

<CCML>-Curso para web al día. 1ª parte

A continuación presentamos un curso para dar a conocer el lenguaje de meta etiquetas <CCML>, el cual se aplica en el programa 'web al día'. Hemos creado este curso para que, partiendo de un sencillo ejemplo, los programadores experimentados de HTML puedan obtener fácilmente una primera impresión de las posibilidades que ofrece <CCML>, así como sus principales formas de estructuración a la hora de programar. Este curso no se ha de entender como una documentación exhaustiva, sino como una introducción a dicho lenguaje.

Conceptos básicos de <CCML>

<CCML> es una extensión, por así decirlo, de HTML. <CCML> es la abreviatura de Content Composition Markup Language. Las etiquetas de <CCML> comienzan siempre con "cc:". Así, por ejemplo, la etiqueta <CCML> para introducir comentarios sería <cc:*>. CCML no diferencia entre mayúsculas y minúsculas.

Archivos utilizados

Cualquier proyecto <CCML> consta siempre de una serie de diferentes archivos que sirven en su conjunto para definir el diseño. La extensión de dichos archivos siempre será ".ccml". El único archivo cuyo nombre está desde un principio definido es "config.ccml", en el cual se encuentra determinado el resto de archivos. En la siguiente tabla pasamos a describir la función de los archivos:

Archivo	Función
config.ccml	Define todos los componentes del diseño
global.ccml	Define gráficos globales
link.ccml	Define los enlaces
navigation.ccml	Define la estructura de navegación de todas las páginas (siempre visible) y con ello los distintos marcos de éstas.
page.ccml	Define el contenido de una página
palette*.ccml	Define las paletas de colores
paragraph_form.ccml	Define un párrafo de tipo: Formulario
paragraph_list.ccml	Define un párrafo de tipo: Lista
paragraph_movie.ccml	Define un párrafo de tipo: Vídeo
paragraph_picture.ccml	Define un párrafo de tipo: Gráfico
paragraph_table.ccml	Define un párrafo de tipo: Tabla
paragraph_teaser.ccml	Define un párrafo de tipo: Titular
paragraph_text.ccml	Define un párrafo de tipo: Texto
splash.ccml	Define una página de inicio a modo de Intro
substyle*.ccml	Define distintas variantes de diseño
teaser_caption.ccml	Elemento del titular "Titular"
teaser_captionabstract.ccml	Elemento del titular "Titular, Texto breve"
teaser_captionabstractpicture.ccml	Elemento del titular "Titular, Texto breve, Imagen"
teaser_lookup.ccml	Elemento del titular "Lista A-Z"

El archivo "config.ccml"

Como ya se ha comentado, este archivo posee la definición principal del diseño. En él encontrará una serie de etiquetas <cc:file>. Dichas etiquetas tienen dos atributos: el atributo "obj" para elegir el objeto que debe ser definido y el atributo "src" para seleccionar el archivo correspondiente:

```
<cc:file obj="global" src="global.ccml">
<cc:file obj="navigation" src="navigation.ccml">
<cc:file obj="page" src="page.ccml">

<cc:file obj="paragraph.picture" src="paragraph_picture.ccml">
<cc:file obj="paragraph.table" src="paragraph_table.ccml">
<cc:file obj="paragraph.list" src="paragraph_list.ccml">
<cc:file obj="paragraph.text" src="paragraph_text.ccml">
<cc:file obj="paragraph.teaser" src="paragraph_teaser.ccml">
<cc:file obj="paragraph.form" src="paragraph_form.ccml">
<cc:file obj="paragraph.movie" src="paragraph_movie.ccml">

<cc:file obj="link" src="link.ccml">
<cc:file obj="splash" src="splash.ccml">

<cc:file obj="teaser.caption" src="teaser_caption.ccml">
<cc:file obj="teaser.captionabstract" src="teaser_captionabstract.ccml">
<cc:file obj="teaser.captionabstractpicture" src="teaser_captionabstractpicture.ccml">
<cc:file obj="teaser.lookup" src="teaser_lookup.ccml">
```

En este archivo se definen además una paleta y una variante de diseño. Para la definición de la variante de diseño también se indica el nombre de la paleta deseada.

```
<cc:palette caption="Standard" src="palette_default.ccml">
<cc:style caption="Standard" src="style_default.ccml" palette="palette_default.ccml">
```

El archivo "global.ccml"

Este archivo se utiliza para definir elementos tales como imágenes, hojas de estilo, etc. En nuestro ejemplo no encontrará ningún uso a este respecto y por tanto está vacío.

El archivo "navigation.ccml"

En este archivo se describe la estructura básica de todas las páginas. Es el centro neurálgico de todo diseño. En un principio, este archivo es un fichero normal en HTML. La primera particularidad es encuentra en la definición de la etiqueta <title>:

```
<title><cc:print value="&page.content.caption"></title>
```

La etiqueta <cc:print> sirve para escribir propiedades de un objeto en el código HTML. En el ejemplo de arriba, el objeto es "page.content" y su propiedad es "caption", es decir, el título de la página actual. El ampersand "&" comunica a 'web al día' que con "page.content.caption" se hace referencia a un objeto y que no se trata de una cadena de caracteres. Si olvida el "&", el título de la página será siempre "page.content.caption". Al crear una página web, 'web al día' aplica aquí el título real de la página.

