mostrados cada uno de los controles que hemos visto desde este navegador, para hacerse una idea de cómo se verán los formularios que de sus páginas. Dado que no siempre es fácil encontrar un ordenador donde comprobar los resultados con Lynx, mostraremos una serie de capturas en las que hemos usado este navegador y las comentaremos brevemente.

Lo primero que debemos tener en cuenta es que el botón de envío no será gráfico, sino que será similar a un enlace hipertexto, como ya comentamos en su momento. Las cajas de texto y las cajas de texto para claves se muestran con una línea horizontal, y en estas últimas lo escrito es sustituido por asteriscos. En la figura 12.17 vemos un ejemplo, un cursor parpadeante (que no puede apreciarse en una imagen) indica el lugar donde podemos empezar a escribir:

Figura 12.17. Los navegadores en modo texto como Lynx también son capaces de mostrar formularios, que aunque no sean gráficos tienen la misma funcionalidad.



El código usado es el mismo que el de la figura 12.9, en aquel caso el navegador usado para mostrar el resultado fue Internet Explorer™, las diferencias son apreciables.

Pero más interesante aún puede ser ver con este navegador nuestro ejemplo 1 de este capítulo (figura 12.14) con Lynx. El resultado que obtenemos puede ser apreciado en la figura 12.18 donde vemos que este navegador también puede mostrar botones de radio y cajas de selección.

Figura 12.18. En esta figura podemos apreciar cajas de texto y de clave, botones de elección y cajas de selección tal y como son mostradas por Lynx.

```

Botón
Hoteles El Paraíso
Sistema de reserva de habitaciones:
Introduzca su nombre: Pepe Pérez
Introduzca una clave para su identificación (4 letras)
Tipo de habitación
(*) Individual ( ) Doble
Opciones extra:
[X] Desayuno en cama [ ] Jakuzzi [ ] Climatiza
[IMAGE]-Submit

```

En este caso es también interesante ver que la imagen que usábamos como botón de envío ha sido sustituido por el texto ' [IMAGE]-Submit', pero sigue funcionando.

Innovaciones de Microsoft

La compañía Microsoft™ en sus navegadores Internet Explorer 3.0™ y posteriores ha incorporado una serie de extensiones a la etiqueta INPUT para incrementar las posibilidades de los programadores de HTML. Estas extensiones afectan a todos los controles que hemos visto que hacen uso de esta etiqueta.

La primera extensión resultará conocida a los lectores, se trata del atributo *TITLE*, y su función será mostrar un bocadillo explicativo cuando el usuario pase el cursor por encima del control. Veamos un ejemplo:

```

<FORM>

<P>Introduzca su nombre:

<INPUT TYPE="text"
NAME="nombre"
TITLE="Introduzca aquí su Nombre"
SIZE="30">

</FORM>

```

Como podemos ver en la figura 12.19, cuando se mantiene el cursor un tiempo aparece un bocado con fondo amarillo (estándar de Windows) con el texto 'Introduzca aquí su nombre ':

Figura 12.19. El atributo TITLE es una extensión de Microsoft que permite dar una descripción a los controles que será mostrada en un bocado.



Este atributo puede usarse con entera libertad, a pesar de tratarse de una extensión, porque los usuarios de Internet Explorer™ podrán beneficiarse de ella y los usuarios de navegadores que no lo entiendan no saldrán perjudicados, simplemente se quedarán igual que si no lo hubiésemos usado.

La segunda extensión está formada por dos atributos: *TABINDEX* y *NOTAB*. Cuando nos encontramos ante un formulario formado por varios controles, como el del ejemplo 1 (figura 12.14), se puede usar la tecla tabulador o TAB para pasar de uno a otro. Por defecto al pulsar esta tecla se pasa al siguiente control en el orden en que fueron insertados en el código HTML. El atributo TABINDEX nos permitirá cambiar el orden. La forma de usarlo consiste en incluirlo en *todos* y cada uno de los controles asignándole un número, este número representará la posición de ese control. El atributo NOTAB, por su parte indicará que este control está fuera de la lista, y por lo general será situado el último de todos. Para entender ambos atributos mejor es necesario verlo con un ejemplo, para ello usaremos parte del código del ejemplo 1:

```
<FORM>
```

```
<P>1) Introduzca su nombre:
```

```
<INPUT TYPE="text"  
  NAME="nombre"  
  TABINDEX="1">
```

```
<P>3) Introduzca una clave para su  
identificación (4 letras):
```

```
<INPUT TYPE="password"
  NAME="clave"
  TABINDEX="3">
<P>4) Dirección
<INPUT TYPE="text"
  NAME="direccion"
  NOTAB>
<P>
2) Tipo de habitación
</P>
<INPUT TYPE="Radio"
  NAME="tipoHabitacion"
  VALUE="individual"
  TABINDEX="2">
Individual
<INPUT TYPE="Radio"
  NAME="tipoHabitacion"
  VALUE="doble"
  CHECKED
  TABINDEX="2">
Doble

</FORM>
```

Tras introducir el nombre, cuando el usuario pulse el tabulador pasará a elegir el tipo de habitación, después introducirá la clave y por último la dirección. En el directorio *ejem2* hemos incluido este ejemplo para que el lector pueda probarlo y compruebe el funcionamiento de los atributos `TABINDEX` y `NOTAB`. Debemos recordar que únicamente funcionarán con Internet Explorer 3.0™ o posterior.

La última innovación que ha introducido Microsoft™ son los atributos `READONLY` y `DISABLED`. Estos atributos funcionan exclusivamente con Internet Explorer 4.0™ (y posteriores), por lo que debemos tener cuidado al usarlos.

Cuando usamos el atributo `DISABLED` en un control lo deshabilitamos. En general este atributo únicamente es usado con cajas de texto. Cuando una caja de texto usa este atributo podrá seguir mostrando texto (si usamos el atributo `VALUE` para poner un texto inicial), pero el usuario no podrá cambiarlo. Este control tampoco podrá ser seleccionado, prácticamente actúa como si no estuviera. La función del atributo `READONLY` es muy similar, con la única diferencia de que el control actúa como tal, por ejemplo en una caja de texto podemos seleccionar con el ratón el texto que hay en ella, pero tampoco podrá modificarse el contenido.

Nota

Aunque en la documentación ofrecida por Microsoft™ no aparece así, el atributo `READONLY` sólo funciona correctamente en las cajas de texto, pero no en los botones de elección o en las cajas de selección. Quizá sea un fallo que se solucione en próximas versiones, pero hoy por hoy, para deshabilitar estos dos últimos controles, debemos usar el atributo `DISABLED`.

