

Introducción al diseño de páginas web con estándares y programas *open source*: Tutorial de XHTML, CSS y **NVU**

Por Lluís Codina y Cristòfol Rovira
UPF, Marzo 2007

Primera Parte: El código fuente de una página web

1. Creación de documentos Web

1.1. Páginas Web y código fuente

El contenido de una página web se puede ver de dos formas diferentes:

- Como **código fuente**. Para verlo así, podemos usar un simple editor de textos, como el *Bloc de Notas*, o un editor de páginas Web como *NVU* (volveremos sobre este programa más adelante).

- Como el **resultado** de interpretar el código fuente. Es lo que hacemos de forma natural cada vez que accedemos a una página web con un navegador estándar de Internet como Internet Explorer, Mozilla, Opera, etc.

Las siguientes capturas de pantalla muestran las dos formas distintas de ver una página web:

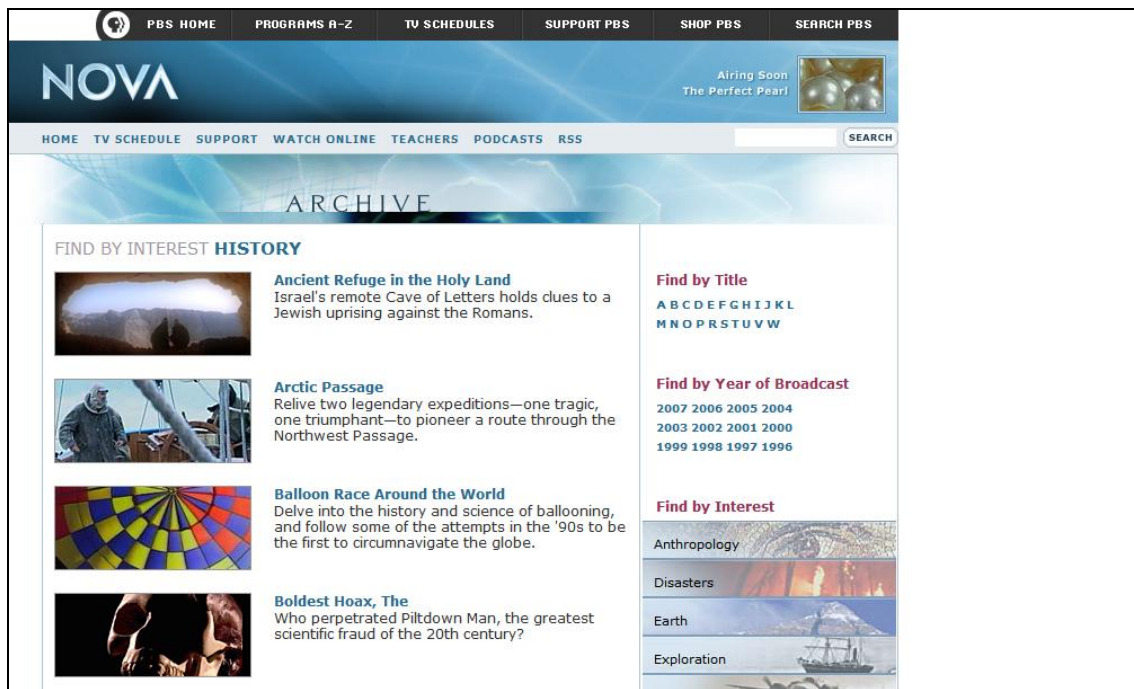


Figura 1: Una página web de la televisión pública de Estados Unidos (PBS) dedicada a documentales científicos

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head><title>NOVA | Archive | History | PBS</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="description" content="Listing of NOVA Web sites by subject - History" />
<link rel="stylesheet" href="/wgbh/nova/css/utility-06.css" type="text/css" media="screen,print" />
<script src="/wgbh/nova/js/novagen-05.js" type="text/javascript" language="Javascript1.1"></script>
</head>
<body>
<div id="ct_pbsnav"><div id="pbsnav">

<!--
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:trackback="http://madskills.com/public/xml/rss/module/trackback/"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description
  rdf:about="http://www.pbs.org/wgbh/nova/archive/int_hist.html"
  trackback:ping="http://www.pbs.org/cgi-registry/trackback.pl/wgbh/nova/archive/int_hist.html"
  dc:identifier="http://www.pbs.org/wgbh/nova/archive/int_hist.html" />
</rdf:RDF>
-->

```

Figura 2: La misma página, pero vista ahora como código fuente (haciendo clic en Ver > Código fuente, en este caso con Firefox).

El **código fuente** de una página Web es un documento de texto que combina:

- i), **contenido** de la página web;
- ii), **instrucciones** sobre cómo debe tratar (o mostrar) ese contenido el navegador;
- iii), opcionalmente, **enlaces** para acceder a otros documentos o para incorporar elementos gráficos (p.e. una imagen) a la misma página web.

Con el navegador vemos las tres cosas como si fueran una unidad. Para hacer un símil literario (o cinematográfico), una página web es como una criatura de *Frankenstein*, ya que consiste en la unión (con costuras más o menos disimuladas) de varias cosas distintas.

La totalidad del código fuente (contenido + instrucciones + enlaces) consiste en texto, incluso en el caso de páginas Web con contenido de tipo multimedia. ¿Por qué? Veamos: el contenido consiste, o bien en texto o bien en enlaces para integrar objetos multimedia. Estos enlaces se expresan mediante texto. A su vez, las instrucciones consisten siempre en unos elementos llamados **etiquetas** que consisten, a su vez, en texto. Por ejemplo, para integrar una imagen en una página se utiliza la etiqueta ``. Para mostrar un texto en cursiva en una página Web se rodea dicho texto con las etiquetas `...`, etc.

Visto de otra forma, podemos decir también que la totalidad del código fuente de una página Web consiste en: (a), el **contenido** textual; más (b), **etiquetas**, que, a su vez, pueden ser de dos tipos:

b1. Instrucciones sobre **cómo mostrar** la información que contiene el documento. Ejemplo, la etiqueta `<h1>` para indicar que el texto en cuestión es un título de primer nivel, es decir, el título principal del documento. Así, un contenido rodeado por las etiquetas `<h1>`, ... `</h1>` provocará que se muestre en un cuerpo mayor que el resto del texto y en negritas.

b2. Instrucciones sobre **cómo integrar** elementos a la página web, típicamente imágenes, pero también facilidades para acceder a otras páginas web o para abrir otros documentos. Ejemplo, la instrucción `` serviría para incluir una imagen como parte de una página Web; mientras que la instrucción `...` servirá para acceder a un documento externo, etc.

Naturalmente, no es en absoluto obligatorio ni necesario que una página Web contenga gráficos. Aclaremos estas ideas con los siguientes ejemplos.

1.2. Lenguaje XHTML e interpretación de páginas

Los detalles concretos sobre qué etiquetas se pueden utilizar en una página Web y cómo se utilizan esas etiquetas están establecidos en una norma

internacional denominada **XHTML** (e**X**tended **H**yper**T**ext **M**arkup Language).

Como XHTML contiene etiquetas, que ya hemos visto que son equivalentes a palabras, e indicaciones sobre cómo usar esas etiquetas, que equivalen a una sintaxis, se suele decir que es un **lenguaje**.

Para que una parte de la información de una página Web, por ejemplo, una parte de la cadena de caracteres "Tercer Congreso sobre Cine Europeo" aparezca en cursiva (*Historia del Cine Europeo*), en el documento fuente de la página Web deberá haber, al menos, dos cosas:

1. El contenido en sí mismo, en este caso, la cadena de caracteres: "Tercer Congreso sobre Cine Europeo"
2. Las instrucciones, en este caso, la instrucción concreta: "muestra la cadena de caracteres en negrita"

Estas dos cosas, mediante el lenguaje HTML, se combinan así:

```
...  
Tercer Congreso sobre <em>Cine Europeo</em>  
...
```

El resultado, una vez interpretado por el navegador, se verá así:

```
...  
Tercer Congreso sobre Cine Europeo  
...
```

La instrucción se ha expresado mediante el par de etiquetas `...`. El ejemplo nos muestra, de paso, porqué la mayor parte (con unas pocas excepciones) de las instrucciones o elementos funcionan mediante pares de etiquetas. La primera etiqueta del par muestra dónde empieza a aplicarse la instrucción (``), el segundo muestra dónde acaba la instrucción (``).

Ahora bien, ¿qué es lo que ha hecho exactamente el navegador con ese conjunto de instrucciones e informaciones? Ha hecho lo siguiente: (1) ha leído la información y las etiquetas que la rodean, (2) ha interpretado las etiquetas de acuerdo con su intérprete interno de lenguaje XHTML, (3) ha retirado las etiquetas en el momento de presentar la información en el monitor y (4) ha formateado el texto de acuerdo con las instrucciones para presentarlo de manera adecuada en el monitor.

Por tanto, cuando decimos que XHTML es el lenguaje de la Web, lo que estamos diciendo es que los navegadores Web incluyen un intérprete de XHTML.

En resumen: para ver el **resultado** de interpretar el código fuente usamos el navegador. Para ver el **código fuente** (texto + instrucciones) abrimos el

documento Web con un editor de texto o usamos la opción específica del navegador (**Ver > Código fuente**).

No existe ninguna diferencia, y esto es muy importante, entre una página Web creada con un programa tan simple como el *Bloc de Notas* como con un programa tan sofisticado como *Dreamweaver*: en rigor, el código fuente es independiente del programa con el que se haya creado.

Una página Web puede contener elementos multimedia (como gráficos, sonido, vídeo, etc.) que no forman parte del lenguaje XHTML, pero tales elementos se enlazan y se incluyen en la página web mediante etiquetas del lenguaje XHTML.

1.3. Etiquetas y elementos XHTML

Como ya hemos visto, las etiquetas XHTML que contienen las instrucciones para el navegador, están formadas por los signos "<" y ">" y uno o más caracteres. Por ejemplo, la etiqueta <h1> (por *Heading 1* que significa Título de nivel 1) sirve para identificar el *título principal* del documento y fuerza al navegador Web a mostrarlo con una tipografía determinada (negritas y un cuerpo grande de texto).

Por ejemplo, el *Título principal* de una página Web sobre Historia del Cine Europeo se podría marcar así:

```
...  
<h1>Historia del Cine Europeo</h1>  
...
```

Obsérvese que la etiqueta de cierre se forma añadiendo una barra inclinada (</h1>). El par de etiquetas XHTML (<h1>, </h1>) más el contenido afectado por ellas (Historia del Cine Europeo) se denomina **elemento**. Un conjunto de elementos forman un documento XHTML o una página Web (ambos términos son sinónimos en este contexto).

En relación a las etiquetas y elementos, una página Web o un documento HTML presenta siempre la misma estructura básica:

- Un indicador inicial y final de que la página contiene código fuente XHTML (<html>...</html>);
- Un elemento de cabecera, (<head>...</head>) inmediatamente después de la primera etiqueta <html>, y por tanto al inicio del documento, para contener datos descriptivos (metadatos) sobre el documento;
- el cuerpo del documento en sí (<body>...</body>)

La estructura global queda así (los puntos suspensivos indican el lugar donde podría ir el contenido del documento y otras etiquetas XHTML):

Estructura básica de un documento XHTML (sin declaración DOCTYPE)

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

</body>
</html>
```

Dentro del elemento `<head>` y del elemento `<body>` puede (y debe) haber otros elementos, por ejemplo, `<head>` debe contener al menos el elemento `<title>`, que, por cierto, no es el mismo tipo de título que el del elemento `<h1>` que hemos comentado antes. El contenido de la de la etiqueta `<title>` de esta zona es el título que aparece en la barra superior (generalmente de color azul) del navegador, no el título que aparece dentro de la página del documento Web. Ejemplos posteriores aclararán esto.

También conviene añadir en la sección `<head>` etiquetas que informan sobre el contenido, autor del documento, etc. Estas etiquetas se denominan meta datos. Se verán más adelante.

Además, el documento debe incluir una declaración técnica denominada DOCTYPE. En el caso del lenguaje más utilizado actualmente en la creación de páginas web se trata de la siguiente:

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

La (críptica) declaración anterior consiste en dos partes bien diferenciadas: la designación del tipo de documento completo, en este caso **XHTML Transitional** y la dirección donde puede encontrarse la definición técnica precisa de ese tipo de documento (<http://www.w3.org>, etc.).

Además, como la Web tiene (o aspira a tener) un alcance universal, es importante declarar explícitamente el lenguaje principal del documento, cosa que se hace utilizando la etiqueta `<html>`, de esta forma:

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

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
```

En la declaración anterior, donde indica "es" como lengua principal puede indicar "ca" si la lengua principal del documento es el catalán (o "en" si es en inglés, etc.).

Por tanto, con la declaración DOCTYPE y la expresión de la lengua principal del documento, la estructura básica de una página web queda así:

Estructura básica de un documento XHTML con declaración DOCTYPE

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

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">

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

En la sección `<body>` es donde está el contenido del documento en sí. Este contenido debe estructurarse con otras etiquetas. Una de las más utilizadas son las etiquetas de cabeceras, o *headings*, de las que hay hasta seis niveles, de manera que un documento HTML puede tener secciones y subsecciones utilizando los diversos niveles de `<h n >`, donde n puede ser cualquier valor entre 1 y 6, de esta manera: `<h1>`, `<h2>`... `<h6>`.

2. Creación de un documento HTML simple

Hay tres tipos de programas distintos para crear páginas Web:

1. Editores de código fuente no especializados, como el *Bloc de Notas*
2. Editores de código fuente especializados, como *PSPAD*
3. Editores en modo gráfico, como *NVU*

Cada una de estas opciones proporciona de forma progresiva más ayuda para la creación de páginas Web. Ahora bien, **para aprender a crear páginas Web** las opciones más recomendables son la 1 y la 2, **y nunca la 3.**

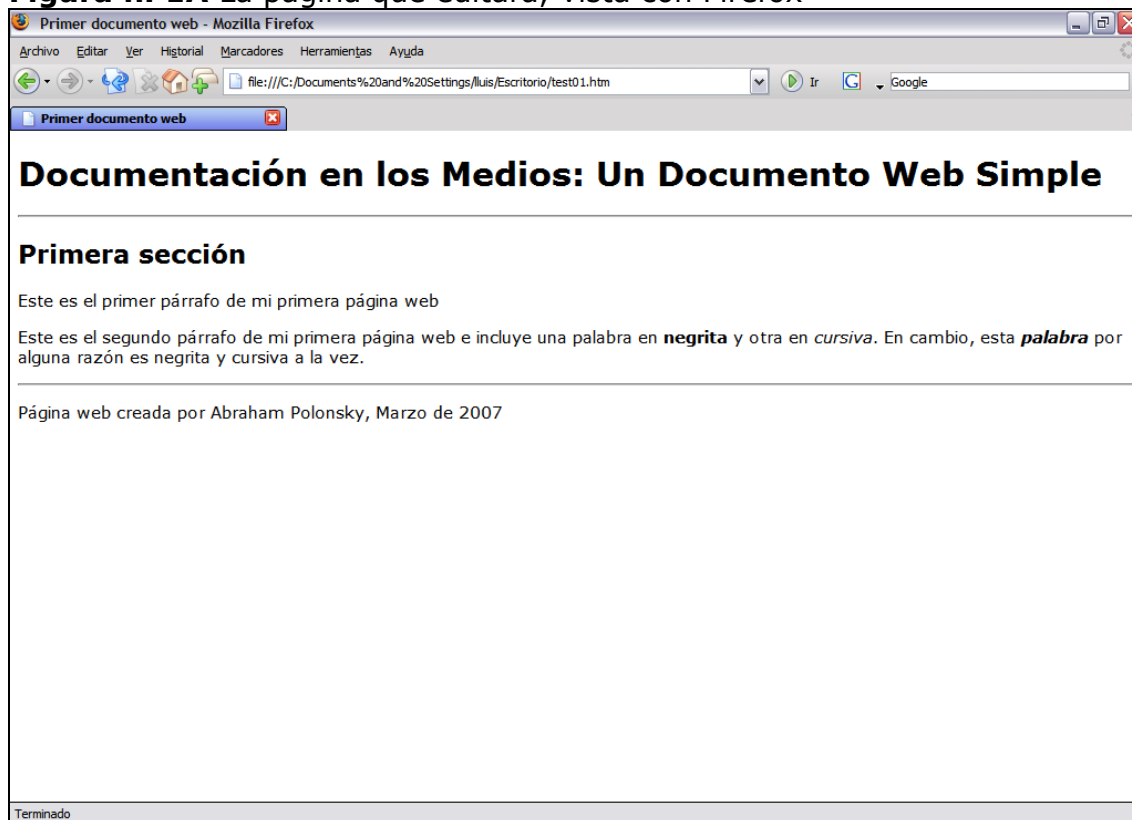
Las opciones 1 y 2 nos obligan a practicar con los fundamentos de la programación XHTML, mientras que la opción 3 nos oculta los detalles de esa programación, ya que esa clase de editores funciona de modo parecido a editores gráficos estándar como Word, evitando que sepamos qué hay realmente en el interior de una página Web. Si luego necesitamos realizar labores de mantenimiento o rectificar errores en la página Web, puede resultar muy difícil o imposible de realizar a todos los efectos prácticos.

Las prácticas siguientes recomendamos que se hagan con el *Bloc de Notas*.

2.1. Primera página web

Active el *Bloc de Notas*. Vamos a crear un documento web con el siguiente texto y estructura (lo que sigue es el resultado, no lo que usted deberá escribir, que se indicará más adelante):

Figura n. 1A La página que editará, vista con Firefox



El código fuente que debe escribir en el documento con su editor de textos es el que aparece en el recuadro siguiente (destacamos las etiquetas en azul para facilitar la lectura del código fuente, pero tenga en cuenta que el Bloc de Notas no colorea el código):

Figura 1B: Código fuente de la página web anterior

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

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">

<head>
  <title>Primer documento web</title>
</head>

<body>
<h1>Documentación en los Medios: Un Documento Web Simple</h1>
<hr />
<h2>Primera sección</h2>
<p>Este es el primer párrafo de mi primera página web</p>
<p>Este es el segundo párrafo de mi primera página web e incluye una
palabra en <strong>negrita</strong> y otra en <em>cursiva</em>. En
cambio, esta <strong><em>palabra</em></strong> por alguna razón es
negrita y cursiva a la vez.</p>
<hr />
<p>Página web creada por Abraham Polonsky, Marzo de 2007</p>
</body>

</html>
```

Para crear la página web anterior, deberá seguir estas instrucciones:

1. **Abrir el *Bloc de Notas*.**

2. **Dar nombre al documento** (aunque aún no haya escrito nada) con el nombre **test01.htm**. Puede guardar el documento en su espacio de disco personal.

Los documentos Web siempre deberán llevar la extensión **htm** (o **html**). Nosotros recomendamos **htm**. Por tanto, el nombre que puede dar al documento es **test01.htm**. Haga: **Archivo > Guardar** en el *Bloc de Notas* y de el nombre indicado (sin cerrar ni salir del *Bloc de Notas*).

3. **Escribir el código fuente.** El recuadro anterior (**Figura 1b**) muestra de nuevo el código fuente que debe entrar en el editor de textos, pero ahora sin colorear. Cambie los datos contextuales que deba cambiar (nombre real del alumno en lugar de Abraham Polonsky y fecha real de realización de la práctica) y sobre todo asegúrese de entrar el código fuente **exactamente** igual. Las tres primeras líneas:

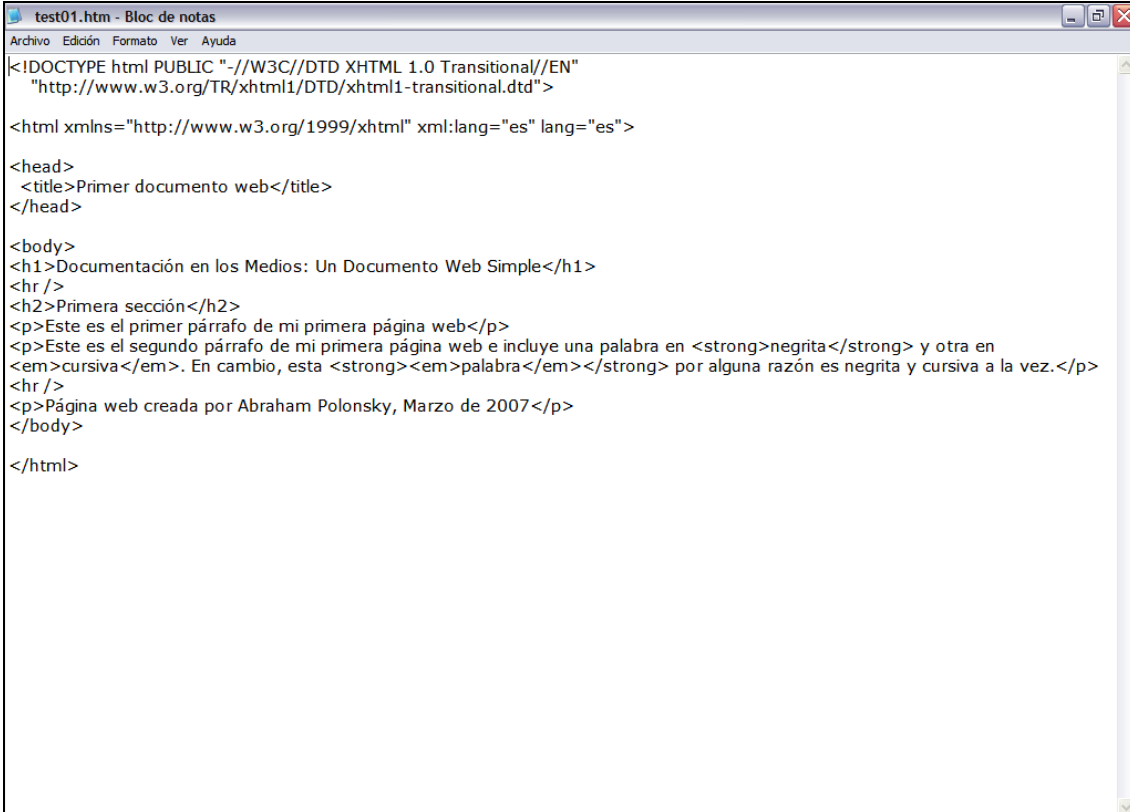
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
```

Puede copiarlas y pegarlas en el *Bloc de Notas*, puesto que siempre son idénticas en cualquier página web (cambie "es" por "ca" si escribe en catalán). El resto del código fuente, desde `<head>` hasta `</html>` es **fundamental** que lo teclee el alumno directamente.

Si se escribe el código fuente anterior con el *Bloc de Notas* el resultado deberá ser igual al de la página siguiente (**Figura 1c**):

Figura 1C (con Bloc de Notas)

A screenshot of a Notepad window titled "test01.htm - Bloc de notas". The window contains the following HTML code:

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

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">

<head>
<title>Primer documento web</title>
</head>

<body>
<h1>Documentación en los Medios: Un Documento Web Simple</h1>
<hr />
<h2>Primera sección</h2>
<p>Este es el primer párrafo de mi primera página web</p>
<p>Este es el segundo párrafo de mi primera página web e incluye una palabra en <strong>negrita</strong> y otra en
<em>cursiva</em>. En cambio, esta <strong><em>palabra</em></strong> por alguna razón es negrita y cursiva a la vez.</p>
<hr />
<p>Página web creada por Abraham Polonsky, Marzo de 2007</p>
</body>

</html>
```

Una vez guardado, compruebe que el resultado corresponde al de la ilustración de la **Figura n. 1a**. Para ello, vaya a la carpeta o unidad de disco donde haya guardado el documento, haga clic en el icono del documento y se abrirá directamente con el navegador de páginas web. No es gran cosa como página web, pero es una página web a todos los efectos (ya aprenderemos cómo mejorar su aspecto).