A continuación le ofrecemos una lista de todas las propiedades del objeto "page":

Propiedad	Significado
.ishome	Indica si esta página es la general (Home).
.metaauthor	Da información acerca de la meta etiqueta Author.
.metadescription	Da información acerca de la meta etiqueta Description.
.metakeywords	Da información acerca de la meta etiqueta Keywords.
.metalang	Indica el atributo de idioma para la meta etiqueta Keywords.
.maintopic.caption	Indica el título del MainTopic en el que se encuentra la página actual.
.maintopic.url	Indica la URL del MainTopic en el que se encuentra la página actual.
.maintopic.abstract	Indica el texto breve del MainTopic en el que se encuentra la página actual.

Propiedad	Significado
.maintopic.number	Indica el número de página del MainTopic en el que se encuentra la página actual.
.topiccount	Indica la cantidad de Topics dentro del MainTopic actual en el que se encuentra la página.
.topic.caption	Indica el título del Topic en el que se encuentra la página actual.
.topic.url	Indica la URL del Topic en el que se encuentra la página actual.
.topic.abstract	Indica el texto breve del Topic en el que se encuentra la página actual.
.topic.number	Indica el número del Topic en el que se encuentra la página actual .
.subtopiccount	Indica la cantidad de SubTopics dentro del Topic actual en el que se encuentra la página actual.
.subtopic.caption	Indica el título del SubTopic en el que se encuentra la página actual.
.subtopic.url	Indica la URL del SubTopic en el que se encuentra la página actual.
.subtopic.abstract	Indica el texto breve del SubTopic en el que se encuentra la página actual.
.subtopic.number	Indica el número del SubTopic en el que se encuentra la página actual.
.content.caption	Indica el título de la página actual.
.content.uid	Indica el UID de la página actual.
.content.nexturl	Indica la URL de la página siguiente.
.content.previousurl	Indica la URL de la página anterior.
.content.nextcaption	Indica el título de la página siguiente.
.content.previouscaption	Indica el título de la página anterior.

Para la definición de las etiquetas <META>, el código es un poco más difícil de descifrar:

```
<meta name="author" content="<cc: print value="&site.metaauthor">">
<meta name="description" content="<cc: print value="&site.metadescri ption">">
<meta name="keywords" lang="<cc: print value="&site.metalang">" content="<cc: print
value="&site.metakeywords">">
```

Aquí se ha insertado código <CCML> en código HTML: Para el Meta-Author se ha insertado por ejemplo la etiqueta <cc:print value="&site.metaauthor"> dentro del código HTML <meta name="author" content="..">.

Un elemento estructural <CCML> muy importante es la siguiente rutina, que también en este curso denominamos 'bucle' (loop):

```
<cc: loop type="maintopic" obj="mymaintopic">
  <a href="<cc: print value="&mymaintopic.url">"><cc: print value="&mymaintopic.caption"></a>
</cc: loop><br>
```

Muy similar a como ocurre con otros lenguajes de programación, el código dentro de una rutina se "ejecuta" varias veces. Aquí se comunica a la etiqueta <cc:loop>, con ayuda de la propiedad "type", de qué tipo de rutina se trata; en este caso una rutina sobre todas las áreas principales o MainTopics (entre <body> y </body>). La referencia a la rutina se guarda en el objeto definido por el usuario "mymaintopic". ¡Hay que tener en cuenta que aquí no puede haber ningún ampersand ("&")! El objeto "mymaintopic" se crea para la duración de todos los procesos de rutina y sus propiedades se definen. El script hace uso de dos de estas propiedades dentro de la rutina: la propiedad "url" y la propiedad "caption". La primera contiene una referencia relativa a la página HTML en cuestión, y la segunda el título de la página correspondiente. Esta rutina muestra por tanto una lista enlazada de todas las páginas del primer nivel de navegación

La siguiente rutina se ocupará del segundo nivel de navegación:

```
<cc: loop type="topic" obj="mytopic" maintopic="&page.maintopic.number">
  <a href="<cc: print value="&mytopic.url">"><cc: print value="&mytopic.caption"></a>
</cc: loop><br>
```

En este caso se ha dado la orden de leer en la propiedad "&page.maintopic.number". En este objeto se puede leer el número del primer nivel de navegación, que está asignado a la página actual. Alrededor de la rutina está fijado un nuevo elemento estructural, la condición:

```
<cc: if cond="&page.topiccount gt 0">
  [...]
</cc: if>
```

Esta condición comprueba si la página actual posee elementos del segundo nivel de navegación. Para ello se utiliza dentro del atributo "cond" una expresión. 'web al día' soporta un versátil lenguaje de expresiones. A continuación presentamos una pequeña muestra de posibles expresiones:

Elemento	Significado	Ejemplo	Resultado
&	Referencia a una variable	&page.content.caption	Mi página personal
		&link.url	../index.html
+	Suma	1+2	3
		11+&page.topiccount*	17
		'Super'+ 'puesto'	Superpuesto
-	Resta	8-2	6
		-3	-3
*	Multiplicación	8*2	16
		2*2*2	8
/	División	49/7	7
		44/0	Error
	Modulo (Resto de una división)	2/2	0
		3/2	1
()	Expresión entre paréntesis	2*(4+1)	10
		10*(4*(2+2))	160
'	Literal	'Mesa'	Mesa
		'1+1'	1+1
=	Comprobación de igualdad	1=1	1
		1=2	
		6=&page.topiccount*	1
		a='a'	1
gt	Mayor que	2 gt 1	1
		z' gt 'a'	1
		3 gt &page.topiccount*	

