

CAPÍTULO 3 ELEMENTOS DE FORMULARIO

HTML5 hace que el desarrollo de formularios sea mucho más sencillo. Se han añadido dos nuevos métodos que pueden ser utilizados en la acción del formulario (`update` y `delete`), pero lo más interesante son los nuevos tipos de `input` y elementos de formulario que mejoran la experiencia del usuario y facilitan el desarrollo de los formularios. Estos nuevos elementos añaden en algunos casos, validaciones propias de sus datos, por lo que ya no es necesario JavaScript para realizar este proceso.

3.1 NUEVOS TIPOS DE INPUT

La especificación de HTML5 define 12 nuevos tipos de `input` que podemos utilizar en nuestros formularios. Esta especificación no define cómo deben mostrarse los nuevos tipos en los navegadores, ni los campos ni las validaciones. De hecho, y gracias a cómo está especificado HTML, los navegadores que no *comprendan* los nuevos tipos de entrada, mostrarán un campo de texto tradicional, por lo que la compatibilidad con navegadores antiguos está garantizada.

3.1.1 TIPO EMAIL

El nuevo tipo `<input type="email">` indica al navegador que no debe permitir que se envíe el formulario si el usuario no ha introducido una dirección de email *válida*, pero no comprueba si la dirección existe o no, sólo si el formato es válido. Como ocurre con el resto de campos de entrada, puede enviar este campo vacío a menos que se indique que es obligatorio.

El atributo `multiple` indica que el valor de este campo, puede ser una lista de emails válidos, separados por comas.

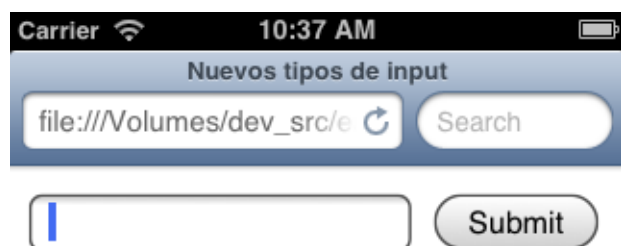




Figura 3.1 Campo de tipo `email` en un dispositivo iOS

3.1.2 TIPO URL

El nuevo tipo `<input type="url">` indica al navegador que no debe permitir que se envíe el formulario si el usuario no ha introducido una URL correcta. Algunos navegadores ofrecen ayudas al usuario, como Opera que añade el prefijo `http://` a la URL si el usuario no lo ha introducido. Una URL no tiene que ser necesariamente una dirección web, sino que es posible utilizar cualquier formato de URI válido, como por ejemplo `tel:555123456`.

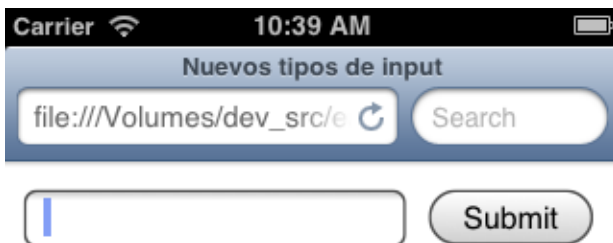


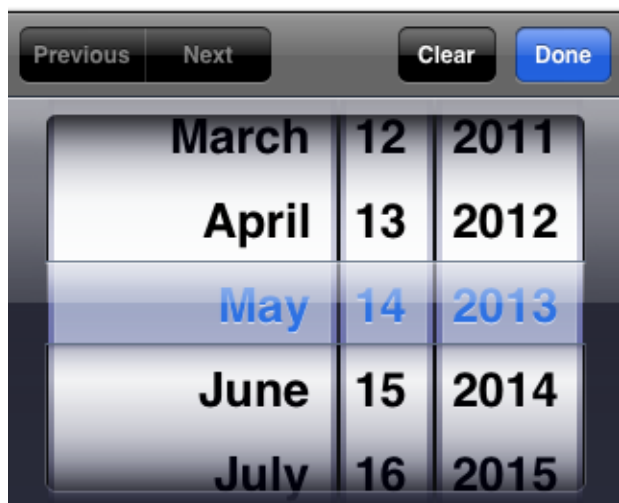
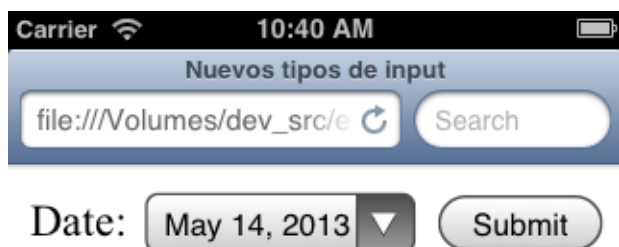
Figura 3.2 Campo de tipo `url` en un dispositivo iOS