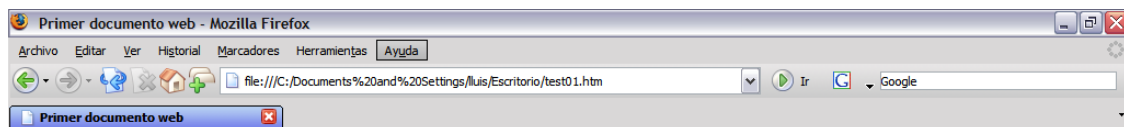
Ahora podemos comentar el significado de las etiquetas que hemos introducido, aparte de las ya comentadas. Se trata de las siguientes:

Tabla 1a: etiquetas básicas

Etiqueta	Explicación
title	No es el título que aparece en el cuerpo del documento, sino el título que aparece en la banda superior del navegador.
h1, h2	Las etiquetas h1, h2, etc., significan literalmente cabecera (o sea, títulos) de nivel 1, nivel 2, etc. Son los diferentes niveles de títulos que aparecen en el cuerpo de muchos documentos relativamente complejos y que deben presentar su contenido más o menos organizado por capítulos, subcapítulos, etc.
hr	Línea horizontal. Traza una línea horizontal. No necesita etiqueta de cierre, pero hay que incluir una barra inclinada al final.
p	Párrafo.
strong	Para marcar un texto en negrita.
em	Abreviatura de "énfasis". Marca un texto en cursiva.

3.1. Modificación de la página

El ejercicio es el siguiente: deberá generar el código fuente que requiere la siguiente página Web (vea las **Instrucciones para la práctica** a continuación de esta captura de pantalla):



Documentación en los Medios: Un Documento Web Simple

Primera sección

Este es el primer párrafo de mi primera página web

Este es el segundo párrafo de mi primera página web e incluye una palabra en **negrita** y otra en *cursiva*. En cambio, esta **palabra** por alguna razón es negrita y cursiva a la vez.

Mis principales intereses

1. Guiones
2. Televisión de calidad
3. Producción de documentales



Página web creada por Abraham Polonsky, Marzo de 2007

Instrucciones para la práctica

1. Recuerde que, donde dice "Abraham Polonsky" puede poner su nombre y primer apellido y que la fecha puede ser la fecha real de la práctica.

3. Donde dice "Intereses principales 1. Guiones 2. Televisión de calidad, etc." puede indicar sus intereses reales en relación al mundo de la comunicación audiovisual o con cualquier aspecto de sus intereses.

4. Observe que los tres elementos de la lista de intereses están separados entre sí por simples saltos simples de línea, no por párrafos. Todo lo anterior se debe a que existe una etiqueta XHTML para crear listas numeradas y cuya sintaxis se indica en la siguiente tabla:

Tabla 2: Etiquetas para listas

Para indicar inicio lista	<code></code> por <i>ordered list</i>
Para indicar final lista	<code></code>
Etiqueta para cada elemento de la lista	<code></code> por <i>list item</i>
Ejemplo	<pre>... Producción Guiones de televisión ..</pre>
Resultado que muestra el navegador	<pre>... 1. Producción 2. Guiones de televisión ...</pre>

Observe que, con una codificación como la anterior, es el navegador el que interpreta el orden de los elementos y añade la numeración. Si borra o añade elementos, no tiene que preocuparse por la numeración, ya que la reordena automáticamente el navegador. También añade de modo automático un sangrado. Intente aplicarla a este ejercicio.

El nuevo código fuente debe quedar así:

```
...
<p>Este es el segundo párrafo de mi primera página web e incluye una
palabra en <strong>negrita</strong> y otra en <em>cursiva</em>. En
cambio, esta <strong><em>palabra</em></strong> por alguna razón es
negrita y cursiva a la vez.</p>

<h2>Mis principales intereses</h2>

<ol>
<li>Guiones</li>
<li>Televisión de calidad</li>
<li>Producción de documentales</li>
</ol>

<hr />
<p>Página web creada por Abraham Polonsky, Marzo de 2007</p>
</body>
...
```

5. Comprobaciones. Una vez codificado y guardado el documento, ábralo con el navegador y compruebe el resultado. Recuerde que, cada vez que haga cambios en su documento XHTML debe guardarlo antes de poder ver los cambios en el navegador. Además, el navegador no mostrará ningún cambio hasta que usted no le obligue a **Actualizar** la página con el botón correspondiente.

6. El poder del código. Cambie las dos etiquetas , ... por las etiquetas ... (unordered list). Es decir, el código fuente deber quedar ahora así:

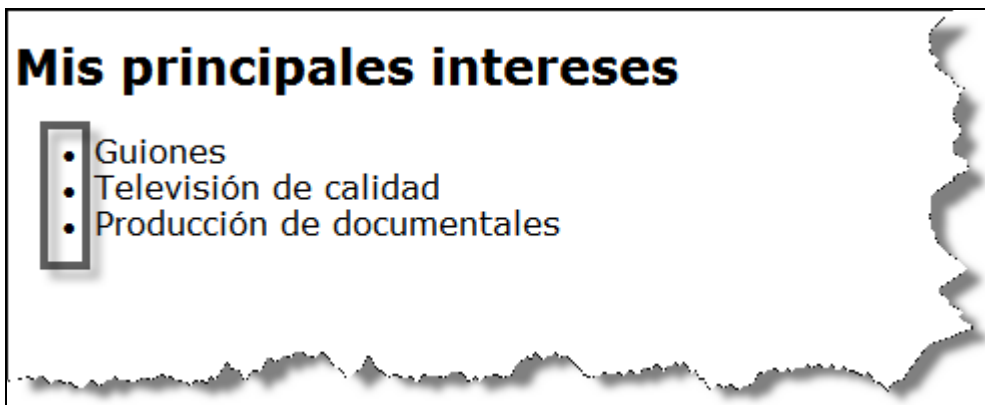
```
...
<p>Este es el segundo párrafo de mi primera página web e incluye una
palabra en <strong>negrita</strong> y otra en <em>cursiva</em>. En
cambio, esta <strong><em>palabra</em></strong> por alguna razón es
negrita y cursiva a la vez.</p>

<h2>Mis principales intereses</h2>

<ul>
<li>Guiones</li>
<li>Televisión de calidad</li>
<li>Producción de documentales</li>
</ul>

<hr />
<p>Página web creada por Abraham Polonsky, Marzo de 2007</p>
</body>
...
```

Compruebe el resultado en el navegador. En lugar de número, ahora debe haber viñetas:



3.2. Enlaces hipertextuales

Ahora practicaremos con uno de los elementos más característicos de l Web: los enlaces hipertextuales. Lo que sigue es una explicación. Aún no tiene que hacer nada, salvo estudiar lo que se indica:

3.2.1. Enlaces

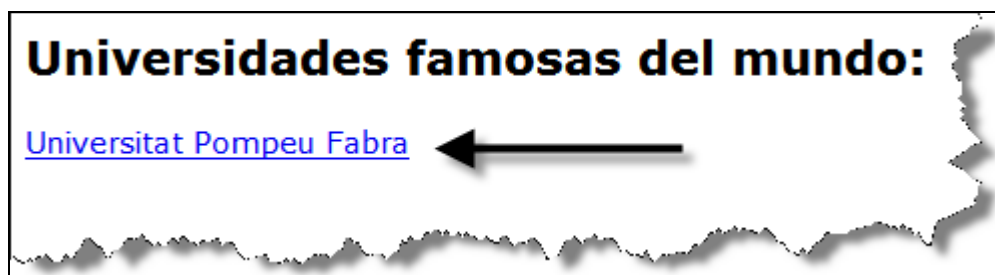
El punto de inicio de un enlace se denomina **anclaje**. Las etiquetas que marcan el punto de inicio de un enlace hipertextual son: `<a href>...` (a por *anchorage*, anclaje). Ejemplo:

```
<a href="camino">texto que sirve de anclaje de inicio para un enlace</a>
```

En el ejemplo de código fuente anterior, la expresión "camino" representa diversas cosas según el caso. "Camino" puede ser otra página web, entonces adopta una forma como "<http://www.upf.edu>", pero también puede ser una sección del mismo documento, o el nombre y el camino de un archivo externo situado en el disco duro del ordenador local. Vamos a ver estos posibles tipos de enlace. Un ejemplo típico de enlace codificado en una página web mediante XHTML:

```
...  
<h2>Universidades famosas del mundo</h2> :  
<a href=http://www.upf.edu>Universitat Pompeu Fabra</a>
```

En el navegador el enlace anterior aparecerá así:



El conjunto `Universitat Pompeu Fabra` nos sirve para señalar algunas características del código XHTML. Es la primera vez que vemos aquí una etiqueta que posee lo que se denomina un **atributo**. El atributo es `href`. Además, el atributo tiene un valor, que siempre aparece entre comillas y después del signo = y que aquí es "`http://www.upf.edu`". Las etiquetas de inicio y de cierre rodean a una porción de texto, en este caso: `Universitat Pompeu Fabra`. El texto rodeado por las etiquetas se denomina contenido afectado.

Por último, el conjunto formado por una etiqueta de apertura, una etiqueta de cierre, uno o más atributos, el valor de los atributos y el contenido afectado se denomina **elemento**. El siguiente diagrama ilustra estas ideas:

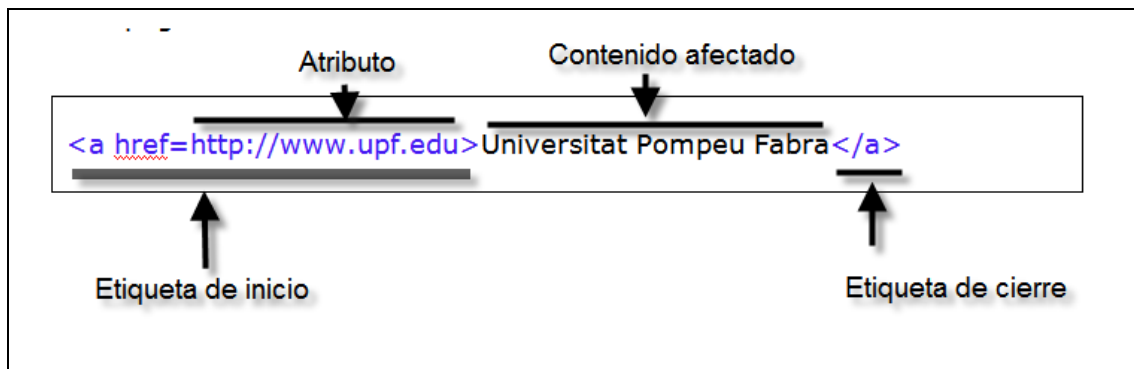


Ilustración 1: Un elemento XHTML se compone de etiquetas, atributos y contenido afectado

Enlaces entre secciones del mismo documento (*bookmarks, named anchors*)

A veces, podemos desear enlazar secciones del mismo documento entre sí. Se trata de una opción sumamente útil que sirve para proporcionar algo de navegación y de interactividad a documentos unitarios muy extensos. Algunos editores de páginas Web en castellano denominan **marcadores** a esta clase de enlaces; los programas en inglés los denominan *bookmarks* o *named anchors*.

La cuestión es que, para poder enlazar entre sí distintas secciones de un mismo documento, primero hay que **nombrar** explícitamente cada una de estas secciones, mediante un atributo `name`, de aquí el nombre *named anchors*. Pero, cada nombre es también una marca, de aquí el término marcadores o *bookmarks*.

Primera parte: nombrar las secciones

Para poner **nombrar** (o marcar) una sección se dispone del atributo `name`. Se utiliza así:

```
...
<a name="capitulo3">Capítulo 3: El cine fantástico en el Festival de Sitges
</a>
...
```

Analicemos esta instrucción. Vemos que contiene la etiqueta en sí misma, en este caso: `<a>...`; un atributo, en este caso: `name`. Por último, hemos indicado un valor: `"capitulo3"`. Las comillas son imprescindibles.

El significado en lenguaje natural de esta instrucción equivale a esto: `"capitulo3"` es el nombre que servirá para enlazar al Capítulo 3. No es necesario que la marca tenga el mismo nombre que la sección. Es

aconsejable usar nombre nemotécnicos para las marcas, pero nada más. Observe que hemos obviado el acento en el nombre de la marca. Es aconsejable no usar nombres con acentos. También podríamos haber puesto un número en lugar de un nombre, etc. La cuestión es que, después, nos acordemos de qué secciones corresponden a cada marca si tenemos un documento muy largo con muchas marcas.

¿Por qué no podríamos enlazar directamente al Capítulo 3 sin ponerle una marca? ¿Acaso no tiene ya un nombre? La razón es que los ordenadores no pueden pensar ni realizar inferencias. Para el ordenador, y por tanto para el navegador, "Capítulo 3" es solamente una cadena de caracteres más de las muchas que hay en la página Web. Si no ponemos una marca no es capaz de detectarla como una sección susceptible de ser enlazada.

Atención: otra cuestión muy importante, el atributo `name` no produce una acción de enlace, sino que nombra/marca una sección de un documento para que pueda **ser objeto** de una acción de enlace. La colocación de este marcador no producirá ningún cambio en el aspecto del texto que queda rodeado por las etiquetas.

Por último, se supone que si este documento tiene 5 secciones, hemos asignado cinco nombres/marcas a cada sección como hemos mostrado con el ejemplo de la hipotética sección 3.

Segunda parte: enlazar las secciones

Para acceder a una sección del documento identificada con un nombre o marcador, necesitamos que, en alguna otra parte del documento, haya una etiqueta `<a>` con el atributo `href`, según se muestra en esta porción de código fuente:

```
...
<h2>Capítulo 2: El cine fantástico en Cataluña</h2>
<p>En relación al fantástico, en el <a href="#capitulo3">Capítulo 3</a>
de esta obra nos referiremos con detalle a la historia del Festival de
Sitges</p>
...
```

lo cual se verá así en el navegador:

```
...
Capítulo 2: El cine fantástico en Cataluña

En relación al fantástico, en el Capítulo 3 de esta obra nos referiremos con
detalle a la historia del Festival de Sitges
...
```

Observaciones:

1. El uso del atributo `href` para esta clase de enlaces a la sección de un documento, siempre tiene esta sintaxis: `href="valor"`, por ejemplo,

`href="#capitulo3"`, es decir, delante del nombre y todo rodeado por comillas, se pone el símbolo #. (El símbolo # se activa con AltGr + tecla del número 3).

2. Los enlaces con el atributo `href`, cualquiera que sea el destino, modifican el aspecto del texto que sirve de punto de activación del enlace. Por defecto, convierten el texto en texto subrayado de color azul, así como cambian la forma del cursor cuando se desliza sobre el texto.

3.2.2. Enlaces entre páginas Web

Como ya sabemos, podemos enlazar una página Web a una o más páginas Web externas a través de la WWW. La etiqueta tiene la forma que se muestra en esta porción de código fuente:

```
...  
<h2>Universidades famosas del mundo</h2>  
<h3>Cataluña</h3>  
<a href="http://www.upf.es">UPF: La mejor universidad del mundo</a>  
...
```

Lo cual tendrá este aspecto en el navegador:

```
...  
Universidades famosas del mundo  
  
Cataluña  
  
UPF: La mejor universidad del mundo  
...
```

Observaciones:

1. La etiqueta `<a>...` marca el lugar donde, si se hace clic, se activará el enlace. Este lugar de activación del enlace se denomina **anclaje de inicio**. El atributo `href` indica cuál será el destino del enlace, por esa razón, se denomina **anclaje de destino**. El anclaje de destino es la página Web o la sección del documento que se mostrará en el monitor si se hace clic sobre el anclaje de inicio. En nuestro ejemplo, el lugar de activación del enlace es el texto [UPF: La mejor universidad del mundo](http://www.upf.es). El destino, en este caso, es la Web de la UPF: <http://www.upf.edu>

2. Las direcciones de destino, como la dirección www.upf.edu, deben ir precedidas por `http://`, como en `href="http://www.upf.edu"`.

3.2.3. Enlaces a ficheros o documentos situados en un ordenador local

La tercera opción de la que disponemos para realizar enlaces es la que consiste en enlazar a una página Web (o un fichero) que tenemos en un ordenador local. Podemos hacerlo como muestra esta porción de código fuente:

```
..  
<p>Enlace a un documento en el mismo ordenador<br>  
<a href="pag02.htm">Página n. 02</a></p>  
...
```

Observe que, al tratarse de enlazar a un documento local, no hemos necesitado introducir <http://>, sino que, directamente, hemos indicado el nombre y la extensión del documento, en este caso, [pag02.htm](#).

Direcciones absolutas

Uno de los problemas más habituales en la creación de páginas Web es la dificultad de comprensión de la diferencia entre enlaces absolutos y relativos. Un enlace absoluto indica la dirección o camino completo de un archivo. Por ejemplo, supongamos que en la página llamada Index (index.htm) ponemos un enlace a la página llamada Página002 (pag002.htm). El siguiente sería un ejemplo de enlace con dirección absoluta entre ambas páginas:

Página Index (index.htm) con enlace absoluto hacia la Página002 (pag002.htm)

```
..  
<p>Enlace a un documento en el mismo ordenador<br>  
<a href="c:/tests/documentos/pag002.htm">Página n. 002</a></p>  
...
```

La dirección absoluta anterior, con indicación completa de la dirección del documento pag002.htm, incluyendo por tanto la letra de la unidad de disco, [c:](#), y la carpeta o directorio, [/documentos/](#), daría problemas.

Si se publica el documento en Internet, el enlace fallará ya que una vez publicado el documento en Internet la página que se pretende enlazar (pag002.htm) ya no estará donde dice el código fuente (unidad [c:](#), directorio [documentos](#)), sino en otra unidad (por ejemplo, la unidad [h:](#)) del ordenador remoto que actúa de servidor de páginas Web.

En segundo lugar, el directorio, por supuesto, puede tener el mismo nombre, pero en ese caso habremos de asegurarnos de que sea así; de lo contrario, es muy poco probable que, aleatoriamente, coincidan. Cuando planeamos publicar nuestra Web deberemos tener una estructura de nombres y jerarquías de directorios idéntica en nuestro ordenador y en el espacio de disco del ordenador remoto donde se publicará nuestra Web (excepto las letras de las unidades de disco, que no hace falta indicarlas)

Direcciones relativas

¿Cómo debería indicarse dicho enlace? Lo mejor es usar siempre direcciones relativas. ¿Relativas a qué? Al contexto de las páginas enlazadas, para lo cual lo más simple es no indicar el camino completo hasta el archivo sino únicamente el nombre del archivo y la carpeta si no es la misma donde están las dos páginas.

En el ejemplo anterior, suponemos que la página que establece el enlace (index.htm) está situada en c:/tests/ y que la página que se desea enlazar (pag002.htm) está situada en el directorio documentos que forma parte de c:/tests, por tanto, está en c:/tests/documentos. Lo primero que haremos, será retirar la indicación de la unidad de disco y prepararemos un enlace relativo así:

Página Inicio (inicio.htm) con enlace relativo hacia la Página 002 (pag002.htm)

```
..  
<p>Enlace a un documento en el mismo ordenador<br>  
<a href="/tests/documentos/pag002.htm">Página n. 002</a></p>  
...
```

Ahora, el enlace anterior funcionará con todos los navegadores y, si publicamos la página en Internet y nos aseguramos que en el ordenador del servidor tengamos la misma estructura de carpetas (en este caso, /test/documentos/) dentro de la unidad que almacena nuestra página Web, también funcionará desde Internet.

Pero, ¿qué sucede, si deseamos que desde la Página 002 (pag002.htm) haya un enlace a la página Index (index.htm). Ahora, la perspectiva a cambiado (por algo los llaman relativos). El enlace ahora podría ser así:

Página 002 (pag002.htm) con enlace relativo hacia la Página Inicio (inicio.htm)

```
..  
<p>Volver a la página Index<br>  
<a href="../inicio.htm">Index</a></p>  
...
```

Observe que hemos indicado la dirección relativa de inicio.htm con "../inicio.htm". Los dos puntos y la barra (../) indican justamente que la página inicio está en un directorio superior (superior en relación al lugar desde el cual se lanza el enlace ahora).

Naturalmente, también puede suceder que las dos páginas que deseamos enlazar se encuentren en la misma carpeta o directorio. Es el caso más favorable, ya que entonces no necesitamos indicar nada más que el nombre del archivo, de este modo:

Página Index (index.htm) con enlace relativo hacia la Página 002 (pag002.htm) estando ambas en el mismo directorio

```
..  
<p>Enlace a un documento en el mismo ordenador<br>  
<a href="pag002.htm">Página n. 002</a></p>  
...
```

A efectos de las siguientes prácticas para esta asignatura puede utilizar únicamente nombres de archivo, sin referencia a unidades de disco ni carpetas a condición de que ponga todos sus archivos en las misma carpeta.

3.3. Ejercicio: enlace externo

Abra un editor de páginas web. Para esta práctica, recomendamos seguir utilizando el *Bloc de Notas*. La razón es la siguiente, como ya hemos señalado, cuando se está aprendiendo a crear páginas Web, conviene retrasar todo lo posible el uso de un editor en modo gráfico.

Esta es la única forma de garantizar que, cuando tenga que solucionar algún problema en su página Web (siempre generan problemas los editores en modo gráfico) sabrá cómo hacerlo. Además, se produce la paradoja siguiente: solamente quienes saben editar código fuente pueden hacer un buen uso de la opción en modo gráfico, y cuanta más experiencia se tiene en código fuente, mejor se sabe explotar las características del modo gráfico, ya que el modo gráfico dista de ser intuitivo en la mayor parte de las ocasiones.