Elemento	Significado	Ejemplo	Resultado
lt	Menor que	1 lt 6	1
		6 lt 2	
and	Y lógico	1 and 1	1
		1 and 0	
or	O lógico	1 or 0	1
		0 or 0	

La estructura del siguiente bucle del tipo "subtopic" para los elementos de navegación del tercer nivel, sería después prácticamente intrascendente:

```
<cc: if cond=" &page.subtopiccount gt 0">
  <cc: loop type="subtopic" obj="mysubtopic" maintopic=" &page.maintopic.number"
  topic=" &page.topic.number">
    <a href=" <cc: print val ue=" &mysubtopic.url ">"><cc: print val ue=" &mysubtopic.caption"></a>
  </cc: loop><br>
</cc: if>
```

La única peculiaridad estaría en el nuevo atributo "topic", cuyo significado debe entenderse análogo al atributo "maintopic", y al que lógicamente también se le incluye una referencia a "page.topic.number". Dentro de las rutinas "topic", "maintopic" y "subtopic" se puede recurrir, entre otros, a los siguientes objetos:

Propiedad	Significado
.number	Indica el número del proceso de rutina.
.letter	Indica la letra (a-z) del proceso de rutina.
.caption	Indica la descripción del MainTopic.
.url	Indica la URL del MainTopic.
.abstract	Indica el breve texto del MainTopic.

Después se encuentra en el código fuente la siguiente línea:

```
<cc: print val ue=" &page.content.text">
```

Aquí se edita el contenido real de la página, tal y como está definido en el archivo "page.ccm1".

Al final del archivo se encuentra un cuarto bucle que muestra una lista de todas las subpáginas de la página web, las llamadas "Common Pages". Estas páginas son apropiadas para elementos generales como los créditos, contacto, etc...

```
<cc: loop type="commonpages" obj="mycommonpage">
  <a href=" <cc: print val ue=" &mycommonpage.url ">"><cc: print val ue=" &mycommonpage.caption"></a>
</cc: loop><br>
```

No es necesario hablar mucho acerca de esta rutina. Su estructura corresponde a las rutinas Topic.

El archivo "page.ccm1"

Este archivo sirve para dar formato al contenido real de la página. Se entiende que una página es una serie de varios párrafos con distintos formatos. Por consiguiente, este archivo consta principalmente de una rutina de párrafo.

Con la línea

```
<h1><cc: print value="&page.content.caption"></h1>
```

se muestra el título de la página actual. Posteriormente sigue la rutina de párrafo ya comentada:

```
<cc: loop type="paragraph" obj="para">
  [ . . ]
</cc: loop>
```

En el objeto "para" se presentan las siguientes propiedades:

Propiedad	Significado
.number	Indica el número del proceso de rutina.
.letter	Indica la letra (a-z) del proceso de rutina.
.text	Indica el texto del párrafo.
.caption	Indica el título del párrafo.
.sidepicture.src	Indica el archivo de origen de la imagen en la columna del margen.
.sidepicture.caption	Indica el título de la imagen en la columna del margen.
.sidepicture.text	Indica la descripción de la imagen en la columna del margen.
.sidepicture.uid	Indica el Unique Identifier (UID) de la imagen de la columna del margen.

Primero se edita el título del párrafo:

```
<h2><cc: print value="&para.caption"></h2>
```

Después se pregunta si está definido un gráfico dispuesto en el margen (sidepicture). Esto se puede reconocer si la propiedad UID está aplicada. La propiedad UID indica el "unique identifier" de un gráfico, un número único universal que caracteriza de manera inconfundible al gráfico correspondiente. Si la propiedad UID está aplicada, hay un gráfico lateral disponible. Con las siguientes líneas

```
<cc: picture uid="&para.sidepicture.uid" suffix="_160" width="160" height="160" obj="parapic">
<cc: picture uid="&para.sidepicture.uid" obj="bigpic">
```

se crean dos versiones de la imagen. La primera línea crea una pequeña versión, mientras que la segunda crea una versión del tamaño original. La imagen es accesible mediante las referencias de objeto "parapic" y "bigpic". Con ayuda de los atributos "width" y "height" se limita, para la imagen de pequeño tamaño, la altura y el ancho máximos de la imagen. Además se adjunta el sufijo "_160" al nombre del archivo y a la denominación del objeto. El objeto por tanto se llama "parapic_160". En los objetos de imagen se presentan las siguientes propiedades:

Propiedad	Significado
.url	Indica la URL del objeto de imagen.
.width	Indica el ancho del objeto de imagen.
.height	Indica la altura del objeto de imagen.

Con la función <cc:printpicture> se escribe una completa referencia de imagen HTML, incluyendo las propiedades "url", "width" y "height". Mediante la función <cc:printpictureurl> se escribe solamente una referencia relativa a la imagen.

Con las líneas

```
<a href="<cc: printpicture url obj="bigpic">" >
  <cc: printpicture obj="parapic_160">
</a>
```

se añade la imagen de menor tamaño y se crea un hipervínculo a la imagen de mayor tamaño.