Figura 3.2 Campo de tipo `url` en un dispositivo iOS

3.1.3 TIPO DATE

El nuevo tipo `<input type="date">` es de los más esperados y útiles. En muchos de los sitios web es normal disponer de campos específicos de fecha, donde el usuario debe especificar fechas (para un concierto, vuelo, reserva de hotel, etc). Al existir tantos formatos de fecha diferentes (DD-MM-YYYY o MM-DD-YYYY o YYYY-MM-DD), esto puede suponer un inconveniente para los desarrolladores o los propios usuarios.

Este nuevo tipo de campo resuelve estos problemas, ya que es el navegador el que proporciona la interfaz de usuario para el calendario, e independientemente del formato en el que se muestre, los datos que se envían al servidor cumplen la **norma ISO** para el formato de fechas.

Figura 3.3 Campo de tipo `date` en un dispositivo iOS

3.1.4 TIPO TIME

El nuevo tipo `<input type="time">` permite introducir una hora en formato 24h, y validarlo. De nuevo, es el navegador el encargado de mostrar la interfaz de usuario correspondiente: puede ser un simple campo donde es posible introducir la hora y los minutos, o mostrar algo más complejo como un reloj de agujas.



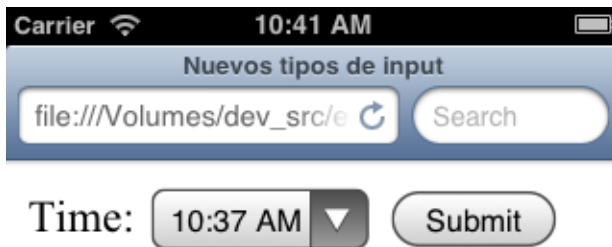


Figura 3.4 Campo de tipo `time` en un dispositivo iOS

3.1.5 TIPO DATETIME

Este nuevo tipo de campo es la combinación de los tipos `date` y `time`, por lo que se valida tanto la fecha como la hora introducida.

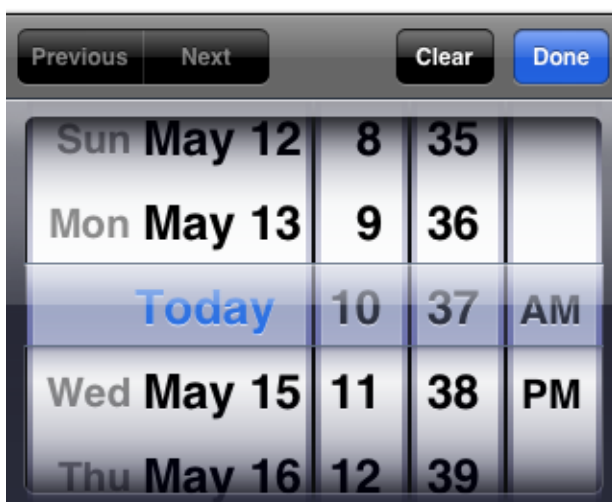
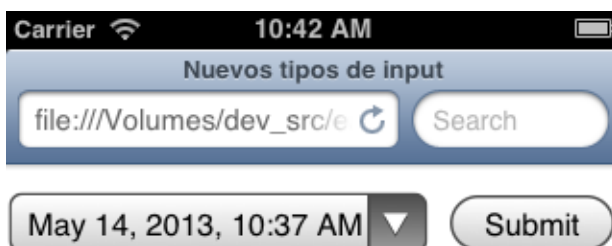




Figura 3.5 Campo de tipo `datetime` en un dispositivo iOS

3.1.6 TIPO MONTH

El nuevo tipo `<input type="month">` permite la selección de un mes en concreto. La representación interna del mes es un valor entre 1 y 12, pero de nuevo queda en manos del navegador la manera de mostrarlo al usuario, utilizando los nombres de los meses por ejemplo.