Abra el documento HTML que creó, de nombre **test01.htm**. Añada ahora, al final del documento, debajo de su lista de intereses, la siguiente codificación:

```

...
<p>Este es el segundo párrafo de mi primera página web e incluye una
palabra en <strong>negrita</strong> y otra en <em>cursiva</em>. En
cambio, esta <strong><em>palabra</em></strong> por alguna razón es
negrita y cursiva a la vez.</p>

<h2>Mis principales intereses</h2>

<ul>
<li>Guiones</li>
<li>Televisión de calidad</li>
<li>Producción de documentales</li>
</ul>

<hr />
<h2>Enlaces</h2>
<p><a href="allmovie.com">AllMovie</a><br />
Una base de datos de cinematografía</p>

<hr />
<p>Página web creada por Abraham Polonsky, Marzo de 2007</p>
</body>
...

```

Guarde el documento y abra el documento con el navegador web y compruebe si funciona. El resultado debería verse así:

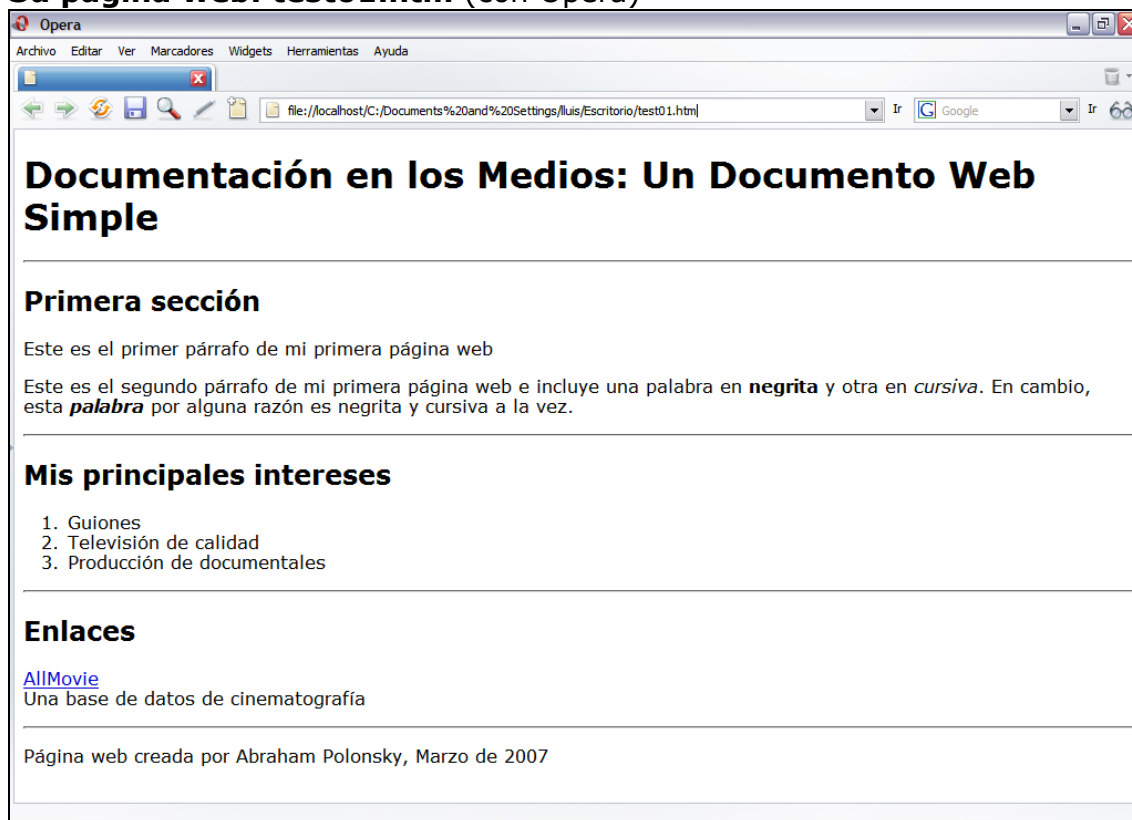


Repasemos: en primer lugar, el texto "Enlaces" debe aparecer en negrita. El texto "AllMovie" debe aparecer de color azul y subrayado, mientras que "Una base de datos de cinematografía" debe aparecer de forma normal (letra redonda). Por cierto, hemos introducido una nueva etiqueta: `
` que provoca un cambio simple de línea.

En segundo, lugar, con el documento abierto con el navegador, al situar el cursor sobre el anclaje de inicio del enlace (AllMovie), debe tomar la forma de una mano. En tercer lugar, haciendo clic sobre el enlace, debe abrirse la página de inicio de AllMovie.

Su documento **test01.htm**, debería tener ahora esta forma (los datos de contexto deberán ser los que correspondan en cuanto a nombre, fecha, e intereses y no necesariamente los que aparecen aquí):

Su página web: test01.htm (con Opera)



Si la codificación no ha tenido éxito, deberá repasar el código y subsanar los errores. En ese caso, conmute entre navegador y editor y vaya comparado los resultados. No olvide que debe guardar cada cambio y que debe forzar al navegador a cargar de nuevo la página cada vez que haga un cambio en ella.

Para completar este ejercicio, añada dos enlaces más a dos páginas web relacionadas con los intereses declarados en su página. Siga el mismo procedimiento de presentación del enlace y de descripción breve del mismo. En total, su página test01.htm deberá tener tres enlaces externos.

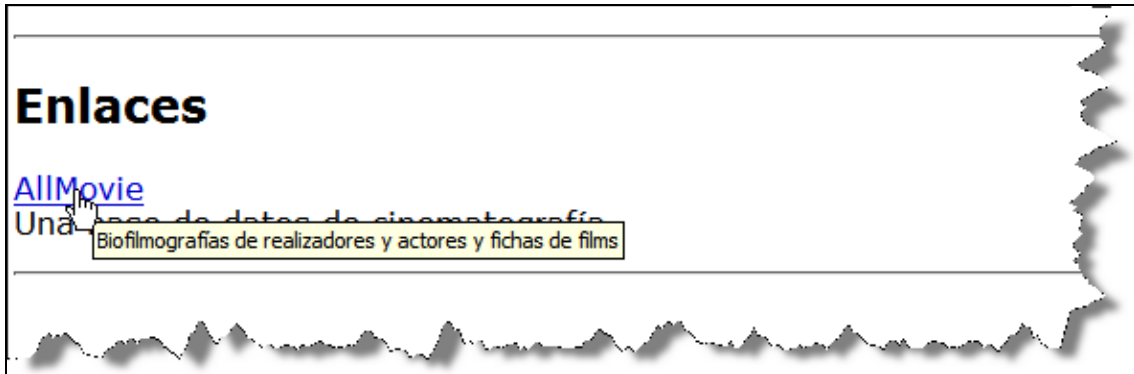
3.3. Títulos de los enlaces

La etiqueta del enlace puede tener un atributo **title**. El efecto de este atributo es desplegar un texto explicativo cuando el cursor se detiene sobre el enlace. A veces, el resultado de hacer clic es muy previsible, pero otras veces se hace necesario anticipar al visitante de la página web con qué se encontrará al hacer clic en el enlace, y esta es la función de **title**.

El código fuente para añadir título a un enlace es el siguiente:

```
...  
<hr />  
<h2>Enlaces</h2>  
<p><a href="allmovie.com" title="Biofilmografías de realizadores y  
actores y fichas de films">AllMovie</a><br />  
Una base de datos de cinematografía</p>  
...
```

El resultado, con el cursor situado en el enlace es el siguiente:



Añada un título, como indica el ejemplo anterior, a cada uno de los tres enlaces de su página Web, con la información sobre cada enlace que crea más conveniente.

Segunda Parte: XHTML y CSS

1. Qué es XHTML

El *eXtended Hypertext Markup Language* (XHTML) es la última versión del estándar internacional para crear páginas web. Sustituye al antiguo lenguaje HTML con el que aún están hechas la mayor parte de páginas de la Web. Como todavía hay muchos documentos codificados en HTML e incluso algunos editores de páginas web lo siguen utilizando, reseñamos a continuación las siete principales diferencias de XHTML en relación a HTML:

1. En XHTML, las etiquetas, los atributos, los valores, etc., de cada elemento, excepto la declaración DOCTYPE, se escriben siempre en minúsculas (en HTML era indiferente).
2. Los documentos XHTML deben tener todos los elementos estructurales: `<html>`, `<head>` y `<body>` (en HTML los navegadores interpretan los documentos aunque carezcan de alguno de los elementos anteriores)
3. El elemento `<title>` debe aparecer siempre en `<head>`.
4. Todos los elementos deben tener una etiqueta de cierre (en HTML el cierre de la etiqueta de párrafo (`</p>`), por ejemplo, era optativo). De este modo, en XHTML los elementos vacíos, como `<hr>` se deben cerrar como en `<hr />`.
5. Los valores de los atributos deben ir siempre entre comillas, aunque sean valores simples.
6. Las etiquetas se deben anidar correctamente. Por ejemplo, `negritas y cursivas` es correcto, pero `negritas y cursivas` no lo es porque están cruzadas entre ellas: en este caso, si la etiqueta `` es la primera, su etiqueta de cierre `` debe ser la última.
7. Todo documento XHTML debe tener al inicio de todo una serie de declaraciones como DOCTYPE (en HTML era optativa).

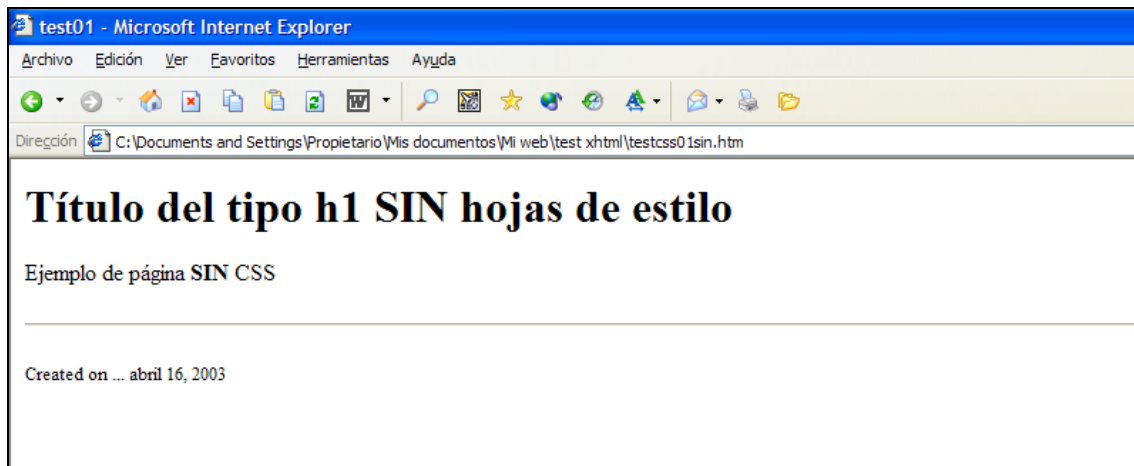
La forma actual de crear páginas web consiste en utilizar XHTML en alguna de sus variantes (nosotros recomendamos la que hemos usado aquí, denominada **Transitional**) más un sistema para dar formato a las páginas denominado **CSS**.

2. Qué es CSS

Cascading Style Sheets (CSS) es una forma muy potente de controlar el aspecto de un documento XHTML (o HTML) con mínimo esfuerzo. En lugar de especificar, para cada elemento cómo deseamos que sea su apariencia mediante atributos antiguos (y actualmente desaconsejados) de HTML del tipo `font` o `color`, etc., podemos crear una única lista centralizada de especificaciones sobre cómo deseamos que aparezcan esos elementos en cualquier lugar de nuestra(s) página(s) web. De este modo si, por motivos de actualización o mantenimiento, deseamos modificar el aspecto de un mismo elemento en todas las páginas, basta con realizar un único cambio en la lista centralizada.

Además, CSS permite controlar a mayor nivel cómo debe interpretar el navegador cada elemento. Por ejemplo, podemos cambiar la forma en la cual se interpretan los elementos <h1>, <h2>, etc., o hacer que cambie la apariencia de los enlaces, etc.

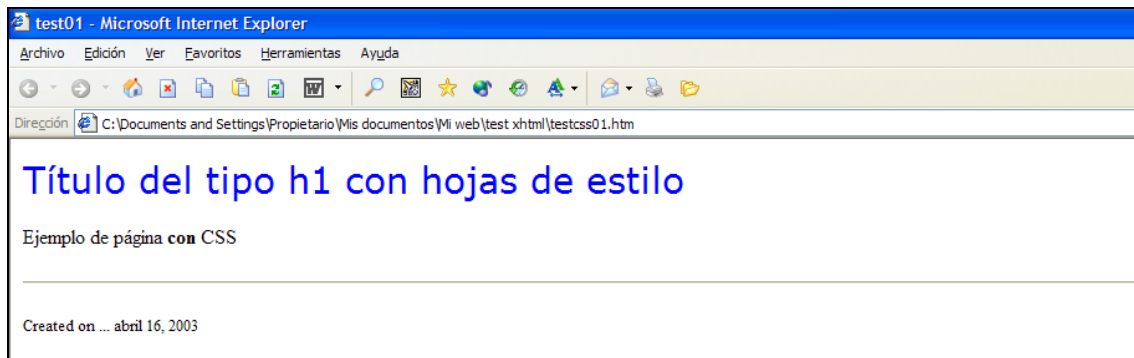
El método es muy sencillo: se puede añadir, dentro del elemento <head> de un documento XHTML un par de etiquetas de estilo (<style>...</style>), dentro de las cuales podemos especificar el aspecto de elementos de la página. Por ejemplo, sin hojas de estilo, la interpretación del navegador de un encabezado tipo <h1> es la siguiente:



Ahora, veremos cómo es un ejemplo de codificación con CSS. Primero, presentamos cómo podemos modificar el aspecto de las etiquetas con CSS <h1>:

```
<head>
<style type="text/css">
h1 {color: blue; font-family: Verdana, Arial; font-weight: normal;
font-size: 22pt}
</style>
</head>
```

La declaración de estilo anterior provocará que todos los encabezados tipo <h1> de una página web tengan las mismas características. El resultado será éste:



Si la página o el sitio web tiene 10 de tales encabezados (o mil) de tipo `<h1>` todos adoptarán de manera automática la misma apariencia sin necesidad de añadir 10 veces (o mil veces) las mismas especificaciones a cada encabezado por separado.

3. Prácticas con CSS

3.1. Un documento XHTML básico

Para los siguientes ejercicios, utilice un editor de páginas web de tipo código fuente (no WYSIWYG), como *Bloc de Notas*.

Entre el siguiente código fuente. Puede copiar y pegar las tres primeras líneas en su documento, para no tener que teclearlas, puede copiar también el texto, pero **no** debería teclear las etiquetas que lo rodean si desea entrenarse en estos temas; es importante que las entre usted, de lo contrario, perjudicará su aprendizaje (naturalmente, puede adaptar los datos de autoría y fecha a su contexto real):

Primer documento XHTML con CSS

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

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<title>El montaje en el cine</title>
</head>
<body>
<h1>El montaje en el cine</h1>
<hr />
<p>Un ejemplo de página xhtml con una cita de Pudovkin:</p>
<blockquote>Repetiré una vez más que el montaje es la fuerza creadora de la realidad
fílmica, y que la naturaleza sólo aporta la materia con que formarla. Esa, precisamente, es
la relación entre montaje y cine.</blockquote>
<p><hr />
(V. I. Pudovkin, 1929)<br />Editor de esta página: A. Polonsky<br />
Marzo 2007</p>
</body>
</html>
```

En el navegador, una vez guardado el documento (puede darle el nombre **test02.htm**) tendrá este aspecto:

El montaje en el cine - Opera

Archivo Editar Ver Marcadores Widgets Herramientas Ayuda

El montaje en el cine

file:///localhost/C:/Documents%20and%20Settings/luis/Escritorio/test02.htm

Google

El montaje en el cine

Un ejemplo de página xhtml con una cita de Pudovkin:

Repetiré una vez más que el montaje es la fuerza creadora de la realidad fílmica, y que la naturaleza sólo aporta la materia con que formarla. Esa, precisamente, es la relación entre montaje y cine.

(V. I. Pudovkin, 1929)
Editor de esta página: A. Polonsky
Marzo 2007

3.2. Modificaciones con CSS

Vuelva a abrir el documento **test02.htm** en el Bloc de Notas. Ahora, añada la siguiente codificación dentro de <head> (y después de <title>). Al ser la primera vez que incluye una hoja de estilo en un documento, es aconsejable no limitarse a copiar y pegar este código y entrarlo a mano (cuando tenga más conocimientos de CSS ya podrá limitarse a copiar y pegar, pero ahora como hacer esto perjudicaría su aprendizaje, es mejor entrar el código a mano):

Ejemplo de código CSS

```
<style type="text/css">
body{font-family: Verdana, Arial; font-size: 12pt;}
h1 {color: blue; font-weight: normal; font-size: 22pt;}
hr { border-color: blue; }
blockquote {font-style: italic;}
</style>
```

El resultado completo del nuevo código fuente del documento **test02.htm** debe ser ahora el siguiente (hemos resaltado el nuevo código añadido):

Ejemplo de XHTML con código CSS

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

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<title>El montaje en el cine</title>

<style type="text/css">
body{font-family: Verdana, Arial; font-size: 12pt;}
h1 {color: blue; font-weight: normal; font-size: 22pt;}
hr { border-color: blue; }
blockquote {font-style: italic;}
</style>

</head>
<body>
<h1>El montaje en el cine</h1>
<hr />
<p>Un ejemplo de página xhtml con una cita de Pudovkin:</p>
<blockquote>Repetiré una vez más que el montaje es la fuerza creadora
de la realidad fílmica, y que la naturaleza sólo aporta la materia con que
formarla. Esa, precisamente, es la relación entre montaje y
cine.</blockquote>
<p><hr />
(V. I. Pudovkin, 1929)<br />Editor de esta página: A. Polonsky<br />
Marzo 2007</p>
</body>
</html></body>
</html>
```

Guarde el documento, con el mismo nombre. Abra el documento con el navegador, o haga clic en el botón **Actualizar** del navegador si ya lo tenía cargado. El resultado debe ser ahora este:



Observe los cambios: toda la fuente del documento a cambiado a Verdana, el título tipo <h1> ahora ha cambiado a color azul y ya no está en negritas y la cita ha pasado a cursiva. Las dos barras horizontales se muestran ahora de color azul. Si su documento tuviera cientos de encabezados, la transformación hubiera sido igual de rápida y eficiente con la edición de una pocas líneas de código fuente.

3.3. Propiedades CSS

Para los ejercicios de esta Unidad, podrá utilizar los siguientes elementos de estilo (el alumno podrá añadir otros, si lo desea, más adelante se indicarán algunas fuentes adicionales de información sobre el tema):

Elementos de hojas de estilo

Selector	Codificación	Explicación
body	body {font-family: Verdana, Arial; font-size: 12pt; margin-right: 15px; margin-left: 15px}	Para proporcionar una fuente homogénea y muy legible a todo el documento. Por si el ordenador del navegador cliente no tiene Verdana, hemos previsto que, en segunda opción use Arial que está siempre presente en todos los ordenadores. Además, nos aseguramos que queden márgenes a ambos lados del documento, para que sea aún más legible
h1 h2 h3	h1, h2, h3 {color: blue}	Podemos indicar de esta forma una propiedad que queremos que afecte a diversos elementos. De este modo, indicamos que queremos que todos los encabezados sean de color azul
h1	h1 { font-weight: normal; font-size: 22pt}	Damos un tamaño de 22 puntos a h1 y eliminamos la negrita
h2	h2 { font-weight: normal; font-size: 18pt}	Idem, con 18 pt para este nivel
h3	h3 { font-weight: normal; font-size: 14pt}	Idem, 14 pt
a	a:hover {color: red; font-weight: bold; text-decoration: none}	Modifica el aspecto de un enlace cuando el usuario desplaza el cursor sobre el mismo. Proporciona algo más de interactividad a los enlaces
hr	hr {border-color: blue}	Causa que las líneas horizontales sean de color azul y que tengan una altura de un píxel

Observe que los distintos valores de cada selector se separan mediante ;
No es necesario poner ; al final del último selector.

Ejemplo de codificación CSS en un documento XHTML

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">

<head>

<style type="text/css">
body {font-family: Verdana, Arial; font-size: 12pt; margin-right: 15px;
margin-left: 15px}
h1, h2, h3 {color: blue}
h1 { font-weight: normal; font-size: 22pt}
h2 { font-weight: normal; font-size: 18pt}
h3 { font-weight: normal; font-size: 14pt}
a:hover {color: red; font-weight: normal; text-decoration: none}
hr {border-color: blue}
</style>

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

3.4. Sintaxis CSS

Como habrá observado por los ejemplos de codificación anteriores, la sintaxis de CSS es sumamente sencilla, pero muy sensible. Debe respetarse escrupulosamente, o no funcionará. En general es así: se indica el nombre de elemento cuyo aspecto deseamos especificar, por ejemplo, **body**. Este elemento, en CSS se denomina selector. Después del selector hay un espacio, y entre corchetes (**{...}**) se indica la propiedad, por ejemplo, **font-family**, después, dos puntos (:), y el valor de la propiedad, por ejemplo, **verdana**. Si hay más de una propiedad, se separan mediante punto y coma (;) y se cierra la declaración con corchetes como ya hemos visto (**{...}**). Finalmente, todas las declaraciones de estilo se encierran entre las etiquetas **<style>...</style>**. Es aconsejable indicar la propiedad **type="text/css"** en la etiqueta de apertura, de modo que la pareja de etiquetas recomendable es ésta:

```
<style type="text/css">...</style>.
```

3.5. Otras posibilidades y fuentes de información

Con CSS se puede controlar otros aspectos de una página web que aquí no hemos visto. Además, las hojas de estilo se pueden guardar en documentos separados de la página web, de manera que una misma hoja de estilo puede afectar a varios documentos escribiendo la hoja de estilo una vez.

Para ello, el documento que contiene la hoja de estilo debe consistir en un archivo de texto simple con la extensión **css**, por ejemplo **estilo1.css**. Este archivo puede contener la misma codificación del ejemplo anterior:

Ejemplo de archivo externo CSS

```
body {font-family: Verdana, Arial; font-size: 12pt; margin-right: 15px;
margin-left: 15px}
h1, h2, h3 {color: blue}
h1 { font-weight: normal; font-size: 22pt}
h2 { font-weight: normal; font-size: 18pt}
h3 { font-weight: normal; font-size: 14pt}
a:hover {color: red; font-weight: normal; text-decoration: none}
hr {color: blue; height: 1px}
```

Como puede verse, en este caso no se debe encerrar entre las etiquetas `<style>` las especificaciones de estilo que se guardan en un archivo externo ni hace falta añadir ningún tipo de declaración inicial, simplemente guardar el archivo en formato texto y con extensión **css**.