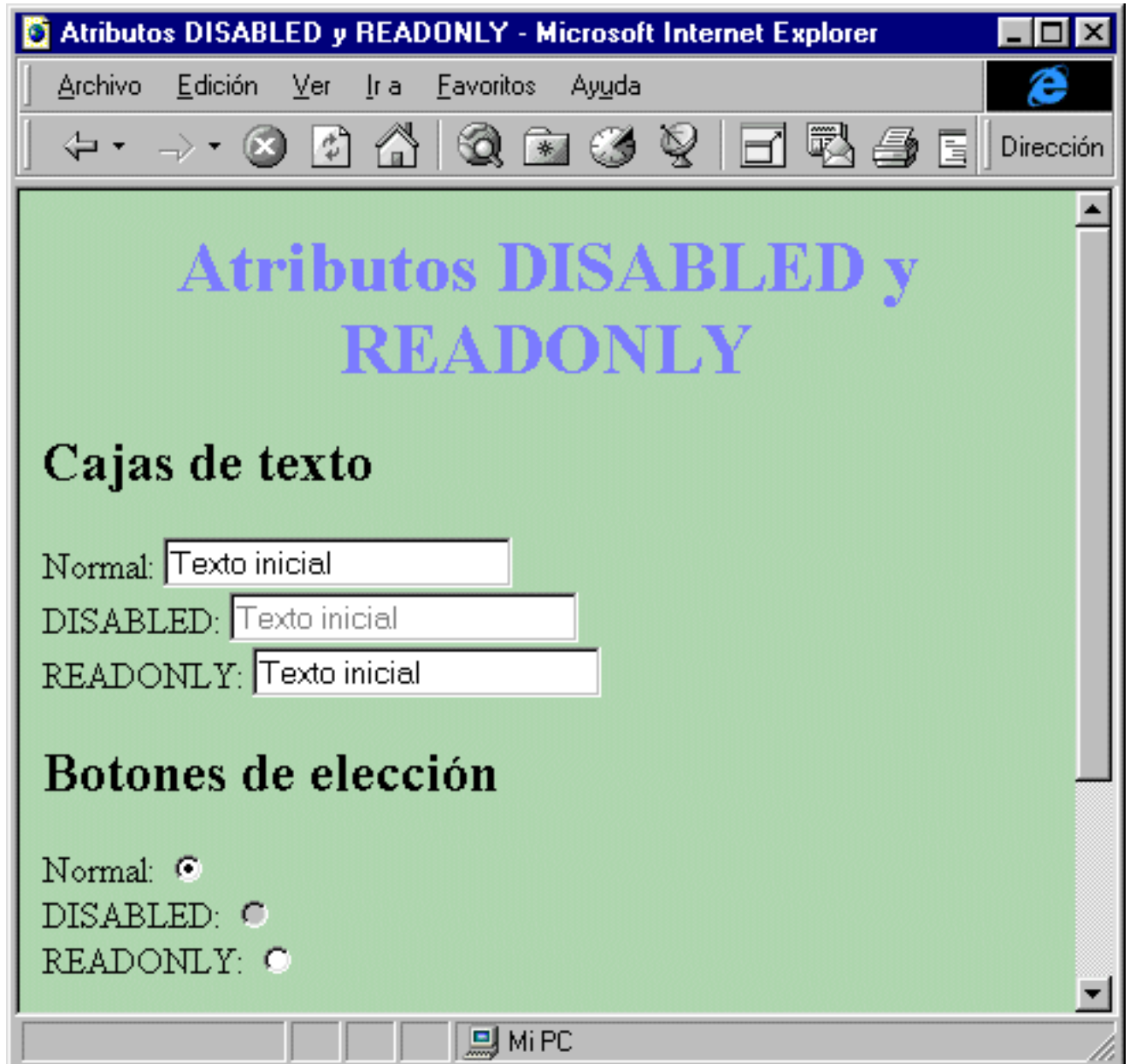
Veamos unos ejemplos con cajas de texto y botones de selección:

```
<H2>
Cajas de texto
</H2>
Normal:
<INPUT TYPE="text"
```

```
    NAME="texto1"
    VALUE="Texto inicial">
<BR>
DISABLED:
<INPUT TYPE="text"
    NAME="texto2"
    VALUE="Texto inicial"
    DISABLED>
<BR>
READONLY:
<INPUT TYPE="text"
    NAME="texto2"
    VALUE="Texto inicial"
    READONLY>
<BR>
<H2>
Botones de elección
</H2>
Normal:
<INPUT TYPE="radio"
    NAME="boton"
    CHECKED>
<BR>
DISABLED:
<INPUT TYPE="radio"
    NAME="boton"
    DISABLED>
<BR>
READONLY:
<INPUT TYPE="radio"
    NAME="boton"
    READONLY>
<BR>
```

En la figura 12.20 podemos ver el resultado de visualizar este código con Internet Explorer 4.0™.

Figura 12.20. Los atributos **DISABLED** y **READONLY** permiten deshabilitar un control de manera que sólo sirva para mostrar datos.



En los casos de la caja de texto y el botón de elección se observa claramente que el navegador le da una apariencia distinta para resaltar que está deshabilitado. En el directorio disabled se ha incluido este ejemplo añadiendo también las cajas de selección para que el lector pueda probarlos.

OTROS CONTROLES

En esta sección hemos incluido dos controles ligeramente diferentes a los que hemos visto hasta ahora, las cajas de texto multilinea y los cuadros de selección. Ambos tienen sus propias etiquetas de HTML, y por tanto no usarán INPUT.

Cajas de texto multilínea: TEXTAREA

Las cajas de texto multilínea o áreas de texto son unos campos que funcionan de manera similar a un editor de texto muy sencillo en el que el usuario puede escribir. Al contrario de lo que sucedía con las cajas de texto convencionales (<INPUT TYPE="text">), en esta ocasión será posible escribir varias líneas de texto, lo que es muy útil para campos en los que se requiere una respuesta extensa. Por ejemplo, si deseamos que el usuario pueda escribir el texto de un *mail*, o escribir una serie de comentarios largos, las áreas de texto serán tremendamente útiles.

La etiqueta usada para insertar este nuevo control es *TEXTAREA*, que consta de cuatro atributos:

- *NAME*: El nombre que queremos asignarle al control. Como siempre, este nombre será enviado junto con los datos del área de texto al mandar el formulario.
- *ROWS*: El número de líneas de la caja de texto.
- *COLS*: El número de caracteres visibles de cada línea. Este atributo es similar al atributo *SIZE* que vimos en las cajas de texto convencionales.

La etiqueta *TEXTAREA* está compuesta por una instrucción de inicio y una instrucción de fin, siendo obligatorio el uso de ambas. Entre ellas únicamente podrá insertarse texto llano (sin ninguna etiqueta *HTML*) y éste será mostrado como contenido inicial del área de texto. Veamos un ejemplo:

Por favor haga sus comentarios sobre esta página.

```
<FORM>
```

```
<TEXTAREA NAME="texto"
  ROWS="10"
  COLS="50">
```

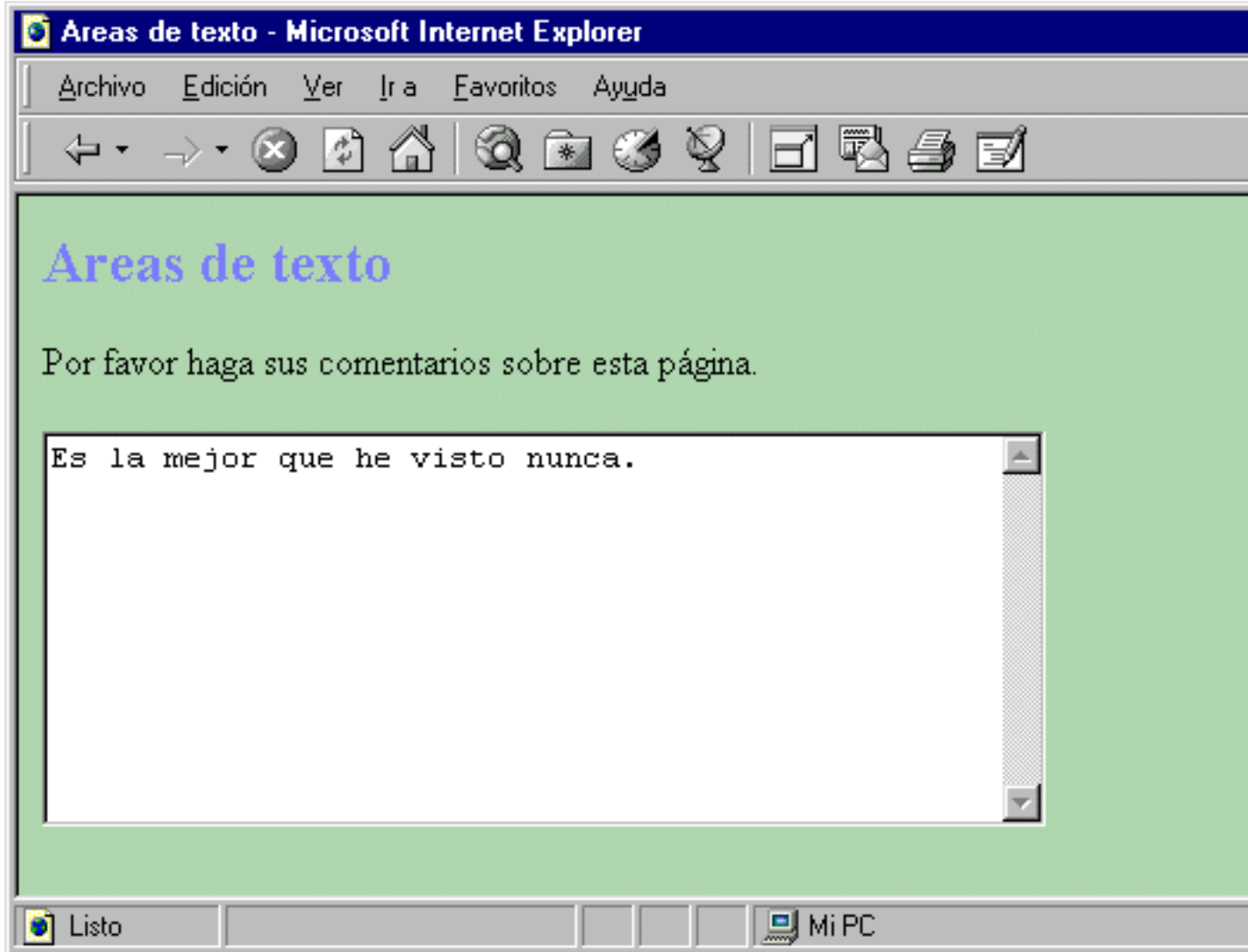
```
Es la mejor que he visto nunca.
```

```
</TEXTAREA>
```

```
</FORM>
```

