

Introducción a XHTML

Jesús Arias Fisteus

02/10/2003

1. Introducción

XHTML es una redefinición de HTML 4 en XML. A pesar de detener distinto nombre, conserva la mayoría de los elementos y atributos de HTML. Las diferencias más destacables están en la ausencia de elementos y atributos relacionados con estilo (fuentes, colores, etc.), que ya en HTML 4 están desaconsejados, y en la existencia de unas normas más estrictas en la colocación de etiquetas de elementos, que se explican a continuación.

La semántica de los elementos y atributos de XHTML es exactamente la misma que en HTML 4. Por ello, la especificación de HTML 4 [1] es una buena referencia, con numerosos ejemplos, para crear documentos XHTML.

Hay, actualmente, varias versiones de XHTML. Todo lo expuesto en este documento será válido para todas ellas:

- XHTML 1.0: el más parecido a HTML 4, con tres variantes, *transitional*, *strict* y *frameset*.
- XHTML 1.1: nueva versión, que parte de XHTML 1.0 *strict*. No permite el uso de elementos y atributos relacionados con el estilo (pero sí, obviamente, hojas de estilo, como CSS).
- XHTML Basic: versión simplificada de XHTML 1.1 pensada para dispositivos de capacidad limitada de procesamiento, como televisores, teléfonos, móviles, PDAs, etc.

1.1. Elementos

Un documento HTML está formado por una jerarquía de elementos y texto. Cada elemento puede contener, a su vez, otros elementos y texto. Un elemento está delimitado por marcas (*tags*), que pueden ser de tres tipos:

- Marcas de inicio de elemento: contienen el nombre del elemento y sus atributos. Por ejemplo: `<table width="300">`.
- Marcas de fin de elemento: contienen sólo el nombre del elemento. Por ejemplo: `</table>`.

- Marcas de elemento vacío: útil cuando un elemento no tiene contenido. Por ejemplo: `
`. Es equivalente a `
</br>`. Estas marcas también pueden contener declaración de atributos.

El contenido de un elemento es la porción del documento que se encuentra entre su marca de inicio y su marca de fin. El tipo de contenido de cada elemento está especificado, y debe ser respetado. Por ejemplo, el elemento *html* debe contener, en este orden, un elemento *head* y un elemento *body*.

Al contrario que en HTML, en XHTML los nombres de los elementos deben estar siempre en minúsculas. Además, es obligatorio que todos los elementos tengan una marca de inicio y de finalización (o marca de elemento vacío, si es el caso), y se encuentren correctamente anidados.

1.2. Atributos

Los elementos, además de contenido, pueden tener atributos. Un atributo está formado por un nombre y un valor. Los atributos de un elemento se especifican en su marca de inicio (o de elemento vacío), indicando, en este orden, su nombre, el símbolo = y su valor, entre comillas (dobles o simples). Por ejemplo: ``.

En caso de que sea necesario especificar un tipo de comillas dentro del valor de un atributo, se puede utilizar el otro tipo para englobarlo, sin dar lugar a confusiones. Por ejemplo:

```

```

El nombre y tipo de los elementos que puede tener un determinado elemento, así como su significado, se encuentran especificados, y deben ser respetados.

Al contrario que en HTML, en XHTML los nombres de los atributos deben estar siempre en minúsculas.

1.3. Referencias a entidades

Las referencias a entidades permiten incluir caracteres que no pueden aparecer directamente en el documento, normalmente debido a alguna de estas razones: es un carácter *reservado* (<, >, &), el editor no es capaz de representarlo, o el sistema de codificación del documento no lo permite (por ejemplo, en ASCII no es posible introducir tildes o la letra ñ).

En la siguiente tabla se recoge la forma de referenciar a las entidades más habituales:

Referencia	Carácter
<code>&amp;</code>	<code>&</code>
<code>&lt;</code>	<code><</code>
<code>&gt;</code>	<code>></code>
<code>&aacute;</code>	<code>á</code>
<code>&eacute;</code>	<code>é</code>
<code>&iacute;</code>	<code>í</code>
<code>&oacute;</code>	<code>ó</code>
<code>&uacute;</code>	<code>ú</code>
<code>&ntilde;</code>	<code>ñ</code>
<code>&iquest;</code>	<code>¿</code>