Este archivo se vincula con el documento mediante un enlace en el código fuente del documento original mediante la etiqueta `link` y las propiedades `type`, `rel`, `href`. Ejemplo:

Ejemplo de vinculación con link

```
...
<head>
...
<link rel="stylesheet" type="text/css" href="estilo1.css" />
...
</head>
...
```






Naturalmente, para que el ejemplo anterior funcione bien, ambos archivos, la página web y el archivo **css**, deben estar guardados en el mismo directorio o carpeta.

Para este taller, nos limitaremos a los aspectos discutidos hasta aquí sobre hojas de estilo, sin considerar otras posibilidades. No es imprescindible usar hojas de estilo externas. Hemos mostrado aquí esta posibilidad para información del alumno que desee utilizarla. Recordamos que la fuente oficial de información sobre CSS es: **www.w3.org**.

3.5.1 Editores CSS

Las últimas versiones de los editores de páginas web suelen incluir utilidades y ayudas para definir hojas de estilo, bien sea del tipo interno o externo. Tales programas convierten la tarea de definir hojas de estilo complejas en una cuestión de "ir de compras", mediante elecciones sucesivas de opciones. Por ejemplo, editores comerciales como *Dreamweaver* o *NVU* incluyen programas que facilitan la creación y mantenimiento de hojas de estilo, pero existen también programas freeware que facilitan las mismas funciones.

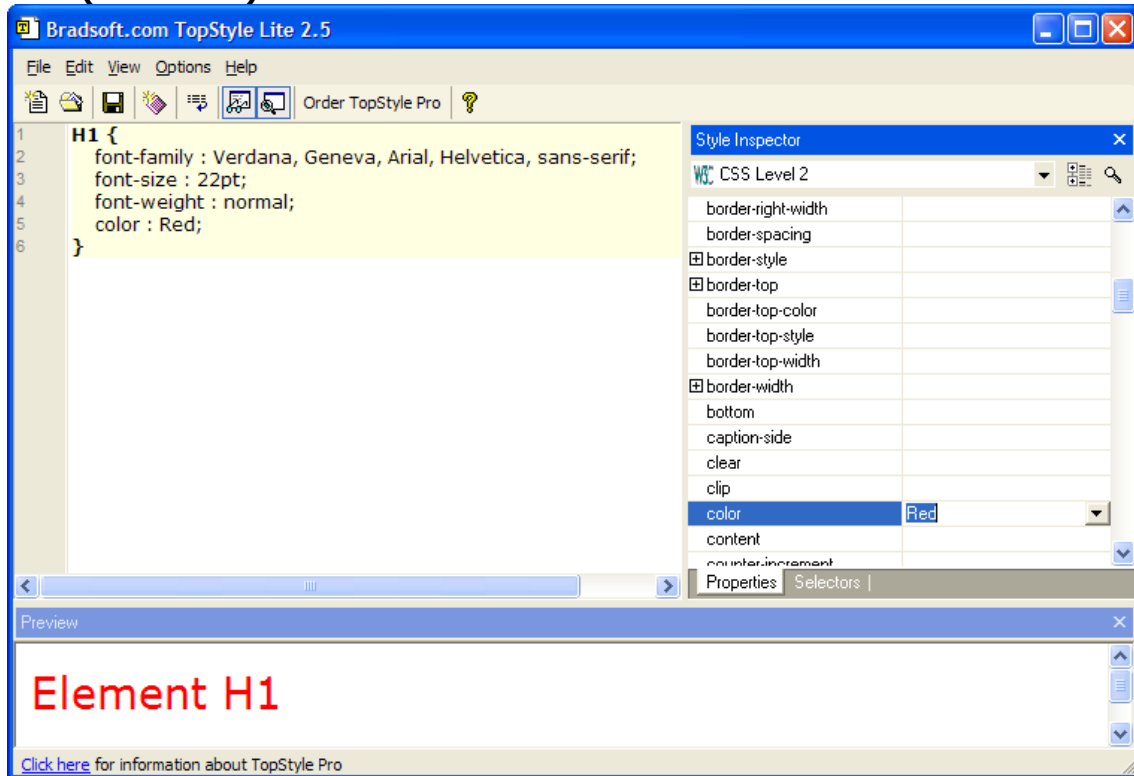
El alumno puede encontrar programas que le ayuden a definir sus hojas de estilo, si lo desea, en Snapfiles (www.snapfiles.com) y una vez en ese sitio, en Freeware > Web Authoring > Style Sheet Editors. En esta sección, se recomienda considerar Balthisar Cascade:

Balthisar Cascade		
 Download	Our Rating: ★★★★★ Popularity:  Updated: 12/20/2002	
v 2.0 beta 12 Freeware	Balthisar Cascade is intended for the creation of specification 2 Cascading Style Sheets (CSS2) for inclusion into HTML documents on networks such as the internet. Balthisar Cascade includes most of the features an amateur or professional HTML developer needs to create publishable hypertext documents. It follows the CSS2 specification as set forth by the World Wide Web Consortium. Cascade currently supports CSS2 strictly, i.e., it does not support proprietary extensions to CSS as used in Microsoft's Internet Explorer, and it does not support JavaScript style sheets at all. Some of Balthisar Cascade's features include: Creation of Classes, Creation of Subclasses, Creation of Pseudo-Classes and Pseudo-Elements, The ability to export data as an HTML file or a CSS2 file... and much more	
 Add to CD View CD	1415 kb  Windows (All)	Screenshot Developer  Full Page

También es sumamente recomendable (quizás, aún mejor que el anterior) el programa **TopStyle Lite** (<http://www.bradsoft.com/topstyle/download/litedload.asp>).

En ambos casos, el programa genera un código que podemos copiar y pegar en nuestras páginas web o exportarlo a un archivo de texto para disponer de un archivo externo de hojas de estilo (archivo extensión **css**):

Ejemplo de especificación de estilo del elemento h1 con TopStyle Lite (freeware)



3.6. Metadatos

Un aspecto fundamental en la arquitectura de la información es el uso de metadatos. Los metadatos son datos sobre datos, es decir, información de la página web sobre la propia página web que debe añadir el autor o editor de la página.

Hay diversos tipos de metadatos. A efectos de este taller consideraremos los siguientes:

Tipos de metadatos y su codificación típica

Metadato	Codificación	Explicación
Título de la página	<title>Título del documento</title>	Es el dato desplegado en la barra superior del programa navegador. Es usado también por los motores de búsqueda para presentar el título de la página en los resultados de una búsqueda
Descripción de la página	<meta name="description" content="descripción del sitio o de la página" />	Es usado por los motores de búsqueda indizar el sitio y para presentar la descripción del sitio en los

		resultados de una búsqueda
Palabras clave	<meta name="keywords" content="palabra1, palabra2,..." />	Sirve para representar el contenido de la página, para indizar la página web por motores internos y por algunos motores de búsqueda
Autor	<meta name="autor" content="nombre" />	Para indicar el nombre del autor. Usado por motores de búsqueda
Título del enlace	... 	Provoca la aparición de un recuadro explicativo cuando el usuario desplaza el cursor sobre el enlace. Usado también por los motores de búsqueda para indizar la página

Los metadatos siempre se añaden en la zona <head>, excepto, el atributo <title>, que se añade en los enlaces. Ejemplo de codificación de metadatos (obviamos el ejemplo del título de un enlace, visto en el taller anterior):

Ejemplo de metadatos en una página XHTML

```

<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>El montaje en el cine</title>
<meta name="description" content="Una cita de V.I. Pudovkin sobre el montaje
cinematográfico" />
<meta name="keywords" content="cine, montaje, Pudovkin, montaje cinematográfico,
historia del cine" />
<meta name="author" content="A. Polonsky" />
</head>
<body>
<h1>El montaje en el cine</h1>
<hr />
<p>Un ejemplo de página xhtml con una cita de Pudovkin:</p>
<blockquote>Repetiré una vez más que el montaje es la fuerza creadora de la realidad
fílmica, y que la naturaleza sólo aporta la materia con que formarla. Esa, precisamente, es
la relación entre montaje y cine.</blockquote>
<p><hr />
(V. I. Pudovkin, 1929)<br />Editor de esta página: A. Polonsky<br />
Abril de 2003</p>
</body>
</html>

```

El resultado de la interpretación de los metadatos no proporciona ningún resultado visible en el navegador de cara a los usuarios, salvo en el caso del atributo **title** de los enlaces, pero tiene efecto sobre el tratamiento que dan

a la página los motores de búsqueda; además, facilitan el mantenimiento y edición de las páginas. Cuando se publica un sitio web es muy importante que cada página tenga sus propios metadatos. En el caso de XHTML, el metadato **title** es, de hecho, imprescindible.

4. Ejercicio de arquitectura de la información

Simularemos la creación de una sede Web sobre guiones de cine y la diseñaremos usando XHTML y CSS.

4.1. Navegación y metadatos

El sitio web que haremos debe tener algunas características muy importantes desde el punto de vista de la arquitectura de la información:

1. *Navegación constante*. Significa que, en todas las secciones o páginas del sitio web debe haber un mismo menú (navegación constante) que permita, al menos, acceder desde cualquier lugar de la web a la página principal, a las secciones principales y a la página de créditos. Esto, naturalmente, sin perjuicio que algunas páginas puedan tener, además, un menú local si fuera necesario (el menú local no es necesario para esta práctica).

2. *Identificación*. Todas las páginas o secciones del sitio deben tener: título, fecha de creación o actualización, datos de autoría o acceso a datos de autoría.

3. *Metadatos*. Todas las páginas, y especialmente, la página principal (index.htm) deben tener los siguientes metadatos en la sección <head> (<title> debe ser siempre el primero):

- title
- author
- description
- keywords

Además, todos los enlaces, internos o externos deben tener el atributo **title** (ver ejercicio de la Unidad 2).

4. *Hoja de estilo*. Todos los aspectos de apariencia de la página y/o de comportamiento de los enlaces debe estar controlado por una hoja de estilo. Puede estar vinculada desde un archivo externo o incrustada en la sección <head>, a preferencia del alumno, así como puede editar la hoja de estilo de manera artesanal ("a mano") o con ayuda de un programa especializado como **TopStyle Lite** (ver el apartado 4.5.1.).

4.2. Preparación de los materiales

En segundo lugar, necesitamos preparar los materiales de base de nuestra página Web. Tomaremos extractos de la obra de Tobias¹ sobre guiones y tramas para simular los contenidos de la Web

4.1. Preparación de materiales

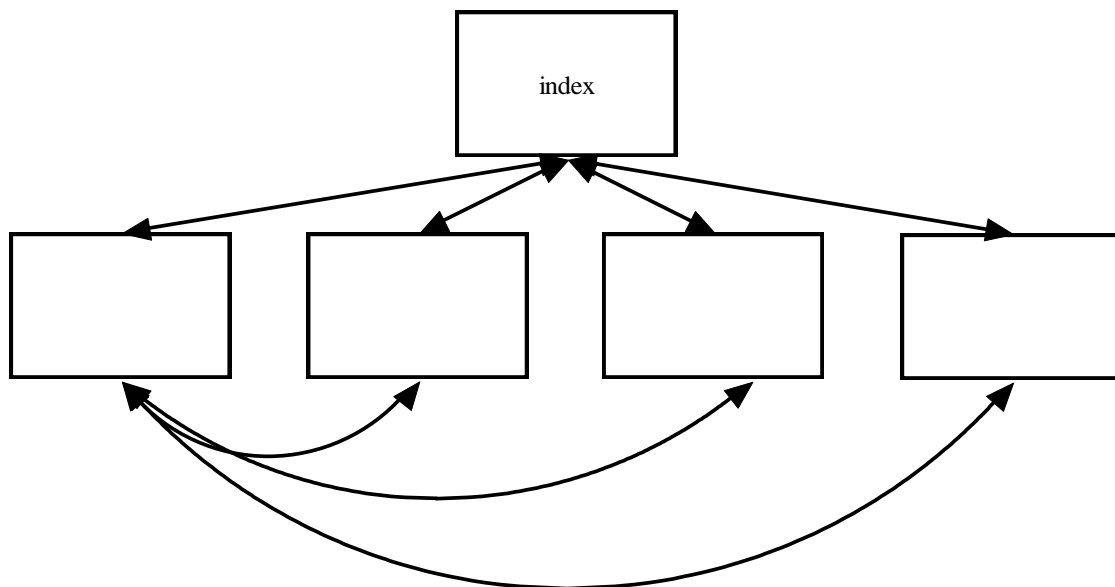
Para esta práctica crearemos un hipertexto con XHTML y CSS que constará en total de cinco secciones. Por tanto, necesitaremos cinco documentos: tres documentos con contenidos (pag01 a pag03), una página principal (index) y un documento para los créditos (pag04):

1. index.htm

¹ Ronald B. Tobias. *El guión y la trama*. Madrid: Ediciones Internacionales Universitarias, 1999 (título original: *20 Master plots and how to build them*, 1993)

2. pag01.htm
3. pag02.htm
4. pag03.htm
5. pag04.htm

La estructura de este hipertexto corresponderá, por tanto, a una estructura jerárquica de un solo nivel y con enlaces en red:



Estructura del hipertexto del ejercicio

El documento index.htm es el último que haremos. Comenzaremos por los documentos de contenido y página de créditos. Lo más práctica será crear un directorio nuevo (puede llamarlo guiones) donde pondremos todos los materiales. No haga subcarpetas. La recomendación es tenerlo todo en un solo directorio, de modo que los enlaces puedan consistir únicamente en nombre de documentos (sin necesidad de indicar nombres de directorio ni estructuras jerárquicas).

Contenido de los documentos

Con un editor de páginas web, en modo código procederá a crear tres documentos HTML con el contenido que se indica a continuación² (puede copiar el texto a su editor de páginas web, de modo que solamente tenga que añadir la codificación del código fuente, pero no estos textos):

² Los textos de los tres documentos son transcripciones de la obra citada de Tobias.

pag01.htm

[inicio](#) | [trama1](#) | [trama2](#) | [trama3](#) | [créditos](#)

El gui3n y la trama

Trama 1: Búsqueda

La **trama de búsqueda**, como su nombre indica, es la búsqueda que efectúa el protagonista de una persona, un lugar o una cosa, sea tangible o intangible. Puede ser el *Santo Grial*, el *Valhalla*, la inmortalidad, la *Atlántida* o el *Reino Medio*. El personaje principal busca específicamente (y no casualmente) algo que espera o desea cambiará su vida de una manera significativa.

El objeto de la búsqueda del protagonista se refleja poderosamente en su carácter y, habitualmente, lo transforma de algún modo, afectando así el cambio del personaje, lo cual es importante al término del relato. En todos los casos, el protagonista es muy diferente al final de la historia de como era en un principio.

Observaciones: la página tiene un menú de navegación en la parte superior, que permitirá acceder a la página principal, a las otras secciones y a la página de créditos. Observe que, el elemento del menú correspondiente a la página que se está viendo, está tratado de otra forma (no contiene un enlace, ya que sería un enlace a sí mismo y tiene otro color).

Los contenidos de las otras páginas son los siguientes (obviamos a efectos de estas ilustraciones la navegación y la autoría/fecha):

pag02.htm

[inicio](#) | [trama1](#) | [trama2](#) | [trama3](#) | [créditos](#)

El gui3n y la trama

Trama 2: Aventura

La **trama de la aventura** se parece en varios aspectos a la de búsqueda, pero existen profundas diferencias entre ambas. La de búsqueda es una trama de personajes. Es una trama mental. La de aventura, por otro lado, es una trama de acción, es una trama de lo físico.

La diferencia radica principalmente en el enfoque. En la trama de búsqueda, desde el principio hasta el final, el énfasis recae en la persona que efectúa la búsqueda; en la de aventura, el enfoque recae en el viaje mismo.

pag03.htm (obviamos la navegación constante, pero la página del alumno debe contenerla)

Trama 3: Persecución

Hay dos juegos que nunca parecen cansar la imaginación de los niños: el juego del escondite y el *corre, corre que te pillo*. Intenta recordar la sensación de excitación al estar a la caza y hallar dónde se escondían los demás.

La **trama de persecución** es la versión literaria del juego del escondite. La premisa básica de la trama es simple: una persona caza a otra. Sólo se necesitan un par de personajes, el perseguido y el perseguidor. Y ya que es una trama física, la caza es más importante que aquellos que toman parte en ella.

El cuarto documento será un poco diferente y deberá contener sus datos, por ejemplo:

pag04.htm

Créditos

Contenido: página realizada con extractos de la obra de Ronald B. Tobias: *20 Master plots and how to build them* (traducción al castellano; El guión y la trama, 1999).

Editor: página realizada por [Abraham Polonsky](#), [Estudios de Comunicación Audiovisual](#), Barcelona ([UPF](#)). Marzo 2007

Observaciones: el nombre debe contener un enlace al correo electrónico del autor (enlace de tipo **mailto**). Estudios de Comunicación Audiovisual, debe contener un enlace a la página web de los Estudios e igual con la palabra UPF.

Los enlaces de tipo **mailto** generan la apertura automática de un correo electrónico y copian la dirección del destinatario. La forma de un enlace **mailto** es la siguiente:

```
...  
<a href="mailto:abraham.polonsky@hotmail.com">Abraham  
Polonsky</a>
```

...

Lo que en pantalla se verá así:

```
...  
Abraham Polonsky
```

...

4.3. Página inicial

La página inicial puede tener, más o menos, este contenido (la apariencia no hace falta que sea idéntica):

index.htm

[trama1](#) | [trama2](#) | [trama3](#) | [créditos](#)

El Guión y la Trama

Fundamentos de la escritura dramática audiovisual

Principales tramas de la escritura dramática audiovisual:

1. [Búsqueda](#)
2. [Aventura](#)
3. [Persecución](#)

Página editada por [A. Polonsky](#), con extractos de la obra de R. B. Tobias
Marzo 2007

Selectores de clase

La ilustración de la página principal de esta práctica nos sirve para ilustrar un tipo de selector que no hemos comentado: los selectores de clase. Los selectores de clase los crea libremente el autor de la página, es decir, no es necesario que coincidan con el nombre de un elemento XHTML. En este caso, se supone que hemos utilizado un selector de clase para lograr centrar algunas partes de las páginas de este ejercicio. Un selector de clase se identifica con un punto (.) seguido de un nombre, por ejemplo: [.centrado](#).

Es lo que podemos hacer si deseamos centrar el título y la navegación constante en la página de ejemplo. Después, podemos asignar la clase "centrado" como un atributo mediante el código `class="centrado"` al elemento que queremos centrar. Veamos cómo sería el código correspondiente:

```

...
<style type="text/css">
body{font-family: Verdana, Arial; font-size: 12pt;}
h1 {color: blue; font-weight: normal; font-size: 22pt;}
hr { border-color: blue; }
blockquote {font-style: italic;}
.centrado { text-align: center; }
</style>

</head>
<body>

<p class="centrado">enlace1 | enlace2 | enlace3 | </p>

<h1 class="centrado">El Guión y la Trama <br />
Fundamentos de la escritura dramática audiovisual </h1>
...

```

Si nos fijamos en el código anterior, primero hemos declarado la clase `.centrado` en la hoja de estilo y después hemos asignado esa clase a los dos elementos que queremos centrar. Las hojas de estilo CSS con este sistema y otros que aquí no hemos considerado, proporcionan una total libertad al creador de la página.

4.4. Objetivos de esta práctica

Utilizando las etiquetas de las prácticas anteriores, más la del código fuente del ejemplo precedente, deberá componer una página Web con los cinco documentos anteriores que cumpla con los siguientes objetivos:

1. Cada una de las tramas debe ser una página independiente, así como debe haber una página principal
2. La página principal contiene un enlace a todas las demás páginas
3. La página principal y todas las páginas contienen metadatos
4. Todas las páginas contienen, título, fecha y autor
5. Todas las páginas tienen navegación constante
6. Todos los enlaces son relativos (solo nombre archivo) y todas las páginas están en la misma carpeta
6. Todos los enlaces tienen atributo título
8. Hay una página de créditos
9. Todos los elementos de presentación están controlados por una hoja de estilo (interna o externa)
10. El nombre del autor debe contener un enlace a su correo electrónico (enlace tipo **mailto**) y las palabras de los Estudios y de la UPF deben tener a las webs correspondientes.

Una vez editado el hipertexto con XHTML y CSS sobre guiones, y conseguidos los diez objetivos anteriores, se ha finalizado esta Unidad Didáctica

(Unidad Didáctica 1. L. Codina, C. Rovira, marzo 2007)

Tercera Parte: Tutorial de NVU

0. Introducción

NVU (se pronuncia "nview") es un editor de páginas web de tipo gráfico (WYSIWYG, o sea: *What You See Is What You Get*).