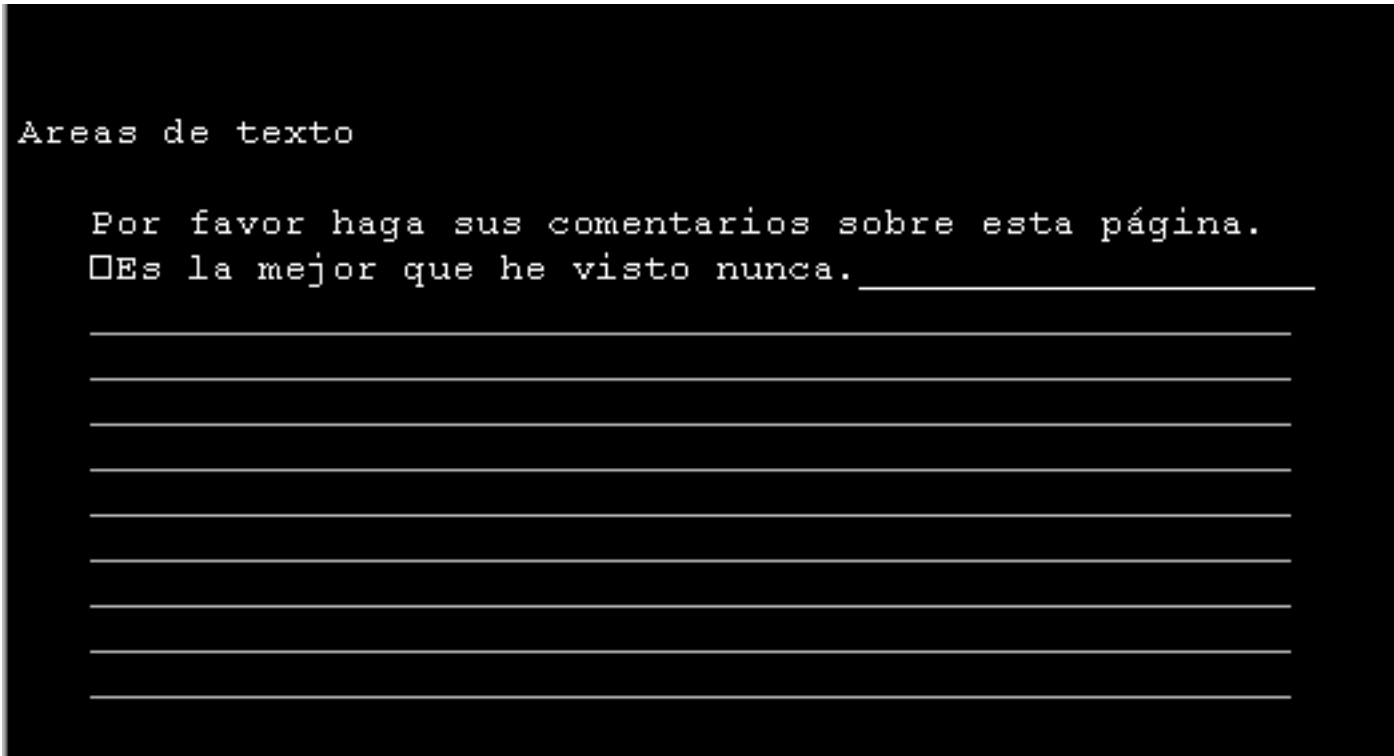
En la figura 12.21 podemos ver este código tal y como es mostrado por Internet Explorer 4.0™:

Figura 12.21. Las áreas de texto (<TEXTAREA> </TEXTAREA>) permiten al usuario introducir texto de varias líneas como si de un sencillo editor de texto se tratara.



Y en la figura 12.22 el mismo visto con Lynx:

Figura 12.22. Lynx muestra las áreas de texto como un conjunto de líneas (tantas como hemos indicado con el atributo COLS) sobre las que el usuario podrá escribir.



Pulsando sobre el área de texto podremos empezar a escribir, al igual que hacíamos en las cajas de texto, pero con la diferencia de que ahora podremos pulsar ENTER cuando queramos para cambiar de línea.

Si no pulsamos ENTER y llegamos al final de la línea el navegador debe decidir que hacer. Esta decisión variará, por lo general, del navegador que estemos usando. Algunos, como Internet Explorer 4.0™, cambian automáticamente a la siguiente línea (*text wrapping*), otros simplemente desplazarán el texto como ocurría con las cajas de texto y no cambiará de línea hasta que pulsemos ENTER y una minoría simplemente ignorará lo que pulsemos hasta que cambiemos de línea pulsando esta misma tecla.

El navegador Netscape™ introdujo una extensión al lenguaje HTML estándar, que actualmente es soportada también por Internet Explorer™, y que nos permitirá tener un mayor control sobre la forma de actuar del navegador cuando se llegue al final de la línea. Esta extensión se basa en un nuevo atributo, *WRAP*, que puede tomar los siguientes valores:

- *WRAP=OFF*: Es el valor por defecto en Netscape Navigator™ y provocará que el texto sea mostrado en una sola línea, desplazándose hacia la izquierda si es necesario, hasta que el usuario pulse ENTER.
- *WRAP=SOFT*: Este es el valor por defecto en Internet Explorer™ y provoca que el navegador pase automáticamente a la siguiente línea cuando se llega al final del área reservada. Sin embargo cuando se pulsa el botón de envío, antes de mandar el contenido, el propio navegador vuelve a juntar todo el texto que fue escrito seguido en una sola línea.
- *WRAP=HARD*: Al igual que el valor anterior, éste provoca que el navegador pase automáticamente a la siguiente línea, pero ahora la información se mandará al servidor Web con los saltos de línea, de la misma manera que si hubiésemos pulsado ENTER nosotros mismos al llegar al final de cada línea.

En la figura 12.23 vemos una página con los tres tipos de área de texto que hemos incluido en el directorio *textarea* para que el lector pueda probarlos. La diferencia entre los dos últimos tipos no es apreciable por el usuario, pero sí existen diferencias en el mensaje que se envía al servidor.

Nota

La etiqueta TEXTAREA también admite el atributo ALIGN y las extensiones introducidas por Microsoft™ que hemos visto anteriormente. Nos referimos a los atributos DISABLED, READONLY, TABINDEX, NOTAB y TITLE.

Concluiremos las áreas de texto con un resumen de la etiqueta TEXTAREA y sus atributos:

```
<TEXTAREA NAME="nombre"
ROWS="filas"
COLS="columnas"
ALIGN="alineamiento"
WRAP="off | soft | hard">

Texto inicial
</TEXTAREA>
```

Cuadros de selección

Éste es el último control para crear formularios que nos ofrece el lenguaje HTML. Su función será dar a elegir entre una serie de opciones de manera que el usuario pueda elegir una o varias de ellas. Existen dos formas de mostrar estas opciones, como una *lista con desplazamiento*, o mediante una *persiana desplegable*. Enseguida veremos cómo es cada una y cómo crearlas.

Para insertar un cuadro de selección usaremos la etiqueta *SELECT*, que consta de una instrucción de inicio y de una instrucción de fin, entre las cuales introduciremos las diferentes opciones para el usuario. Para insertar estas opciones usaremos una nueva etiqueta, *OPTION*, que consta de una única instrucción. Veamos un ejemplo que ilustra el uso de uso de este tipo de control:

```
<FORM>

Elija un color:<BR>
<CENTER>
<SELECT NAME="color">
  <OPTION>Verde
<OPTION>Negro
<OPTION>Rojo
<OPTION>Azul
</SELECT>
</CENTER>

  <INPUT TYPE="submit"
  VALUE="Enviar elección">

</FORM>
```

Cuando se pulse el botón 'Enviar elección' se mandará al servidor el texto que sigue a la etiqueta OPTION, que se haya escogido (Por ejemplo 'color=Rojo').

Como podemos ver en la figura 12.24, este control suele ser mostrado, en los navegadores gráficos, como una persiana desplegable, es decir, únicamente la opción actualmente seleccionada es visible, y pulsando sobre una flecha se muestran el resto de opciones.

Figura 12.24. Los cuadros de selección permiten mostrar varias opciones para que el usuario pueda elegir. En esta ocasión vemos su forma de persiana desplegable.