A continuación se comprueba si la imagen posee un título o no.

```
<cc: if cond="&para. si picture. caption">
```

Si es así se mostrará éste junto con el texto de la imagen:

```
<h3>
<cc: print value="&para. si picture. caption">
</h3>
<p>
<cc: print value="&para. si picture. text">
</p>
```

Después se editará el texto en cuestión del párrafo:

```
<p>
<cc: print value="&para. text">
</p>
```

Al pie de la página se necesitarán enlaces a las páginas siguientes en caso de que la página se esté utilizando dentro de una secuencia de subpáginas. Esto se llevará a cabo mediante las propiedades "previousurl", "nexturl" y "previouscaption", "nextcaption".

```
<cc: if cond="&page. content. previousurl ">
  <p>&lt; &lt; <a href="<cc: print value="&page. content. previousurl ">" >
    <cc: print value="&page. content. previouscaption">
  </a>
</p>
</cc: if>

<cc: if cond="&page. content. nexturl ">
  <p>&gt; &gt; <a href="<cc: print value="&page. content. nexturl ">" >
    <cc: print value="&page. content. nextcaption">
  </a>
</p>
</cc: if>
```

El archivo "paragraph_text.ccml"

La representación real de los párrafos tiene lugar en los ficheros "paragraph*.ccml". El más fácil es el archivo "paragraph_text.ccml". Aquí se edita simplemente el texto del párrafo, que está disponible en el objeto "paragraph" dentro de la propiedad "text".

```
<p><cc: print value="&paragraph. text" </p>
```

Para la elaboración de enlaces del párrafo, se recurrirá al archivo "link.ccml".

El archivo "link.ccml"

En este archivo se diferencian cuatro tipos de enlaces distintos.

Links externos (reconocibles mediante la propiedad "isoffsite") enlazan con contenidos externos a la página y se abren, en este ejemplo, en una nueva ventana.

```
<cc: if cond="&link. isoffsite">
  <a href="<cc: print value="&link. url ">" target="_blank"><cc: print value="&link. caption"></a>
</cc: if>
```

Links internos (reconocibles mediante la propiedad "isonsite"), que se abren por el contrario de manera "normal".

```
<cc: if cond="&link. isonsite">
  <a href="<cc: print value="&link. url ">"><cc: print value="&link. caption"></a>
</cc: if>
```

Para las descargas (reconocibles mediante "isdownload") se representa el tamaño. La etiqueta <cc:download> coloca el archivo en cuestión en el directorio "downloads". El objeto de descarga soporta además dos propiedades:

Propiedad	Significado
.url	Indica la URL del objeto de descarga.
.size	Indica el tamaño del archivo de descarga en Kbytes.

```
<cc:if cond="&link.isdownload">
  <cc:download uid="&link.uid" obj="mydownload">
    <a href="<cc:print value="&mydownload.url">"><cc:print value="&link.caption"></a> [<cc:print value="&mydownload.size"> KB]
  </cc:if>
```

Mientras, para las imágenes, primero se crea un objeto de imagen en su tamaño original y posteriormente se enlaza.

```
<cc:if cond="&link.ispicture">
  <cc:picture uid="&link.uid" obj="mypicture">
    <a href="<cc:print value="&mypicture.url">"><cc:print value="&link.caption"></a>
  </cc:if>
```

Aquí presentamos una lista de todas las propiedades del objeto de enlace:

Propiedad	Significado
.url	La URL del enlace.
.caption	El título del enlace.
.isonsite	El enlace está dentro de una página.
.isoffsite	El enlace irá fuera de la página.
.isdownload	El enlace acude a una archivo.
.src	Indica el archivo de origen de la descarga.

El archivo "paragraph_list.ccm1"

Las enumeraciones constan de varias líneas que se editan mediante una rutina del tipo "list":

```
<ul >
<cc:loop type="list" obj="mylist">
  <li><cc:print value="&mylist.text"></li >
</cc:loop>
</ul >
```

Dentro del objeto "mylist" se presentan las siguientes propiedades:

Propiedad	Significado
.number	Indica el número del proceso de rutina.
.letter	Indica la letra (a-z) del proceso de rutina.
.text	Indica el texto de la entrada de la lista.

El archivo "paragraph_picture.ccm1"

Las imágenes constan de una imagen de tamaño medio y enlazan con su respectiva imagen del tamaño original. La manera de proceder es similar a la de las imágenes situadas en el margen de los párrafos. Primero se crean objetos de imagen:

```
<cc: picture uid="&paragraph.picture.uid" suffix="_270" obj="bigpic" width="270" height="400">
<cc: picture uid="&paragraph.picture.uid" obj="hugepic">
```

Posteriormente se representará la imagen mediana que enlaza a la grande:

```
<a href="<cc: printpictureurl obj="hugepic">"><cc: printpicture obj="bigpic_270"></a>
```

Después se procede a la posible inserción de un subtítulo en la imagen:

```
<cc: if cond="&paragraph.picture.caption">
  <h3><cc: print value="&paragraph.picture.caption"></h3>
  <p><cc: print value="&paragraph.picture.text"></p>
</cc: if>
```

En el párrafo de imagen se pueden presentar las siguientes propiedades de objeto:

Propiedad	Significado
.picture.uid	Indica el UID del archivo de origen de la imagen.
.picture.text	Indica el texto de la imagen.
.picture.caption	Indica el título de la imagen.