Para escribir en castellano es, sin embargo, más cómodo declarar el sistema de codificación de caracteres *ISO Latin 1* (ISO-8859-1), que permite utilizar todos los caracteres de los lenguajes de Europa occidental, sin necesidad de utilizar entidades. Para ello, se puede introducir en *head* el siguiente elemento:

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

1.4. Texto

El texto de un documento HTML se introduce como contenido de aquellos elementos que lo permitan. Los caracteres de fin de línea son tratados como espacios en blanco. Por otra parte, dos o más espacios en blanco consecutivos son tratados como un único espacio en blanco. Los navegadores se encargan de dividir el texto en líneas dependiendo del ancho de la ventana de representación, tamaño de letra, etc.

Si se desea forzar un cambio de línea, puede recurrirse al elemento *br* (marca `
`), pero sólo en situaciones justificadas. En la mayoría de las ocasiones, basta con dividir el texto en párrafos (es decir, englobar cada párrafo dentro de un elemento *p*). Por ejemplo:

```
<p>
El texto de un documento HTML se introduce como contenido
de aquellos elementos que lo permitan. Los caracteres de fin
de línea son tratados como espacios en blanco.
</p>
```

```
<p>
Por otra parte, dos o más espacios en blanco
consecutivos son tratados como un único espacio en blanco.
</p>
```

1.5. Comentarios

Los comentarios permiten incluir en el documento texto que los navegadores deben ignorar. Así, por ejemplo, el creador del documento puede dejar indicaciones útiles para

la siguiente vez que tenga que modificarlo, ocultar temporalmente texto o elementos, etc. Se especifican entre “<!--” y “-->”. Por ejemplo:

```
<!-- un comentario en una línea -->
<!-- comentario que ocupa más de una
línea -->
```

1.6. Elementos en bloque y en línea

Existen dos tipos básicos de elementos, por su forma de ser representados: elementos de bloque y elementos en línea. Los primeros siempre son representados comenzando una nueva línea (por ejemplo, párrafos, tablas, listas, etc.) Los segundos se representan siempre en la misma línea que el resto del texto o elementos a representar, sin provocar un salto de línea, salvo que sea necesario por falta de espacio en la línea anterior (por ejemplo, imágenes, elementos para especificar fuentes, cursiva, negrilla, etc.)

2. Elementos más utilizados

2.1. Estructura del documento

Todo el documento HTML está contenido dentro del elemento *html*. Por otra parte, los documentos HTML contienen dos partes claramente diferenciadas: la cabecera (*head*) y el cuerpo (*body*). La cabecera contiene declaraciones globales para todo el documento (título, sistema de codificación de caracteres, hojas de estilo, meta-datos, etc.) El cuerpo contiene la parte directamente representable del documento (texto, imágenes, tablas, etc.) Cada una de estas partes está contenida dentro de un elemento. La estructura básica es, por tanto:

```
<html>
  <head>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

2.2. Títulos

Normalmente, un documento HTML debe tener un título, que se suele mostrar en la barra superior de la ventana del navegador. Éste se especifica mediante el elemento *title*, dentro de la cabecera:

```
<html>
  <head>
```

```

    <title>Hola Mundo!</title>
  </head>
  <body>
    ...
  </body>
</html>

```

Por otra parte, es muy frecuente la necesidad de estructurar un documento en secciones (capítulos, apartados, etc), y que cada una de éstas tenga un título. En XHTML se pueden utilizar los elementos *h1*, *h2*, ..., *h7*. Estos elementos, de tipo bloque, aceptan como contenido el título de la sección. Cuanto menor es el número del nombre del elemento, de más alto nivel será la sección, como se puede ver en el siguiente ejemplo:

```

...
<body>
  <h1>Noticias de última hora</h1>
  <h2>Nacional</h2>
  <h3>Palacio asegura no haber recibido ninguna...</h3>
  <p>
    Ayer, Marruecos acusó a España en la prensa...
  </p>