Existe un atributo, *SELECTED*, que nos permitirá elegir la opción que debe estar seleccionada por defecto. Por ejemplo, si en el código anterior queremos que el color negro aparezca seleccionado inicialmente tendremos que escribir:

```
<SELECT NAME="color">
<OPTION>Verde
<OPTION SELECTED>Negro
<OPTION>Rojo
<OPTION>Azul
```

```
</SELECT>
```

Al principio hemos comentado que los cuadros de selección pueden mostrarse como persianas desplegadas, o bien como listas con desplazamiento. Hasta ahora sólo hemos visto el primer caso en el que únicamente la opción seleccionada es visible. En ocasiones puede ser conveniente que todas, o al menos algunas de las posibilidades sean mostradas directamente, sin tener que pulsar un botón. Para conseguir esto el lenguaje HTML ofrece el atributo *SIZE*, que determina el número de opciones que pueden ser vistas simultáneamente. Si este valor es 1 (o no se usa el atributo *SIZE*), se mostrará una persiana desplegable, como hemos visto hasta ahora. Si es mayor, se mostrará una lista, que podrá tener barras de desplazamiento si no caben todas las opciones disponibles. Veamos un ejemplo:

```
<FORM>

Lista de la compra:<BR>
<CENTER>
<SELECT NAME="compra"
  SIZE="6">

  <OPTION VALUE="Fruta">
Fruta

  <OPTION VALUE="Verdura">
Verdura

  <OPTION VALUE="Ternera">
Ternera

  <OPTION VALUE="Mantequilla">
Mantequilla

  <OPTION VALUE="Salchichas">
Salchichas

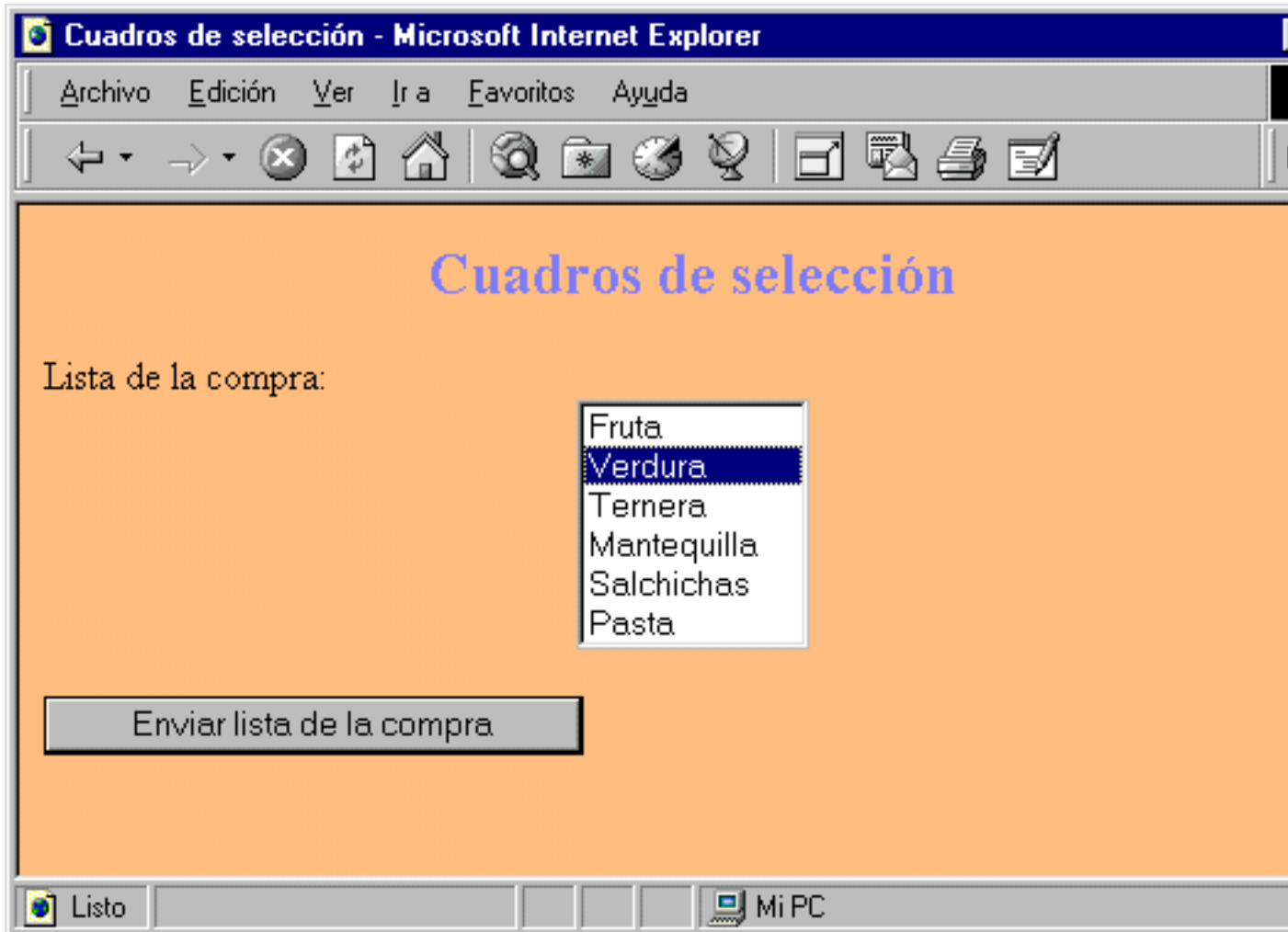
  <OPTION VALUE="Pasta">
Pasta
</SELECT>
</CENTER>
<P>

  <INPUT TYPE="submit "
  VALUE="Enviar lista de la compra">

</FORM>
```

En la figura 12.25 vemos este ejemplo:

Figura 12.25. Usando el atributo **SIZE** con un valor mayor que uno conseguimos que el cuadro de selección se muestre como una lista.



En este caso el número de opciones es igual al tamaño que le hemos dado al control, por lo que no han sido necesarias barras de desplazamiento.

Si no indicamos lo contrario, los cuadros de selección actúan como los botones de radio, es decir, sola una opción puede ser elegida simultáneamente. Afortunadamente, es posible cambiar este comportamiento, y permitir la selección de varias opciones, usando el atributo *MULTIPLE* en el interior de la etiqueta *SELECT*. En el ejemplo de la lista de la compra esta posibilidad es de gran utilidad, añadiendo este atributo al código anterior queda:

```
<FORM>
```

```
Lista de la compra:<BR>
```

```
<CENTER>
```

```
<SELECT NAME="compra"
```

```
  SIZE="6"
```

```
  MULTIPLE>
```

```
<OPTION VALUE="Fruta">
Fruta
```

```
<OPTION VALUE="Verdura">
Verdura
```

```
<OPTION VALUE="Tenera">
Tenera
```

```
<OPTION VALUE="Mantequilla">
Mantequilla
```

```
<OPTION VALUE="Salchichas">
Salchichas
```

```
<OPTION VALUE="Pasta">
Pasta
```

```
</SELECT>
```

```
</CENTER>
```

NOTA (usuarios de Windows): Para seleccionar varias opciones mantengan pulsada la tecla <CODE>Control </CODE> y selecciónenlas con el ratón.

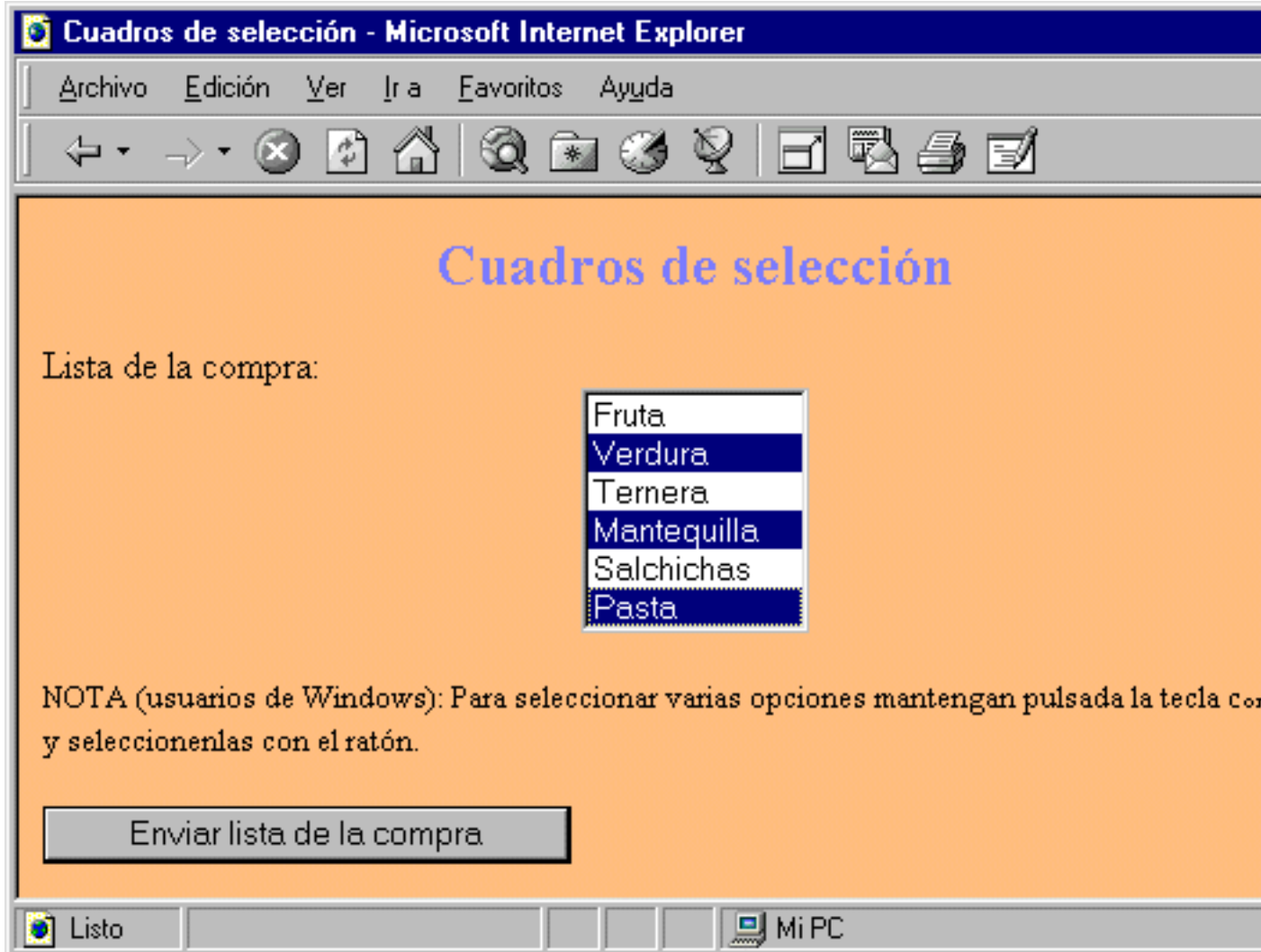
```
<P>
```

```
<INPUT TYPE="submit"
VALUE="Enviar lista de la compra">
```

```
</FORM>
```