El archivo "paragraph_table.ccm1"

En este archivo se editarán tablas a cuyo contenido se accede mediante dos bucles intrincados: A través de las rutinas de columnas y líneas. En realidad, la edición de una tabla completa es intrascendental: Se utilizan los tipos de rutinas "row" y "col", dando en el bucle "col" el atributo "row". De esta forma queda claro de qué variable de objeto "col" se debe tomar la línea actual.

```
<table>
<cc: loop type="row" obj="myrow">
  <tr>
    <cc: loop type="col" obj="mycol" row="&myrow.number">
      <td>
        <cc: print value="&mycol.text">
      </td>
    </cc: loop>
  </tr>
</cc: loop>
</table>
```

A continuación presentamos las propiedades de que disponen las rutinas "col" y "row":

Propiedad (row)	Significado
.number	Indica el número de proceso de rutina.
.letter	Indica la letra (a-z) del proceso de rutina.
Propiedad (col)	Significado
.number	Indica el número de proceso de rutina.
.letter	Indica la letra (a-z) del proceso de rutina.
.text	Indica el texto de la celda de la tabla.
.picture	Indica el UID de una imagen de celda.

La propiedad "picture" permite que celdas de una tabla puedan contener imágenes en lugar de texto. Éstas están disponibles dentro del UID. A continuación, un ejemplo ilustrativo:

```
<cc: i f cond="&mycol . pi ctur e">
  <cc: pi ctur e ui d="&mycol . pi ctur e" obj =" tabl epi c" wi dth="600 / &paragraph. tabl e. col count"
  suffi x=" col s' +&paragraph. tabl e. col count">
    <cc: pi ctur e ui d="&mycol . pi ctur e" obj =" bi gpi c">
      <a href="<cc: pri ntpi ctur eurl obj =" bi gpi c">"><cc: pri ntpi ctur e obj =" tabl epi c"
  suffi x=" col s' +&paragraph. tabl e. col count"></a>
    <cc: el se>
      <cc: pri nt val ue="&mycol . text">
</cc: i f>
```

Aquí también se crean dos imágenes, calculándose el ancho de la imagen más pequeña a partir del ancho de la tabla dividido por el número de columnas. Casi con toda seguridad también se añade al nombre del archivo un sufijo correspondiente para evitar duplicaciones.

A continuación presentamos dos propiedades del objeto "paragraph" que pueden ser de utilidad:

Propiedad	Significado
.table.rowcount	Indica el número de líneas de la tabla.
.table.colcount	Indica el número de columnas de la tabla.

El archivo "paragraph_form.ccml"

En este archivo se define el diseño de los formularios. Un formulario consiste en una serie de campos de formulario que son elaborados en un bucle. Pero también el objeto de formulario por sí mismo dispone de una serie de propiedades globales:

Propiedad	Significado
.form.rowcount	Indica el número de líneas del actual formulario.
.form.action	Indica la URL para la acción del formulario.
.form.method	Indica el método de transferencia del formulario.

En este ejemplo se evalúan estas propiedades tal y como sigue:

```
<form name="sendform" method="<cc: pri nt val ue="&paragraph. form. method">" acti on="<cc: pri nt
val ue="&paragraph. form. acti on">" encty pe="<cc: pri nt val ue="&paragraph. form. encty pe">">
```

Posteriormente se procesan todos los campos con una rutina del tipo "form". Se presentan las siguientes propiedades en el objeto de bucle:

Propiedad	Significado
.number	Indica el número del proceso de rutina.
.letter	Indica la letra (a-z) del proceso de rutina.
.caption	Indica el título del formulario actual.
.value	Indica el valor estándar del elemento actual.
.name	Indica el nombre del elemento actual.
.checked	Indica si el elemento actual debe ser seleccionado de forma estándar.
.required	Indica si se trata, en el elemento actual, de un campo obligatorio.
.validate	Indica si el contenido del campo actual debe ser validado como dirección de E-mail.
.istext	Indica que el elemento actual es un campo de entrada del tipo TEXT.
.ishidden	Indica que el elemento actual es un campo de entrada oculto del tipo HIDDEN.

Propiedad	Significado
.istextarea	Indica que el elemento actual es un campo de entrada de mayor tamaño del tipo TEXTAREA.
.ischeckbox	Indica que el elemento actual es una casilla de verificación (CHECKBOX).
.isradio	Indica que el elemento actual es un botón de opción (RADIO Button).
.isheader	Indica que el elemento actual es un título (Header).
.text	Indica el comentario acerca del objeto actual.

El tipo de campo se puede analizar con ayuda de propiedades tales como ".istext", ".isradio", etc.