  <h3>...</h3>
  <p>
    El ministro de...
  </p>

  <h2>Internacional</h2>
  <h3>...</h3>
  <p>
    Ayer se celebró en...
  </p>
</body>

```

2.3. Párrafos

El elemento *p* es un elemento de bloque que permite estructurar el texto en párrafos. Puede contener texto y cualquier elemento *en línea*.

2.4. Imágenes

Las imágenes se insertan mediante el elemento en línea *img*. La mayoría de los navegadores son capaces de representar imágenes GIF, JPEG y PNG. Este elemento no puede tener contenido, y en él se especifica mediante atributos la siguiente información:

- Fuente de la imagen (*src*): URL, absoluta o relativa, de la imagen (atributo obligatorio).
- Texto alternativo (*alt*): texto descriptivo de la imagen, pensado para navegadores incapaces de representar imágenes y para navegadores que representen el documento mediante sonido (atributo obligatorio).
- Tamaño de la imagen (*width* y *height*): ambos son atributos opcionales que permiten al navegador conocer el tamaño de las imágenes sin necesidad de descargarlas. Su utilización facilita a los navegadores el ir representando la página mientras descarga las imágenes, permitiendo así al usuario ver la página mientras se realiza la descarga.

A continuación se muestran dos ejemplos de utilización:

```


```

2.5. Enlaces

Los enlaces a otras páginas se pueden realizar mediante el elemento en línea *a*. Su contenido es el texto (o imágenes) que el usuario puede pinchar para activar el enlace. El destino del enlace se especifica mediante el atributo *href*. El siguiente ejemplo muestra un enlace a otra página dentro de un párrafo:

```
<p>
  La Escuela Politécnica de la <a href="http://www.it.uc3m.es">
  Universidad Carlos III de Madrid</a> está situada en Leganés.
</p>
```

Al igual que en el caso de las imágenes, el destino puede ser codificado de forma absoluta o relativa. También se puede asociar un enlace a una imagen, como se refleja en el siguiente ejemplo:

```
<a href="http://www.it.uc3m.es">
  
</a>
```

2.6. Listas

Las listas permiten estructurar información en *puntos*, comenzando cada punto en una nueva línea. Existen tres elementos, todos de tipo bloque, para especificar listas, uno para cada uno de los tipos de lista permitidos: listas ordenadas, listas no ordenadas y listas de definiciones.

Las listas ordenadas (elemento *ol*) asignan automáticamente un número a cada *item* de la lista. El contenido de este elemento es uno o más *items de lista* (elemento *li*).

Las listas no ordenadas (elemento *ul*) son como las anteriores, pero no asignan números a los *items*.

Las listas de definiciones (elemento *dl*) tienen como contenido elementos *dt* y *dd*. El primero permite especificar el término a definir, y el segundo, la propia definición.

A continuación se muestran ejemplos de listas, y cómo se verían en un navegador:

<pre> <p> Lista no ordenada: </p> Listas ordenadas. Listas no ordenadas. Listas de definiciones. <p> Lista ordenada: </p> Listas ordenadas. Listas no ordenadas. Listas de definiciones. <p> Lista de definiciones: </p> <dl> <dt>HTML</dt> <dl>HiperText Markup Language</dl> <dt>XML</dt> <dl>eXtensible Markup Language</dl> </dl> </pre>	<p>Lista no ordenada:</p> <ul style="list-style-type: none"> ▪ Listas ordenadas. ▪ Listas no ordenadas. ▪ Listas de definiciones. <p>Lista ordenada:</p> <ol style="list-style-type: none"> 1. Listas ordenadas. 2. Listas no ordenadas. 3. Listas de definiciones. <p>Lista de definiciones:</p> <p>HTML</p> <p style="padding-left: 2em;">HiperText Markup Language</p> <p>XML</p> <p style="padding-left: 2em;">eXtensible Markup Language</p>
--	---

2.7. Estilo del texto

Los elementos para controlar el estilo del texto se dividen en dos categorías: estilo físico y estilo lógico.

Los elementos de estilo físico regulan el formato de los caracteres de su contenido explícitamente (*b*: negrilla; *i*: cursiva; *u*: subrayado; *tt*: texto mecanografiado).

Los elementos de estilo lógico aportan información semántica acerca de su contenido, lo cual puede afectar, indirectamente, a su estilo de presentación. Se recogen en la siguiente tabla:

cite	Cita
code	Código fuente
dfn	Definido
em	Enfatizado
kdb	Palabra clave
samp	Ejemplo
strong	Resalta
var	Variable

2.8. Tablas

Las tablas resultan muy útiles para estructurar la información en filas y en columnas. Existen bastantes elementos relacionados con tablas, pero aquí sólo se explican los más relevantes: *table*, *th* (cabecera), *tr* (fila) y *td* (celda).

En HTML, una tabla se representa como un conjunto de filas. A su vez, cada fila contiene una o más celdas. En las misma tabla, todas las filas deben tener el mismo número de celdas, para poder representar las columnas correctamente alineadas.

Por defecto, cada celda se corresponde con una columna. El atributo *colspan* de *td* permite variar esta correspondencia, estableciendo su valor como el número de columnas por las que se debe expandir una celda.

A continuación se presenta un ejemplo, con su aspecto aproximado:

<pre><table border="1"> <tr> <th>izda.</th> <th>centro</th> <th>drcha.</th> </tr> <tr> <td>1 izda.</td> <td>1 centro</td> <td>1 drcha.</td> </tr> <tr> <td>3 izda.</td> <td colspan="2">3 drcha.</td> </tr> </table></pre>	<table border="1"> <tr> <td>izda.</td> <td>centro</td> <td>drcha.</td> </tr> <tr> <td>1 izda.</td> <td>1 centro</td> <td>1 drcha.</td> </tr> <tr> <td>3 izda.</td> <td colspan="2">3 drcha.</td> </tr> </table>	izda.	centro	drcha.	1 izda.	1 centro	1 drcha.	3 izda.	3 drcha.	
izda.	centro	drcha.								
1 izda.	1 centro	1 drcha.								
3 izda.	3 drcha.									

2.9. Formularios

Los formularios son elementos de bloque que pueden contener *controles*, mediante los cuales el usuario puede interactuar con la página, normalmente para enviar datos al

servidor *web*.

2.9.1. Controles

Cada control tiene un nombre, especificado mediante el atributo *name*. Además, cada control tiene asociado un valor inicial (salvo excepciones, especificado en el atributo *value*) y un valor actual (introducido por el usuario o por un *script*).

Cuando el formulario es enviado, se envía el nombre de cada uno de sus controles, y su valor actual.

2.9.2. Tipos de controles

En un formulario se pueden utilizar distintos tipos de controles:

- *Botón*: se pueden especificar tanto con el elemento *input* (*type*="button") como con el elemento *button*. Puede ser de tres tipos: *submit* (envía el formulario), *reset* (establece el valor inicial en todos los controles del formulario) y *push* (utilizado para cualquier otra cosa, en combinación con *scripts*).
- *Checkbox*: permite especificar valores *on/off*. Se especifican mediante el elemento *input* (*type*="checkbox"). Pueden utilizarse, en el mismo formulario, varios con el mismo nombre. Si se especifica el atributo *checked*="checked", se inicializa a *on*.
- *Botón radio*: también permite especificar valores *on/off*, pero de tal forma que sólo uno de los que comparten el mismo nombre puede estar en *on* simultáneamente. Se especifican mediante el elemento *input* (*type*="radio"). Si se especifica el atributo *checked*="checked", se inicializa a *on*.
- *Menú*: permite al usuario seleccionar una opción entre varias. Se crean mediante los elementos *select*, *optgroup* y *option*.
- *Texto*: permite que el usuario introduzca una o varias líneas de texto. Se especifica mediante el elemento *input* (*type*="text"), cuando se desea sólo una línea, o mediante el elemento *textarea* cuando éste puede introducir cualquier número de líneas.
- *Fichero*: permite al usuario seleccionar un fichero del sistema local de ficheros. Mediante este control se pueden enviar ficheros al servidor. Se especifica mediante el elemento *input* (*type*="file").
- *Valor oculto*: no se muestra al usuario. Su valor será siempre el valor inicial, salvo que algún *script* lo modifique. Permite que los *scripts* puedan pasar valores al servidor, y como ayuda para el control de sesiones.

2.9.3. Envío de un formulario

Cuando el usuario pincha en un botón de tipo *submit*, se produce el envío del formulario. Existen dos métodos de envío, que se especifican mediante el atributo *method* del elemento *form*:

- *GET*: utiliza el método GET de HTTP. Los parámetros se codifican concatenados en el URI especificado en el atributo *action* del elemento *form*. No se debe utilizar este método para operaciones no idempotentes.
- *POST*: utiliza el método POST de HTTP. Los parámetros se codifican en el cuerpo del mensaje HTTP.

Los parámetros a los cuales se hace mención son pares que contienen el nombre y valor actual correspondientes a todos los controles *con éxito* (en general, son aquellos que no se encuentren deshabilitados, y los botones radio y *checkboxes* que se encuentren en *on*).

Por otra parte, hay dos formas de codificar los parámetros a enviar al servidor, seleccionables mediante el atributo *enctype* del elemento *form*:

- *application/x-www-form-urlencoded*: se sustituyen los espacios en blanco por “+”, y todos los valores no alfanuméricos y caracteres reservados por “%” seguido de su valor ASCII (dos dígitos en hexadecimal). Los nombres se separan de los valores por un carácter “=”, y cada par nombre–valor se separa del resto por un carácter “&”. Por ejemplo: `nombre=Juan+P%C3%A9rez&edad=29`. Éste es tipo de codificación por defecto, y el único permitido con el método GET.
- *multipart/form-data*: más eficiente para el envío de grandes cantidades de datos, así como datos binarios. Debe ser utilizado, por ejemplo, para el envío de ficheros al servidor. El cuerpo del mensaje HTTP se separa en varias partes, cada una de las cuáles envía los datos asociados a un control.

2.9.4. Ejemplo

A continuación se muestra el código de ejemplo de un formulario.