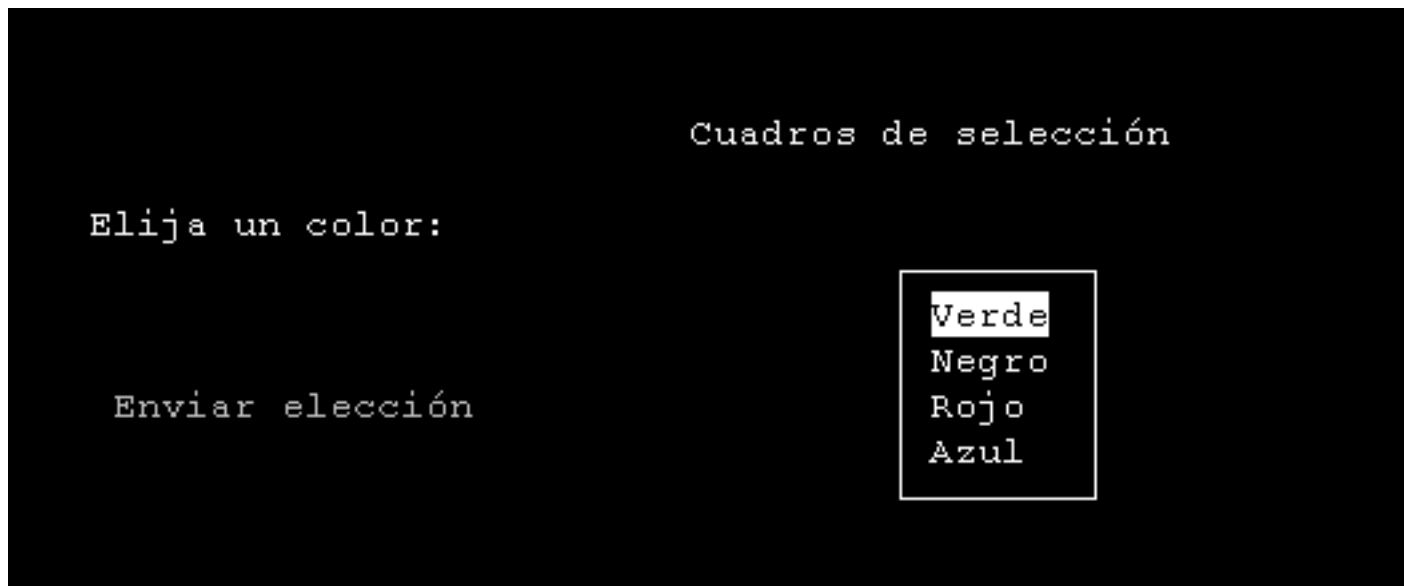
La forma de seleccionar varias opciones simultáneamente dependerá del navegador que usemos. En general consistirá en mantener apretada una tecla mientras se seleccionan las diferentes opciones con el ratón. Para facilitar la tarea de los navegantes puede ser conveniente añadir a la página una nota explicativa sobre la manera de elegir varias opciones, tal y como hemos hecho en este ejemplo. En la figura 12.26 podemos ver como es mostrado Internet Explorer 4.0™:

Figura 12.26. Usando el atributo MULTIPLE el usuario podrá seleccionar varias de las opciones que se ofrecen en la lista.



Y en la figura 12.27 el mismo código, esta vez visto desde Lynx:

Figura 12.27. El navegador Lynx también es capaz de mostrar cuadros de selección, por lo que podremos usarlo con libertad.



Nota

La etiqueta SELECT también consta del atributo ALIGN para elegir su alineamiento. En lo que a extensiones se refiere, también podemos usar el atributo DISABLED en la etiqueta SELECT para desactivarla, aunque solo funcionará con Internet Explorer 4.0™. Igualmente podemos usar el atributo TITLE tanto en SELECT como en OPTION para incluir una descripción que será mostrada en un bocadillo cuando el usuario sitúe el ratón sobre el control, en el primer caso, o sobre la opción específica, en el segundo.

Para concluir con los cuadros de selección sólo nos resta hacer un breve resumen de las etiquetas SELECT y OPTION, y de sus atributos:

```
<SELECT NAME="nombre"
  SIZE="nº_elementos_visibles"
  MULTIPLE
  ALIGN="alineamiento"
  TITLE="texto_bocadillo"
  TABINDEX="n"
  DISABLED>

  <OPTION SELECTED
    VALUE="nombre de esta opción"
    TITLE="texto_bocadillo">

  ...
</SELECT>
```

PRÁCTICA 6-FORMULARIO PARA "Los Alpes"

Ahora que ya conocemos todos los controles existentes para crear formularios es hora de hacer uno con apariencia profesional. Esto no quiere decir que use todos los controles existentes, sino que los use de una manera adecuada, cuidando la presentación y facilitando al usuario la introducción de datos. Como motivo hemos elegido crear una página más del sitio Web de la agencia de viajes Los Alpes, que permita a los visitantes pedir que les envíen por correo un catálogo con todos los viajes disponibles.

En primer lugar procederemos a escribir el código HTML necesario para crear los formularios y posteriormente nos dedicaremos a darle una apariencia profesional, tarea en la que podremos demostrar nuestra capacidad para usar tablas para controlar la disposición de los elementos en una página Web.

- **PASO 1:** Para comenzar partiremos de las páginas que creamos en la práctica 5. Podemos realizar los cambios sobre estas mismas páginas o crear una copia en otro directorio (en el directorio *pract5* se encuentran de nuevo las páginas de esta práctica). Usando nuestro editor de texto/HTML habitual abrimos el archivo *plantilla.htm* donde creamos la plantilla de la página Web de Los Alpes. El aspecto de esta página es muy simple, como podemos ver en la figura 12.28:

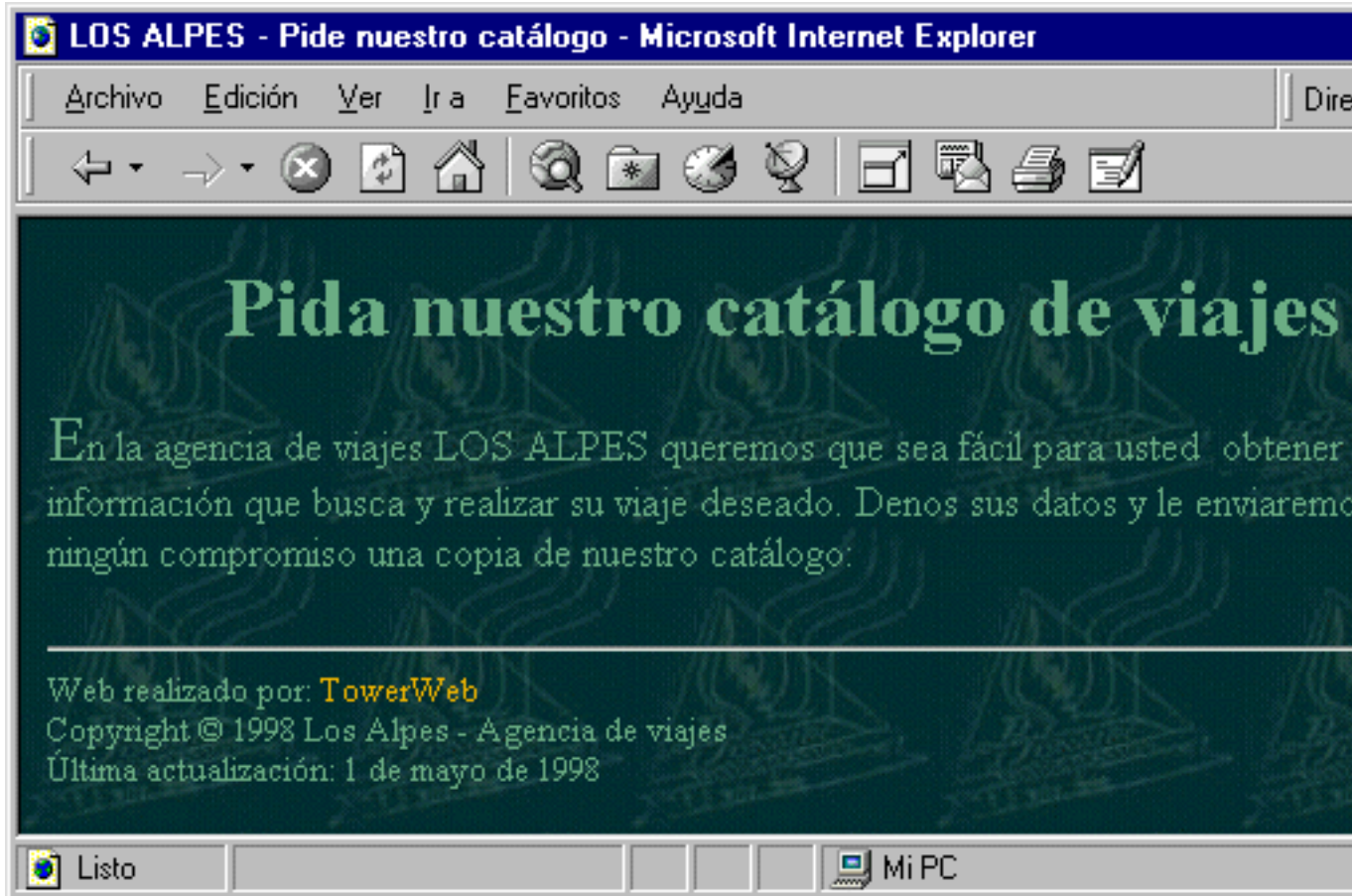
Figura 12.28. Este es el aspecto inicial de la plantilla del sitio Web de Los Alpes.



En primer lugar, antes de realizar ningún cambio, guardamos este documento con el nombre *form.htm*. Sobre este nuevo archivo realizaremos los cambios. Lo primero que haremos será adaptar el código de la plantilla a nuestros objetivos, cambiando la fecha y el título de la página, y escribiendo un texto introductorio del formulario (ver figura 12.29):

D1

Figura 12.29. El primer paso será modificar la plantilla, cambiando el título y la fecha y escribiendo un texto introductorio.



- **PASO 2:** El siguiente paso consistirá en insertar los controles del formulario. Dado que nuestro objetivo será recoger el nombre, dirección, teléfono, etc. del usuario, usaremos en su mayoría cajas de texto, con la única excepción de un cuadro de selección para elegir el tipo de catálogo. El código necesario para insertar estos controles es:

```
<FORM>

<P>Nombre completo:
<INPUT TYPE="text"
  NAME="nombre"
  SIZE="30">
<P>Dirección:
<INPUT TYPE="text"
  NAME="direccion"
  SIZE="50">
<P>Ciudad:
<INPUT TYPE="text"
  NAME="ciudad"
  SIZE="20">
```

```

    Código Postal:
<INPUT TYPE="text"
  NAME="c.p."
  SIZE="6"
  MAXLENGTH="5">
<P>Teléfono:
<INPUT TYPE="text"
  NAME="telefono"
  SIZE="10"
  MAXLENGTH="9">
<P>Fax:
<INPUT TYPE="text"
  NAME="FAX"
  SIZE="10"
  MAXLENGTH="9">
<P>Email:
<INPUT TYPE="text"
  NAME="email"
  SIZE="30">
<P>Catálogo:

<SELECT>

  <OPTION VALUE="nacional">
  Nacional

  <OPTION VALUE="internacional" SELECTED>
  Internacional

  <OPTION VALUE="completo">
  Completo
</SELECT>

<P>
<INPUT TYPE="image"
  BORDER="0"
  SRC="img/enviar.jpg"
  VALUE="Enviar">

</FORM>

```

Y deberemos insertarlo a continuación del texto introductorio. Este código no tiene demasiada complejidad, pudiendo destacarse la limitación a cinco caracteres para el código postal y a nueve el del número de teléfono y fax, con lo que evitamos posibles equivocaciones de los usuarios. En ambos casos hemos usado el atributo `MAXLENGTH` para imponer esta limitación. En esta página hemos decidido, además, usar un botón de envío personalizado, acorde con la apariencia del Web. Este y otro material necesario para la práctica puede encontrarse en el directorio *pract6/material*.

D1 Una vez realizados estos cambios y escrito el código anterior obtenemos el resultado que vemos en la figura 12.30.

Figura 12.30. Usando los conocimientos adquiridos en este capítulo utilizamos las etiquetas `INPUT` y `SELECT` para crear los controles del formulario.

LOS ALPES - Pida nuestro catálogo - Microsoft Internet Explorer

Archivo Edición Ver Ir a Favoritos Ayuda

← → × ↺ 🏠 🔍 📁 🌐 📧 📄 📧

Pida nuestro catálogo de viajes

En la agencia de viajes LOS ALPES queremos que sea fácil para usted obtener información que busca y realizar su viaje deseado. Denos sus datos y le enviaremos ningún compromiso una copia de nuestro catálogo:

Nombre completo:

Dirección:

Ciudad: Código Postal:

Teléfono:

Fax:

Email:

Catálogo:

Enviar

Web realizado por: *TowerWeb*
Copyright © 1998 Los Alpes - Agencia de viajes
Última actualización: 1 de mayo de 1998

Listo Mi PC

- **PASO 3:** Podríamos dejar el formulario así, y publicarlo en Internet; de hecho gran parte de los formularios que encontramos tienen una apariencia similar a esta. Sin embargo en capítulos anteriores hemos aprendido a usar tablas para controlar la disposición de los elementos y es hora de usar estos conocimientos para dar a nuestro formulario una apariencia mucho más profesional. En un Web real un buen diseño incita a un mayor número de navegantes a rellenar el formulario.

Para empezar cambiamos el titular por una imagen (*pract6/material/catalogo.gif*) en la que hemos escrito el mismo texto con un tipo de letra que simula escritura manual. El resto de cambios hacen uso de tablas. Para realizar el diseño suele ser conveniente hacer un boceto sobre el papel o con un programa de dibujo, donde no existen las limitaciones del lenguaje HTML. Una vez hecho el boceto es hora de usar este lenguaje para plasmarlo sobre una página Web. Tras hacer esto en nuestro formulario concluimos que sería adecuado usar una tabla de nueve filas y tres columnas. De las filas la primera es para el título y para el texto introductorio, mientras que el resto son para cada campo del formulario. La primera columna contendrá el texto indicativo del formulario alineado a la derecha (`<TD ALIGN="right">`), la segunda contendrá el control en sí y la tercera servirá únicamente como margen derecho. En la figura 12.31 mostramos el formulario con su nueva apariencia mostrando los bordes de la tabla, para que el lector pueda apreciar la estructura de la tabla:

Figura 12.31. Usando una tabla de nueve filas y tres columnas controlamos la disposición de todos los elementos de la tabla para conseguir un diseño profesional.

LOS ALPES - Pida nuestro catálogo - Microsoft Internet Explorer

Archivo Edición Ver Ir a Favoritos Ayuda

← → × ↺ 🏠 🔍 📁 🌐 📧 📄 📝

Pida nuestro catálogo de viajes

En la agencia de viajes LOS ALPES queremos que sea fácil para usted la obtener información que busca y realizar su viaje deseado. Denos sus datos y le enviaremos sin ningún compromiso una copia de nuestro catálogo.