En campos ocultos se evalúan las propiedades "name" y "value":

```
<cc: i f cond=" &myform. i shi dden" >
  <i nput type="hi dden" name=" <cc: pri nt val ue=" &myform. name" >" val ue=" <cc: pri nt
val ue=" &myform. val ue" >" >
  </cc: i f>
```

En campos de título, se evalúan las propiedades "caption" y "text":

```
<cc: i f cond=" &myform. i sheader" >
  <h3><cc: pri nt val ue=" &myform. capti on" ></h3>
  <p><cc: pri nt val ue=" &myform. text" ></p>
  </cc: i f>
```

En campos de entrada de texto, son de interés los campos "caption", "name" y "value":

```
<cc: i f cond=" &myform. i stext" >
  <cc: pri nt val ue=" &myform. capti on" >
  <i nput type=" text" name=" <cc: pri nt val ue=" &myform. name" >" val ue=" <cc: pri nt
val ue=" &myform. val ue" >" >
  <p><cc: pri nt val ue=" &myform. text" ></p>
  </cc: i f>
```

Lo mismo ocurre con áreas de texto:

```
<cc: i f cond=" &myform. i stextarea" >
  <cc: pri nt val ue=" &myform. capti on" >
  <textarea name=" <cc: pri nt val ue=" &myform. name" >" ><cc: pri nt
val ue=" &myform. val ue" ></textarea>
  <p><cc: pri nt val ue=" &myform. text" ></p>
  </cc: i f>
```

La propiedad "checked" es importante igualmente en el caso de casillas de verificación:

```
<cc: i f cond=" &myform. i scheckbox" >
  <i nput type="checkbox" name=" <cc: pri nt val ue=" &myform. name" >" val ue=" <cc: pri nt
val ue=" &myform. val ue" >" <cc: i f cond=" &myform. checked" >checked</cc: i f>> <cc: pri nt
val ue=" &myform. capti on" >
  <p><cc: pri nt val ue=" &myform. text" ></p>
  </cc: i f>
```

Lo mismo sucede para botones de opción:

```
<cc: i f cond=" &myform. i sradi o" >
  <i nput type="radi o" name=" <cc: pri nt val ue=" &myform. name" >" val ue=" <cc: pri nt
val ue=" &myform. val ue" >" <cc: i f cond=" &myform. checked" >checked</cc: i f>> <cc: pri nt
val ue=" &myform. capti on" >
  <p><cc: pri nt val ue=" &myform. text" ></p>
  </cc: i f>
```

El archivo "paragraph_movie.ccm1"

En el caso de párrafos de vídeo, se disponen de las siguientes propiedades:

Propiedad	Significado
.movie.uid	Indica el UID de la película.
.movie.text	Indica el texto de la película.
.movie.caption	Indica el título de la película.
.movie.width	Indica el ancho de la película.
.movie.height	Indica la altura de la película.
.movie.isflash	Indica que se trata de una animación Flash.
.movie.isquicktime	Indica que se trata una película Apple QuickTime.
.movie.isrealmedia	Indica que se trata de un Clip de RealMedia.
.movie.iswindowsmedia	Indica que se trata de un Clip de Windows Media.

Posteriormente se ha de realizar una diferenciación para los cuatro tipos distintos de vídeos soportados, creándose el objeto de película en cuestión con la etiqueta "Download".

Para Flash:

```
<cc: if cond=" &paragraph.movi e. i sfl ash" >
  <cc: downl oad ui d=" &paragraph.movi e. ui d" obj ="mymovi e" >
  <obj ect cl assi d="cl si d: D27CDB6E-AE6D-11cf-96B8-444553540000"
  codebase="ht tp: //downl oad. macromedi a. com/pub/shockwave/cabs/
  fl ash/swfl ash. cab#versi on=4, 0, 2, 0"
  id="home"
  wi dth=" <cc: pri nt val ue=" &paragraph.movi e. wi dth" >"
  hei ght=" <cc: pri nt val ue=" &paragraph.movi e. hei ght" >" >
  <param name="src" val ue=" <cc: pri nt val ue=" &mymovi e. url " >" >
  <param name="movi e" val ue=" <cc: pri nt val ue=" &mymovi e. url " >" >
  <param name="qual i ty" val ue="hi gh" >
  <embed src=" <cc: pri nt val ue=" &mymovi e. url " >"
  qual i ty="hi gh"
  wi dth=" <cc: pri nt val ue=" &paragraph.movi e. wi dth" >"
  hei ght=" <cc: pri nt val ue=" &paragraph.movi e. hei ght" >"
  type="appl i cati on/x-shockwave-fl ash"
  pl ugi nspace="ht tp: //www. macromedi a. com/shockwave/downl oad/i ndex. cgi ?
  P1_Prod_Versi on=ShockwaveFl ash" >
  </obj ect>
</cc: i f>
```