```
<form action="http://acme.es/submit" method="post"
      encoding="multipart/form-data" >
<h3>Información Personal</h3>
<p>
  Apellidos: <INPUT name="personal_apellidos" type="text">
  Nombre: <INPUT name="personal_nombre" type="text">
  Dirección: <INPUT name="personal_direccion" type="text">
</p>

<h3>Historia Médica</h3>
<p>
```

```



```

3. Hojas de estilo CSS

Las primeras versiones de HTML permitían especificar el estilo de los documentos (por ejemplo, los tipos y tamaños de fuente, colores de texto, colores de fondo, ancho de elementos, etc.) Sin embargo, actualmente, es preferible utilizar, para esto, CSS (*Cascading Style Sheet*). El mejor manual de referencia de CSS es su especificación [2].

3.1. Declaración de hojas de estilo

El lenguaje XHTML es independiente del lenguaje utilizado para especificar las hojas de estilo. Por ello, en la cabecera del documento, es recomendable declararlo. En el caso de CSS la declaración necesaria es la siguiente:

```
<meta http-equiv="Content-Style-Type" content="text/css">
```

Por otra parte, las hojas de estilo a utilizar para un documento pueden ser incluidas en el propio documento (con el elemento *list* en la cabecera), en un documento externo (elemento *link* en la cabecera) o directamente en un elemento (atributo *list*). A continuación se muestran algunos ejemplos:

```
<head>
```

```

...
<link href="special.css" rel="stylesheet" type="text/css">
</head>

<head>
...
<style type="text/css">
  h1 {border-width: 1; border: solid; text-align: center}
</style>
</head>

<p style="font-size: 12pt; color: fuchsia">
Aren't style sheets wonderful?
</p>

```

3.2. Colores y tamaños

Aquí se presentan los tipos de datos más habituales de las propiedades de CSS: colores y tamaños.

3.2.1. Colores

Hay dos formas de especificar un color: por sus coordenadas RGB (rojo, verde y azul), o, para los colores predefinidos, por su nombre.

Las coordenadas RGB se dan indicando la coordenada de cada uno de estos colores (cada coordenada entre 0 y 255). Por ejemplo, `rgb(64,128,255)` especifica un color con 64 de rojo, 128 de azul y 255 de verde.

Los colores predefinidos en CSS son los siguientes: *aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white y yellow.*

3.2.2. Tamaños

Los tamaños permiten especificar altura y anchura de elementos, tamaños de márgenes, tamaños de fuentes, etc. Pueden ser especificados de forma relativa a otra propiedad (*em*, tamaño de fuente; *ex*, valor *x-height* de la fuente; *px*, tamaño de pixel) o de forma absoluta (*in*, pulgadas; *cm*, centímetros; *mm*, milímetros; *pt*, puntos, o 1/72 pulgadas; *pc*, picas, o 12 puntos).

También se pueden especificar longitudes como porcentajes. En el siguiente ejemplo se establece un margen izquierdo del 10% del ancho del párrafo:

```
p { margin-left: 10% }
```

3.3. Asignación de propiedades a elementos

CSS define un conjunto de propiedades y los posibles valores de las mismas. Cada declaración está compuesta por dos campos: *selector* y *declaración*. El selector establece las reglas de selección de elementos. La declaración especifica propiedades para los elementos seleccionados. Por ejemplo, en el siguiente ejemplo, se asigna un color de texto, un tamaño de fuente y texto en *negrilla* a todos los elementos *p* del documento:

```
p { color: rgb(0,0,128);  
    font-size: 14pt;  
    font-weight: bold }
```

Como se ve en él, primero se especifica el selector, y a continuación las declaraciones entre llaves, separadas entre sí por “;”. El nombre y el valor de cada propiedad se separan por “:”. Los tamaños de fuente se suelen especificar en *puntos* (*pt* en CSS). Los colores pueden ser especificados utilizando las coordenadas RGB, o el nombre del color, para aquellos predefinidos

En la siguiente tabla se muestran los patrones más importantes que se pueden utilizar en el selector.

Patrón	Elementos seleccionados
*	Cualquier elemento.
E	Cualquier elemento E.
E F	Cualquier elemento E descendiente de F.
E > F	Cualquier elemento E hijo de F.
E:first-child	Cualquier elemento que sea el primer hijo de un elemento E.
E:link	Cualquier elemento E tal que sea origen de un enlace cuyo destino no haya sido visitado.
E:visited	Ídem, con destino visitado.
E + F	Cualquier elemento F inmediatamente precedido por un elemento E.
E[foo]	Cualquier elemento E en el que esté establecido el atributo <i>foo</i> .
E[foo="warning"]	Cualquier elemento E cuyo atributo <i>foo</i> tenga el valor <i>warning</i> .
E[foo~="warning"]	Si el valor del atributo <i>foo</i> es una lista de valores separados por espacios y uno de ellos es el valor <i>warning</i> .
E.warning	Cualquier elemento E cuyo atributo <i>class</i> tenga el valor <i>warning</i> .

A continuación se muestran ejemplos sencillos de utilización de propiedades de CSS básicas:

```
p { color: red }
```

```

p { background-color: rgb(128,129,255) }
p { background-image: url("http://www.foo.com/image.png" }
p { background: url("http://www.foo.com/image.png" }
p { background: black }

p { text-indent: 3em }           // em = tamaño de fuente
p { text-align: center }       // left, center, right, justify
p { text-decoration: underline } // underline, overline,
                                // line-through, blink, none
p { text-transform: capitalize } // capitalize, uppercase, lowercase,
                                // none

p { font-style: italic }       // normal, italic, oblique
p { font-weight: bold }       // normal, bold, bolder, lighter,
                                // 100, 200, ..., 900
p { font-size: large }        // xx-small, x-small, small,
                                // medium, large, x-large,
                                // xx-large, o un tamaño
                                // relativo o absoluto.

p { margin-left: 10% }        // 10% del ancho
p { margin-top: 10px }        // 10 pixels
p { margin-bottom: 1cm }      // 1 cm
p { margin-right: 5em }       // 5 veces el tamaño de fuente

p { padding-left: 10% }       // otros: padding-right, padding-top,
                                // padding-left, padding-bottom

```

3.4. Agrupación de elementos

En muchas ocasiones resulta interesante aplicar las mismas propiedades a un grupo de elementos o texto. Se pueden agrupar utilizando los elementos *div* y *span*. La diferencia entre ambos es que el primero es de tipo bloque (y, por tanto, fuerza un salto de línea antes y después), mientras que el segundo es de tipo *en línea*. Se muestra, a continuación, un ejemplo que hace uso de ambos:

```

<head>
...
<style type="text/css">
  div.resumen { text-align: justify }
  span.inicio { font-weight: bold }
</style>
</head>
<body>
<div class="resumen">

```

```
<p><span class="inicio">La Bolsa española</span> acumula
nueve jornadas de caídas consecutivas, la peor estadística
desde la Guerra del Golfo. </p>
<p><span class="inicio">El Ibex </span> llegó al triste
récord de cinco mínimos anuales seguidos y restó un
3,39% para situarse en los 5.390 puntos. </p>
</div>
</body>
```

Referencias

- [1] “HTML 4.01 Specification”
<http://www.w3.org/TR/html401>
- [2] “Cascading Style Sheets, level 2. CSS2 Specification”
<http://www.w3.org/TR/REC-CSS2>