Nombre:	<input type="text"/>		
Dirección:	<input type="text"/>		
Ciudad:	<input type="text"/>	Código Postal:	<input type="text"/>
Teléfono:	<input type="text"/>		
Fax:	<input type="text"/>		
Email:	<input type="text"/>		
Catálogo:	<input type="text" value="Internacional"/>		

Enviar

Web realizado por: [TowerWeb](#)
Copyright © 1998 Los Alpes - Agencia de viajes
Última actualización: 1 de mayo de 1998

D1 Otro aspecto que podemos comentar es que el ancho total de la tabla será de 600 píxeles, con lo que conseguimos que se visualice correctamente en todo tipo de monitores. Para que también quede bien en monitores grandes la centramos usando la etiqueta CENTER. Por otro lado, para crear cierta separación entre los diferentes campos podemos usar el atributo CELLSPACING, con el que controlamos la distancia en píxeles entre las diferentes celdas.

La realización de este código es un ejercicio muy importante, ya que este tipo de retoques es muy común en las páginas Web reales. Por esta razón dejamos al lector que intente mejorar el diseño de la página usando las pistas que le hemos dado. En cualquier caso, en el directorio *pract6* se ofrece una posible solución. En la figura 12.32 podemos ver la página completa una vez introducida en la estructura de frames:

Figura 12.32. Este es el aspecto final del formulario una vez introducido en la estructura con frames del sitio Web creado en la práctica 5.

Agencia de viajes LOS ALPES - Microsoft Internet Explorer

Archivo Edición Ver Ir a Favoritos Ayuda

← → × ↻ 🏠 🔍 📁 🕒 📡 📄 📧 🖨️ 📝



Los alp

Agencia de viajes

Principal Viajes a su medida Ofertas Sup

Pida nuestro catálogo de viaje

En la agencia de viajes LOS ALPES queremos que sea fácil para usted la obtener información que busca y realizar su viaje deseado. Denos sus datos y le enviaremos sin ningún compromiso una copia de nuestro catálogo:

Nombre:

Dirección:

Ciudad: Código Postal:

Teléfono:

Fax:

Email:

Catálogo:

Enviar

La mejora en apariencia respecto a la que veíamos en la figura 12.30 es apreciable.

COMO USAR LOS DATOS DE UN FORMULARIO

Una vez dominamos la creación, y diseño, de formularios en las páginas Web probablemente nos estamos preguntando ¿cómo puedo hacer que me lleguen los datos?. La respuesta a esta pregunta va mas allá del lenguaje HTML. Requiere la puesta en acción de unos programillas llamados *CGIs*, que se ejecutan en el servidor y que pueden estar escritos en prácticamente cualquier lenguaje de programación.

Estos programas deben encontrarse en algún lugar del servidor Web. Al enviar un formulario a uno, éste procesará la información y contestará con la respuesta oportuna. Esta respuesta será en general o bien la dirección de una página o bien una página creada por el propio CGI. ¿Cómo se envía información a un CGI?. Hasta ahora habíamos omitido este detalle, pero la etiqueta `FORM` consta de un atributo llamado *ACTION* donde debemos especificar la dirección URL del programa CGI donde deben enviarse los datos. Además debe usarse otro atributo, *METHOD*, para indicarle cómo se mandarán estos datos. Este atributo puede tomar dos valores *GET* y *POST*, omitiendo todos los detalles, podemos decir que el primero se usa con formularios pequeños, mientras que el segundo será el escogido para formularios donde la cantidad de información es grande. Una vez hemos visto estos atributos hemos de decir que para que los ejemplos que hemos visto funcionen cuando los publicamos en Internet habrá que añadirseles a la etiqueta `<FORM>`. Veamos un ejemplo:

```
<FORM METHOD="post"
ACTION="http://www.miservidor.es/cgi-bin/
programaCGI">

<!-- Código para insertar los distintos controles -->

</FORM>
```

En este caso hemos supuesto la existencia de un CGI llamado *programa CGI* situado en el directorio *cgi-bin* (este es el nombre habitual del directorio donde suelen estar los CGIs) de nuestro servidor. El navegador mandará a este programa los datos del formulario codificado con un código especial, cuyo tipo MIME es *application/x-www-form-urlencoded*. En algunos de los ejemplos de este capítulo hemos visto ejemplos de como funciona esta codificación. La etiqueta `FORM`, tiene un atributo, *ENCTYPE*, que nos permitirá cambiar el tipo de codificación. Si no queremos usar ninguna podemos escribir:

```
<FORM ENCTYPE="text/plain"
METHOD="post"...>
</FORM>
```

No usar codificación podrá ocasionar problemas al enviar caracteres especiales como son las letras acentuadas, pero puede ser útil en ocasiones, al no ser necesario decodificar el mensaje.

La creación de los programas CGI excede los límites de esta sección dedicada al lenguaje HTML, pero será cubierta en detalle más adelante debido a la gran importancia que tienen estos programas en la programación de Internet y en concreto de páginas Web. Hasta entonces veremos algunos métodos alternativos que nos permitirán hacer uso de los formularios que hemos creado sin necesidad de CGIs.

Formularios sin CGIs

Sin duda los programas CGI permiten realizar muchas más cosas que cualquier otro método, pero la necesidad de conocer un lenguaje de programación nos lleva a buscar métodos alternativos que serán suficientemente buenos si no tenemos demasiados requerimientos.

Una alternativa a los CGI son los lenguajes de script, que nos permitirán interactuar con el navegante sin hacer uso del servidor. Pero la alternativa en la que nos centraremos aquí será otra, que se basa en la posibilidad de *mandar todos los datos de un formulario a una dirección de correo electrónico*. Esto es suficiente para el ejemplo de la práctica 6, ya que simplemente necesitamos conocer los datos del usuario para mandar el catálogo, no necesitamos interactuar con él en manera alguna. Para conseguir que los datos de un formulario sean enviados por e-mail debemos usar las etiquetas ACTION, METHOD y ENCTYPE de la siguiente manera:

```
<FORM ACTION="mailto:milogin@midireccion.es"
  ENCTYPE="text/plain"
  METHOD="post">
<!-- ... -->
</FORM>
```

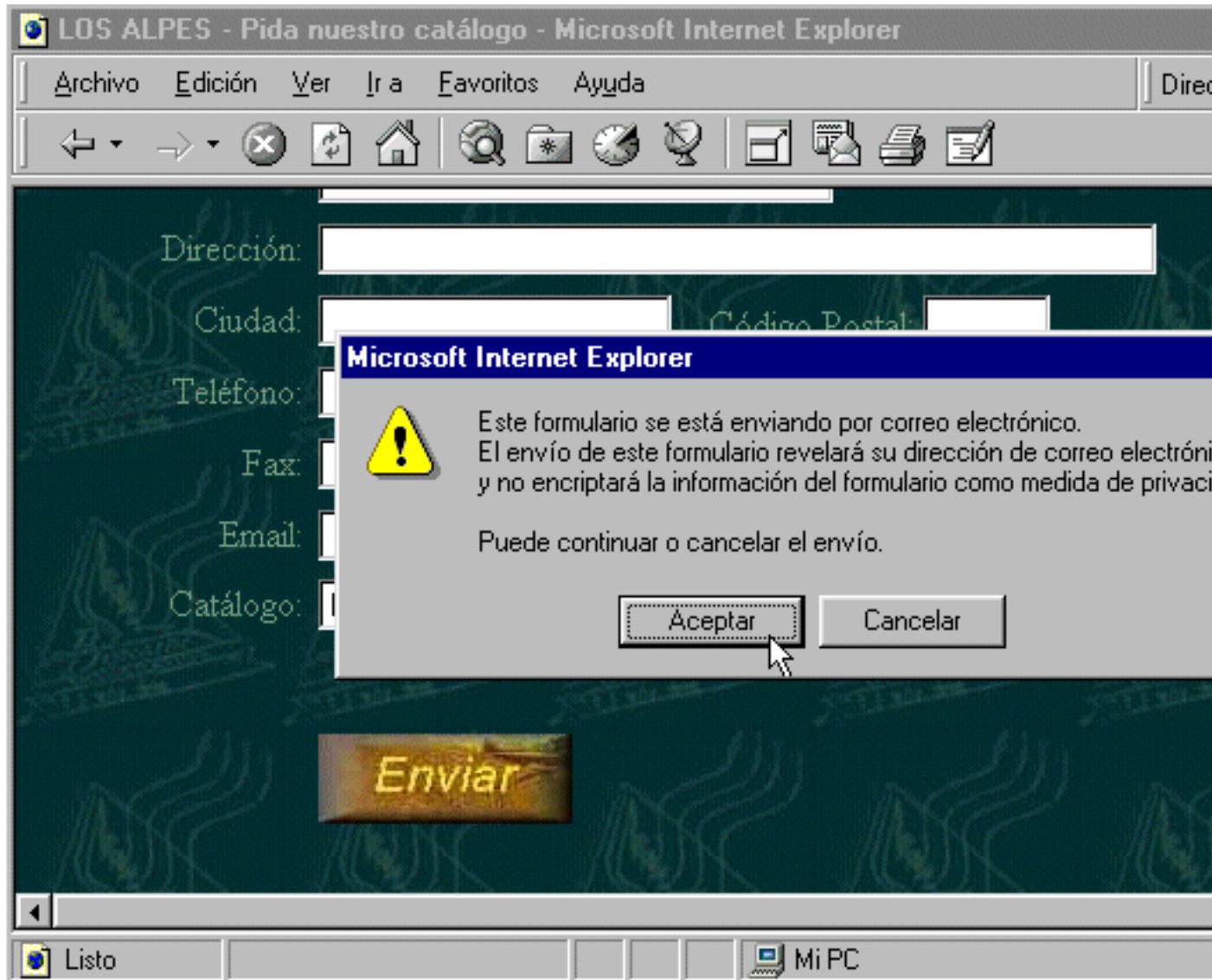
Donde hemos usado como ejemplo la dirección de correo electrónico *milogin@midireccion.es*, que debe ser sustituida por aquella a la que queramos enviar los datos, que en general será la nuestra como creadores del Web.