Lo que hace especialmente interesante a *NVU* es que se trata de un proyecto de tipo *Open Source*. Los programas *Open Source* se caracterizan por dos cosas:

- *Suelen ser gratuitos* o de precio muy asequible. En este caso, NVU es un programa totalmente gratuito.
- *Su código fuente es de dominio público*, lo cual significa no solamente que su conocimiento está disponible para cualquier programador que desee acceder al funcionamiento interno del programa, sino que todo aquel que lo desee puede hacer aportaciones para mejorar el programa o para dotarlo de nuevas características. Por tanto, los programas de tipo *Open Source*, como NVU, se benefician del hecho de que cientos de programadores de todo el mundo aportan su talento para que el programa en cuestión sea cada vez mejor.
- *Respeto por los estándares*. A diferencia de muchos programas comerciales suelen ajustarse bien y respetar de manera muy cuidadosa los estándares abiertos. En este caso, nos estamos refiriendo a la norma XHTML combinada con el uso de hojas de estilo CSS

La versión original (en inglés) de NVU está disponible en esta dirección:

<http://www.nvu.com/index.php>

Versión adaptada al castellano:

<http://www.proyectonave.es/productos/descargas.php>

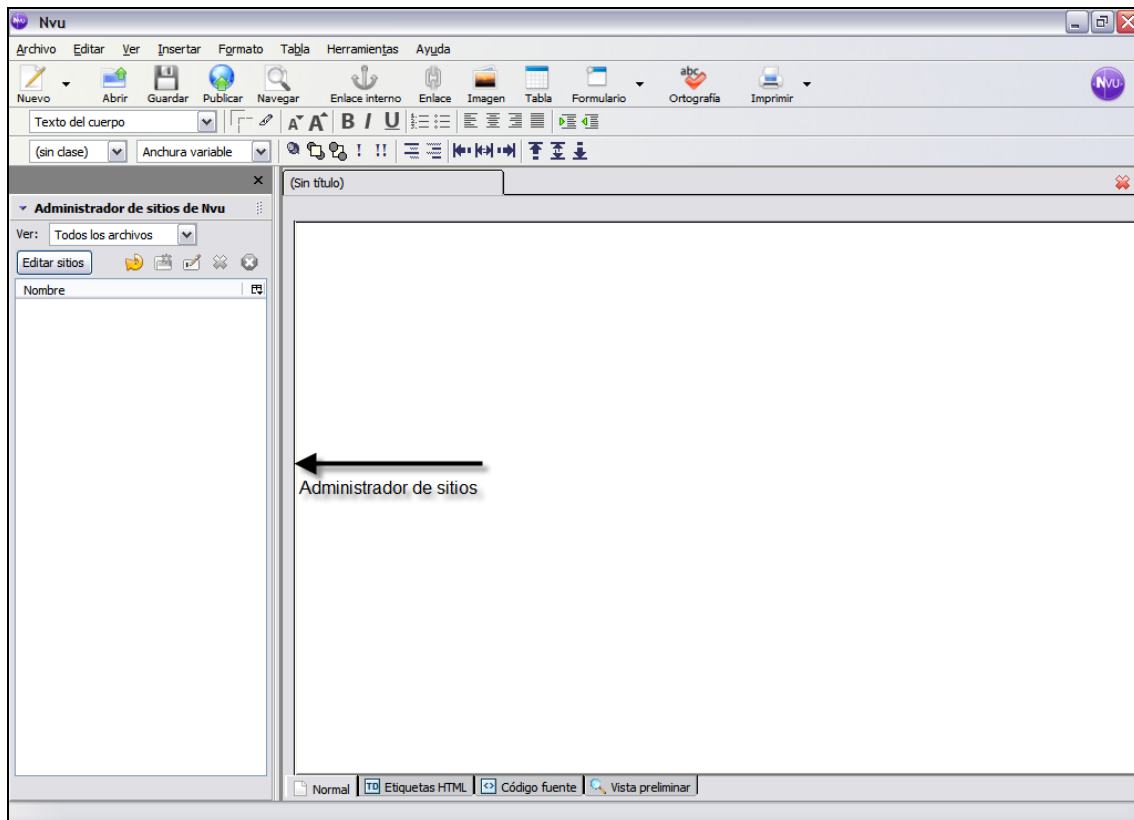
Versión adaptada al catalán:

<http://www.softcatala.org/wiki/Nvu>

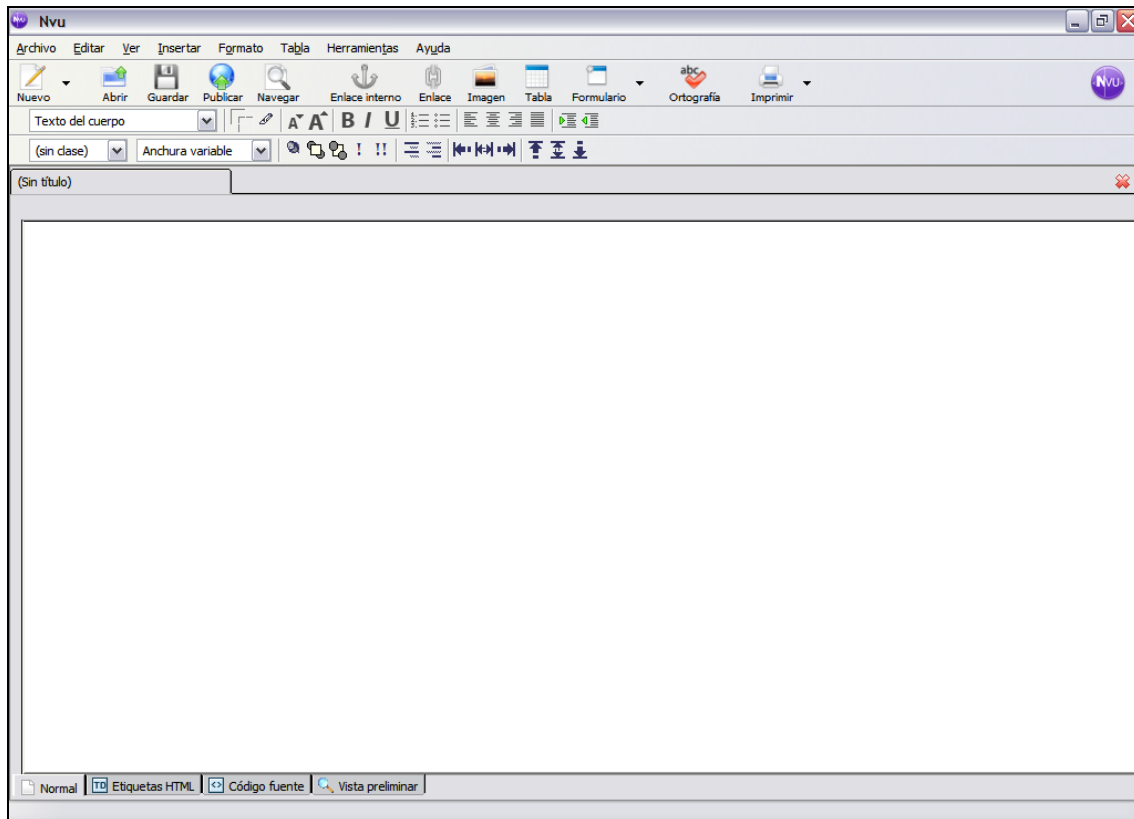
La documentación de esta unidad utiliza la versión en castellano, pero el alumno puede descargar y utilizar la versión en la lengua que le sea más cómoda.

1. Uso del programa

Una vez descargado e instalado el programa hacemos clic sobre el icono del mismo para usarlo. Aparecerá esta página:

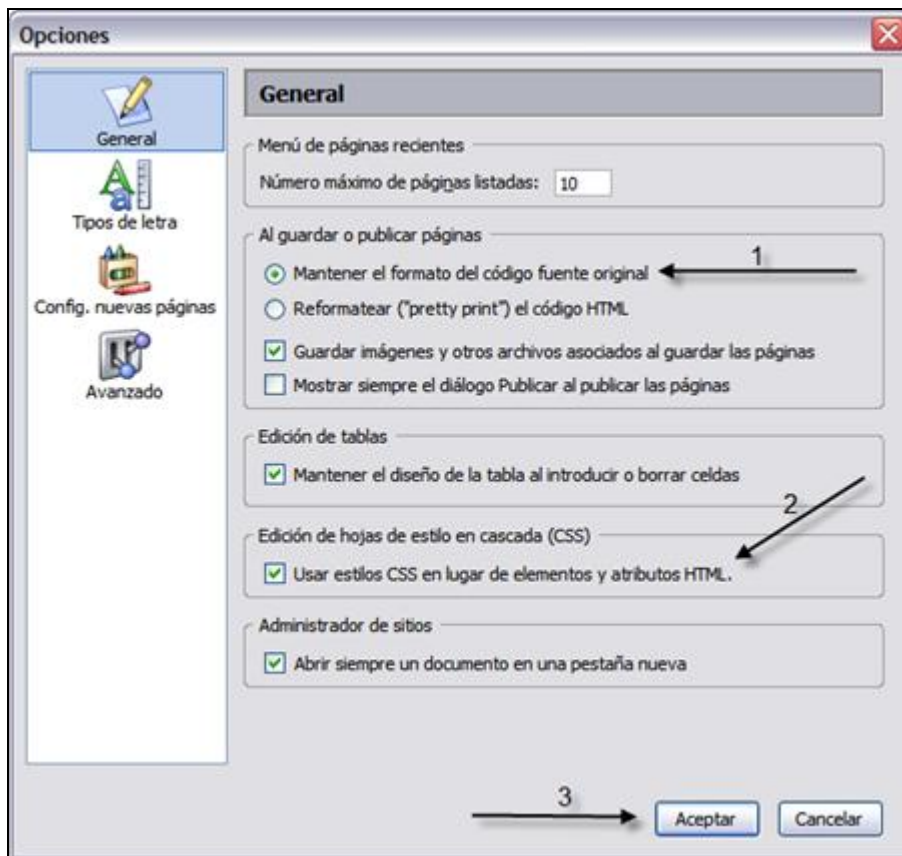


Tal como señala la ilustración, a la izquierda hay un panel que facilita la gestión de los archivos y las páginas que forman parte de un sitio. Para los siguientes ejercicios cerraremos este panel. Se puede cerrar (y abrir) con la tecla de función **F9**. Una vez cerrado el panel tendremos esta pantalla:



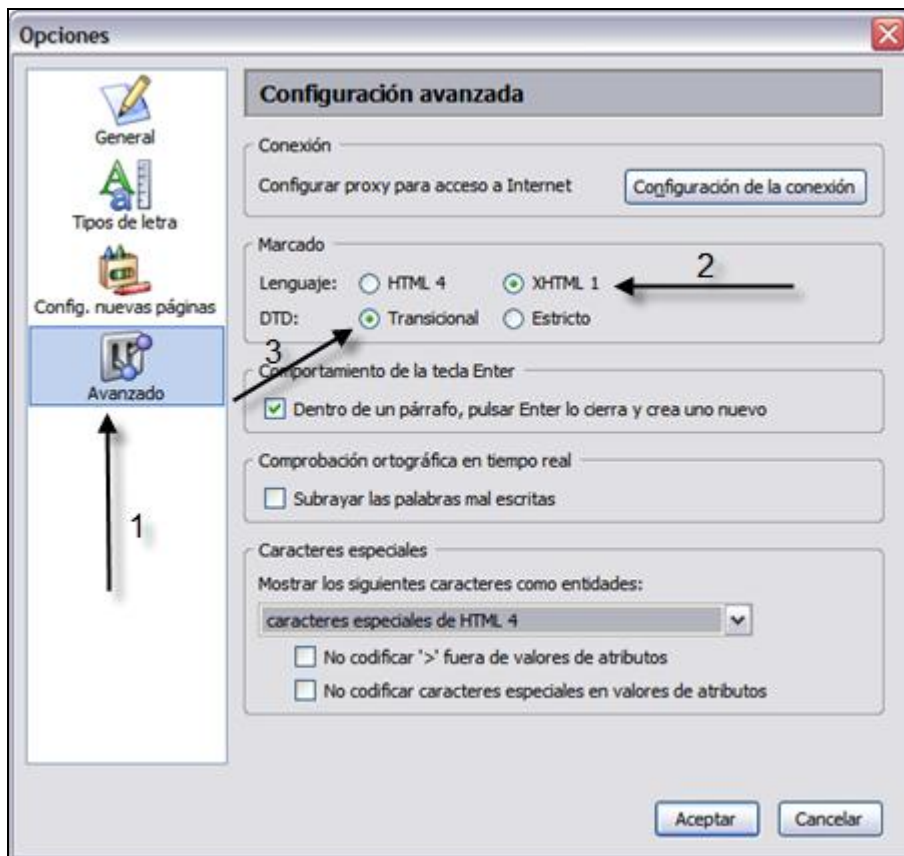
2. Crear un nuevo documento

Antes de crear el primer documento, debe preparar la siguiente configuración de **NVU**. Haga clic en **Herramientas > Preferencias**. En la primera ventana de diálogo que aparecerá (**General**) seleccione: (1) **Mantener el formato del código fuente original** (2) seleccione: **Usar estilos CSS en lugar de...** tal como muestra la siguiente ilustración:

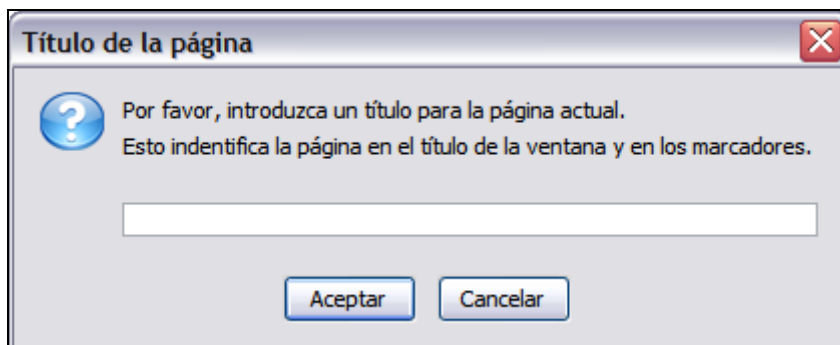


(3) haga clic en **Aceptar**. Es importante que cada vez que inicie NVU se asegure que estas dos opciones están seleccionadas.

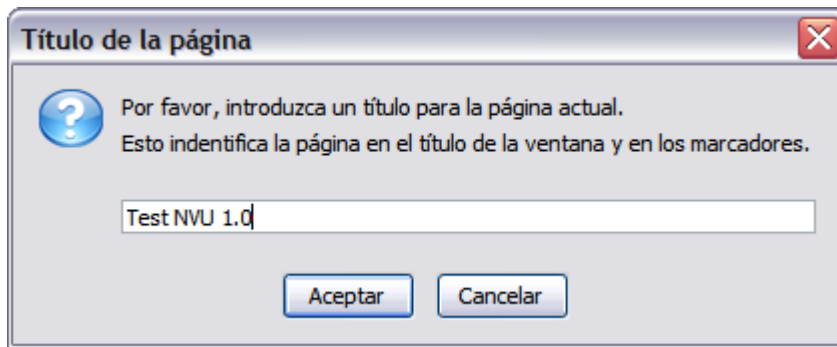
A continuación, vaya a de nuevo a la opción de configuración (**Herramientas > Preferencias**) y seleccione ahora **Avanzado**; marque las opciones: **XHTML 1** y **Transicional** y haga clic en **Aceptar**:



Ahora podemos crear nuestro primer documento en NVU. En primer lugar, procederemos a guardar el documento aunque aún no hayamos escrito nada con **Archivo > Guardar** y aparecerá esta ventana de diálogo:



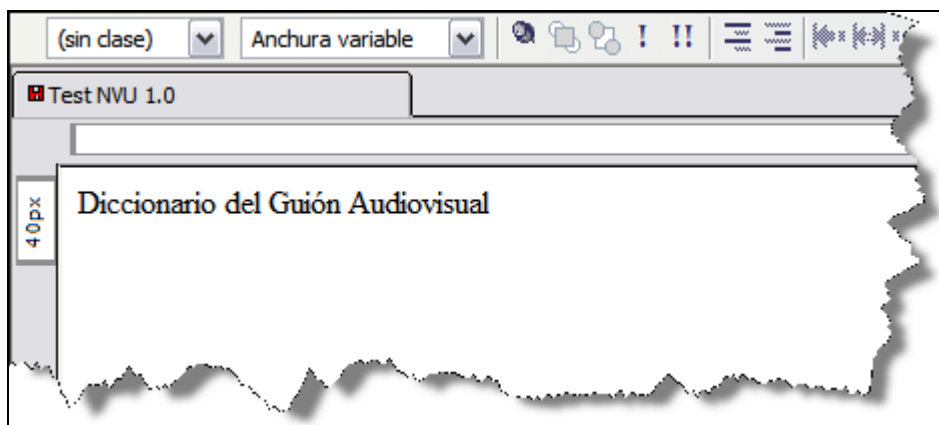
NVU nos pide que demos un título a la página, pero, atención este título no es el nombre del archivo, sino el texto que será asignado a la etiqueta `title` del código fuente de la página. Puede darle el título siguiente: Test NVU 1.0, de este modo:



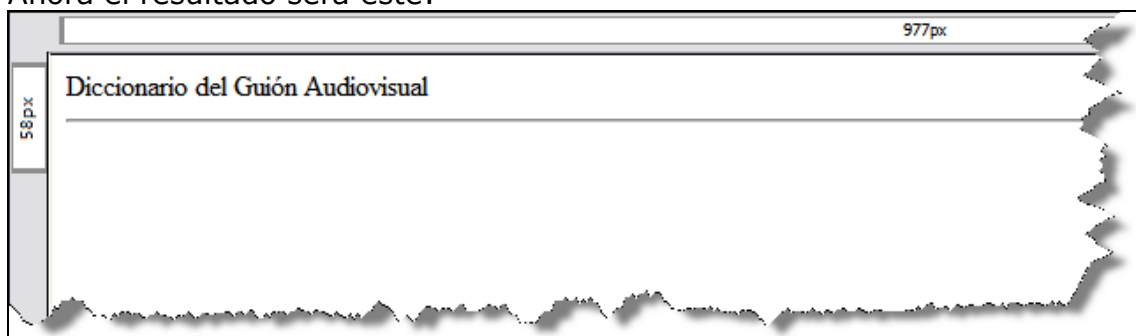
Haga aceptar y ahora el sistema le pedirá el nombre del archivo. Puede dar este nombre: **nvu01.htm**.

Con NVU se puede trabajar de un modo parecido con el que trabajamos con un editor de texto como Word. Vamos a hacer un documento muy simple de prueba que contendrá un título, una barra horizontal, una tabla y un pie.

Entre en NVU el siguiente texto: "Diccionario del Guión Audiovisual" (sin las comillas)



Haga **intro** al final de la frase y seleccione **Insertar > Línea horizontal**. Ahora el resultado será este:

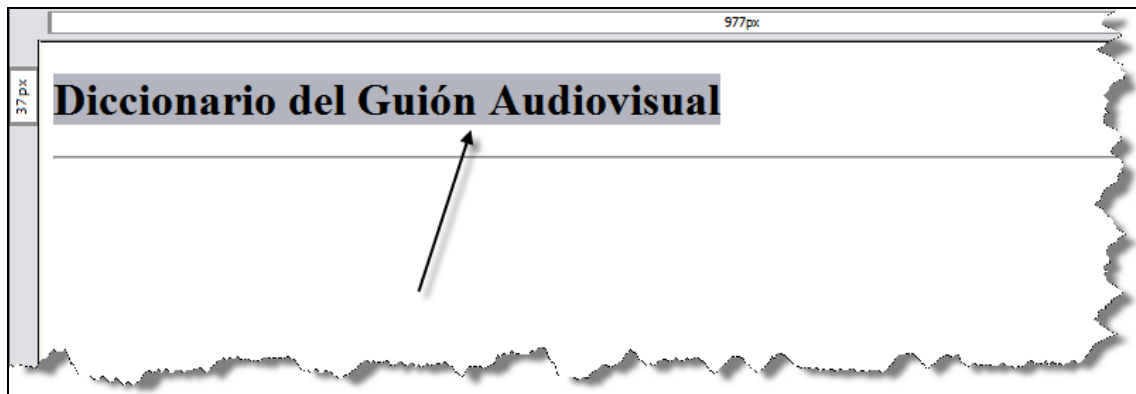


Ahora seleccione con el cursor la frase **Diccionario del Guión Audiovisual**.

Despliegue, con el texto seleccionado la opción **Título 1** de la caja desplegable de **formatos** que está situada en la barra intermedia de menús:



El resultado será este:



3. El código fuente

Vamos a comprobar ahora que NVU se ha ocupado de generar por nosotros el código fuente. Guarde el resultado sin cerrar el documento. Haga clic en la pestaña inferior que permite ver (y editar) el código fuente:

Debería ver un código fuente como éste, generado de forma automática por NVU:


```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2. <html
3.   xmlns="http://www.w3.org/1999/xhtml">
4. <head><meta content="text/html; charset=ISO-8859-1"
5.   http-equiv="content-type" /><title>Test NVU 1.0</title>
6. </head>
7. <body><h1>Diccionario del Guión Audiovisual</h1>
8. <hr style="width: 100%; height: 2px;" /></body>
9. </html>
```

Puede observar, entre otras cosas, que ha generado la declaración DOCTYPE relativa a **XHTML Transitional**, así como las etiquetas correspondientes a los elementos que hemos ido creando: **body**, **h1**, **hr**, etc. Observe, además que en el caso de algunas etiquetas, como **hr**, ha

añadido una declaración de estilo dentro del propio elemento como un atributo de la etiqueta, es el caso de `hr`:
<hr style="width: 100%; height: 2px;" />.

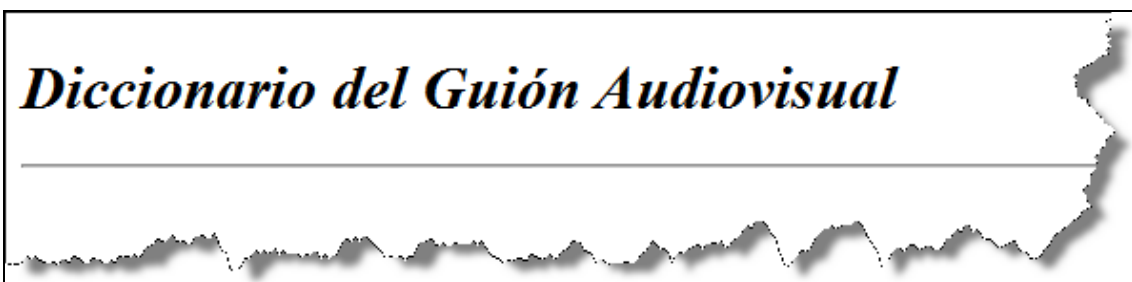
Esta forma de añadir estilo dentro de una etiqueta mediante el atributo `style="..."` se denomina estilo **en línea**.