Para Windows Media:

```
<cc: i f cond=" &paragraph.movi e. i swi ndowsmedi a" >
  <cc: downl oad ui d=" &paragraph.movi e. ui d" obj ="mymovi e" >
  <obj ect id="Medi aPl ayer"
  cl assi d="CLSI D: 22d6f312-b0f6-11d0-94ab-0080c74c7e95"
  codebase="ht tp: //acti vex. mi cros oft. com/acti vex/control s/mpl ayer/
  en/nsmp2i nf. cab#Versi on=6, 4, 7, 1112"
  standby="Loadi ng Mi cros oft Wi ndows Medi a Pl ayer components. . ."
  type="appl i cati on/x-ol eobj ect"
  wi dth=" <cc: pri nt val ue=" &paragraph.movi e. wi dth" >"
  hei ght=" <cc: pri nt val ue=" &paragraph.movi e. hei ght" >" >

  <param name="fi l e name" val ue=" <cc: pri nt val ue=" &mymovi e. url " >" >
  <param name="Ani mati onatStart" val ue=" true" >
  <param name="AutoStart" val ue=" true" >
  <param name="ShowControl s" val ue=" 1" >
  <embed type="appl i cati on/x-mpl ayer2"
  pl ugi nspace="ht tp: //www. mi cros oft. com/i sapi /redi r. dl l ?prd=
  wi ndows&sbp=medi aPl ayer&ar=Medi a&sba=Pl ugi n&"
  src=" <cc: pri nt val ue=" &mymovi e. url " >"
  name="Medi aPl ayer"
  showcontrol s=" 1"
  AutoStart=" 1"
  wi dth=" <cc: pri nt val ue=" &paragraph.movi e. wi dth" >"
  hei ght=" <cc: pri nt val ue=" &paragraph.movi e. hei ght" >"
  </embed>
```

```
</object>
</cc:if>
```

Para QuickTime:

```
<cc:if cond="&paragraph.mov.e.iquicktime">
  <cc:download uid="&paragraph.mov.e.uid" obj="mymov.e">
    <embed src="<cc:print val ue="&mymov.e.url">"
      pluginspage="http://www.apple.com/quicktime/download/"
      type="video/QuickTime"
      cache="true"
      width="<cc:print val ue="&paragraph.mov.e.widht">"
      height="<cc:print val ue="&paragraph.mov.e.height">"
      name="movieview"
      autoplay="true"
      bgcolor="<cc:print val ue="&usercolor.dark">"
      controller="true"
      loop="false">
    </embed>
  </cc:if>
```

Para RealMedia:

```
<cc:if cond="&paragraph.mov.e.isrealmedia">
  <cc:download uid="&paragraph.mov.e.uid" obj="mymov.e">
    <object id="realvideo"
      classid="clsid:CFCDAA03-8BE4-11cf-B84B-0020AFBCCFA"
      width="<cc:print val ue="&paragraph.mov.e.widht">"
      height="<cc:print val ue="&paragraph.mov.e.height">">
      <param name="controls" val ue="imagemindow">
      <param name="autostart" val ue="true">
      <param name="console" val ue="_master">
      <param name="maintainaspect" val ue="true">
      <param name="nojava" val ue="true">
      <param name="src" val ue="<cc:print val ue="&mymov.e.url">">
      <embed nojava="true"
        console="one"
        width="<cc:print val ue="&paragraph.mov.e.widht">"
        height="<cc:print val ue="&paragraph.mov.e.height">"
        controls="ImageWindow"
        src="<cc:print val ue="&mymov.e.url">"
        type="audio/x-pn-realaudio-plugin"
        autostart="true"
        pluginspage="http://www.real.com/player/index.html">
      </embed>
    </object>
    <object id="realcontrol"
      classid="clsid:CFCDAA03-8BE4-11cf-B84B-0020AFBCCFA"
      width="<cc:print val ue="&paragraph.mov.e.widht">"
      height="36">
      <param name="controls" val ue="Control Panel">
      <param name="console" val ue="_master">
      <param name="maintainaspect" val ue="true">
      <param name="nojava" val ue="true">
      <param name="src" val ue="<cc:print val ue="&mymov.e.url">">
      <embed nojava="true"
        console="one"
        width="<cc:print val ue="&paragraph.mov.e.widht">"
        height="36"
        controls="Control Panel"
        src="<cc:print val ue="&mymov.e.url">"
        type="audio/x-pn-realaudio-plugin"
        autostart="true"
        pluginspage="http://www.real.com/player/index.html">
      </embed>
    </object>
  </cc:if>
```

Posteriormente son editados además el título y texto de la película:

```
<h3><cc:print val ue="&paragraph.mov.e.caption"></h3>
<p><cc:print val ue="&paragraph.mov.e.text"></p>
```

El archivo "paragraph_teaser.ccml"

Los titulares son párrafos dentro de los cuales pueden estar hasta tres listas creadas automáticamente una junto a la otra. Por lo demás, la estructura del archivo "paragraph_teaser.ccml" es bastante simple. Sólo consiste principalmente en una única rutina "teaser":

```
<table>
<tr>
<cc:loop type="teaser" obj="myteaser">
  <td align="top">
    <cc:print value="&myteaser.text">
  </td>
</cc:loop>
</tr>
</table>
```

La "tarea" es realizada dentro de los cuatro archivos Teaser que proporcionan información al objeto "myteaser.text". 'web al día' decide automáticamente cuáles de los cuatro archivos son necesarios mediante las configuraciones del usuario.

El archivo "teaser_caption.ccml"

En este archivo se editan titulares que consisten solamente en títulos. Tenga en cuenta que desde aquí se puede acceder a todas las propiedades del objeto de bucle "myteaser" a partir del archivo "paragraph_teaser".

Propiedad	Significado
.number	Indica el número del proceso de rutina.
.letter	Indica la letra (a-z) del proceso de rutina.
.text	Indica el texto obtenido del elemento de titular actual.
.caption	Indica el título del elemento de titular actual.