Nota

La función del atributo ENCTYPE="text/plain" es enviar los datos sin ninguna codificación. Esto es conveniente para facilitar la lectura de los mismos, pero no necesaria. De hecho, en algunos casos aislados puede ser preferible no usar este atributo y usar la codificación por defecto.

Al pulsar sobre el botón de envío el navegador advertirá al usuario (ver figura 12.33) que la página quiere enviar un *e-mail*, si el usuario acepta enviará los datos del formulario.

Figura 12.33. Como método alternativo a los CGI's pueden mandarse los datos del formulario por correo electrónico. En ese caso el navegador pedirá al usuario confirmación de la acción.



Nota

Al contrario que los CGI's, el método alternativo que envía por correo electrónico los datos del formulario puede no funcionar con algunos navegadores. Sin embargo actualmente un gran número de ellos, incluyendo Netscape Navigator™, Internet Explorer™ y Opera™, soporta esta característica, por lo que podemos usarlo con suficientes garantías.

Con este método alternativo podemos empezar a usar todos los formularios que hemos visto hasta ahora, y practicar con ellos. Cuando, más adelante, estudiemos a fondo los CGI's veremos como nuestras posibilidades se incrementan enormemente.

ENVÍO DE FICHEROS USANDO FORMULARIOS

Como apéndice a los formularios incluimos un método introducido por Netscape™ y aceptado por el estándar HTML 3.2 que permite a los visitantes de una página mandar un archivo presente en su ordenador a la página Web.

Con los controles que hemos visto hasta ahora se podía enviar un archivo de texto copiándolo a un área de texto, pero no existía ninguna manera de mandar, por ejemplo, una imagen o un programa. Con este nuevo sistema será posible enviar ficheros de todo tipo.

Los cambios respecto a los controles anteriores no son muchos. En primer lugar debemos hacer uso del atributo ENCTYPE especificando que los datos que se van a enviar desde este formulario al servidor serán de tipo *multipart/form-data*, después debemos usar ACTION para especificar la dirección de un programa del servidor Web que sea capaz de recoger este archivo y realizar las acciones oportunas con él, por ejemplo:

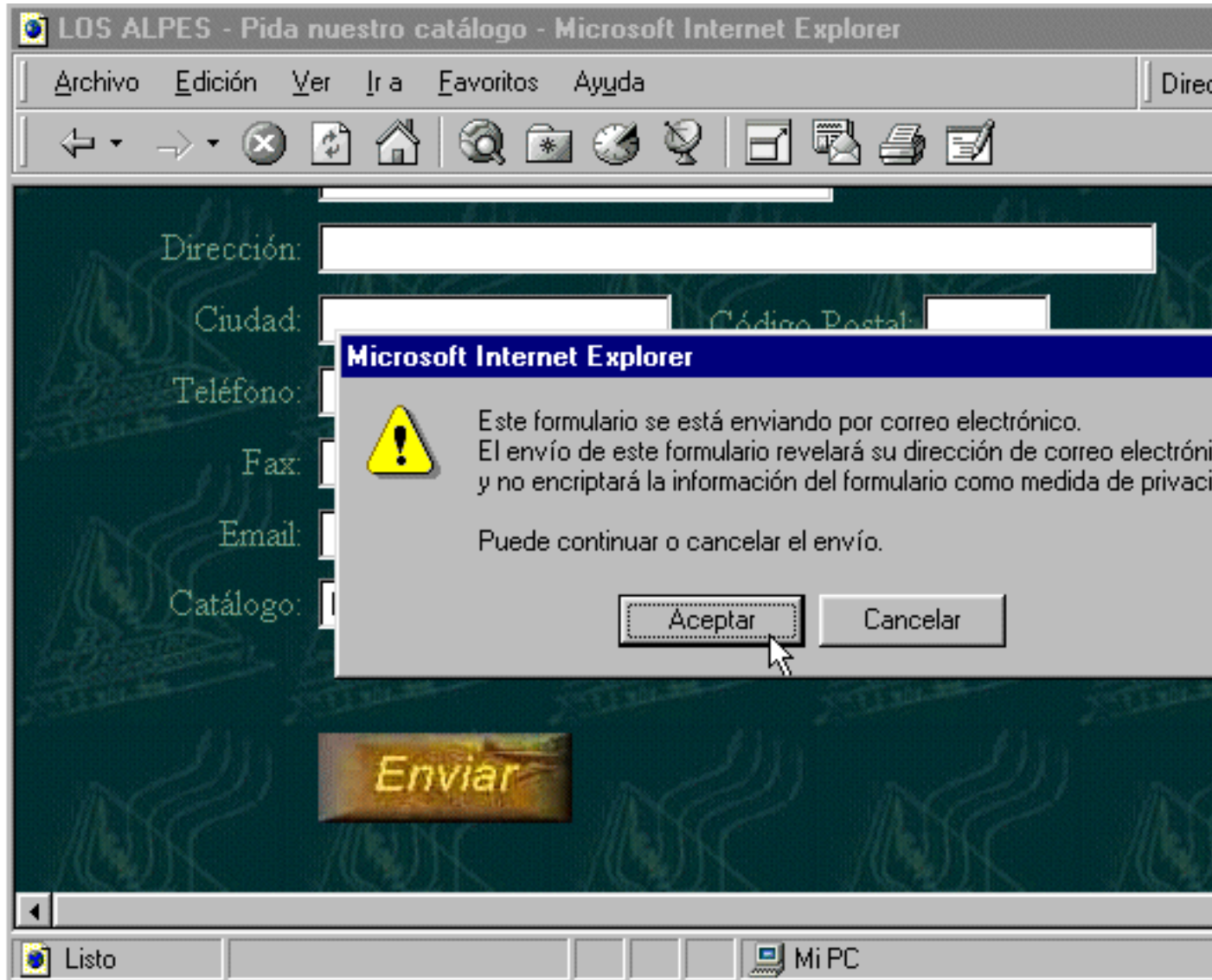
```
<FORM ENCTYPE="multipart/form-data"
ACTION="http://www.miservidor.es/cgi-bin/
cogefichero">
  <!-- ... -->
</FORM>
```

En lo que al control en sí se refiere usaremos de nuevo la etiqueta INPUT, esta vez con *TYPE="file"*. El navegador lo mostrará como una caja de texto junto con un botón que permitirá al usuario buscar entre sus archivos el que quiere enviar. Un ejemplo con el código completo sería:

```
<FORM ENCTYPE="multipart/form-data"
ACTION="http://www.miservidor.es/cgi-bin/
cogefichero">
  Enviar el fichero:
  <INPUT NAME="fichero_usuario"
TYPE="file">
  <INPUT TYPE="submit"
VALUE="Enviar fichero">
</FORM>
```

Usando este código se obtiene el resultado que observamos en la figura 12.33 en la que se muestra también el cuadro de diálogo que ofrece Internet Explorer 4™ cuando se pulsa sobre el botón 'Examinar...':

Figura 12.33. Como método alternativo a los CGI's pueden mandarse los datos del formulario por correo electrónico. En ese caso el navegador pedirá al usuario confirmación de la acción.



Hasta aquí es sencillo. La parte más difícil en el envío de ficheros desde páginas Web consistirá en crear un programa para el servidor que sea capaz de recibirlo. Por lo general puede ser conveniente obtener alguno hecho por profesionales que realice, o al menos nos ayude, a realizar esta labor. Para ello recomendamos al lector que se dirija a la dirección:

- <http://www.bio.cam.ac.uk/cgi-lib/> [<http://www.bio.cam.ac.uk/cgi-lib/>]

Si está interesado en adquirir información para crear el programa usted mismo puede obtener toda la información necesaria en el servidor FTP:

- <ftp://ds.internic.net/rfc/rfc1867.txt> [<ftp://ds.internic.net/rfc/rfc1867.txt>]