3.1. Modificar el código fuente

Con NVU podemos modificar el código fuente modificando directamente las etiquetas en la vista XHTML:  Código fuente. Vamos a hacer una prueba. Modifique directamente el código fuente de la página tal como muestra la ilustración siguiente:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html
  xmlns="http://www.w3.org/1999/xhtml">
  <head><meta content="text/html; charset=ISO-8859-1"
  http-equiv="content-type" /><title>Test NVU 1.0</title>
</head>
<body><h1><em>Diccionario del Guión Audiovisual</em></h1>
<hr style="width: 100%; height: 2px;" /></body>
</html>
```

Guarde la página (sin cerrarla), vuelva a la vista **Normal** y compruebe que ahora el título se muestra en negritas como consecuencia de haber añadido la etiqueta `em`:

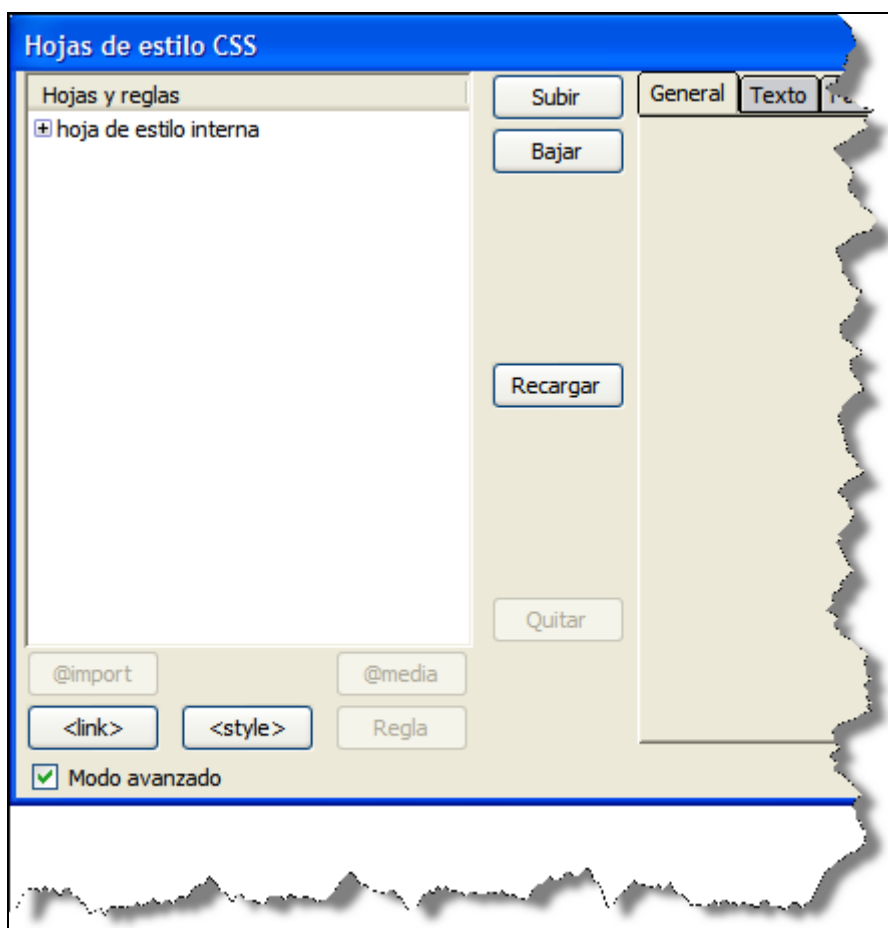


Vuelva a la vista **Código fuente** y elimine las etiquetas `` y `` de forma que el elemento `h1` vuelva a tener su aspecto normal.

4. Usar CSS

NVU facilita el uso de estilos CSS para modificar elementos. Ya hemos visto que aplica CSS **en línea** directamente cada vez que editamos o creamos contenido. Ahora veremos que también podemos crear declaraciones de estilo que afecten a todos los elementos de una clase, por ejemplo, a los elementos `h1`, los elementos `p`, etc.

En la vista **Normal** de NVU, haga **Herramientas > Editor CSS**. Aparece esta ventana:



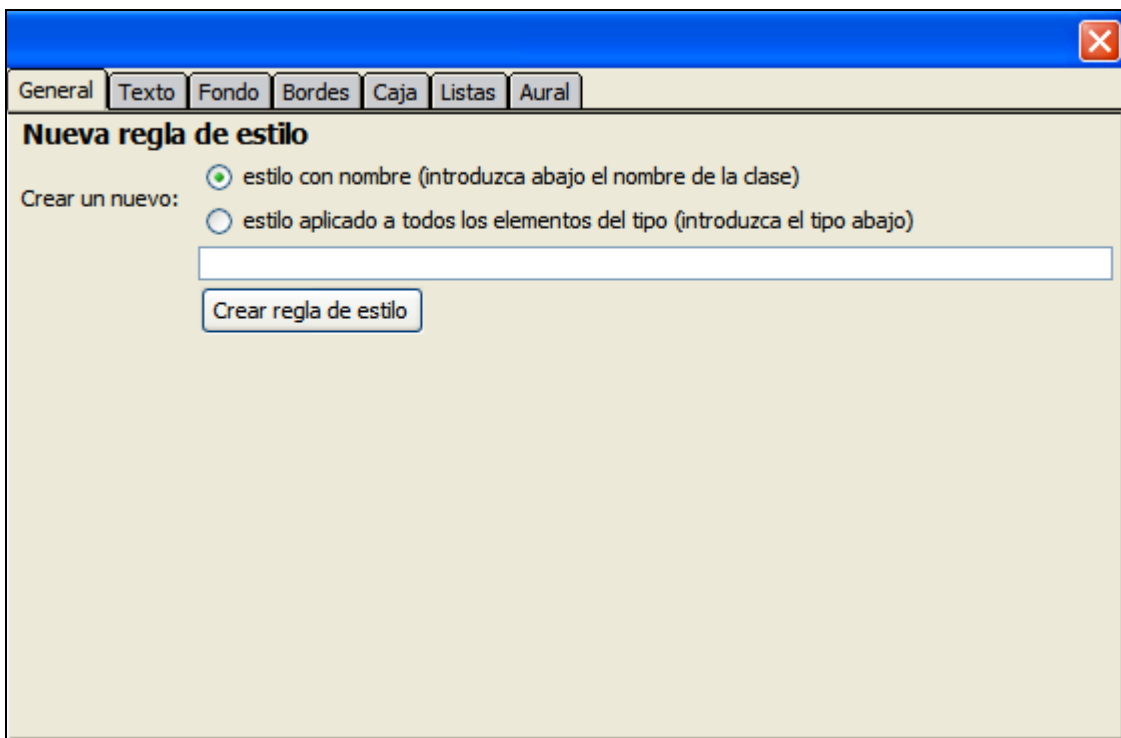
Desmarque la opción **Modo avanzado**:



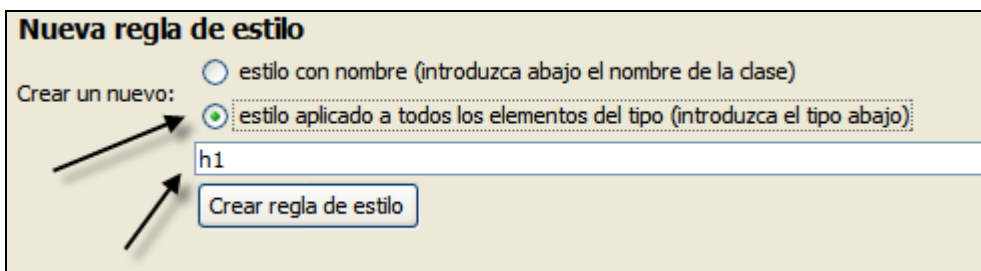
Ahora, haga clic en **Regla**:



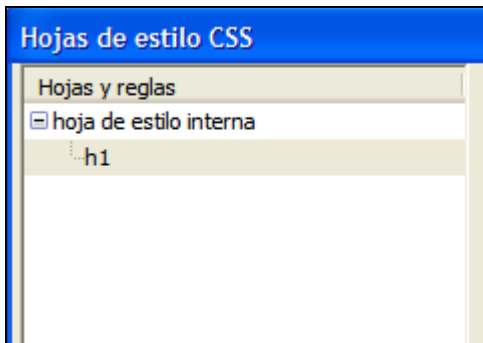
De este modo, se activa el grupo de opciones de la derecha de la ventana de diálogo:



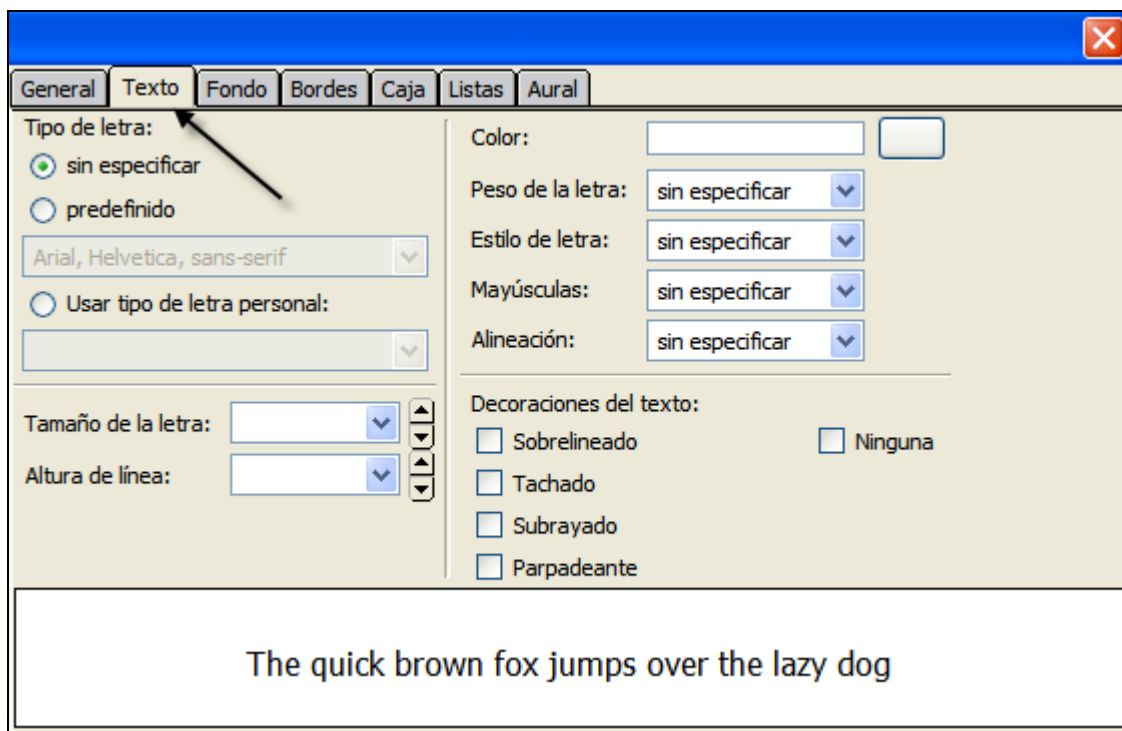
Vamos a definir un estilo para **h1** y para **body**. Haga clic en estilo aplicado a todos los elementos del tipo (introduzca el tipo abajo) y entre h1:



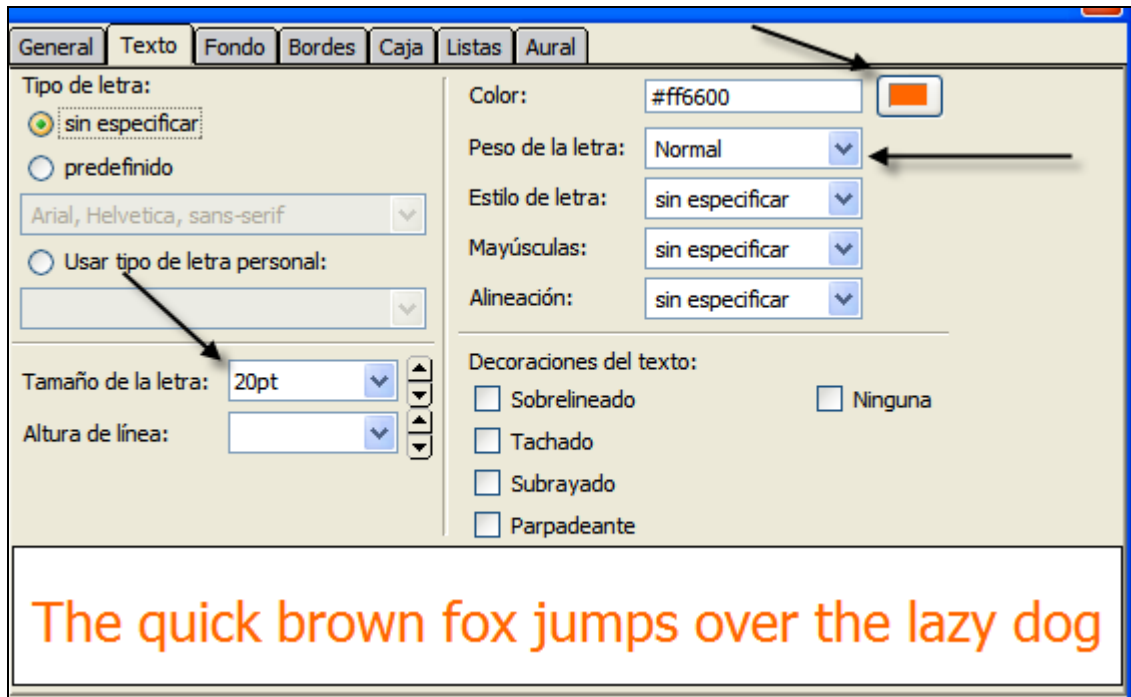
Haga clic en el botón **Crear regla de estilo**. La ventana de la izquierda, que contiene el árbol de los libros de estilo de la página web ha cambiado:



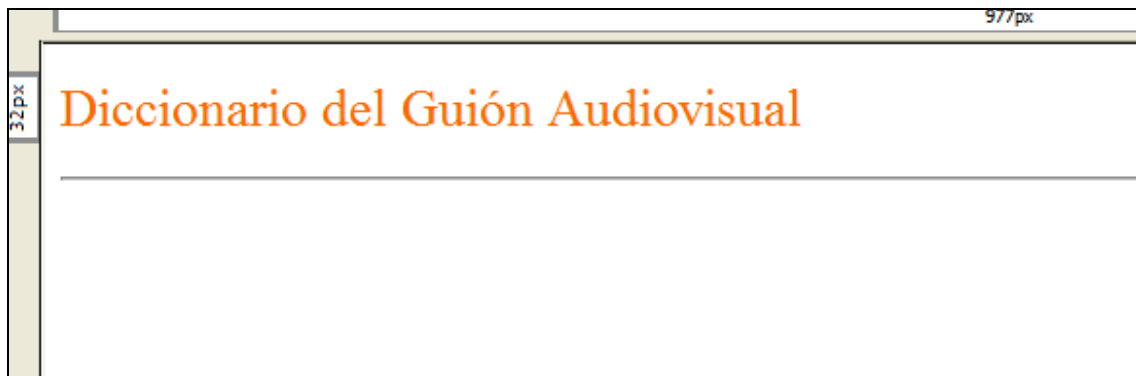
Ahora muestra la posibilidad de declarar un estilo para el elemento `h1`. Haga clic en la pestaña **Texto** del grupo de opciones de la derecha:



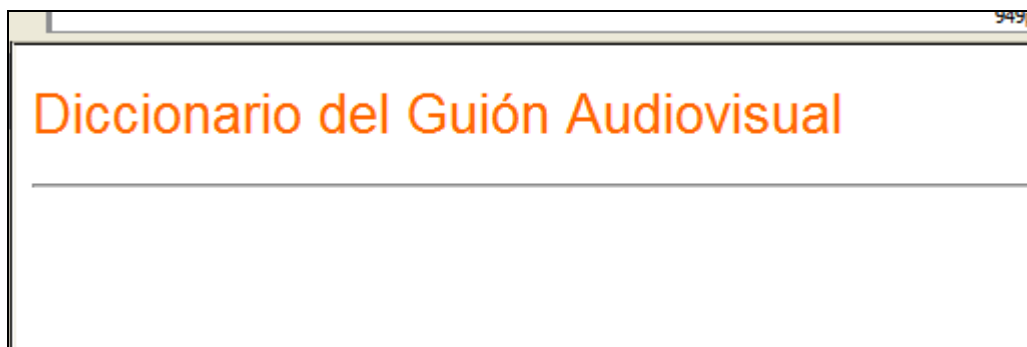
Seleccione en **Color**, un color de su elección, en **Tamaño de la letra** entre el valor **20pt** y en **Peso de la letra** seleccione **Normal**:



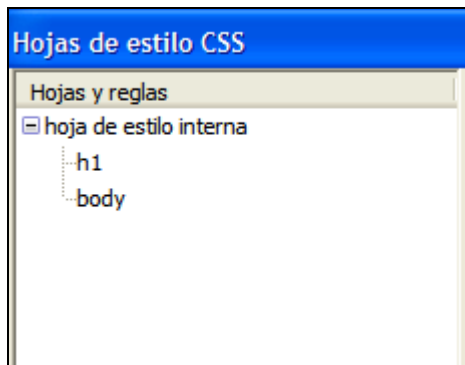
Haga clic en Cerrar y verá de forma automática cómo h1 ha cambiado y ahora la página se ve así:



Vuelva a activar el editor de CSS de NVU y consiga modificar el elemento `body` de manera que utilice ahora la familia de letras **Arial, Helvetica, sans serif**. La página, una vez creada la regla, quedará así:



El editor de CSS mostrará ahora un árbol con dos elementos definidos:

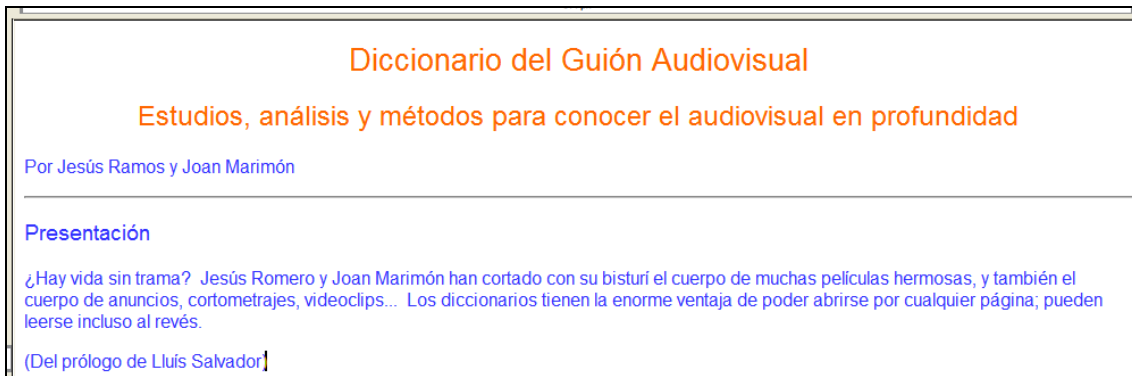


Compruebe que NVU ha generado el código correspondiente de la hoja de estilo en el código fuente:

```
6.  
7. <style  
8.   type="text/css">  
9.   h1 { font-weight: normal;  
10.      color: rgb(255, 102, 0);  
11.      font-size: 20pt;  
12.      }  
13.  
14.   body { font-family: Arial,Helvetica,sans-serif;  
15.      }  
16.  
17. </style></head>  
18.
```

4.1. Nuevas modificaciones

Complete la página con el texto que verá a continuación y haga las modificaciones que necesite mediante CSS para que el resultado ahora sea el siguiente:

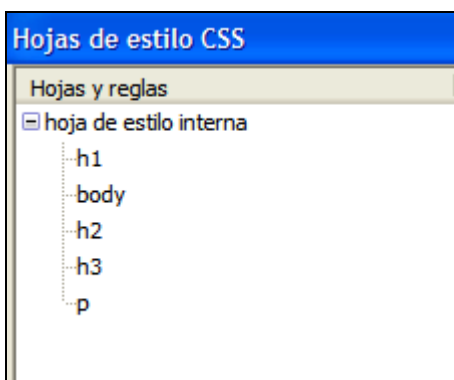


Si le resulta más cómodo, aquí tiene el nuevo texto (puede copiarlo y pegarlo en la página en lugar de teclearlo usando la opción **Pegar sin formato**):

```
Diccionario del Guión Audiovisual
Estudios, análisis y métodos para conocer el audiovisual en
profundidad
Por Jesús Ramos y Joan Marimón
Presentación