Posteriormente se deben procesar solamente todos los elementos del titular en un bucle "teaserelement":

```
<cc:loop type="teaserelement" obj="myteaserelement">
  <h4><a href="<cc:print value="&myteaserelement.url">"><cc:print
  value="&myteaserelement.caption"></a></h4>
</cc:loop>
```

El objeto de bucle del elemento de titular presenta las siguientes propiedades:

Propiedad	Significado
.number	Indica el número del proceso de rutina.
.letter	Indica la letra (a-z) del proceso de rutina.
.caption	Indica el título del elemento de titular actual.
.abstract	Indica el abstracto del elemento de titular actual.
.sidepicture.src	Indica el archivo de origen de la imagen situada en la columna del margen.
.sidepicture.caption	Indica el título de la imagen situada en la columna del margen.
.sidepicture.text	Indica la descripción de la imagen situada en la columna del margen.
.url	Indica la URL del elemento de titular actual.

El archivo "teaser_captionabstract.ccml"

Este archivo se asemeja mucho a los predecesores. Se editan adicionalmente textos breves acerca de los titulares:

```
<cc: loop type="teaserelement" obj="myteaserelement">
  <h4><a class="teaser" href="<cc: print value="&myteaserelement.url ">"><cc: print
value="&myteaserelement.caption"></a></h4>
  <p><cc: print value="&myteaserelement.abstract"></p>
</cc: loop>
```

El archivo "teaser_captionabstractpicture.ccml"

Aquí, adicionalmente, se debe crear y mostrar una imagen previa:

```
<cc: loop type="teaserelement" obj="myteaserelement">
  <cc: if cond="&myteaserelement.screenshot.uid">
    <cc: picture uid="&myteaserelement.screenshot.uid" suffix="_100" width="100"
height="200" obj="teaserpic">
    <a href="<cc: print value="&myteaserelement.url ">"><cc: printpicture
obj="teaserpic_100"></a>
  </cc: if>
  <h4><a href="<cc: print value="&myteaserelement.url ">"><cc: print
value="&myteaserelement.caption"></a></h4>
  <p><cc: print value="&myteaserelement.abstract"></p>
</cc: loop>
```

La manera de proceder es muy similar a la de las imágenes situadas al margen dentro de párrafos.

El archivo "teaser_lookup.ccml"

En este archivo se consideran unas "reglas" ligeramente distintas con respecto a los tres archivos anteriores. Primero, este archivo se procesa 28 veces: Para 27 letras y para entradas numéricas. Pero el desarrollador de <CCML> no debe preocuparse de esto: La rutina "teaserelement" siempre aporta las entradas convenientes.

```
<cc: if cond="&teaser.elementcount gt 0">
  <cc: loop type="teaserelement" obj="myteaserelement">
    <p><a class="teaser" href="<cc: print value="&myteaserelement.url ">"><cc: print
value="&myteaserelement.caption"></a></p>
  </cc: loop>
  <cc: else>
    <p>Keine Einträge gefunden</p>
  </cc: if>
```

Solamente se necesita algo de iniciativa por parte del desarrollador a la hora de estructurar el registro de letras, con el cual el usuario pueda cambiar a una página de registro. Aquí presentamos una rutina del tipo "lookup" que puede llevarse a la práctica:

```
<cc: loop type="lookup" obj="mylookup">
  <cc: if cond="&mylookup.iscurrentletter">
    <cc: print value="&mylookup.caption">
    <cc: else>
      <a href="<cc: print value="&mylookup.url ">"><cc: print value="&mylookup.caption"></a>
    </cc: if>
  </cc: loop>
```

Aquí se analiza, con la propiedad "iscurrentletter", si el proceso actual de la rutina corresponde a la letra de la página de registro actual. Además se utiliza el objeto "teaser": Su única propiedad "elementcount" indica cuántas entradas tiene el actual elemento del titular (Teaser). Si la cantidad es cero, aparecerá un mensaje al respecto.

A continuación aportamos una enumeración de todas las propiedades que presenta este bucle:

Propiedad	Significado
.number	Indica el número del proceso de rutina.
.letter	Inicia la letra (a-z) del proceso de rutina.
.caption	Indica la letra de la columna de búsqueda (Lookup).
.url	Indica la URL de la subpágina.
.iscurrentletter	Indica si la letra actual corresponde con la subpágina actual.

El archivo "splash.ccml"

Este archivo define una página de preconexión o Intro que aparece antes de la página en cuestión. Además actúa dependiendo si hay un enlace a la página principal configurado por el usuario o si sólo se debe mostrar un mensaje de "Página en construcción". Por lo demás, el usuario puede definir un intervalo de tiempo, tras el cual se deba redirigir automáticamente a la página de inicio.

Para acceder a la imagen Splash definida por el usuario, se utilizará la etiqueta <cc:userpicture>. Dicha etiqueta soporta los siguientes atributos:

Atributo	Significados
obj	Indica el nombre del objeto.
type	Indica el tipo: "splash" para la pantalla de Intro (Splashscreen) y "logo" para el logo del usuario.

Dentro del archivo "splash.ccml", se presenta el objeto "splash" con las siguientes propiedades:

Propiedad	Significado
.timer	Segundos tras los cuales debe aparecer la página principal.
.forward	Indica si debe mostrarse un enlace a la página principal.
.bgcolor	Indica el color de fondo de la pantalla de Intro.

Con la propiedad "url" del objeto "home" se puede ahora enlazar a la página de inicio.