¿Hay vida sin trama? Jesús Romero y Joan Marimón han cortado con su
bisturí el cuerpo de muchas películas hermosas, y también el cuerpo de
anuncios, cortometrajes, videoclips... Los diccionarios tienen la
enorme ventaja de poder abrirse por cualquier página; pueden leerse
incluso al revés.
(Del prólogo de Lluís Salvador)
```

Como pistas podemos indicar que hemos creado estilos para los siguientes elementos, tal como muestra el propio editor:



Más pistas: hemos declarado la segunda línea del título (Estudios, análisis, etc.) como un elemento **h2** y Presentación como un elemento **h3**. Una vez entrado el texto, hemos marcado explícitamente cada párrafo como tal, es decir, como **Párrafo** en la ventana de formatos.

5. Uso de tablas

Una forma muy habitual de planificar la estructura de un sitio es mediante tablas. Una estructura habitual consiste en definir una tabla de tres filas y

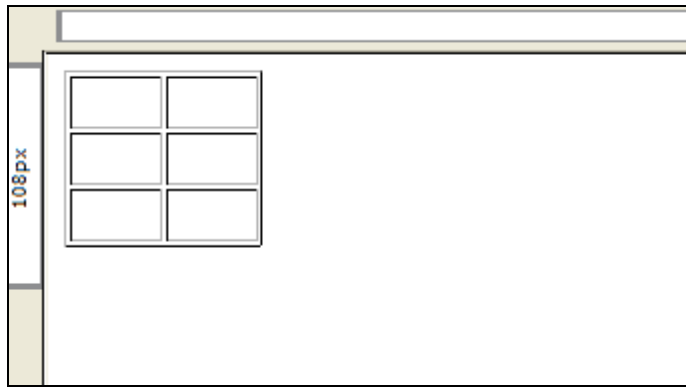
dos columnas en las cuales la primera columna se destina a contener los enlaces de navegación y la segunda columna al contenido. Por su parte, la primera fila se destina al título y la tercera a datos de contexto como la última fecha de actualización, el nombre del autor (si corresponde), etc. La siguiente figura ilustra esta distribución:

	Título
Navegación Item1 Item2 Item3	Contenido _____ _____ _____ _____
	Fecha, Autoría

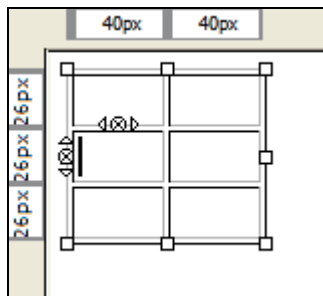
Con NVU es muy fácil crear tablas. Vamos a hacer una estructura como la anterior. Haga un nuevo documento, que llamaremos **guia.htm**. Puede darle el título de Guía del Audiovisual. Una vez nombrado el documento, haga **Tabla > Insertar > Tabla...** Aparecerá esta ventana:



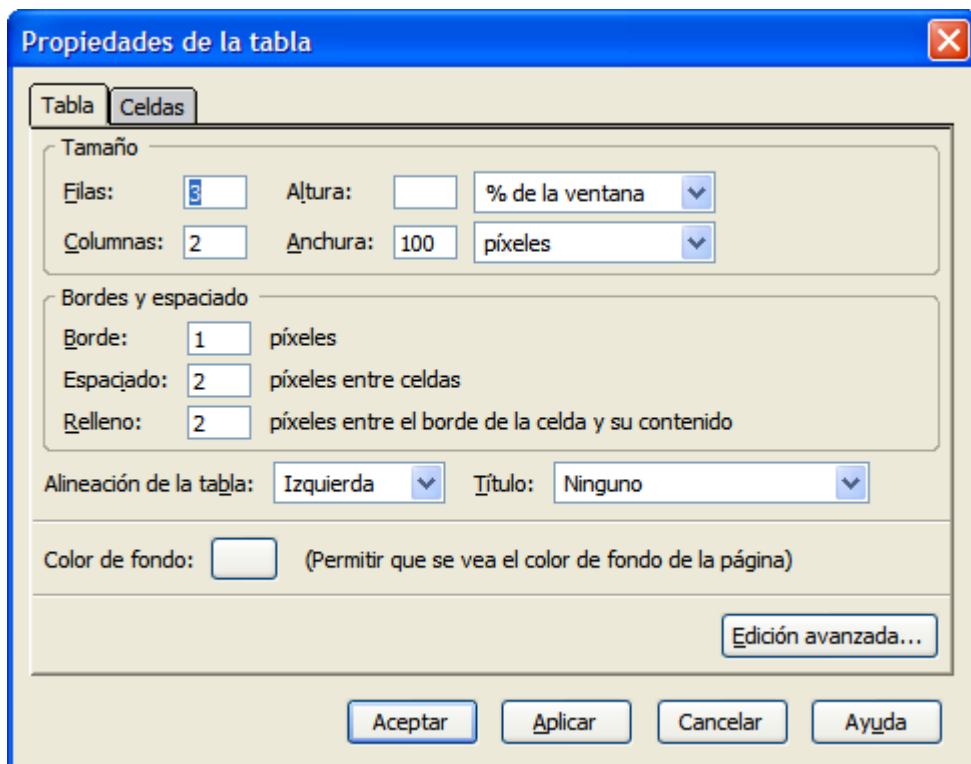
Seleccione tres filas y dos columnas y haga clic en **Aceptar**. Le aparecerá una tabla como ésta:



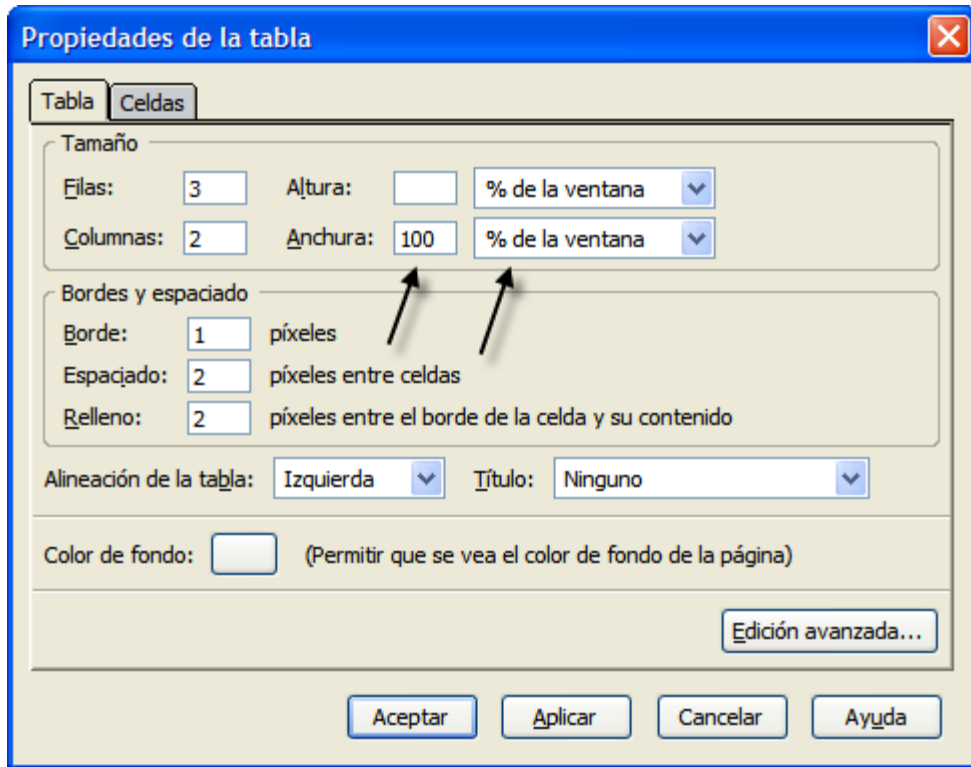
La razón de la que la tabla no ocupe todo el ancho de la página es que le ha asignado una pequeña anchura en píxeles (un píxel es un punto de la pantalla) que es una unidad de medida absoluta. Debemos definir la tabla en medidas relativas, es decir, en tanto por ciento y lo haremos asignándole el 100 por cien de la pantalla. Para ello, haga clic en la tabla para seleccionarla:



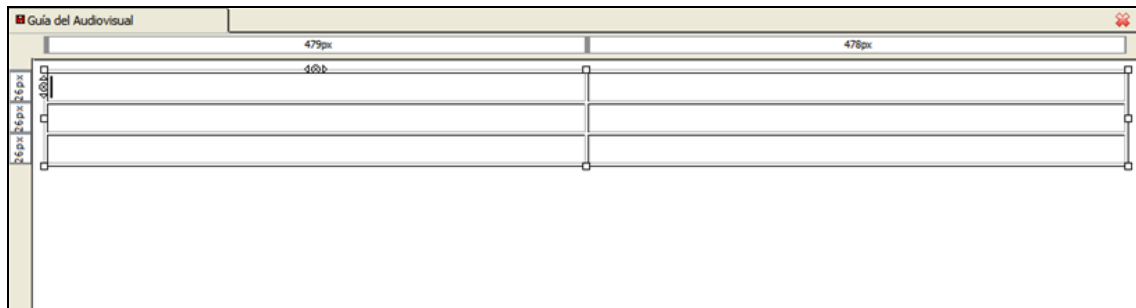
Haga clic en **Tabla > Propiedades de tabla**. Aparecerá esta ventana:



En el campo **Anchura** indique **100 % de la ventana** y haga clic en **Aceptar**:



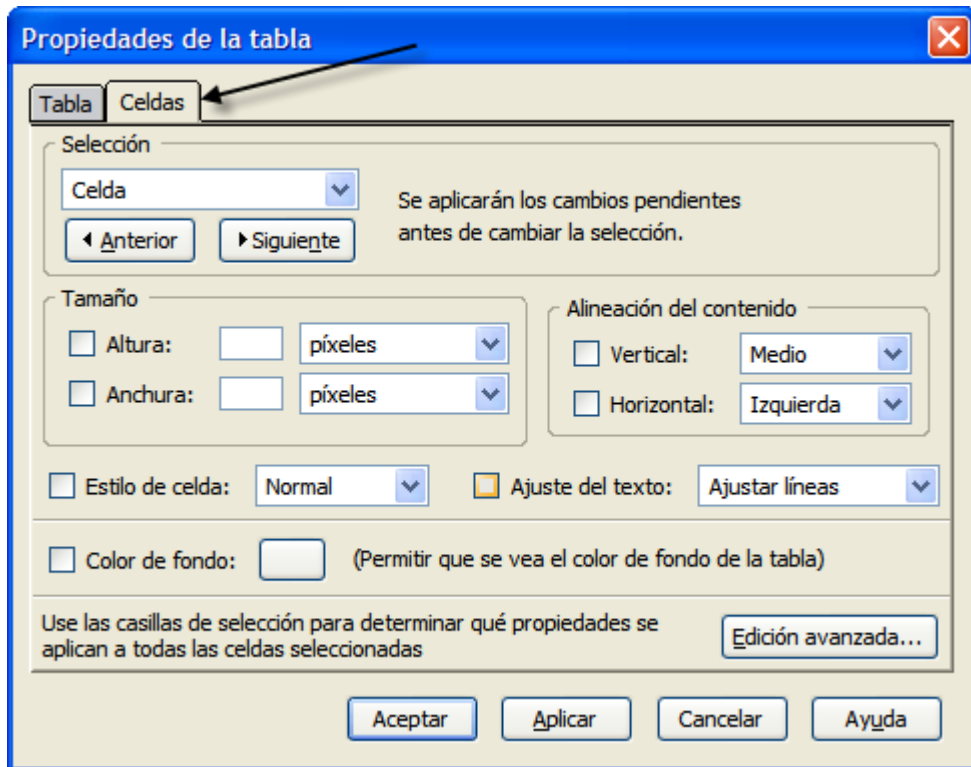
Ahora la tabla se expande para ocupar el espacio disponible:



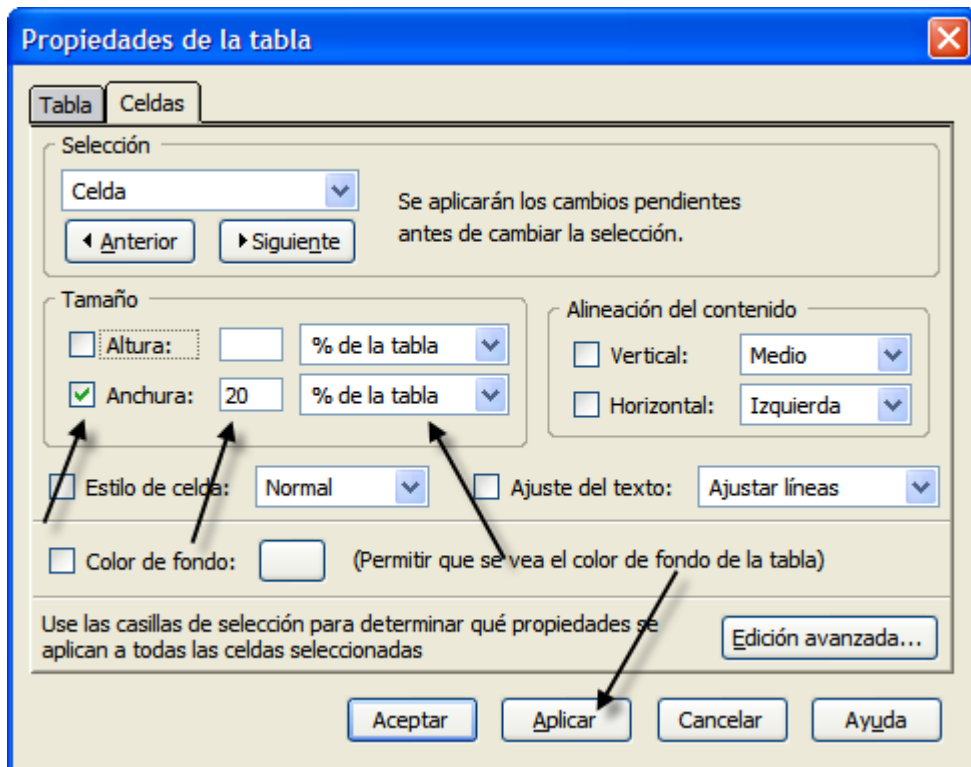
El problema es que la anchura de las columnas sigue estando definida mediante píxeles. Para acabar de diseñar el esquema de nuestra página debemos asignar un ancho relativo a cada columna. Normalmente, a la primera columna, la de navegación, se asigna un 20% y a la segunda columna, la de contenido un 80%:

Primera columna: 20%	Segunda columna: 80%

Vamos a aplicar este formato a la tabla que estamos formateando con NVU. Volvemos a seleccionar la tabla y volvemos a activar la ventana de propiedades de la tabla, pero en esta ocasión hacemos clic en la pestaña **Celdas**:

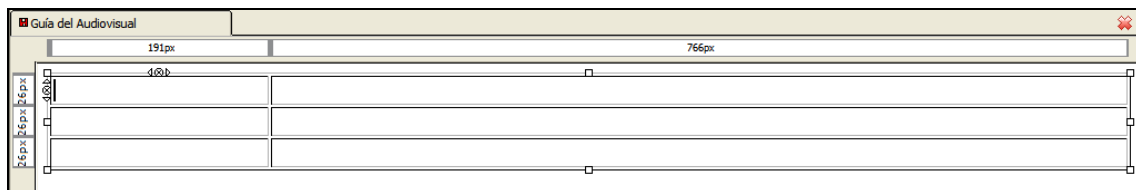


En **Tamaño** marcamos la casilla **Anchura** e indicamos **20% de la tabla** y hacemos clic en **Aplicar** (no en Aceptar para que aplique los cambios pero no se cierre aún la ventana puesto que hemos de modificar aún la segunda columna):



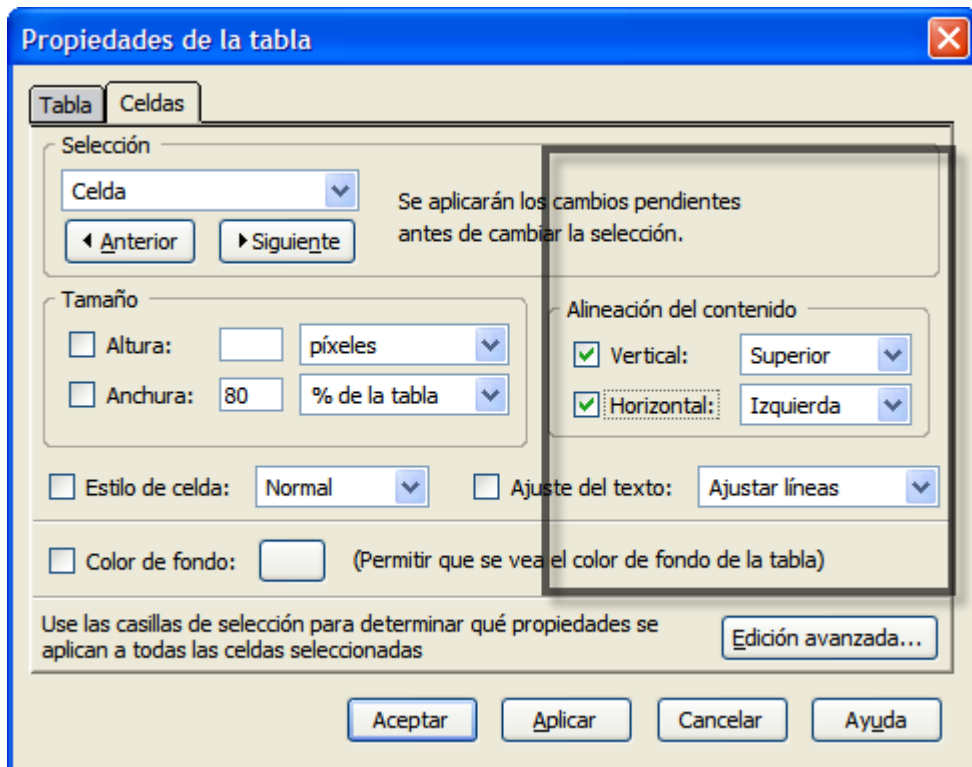
Ahora hacemos clic en el botón **Siguiente** y quedará seleccionada la segunda columna. Procedemos igual que en el caso anterior, pero ahora aplicamos un 80% de anchura. Ahora ya puede hacer clic en **Aceptar** y cerrar la ventana de propiedades de la tabla.

Ahora el formato de la tabla debe ser el siguiente:

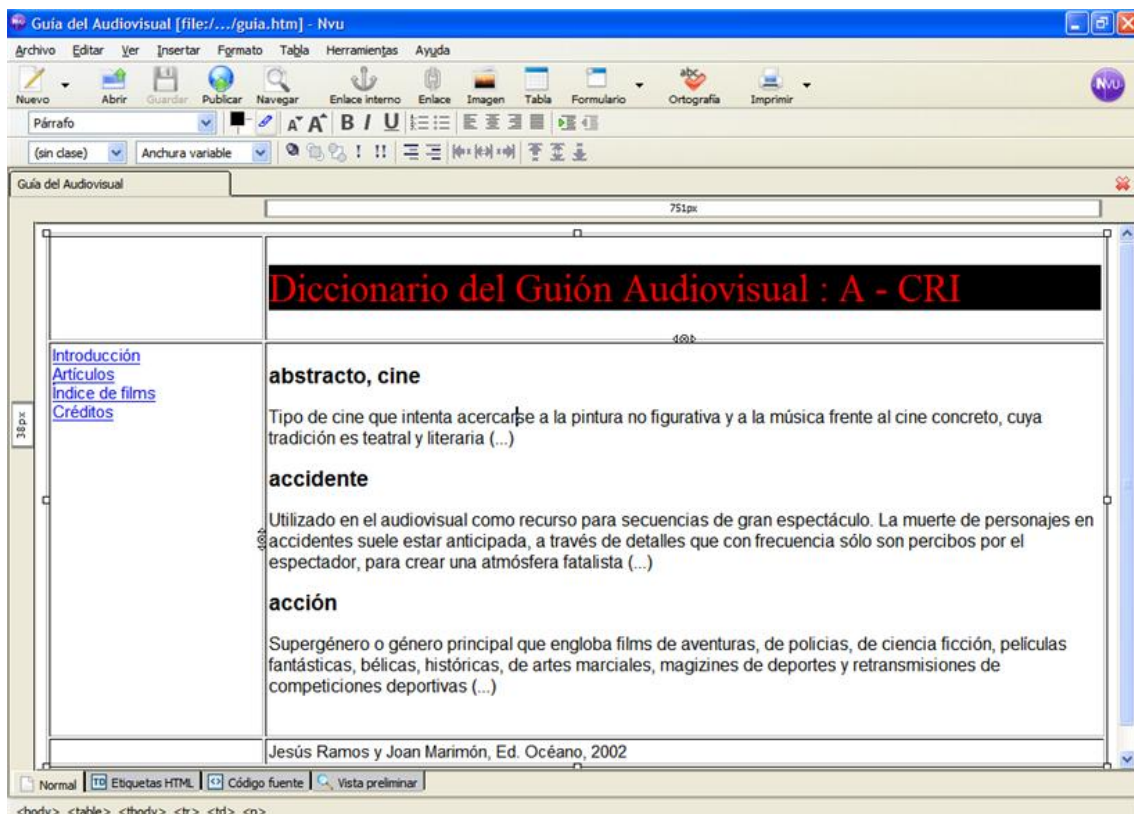


Vamos a crear la maqueta de un sitio web que contendría un supuesto diccionario audiovisual en línea (nos basamos en el Diccionario del Guión Audiovisual de J. Ramos y J. Marimón).

Modifique el contenido de la página que está editando de manera que quede, más o menos como verá en la captura siguiente (titulada **Maquetación con tablas-1**). Para alinear el texto dentro de la tabla utilice la ventana de **Propiedades de la tabla > Celdas > Alineación del contenido**:



Los enlaces de la navegación los puede simular mediante enlaces al mismo sitio (por ejemplo, a allmovie.com). Al final, debe quedar un resultado como éste, más o menos (el texto lo tiene un poco más abajo para que no tenga que teclearlo):

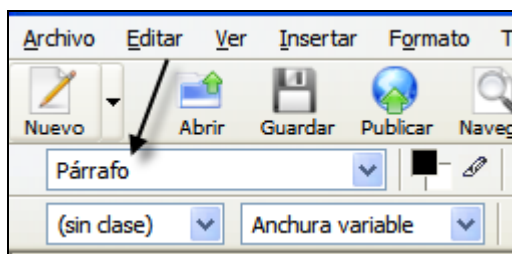


Maquetación con tablas-1

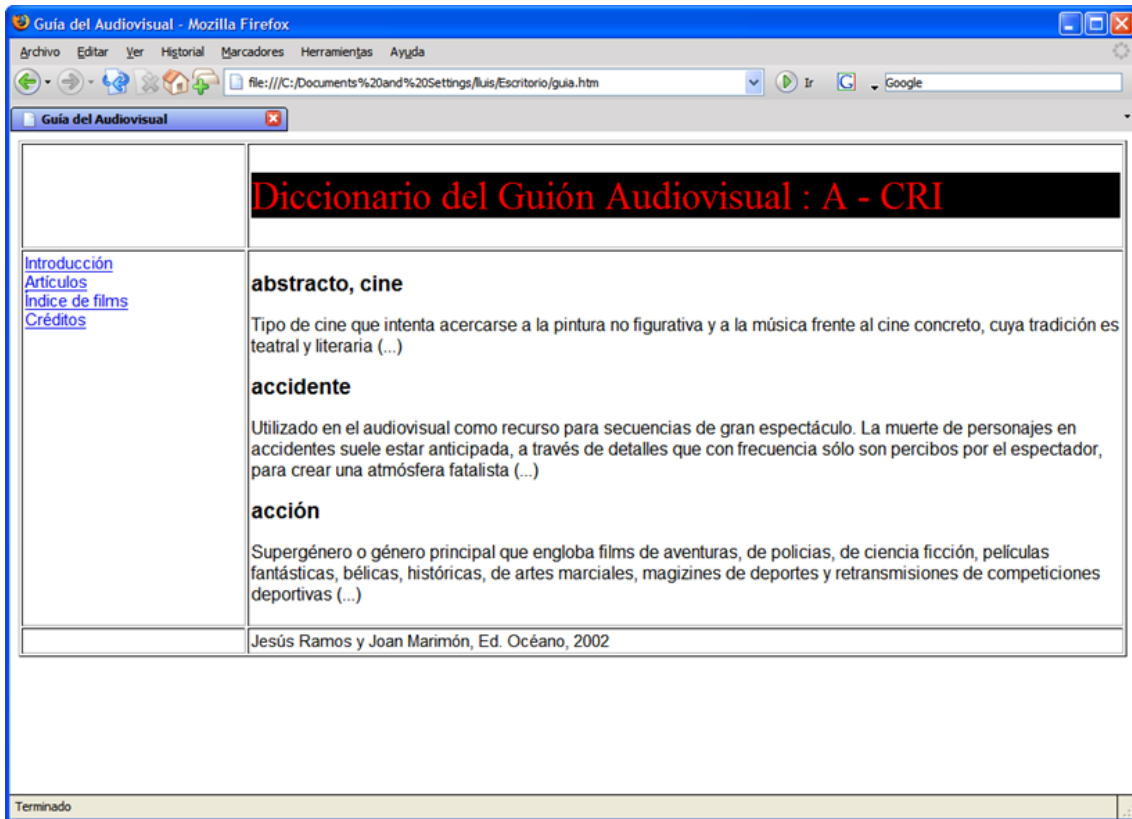
Para no tener que teclear todo el texto, puede copiarlo de aquí:

<p>abstracto, cine</p> <p>Tipo de cine que intenta acercarse a la pintura no figurativa y a la música frente al cine concreto, cuya tradición es teatral y literaria (...)</p> <p>accidente</p> <p>Utilizado en el audiovisual como recurso para secuencias de gran espectáculo. La muerte de personajes en accidentes suele estar anticipada, a través de detalles que con frecuencia sólo son percibidos por el espectador, para crear una atmósfera fatalista (...)</p> <p>acción</p> <p>Supergénero o género principal que engloba films de aventuras, de policías, de ciencia ficción, películas fantásticas, bélicas, históricas, de artes marciales, magazines de deportes y retransmisiones de competiciones deportivas (...)</p>

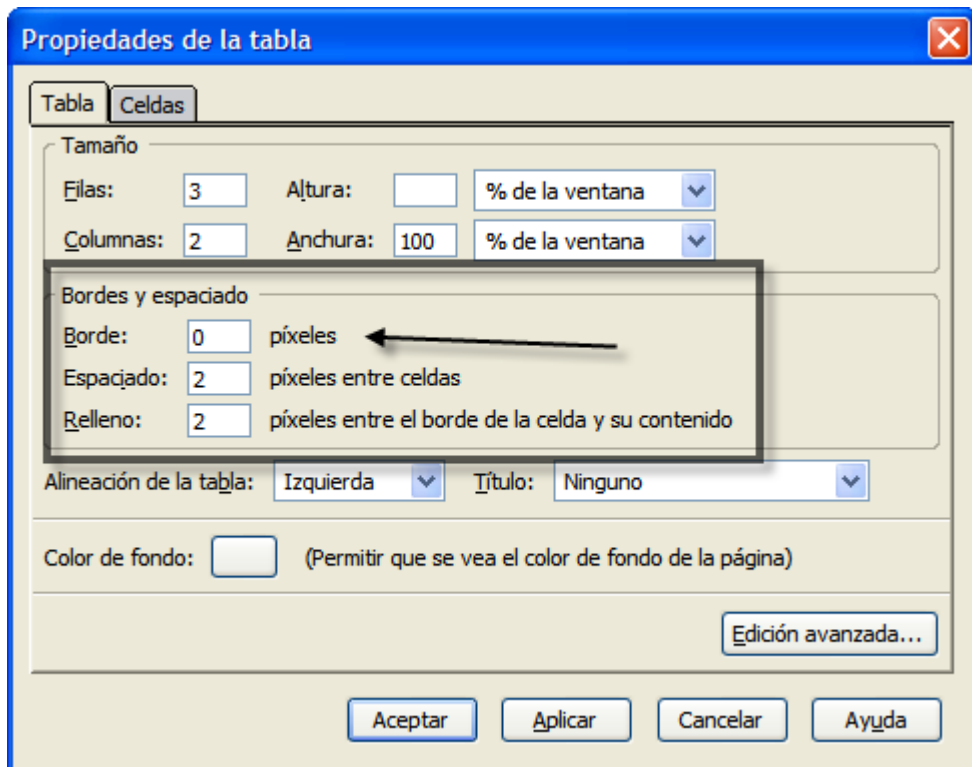
Una vez entrado el texto, ya sea que lo haya tecleado o que lo haya copiado y enganchado recuerde que debe asegurarse que cada párrafo está marcado como tal, o sea, de ser necesario marque cada párrafo, despliegue los estilos de la ventana de formato y seleccione **Párrafo**:



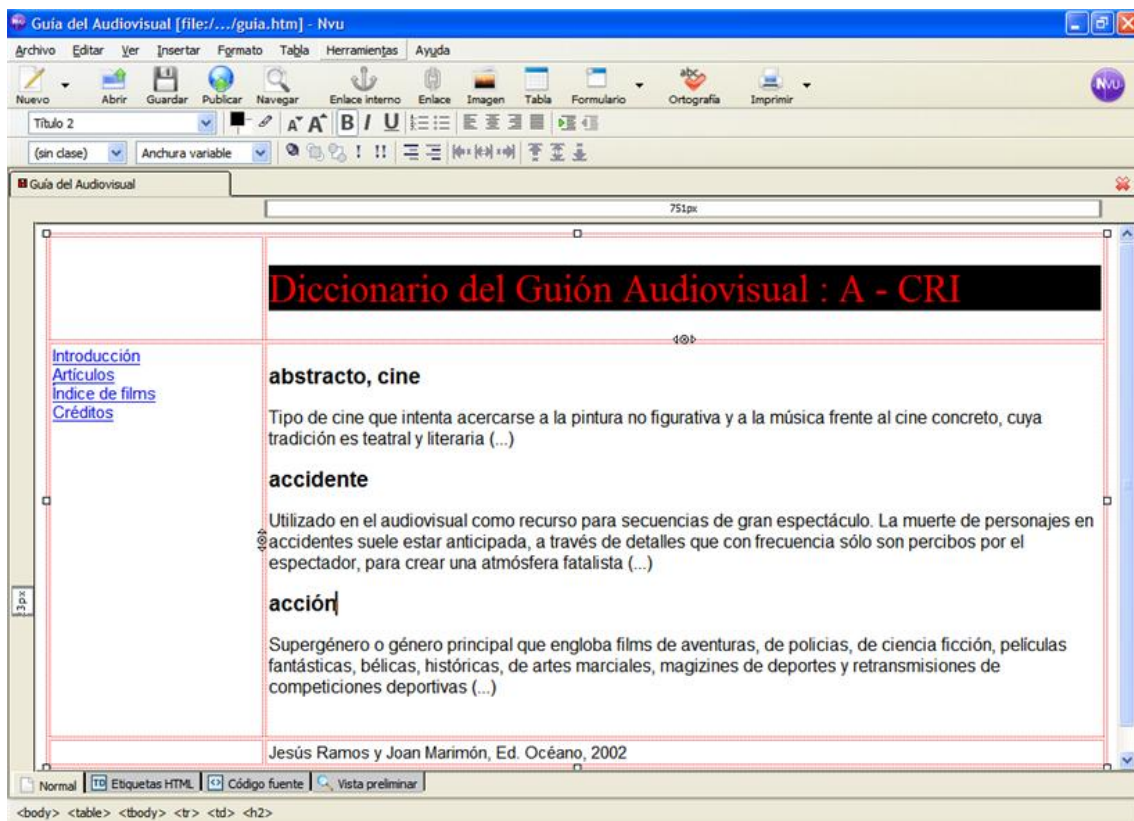
Marque igualmente el título principal como **Título 1** y los encabezados de cada artículo (abstracto, accidente, acción) como en **Título 2**, etc. Después defina estilos CSS para cada elemento, etc. El resultado, visto con el navegador debe quedar así:



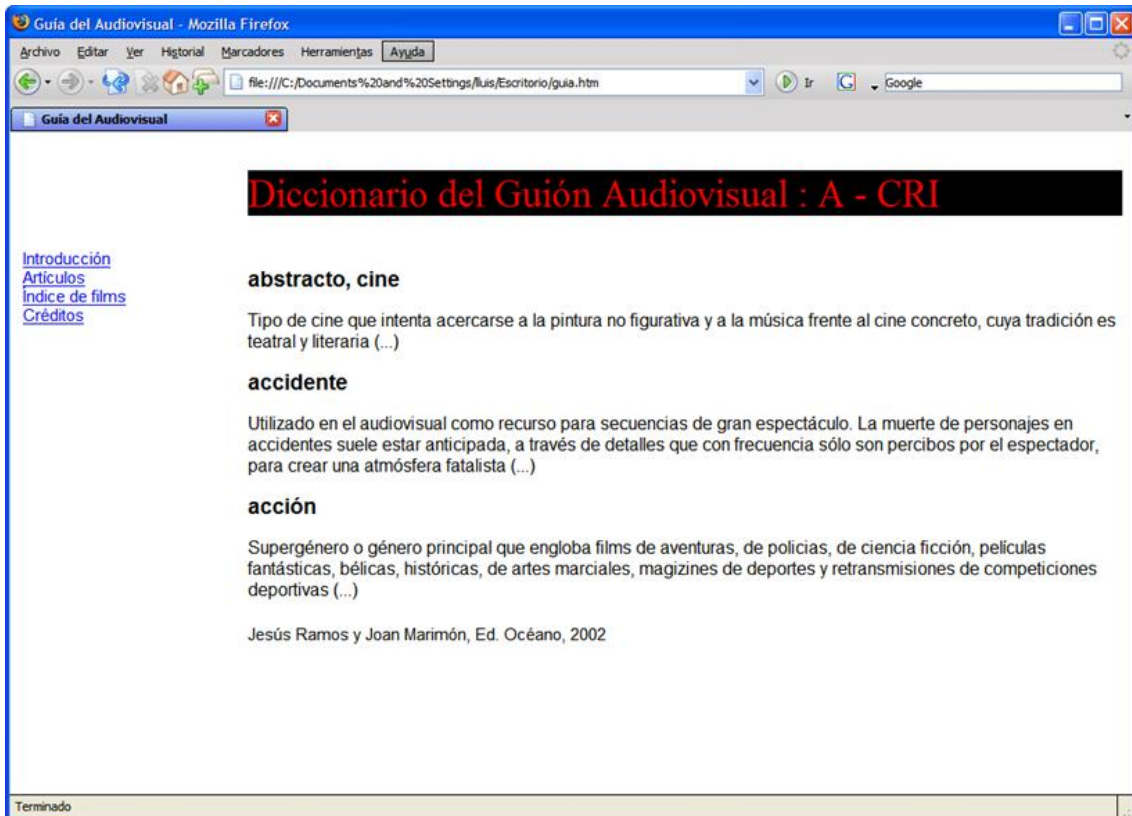
En general, el hecho de que puedan verse las guías de la tabla no se considera muy estético. Para solucionarlo, haremos un último cambio antes de dar por concluido el ejercicio. Haremos un cambio en las propiedades de la tabla de manera que las guías sean invisibles. Para ello, hacemos clic en la tabla para seleccionarla, activamos de nuevo las propiedades de la tabla y indicamos **Borde: 0 píxels:**



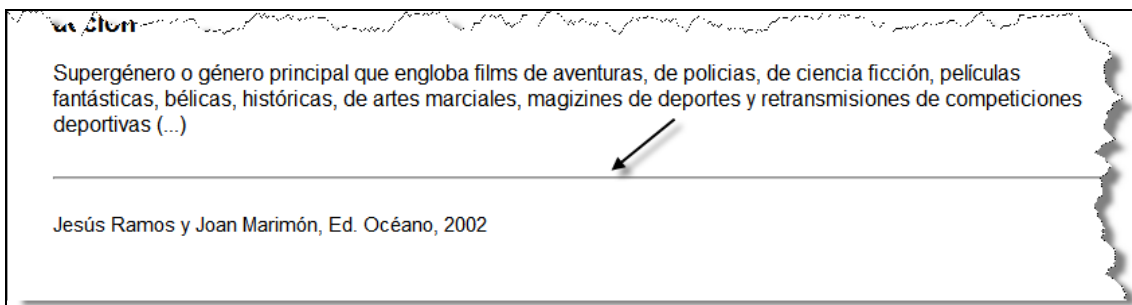
El resultado ahora en NVU será este:



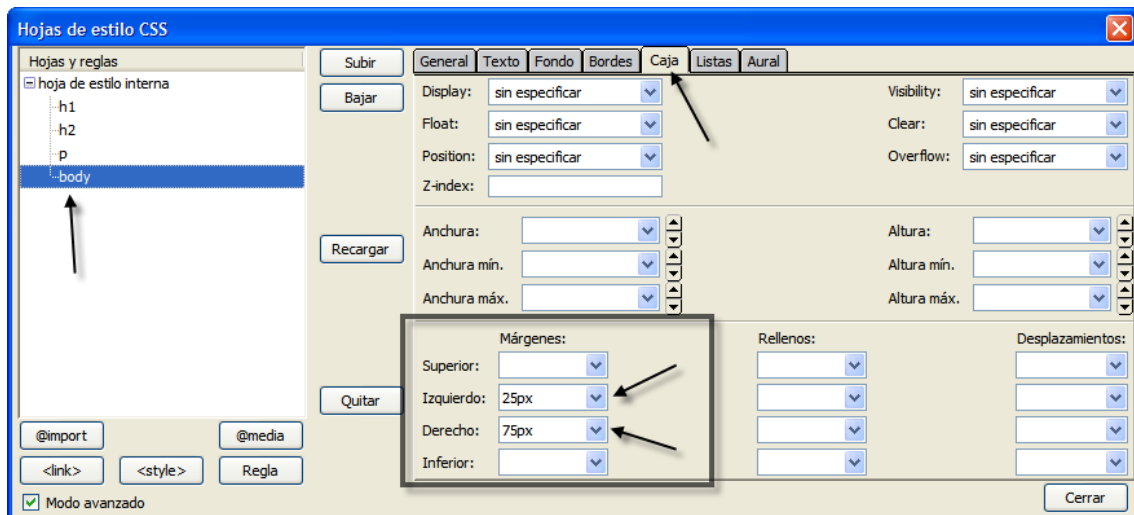
Y en el navegador, será este (no olvide guardar los cambios en NVU y obligar a recargar la página al navegador):



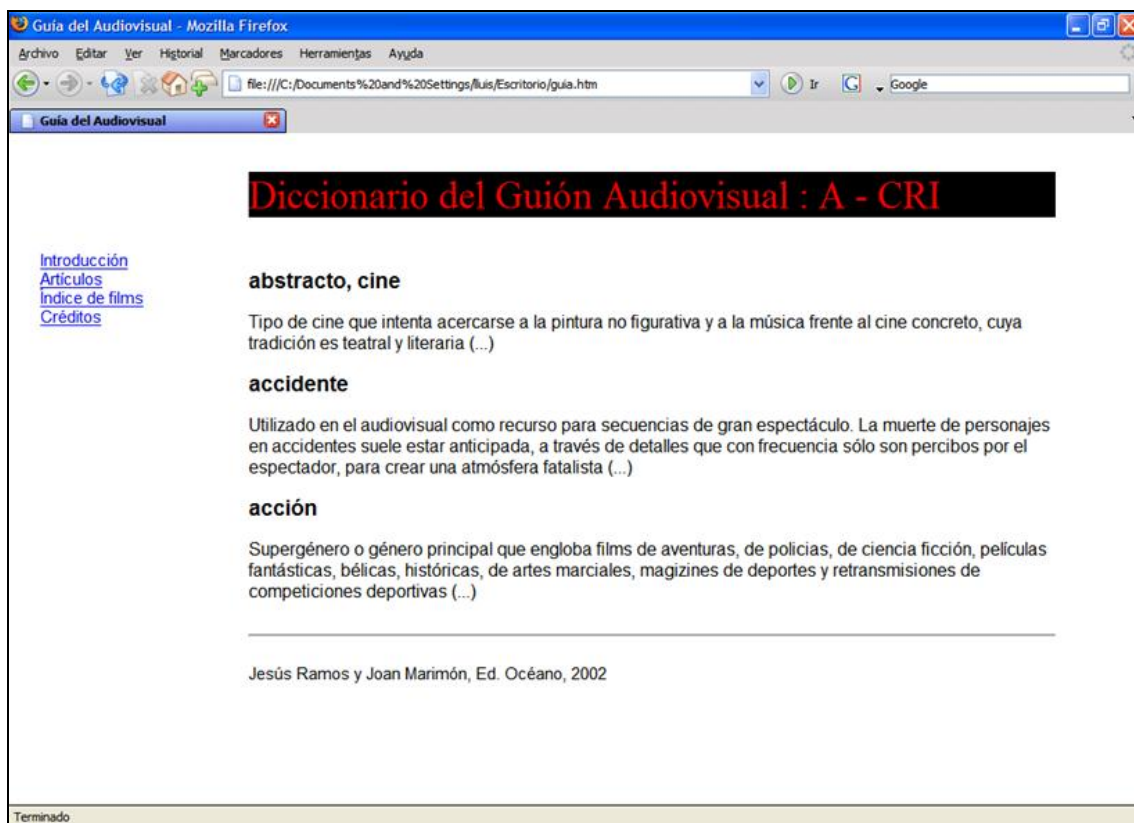
Ahora, no obstante, necesitaremos al menos una línea horizontal para separar de forma más clara algunos elementos. Podemos situar tal línea para separar el cuerpo del **Diccionario** de los datos de autoría:



Por último, para mejorar la legibilidad, vamos a asignar un margen general al elemento **body** a derecha e izquierda de la ventana, a fin de que quede más espacio en blanco a los lados. Asignaremos 25 píxels a la izquierda y 75 píxels a la derecha mediante la propiedad **Márgenes** de la pestaña **Caja**:



El resultado final, visto en el navegador será este:



Vista parcial del código fuente tal como lo ha creado, de forma totalmente automática NVU para la pagina de la captura anterior:

```

1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "h
2. <html
3.   xmlns="http://www.w3.org/1999/xhtml"><head><meta
4.   content="text/html; charset=ISO-8859-1"
5.   http-equiv="content-type" /> <title>Gu&iacute;a
6. del Audiovisual</title><style type="text/css">
7.   h1 { font-weight: normal;
8.     color: rgb(255, 0, 0);
9.     background-color: rgb(0, 0, 0);
10.    font-family: Times New Roman,Times,serif;
11.    font-size: 26pt;
12.  }
13.
14.  h2 { font-family: Arial,Helvetica,sans-serif;
15.    font-size: 14pt;
16.    color: rgb(0, 0, 0);
17.  }
18.
19.  p { font-family: Arial,Helvetica,sans-serif;
20.    font-size: 12pt;
21.    color: rgb(0, 0, 0);
22.  }
23.
24.  body { font-family: Arial,Helvetica,sans-serif;
25.    font-size: 11pt;
26.    margin-right: 75px;
27.    margin-left: 25px;
28.  }
29.
30.
31. </style></head>
32. <body style="text-align: left; width: 100%;"
33. border="0" cellpadding="2" cellspacing="2"><tbody><tr><td

```

Cuarta Parte: Cómo progresar a partir de aquí y referencias

1. Edición con estándares

El respeto por los estándares *incluye* actualmente el uso del lenguaje XHTML (en su modalidad Strict o Transitional) junto con CSS para editar páginas, más o menos tal como hemos intentado mostrar en este tutorial. Ahora bien, visto de otra forma *excluye* expresamente utilizar HTML con todas sus formas obsoletas de modificar la apariencia de la página con atributos como *font*, *center*, etc., e *implica* evitar en todo lo posible el uso de marcos (*frames*). Todo esto representa lo que podríamos denominar una forma razonable de adopción de estándares. No es su forma más avanzada.

Si deseamos cumplir al máximo con los estándares actuales debemos utilizar XHTML 1.0 bajo la modalidad *Strict* (en lugar de la modalidad *Transitional* como hemos hecho aquí). Tampoco deberíamos utilizar tablas

para maquetar páginas, sino utilizar en su lugar las propiedades de posicionamiento (las así llamadas capas) que proporciona CSS.

Sin embargo, hay tres importantes barreras para cumplir los estándares de una forma digamos total o extrema:

1. *Los editores de páginas web.* Los programas de creación y edición de páginas web están poco o mal preparados para trabajar con la versión extrema de los estándares. Aunque pueden incluir sin ningún problema la declaración DOCTYPE correspondiente a XHTML Strict, después no proporcionan ninguna forma de facilitar la creación de páginas que sea acorde cien por cien con ese DOCTYPE.
2. *Los navegadores.* Casi ningún navegador cumple al cien por cien los estándares (estamos pensando en particular en el posicionamiento con CSS) de manera cien por cien adecuada, ni siquiera las versiones aparecidas a lo largo del año 2006. El navegador que mejor parece asumir el estándar CSS es realmente minoritario: Opera. Por desgracia nadie puede diseñar pensando en Opera. Más el noventa por ciento del tráfico web proviene de los navegadores Firefox y Explorer, y ninguno de estos, ni siquiera en sus versiones actuales soportan bien el estándar CSS. Es más, en el caso que, a partir de un momento dado, digamos a partir de mañana mismo, aparecieran sendas versiones de Firefox y Explorer cien por cien compatibles con CSS, el parque de navegadores no se actualizaría de un día para otro.
3. *Los autores no profesionales de páginas web.* Por desgracia, los dos factores anteriores combinados generan un tercero: los autores no profesionales de páginas web. Se supone que la web es un espacio abierto a todos. Este *todos* se reduciría a una ínfima parte de la población si pretendemos que no se publique ninguna página web como sea utilizando XHTML Stric y posicionamiento CSS. El crecimiento de la web se detendría en seco. En realidad, hay algo absurdo en todo esto: resulta que para cumplir los estándares ahora la publicación de páginas web está menos al alcance que nunca. En realidad, ni siquiera los profesionales del diseño de páginas web están preparados para trabajar sin tablas y usar únicamente posicionamiento CSS.

La solución de compromiso, o la solución razonable dirigida al gran público, es decir, la solución que combina el máximo respeto posible a los estándares con la necesidad de proporcionar herramientas para *todos* y no limitadas a los ingenieros del MIT es la que hemos adoptado aquí: XHTML Transitional + CSS junto con un uso moderado de tablas para maquetación de páginas. Es una recomendación con la que se muestran de acuerdo autores que destacan en la defensa de estándares, como Jeffrey Zeldman.

Lo anterior no debería ser una excusa para los profesionales del diseño y la creación web. Este colectivo no puede ignorar que solamente protegerán de forma adecuada la inversión de sus clientes si entregan diseños con posicionamiento CSS y abandonan definitivamente los marcos (muy

especialmente) y las tablas (salvo cuando haya que presentar datos tabulares, claro).

2. El siguiente paso

Como se desprende de lo anterior, todo aquel que desee ir más allá de lo que se enseña aquí deberá profundizar en el uso de CSS y utilizar XHTML Strict de forma decidida. La bibliografía que se muestra a continuación intenta ser de ayuda.

3. Bibliografía recomendada sobre edición web con estándares

Charles Wyke-Smith. *Stylin' with CSS: A designer's guide*. New Riders, 2005

Dave Seha, Molly E. Holzchlag. *The Zen of CSS design*. New Riders, 2005

Jason Cranford Teague. *DHTML y CSS avanzado*. Anaya, 2005

Jeffrey Zeldman. *Designing with web standards* (second edition). New Riders, 2007

Jennifer N. Robbins. *Diseño Web: Guía de referencia*. Anaya, 2006

Peter Müller. "CSS, un juego de niños: Crea tu sitio web". *PC Cuadernos técnicos*, n. 33, 2007

Steven Champeon et al. *Cascading Style Sheets*. Anaya: 2003

Este tutorial sobre XHTML, CSS y NVU forma parte de los materiales didácticos preparados por los profesores L. Codina y C. Rovira para la asignatura de Documentación en los Medios de la Facultad de Comunicación Audiovisual de la Universidad Pompeu Fabra, Curso 2006-2007.

Forma recomendada de citación: [Lluís Codina, Cristòfol Rovira. Introducción al diseño de páginas web con estándares y programas open source: Tutorial de XHTML, CSS y NVU. Barcelona: UPF, Marzo 2007, 71 p. (Acceso: www.lluiscodina.com)]