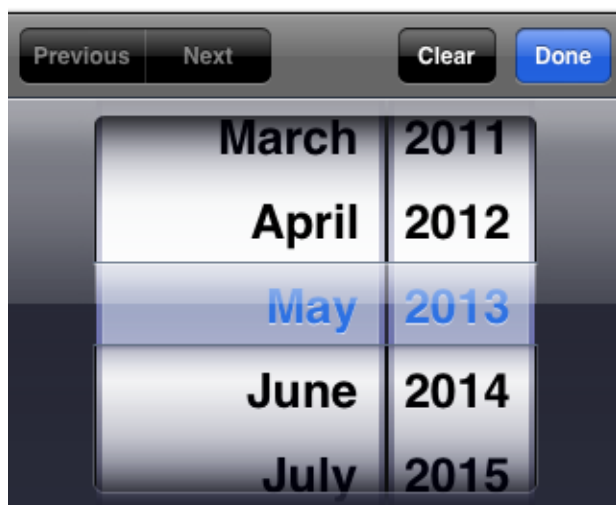
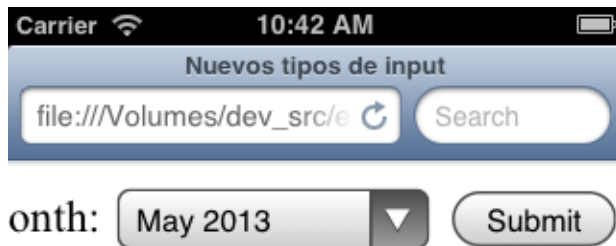
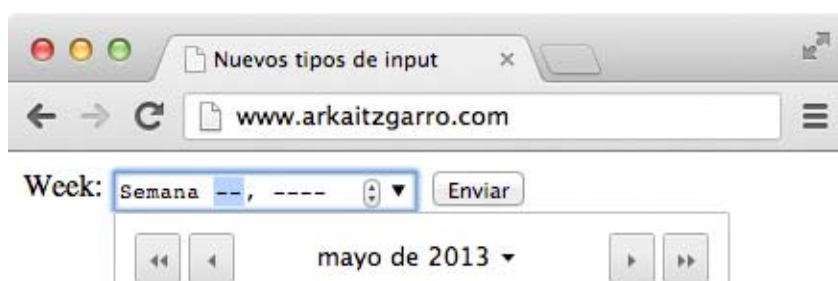


Figura 3.6 Campo de tipo `month` en un dispositivo iOS

3.1.7 TIPO WEEK

El nuevo tipo `<input type="week">` permite la selección de una semana del año concreta. La representación interna del mes es un valor entre 1 y 53, pero de nuevo queda en manos del navegador la manera de mostrarlo al usuario. La representación interna de la semana 7, por ejemplo, es la siguiente: 2013-W07.



WEEK: Semana --, ----

mayo de 2013

Semana	lun	mar	mié	jue	vie	sáb	dom
18	29	30	1	2	3	4	5
19	6	7	8	9	10	11	12
20	13	14	15	16	17	18	19
21	20	21	22	23	24	25	26
22	27	28	29	30	31	1	2
23	3	4	5	6	7	8	9

Figura 3.7 Campo de tipo week en Google Chrome

3.1.8 TIPO NUMBER

Como es de esperar, el nuevo tipo `<input type="number">` valida la entrada de un tipo de dato numérico. Este tipo de campo encaja perfectamente con los atributos `min`, `max` y `step`, que veremos más adelante.




Figura 3.8 Campo de tipo `number` en un dispositivo iOS

3.1.9 TIPO RANGE

El nuevo tipo `<input type="range">`, muestra un control deslizante en el navegador. Para conseguir un elemento de este estilo, era necesario un

El nuevo tipo `<input type="range">`, muestra un control deslizante en el navegador. Para conseguir un elemento de este estilo, era necesario un gran esfuerzo para combinar imágenes, funcionalidad y accesibilidad, siendo ahora mucho más sencillo. Este tipo de campo encaja de nuevo perfectamente con los atributos `min`, `max` y `step`, que veremos más adelante.

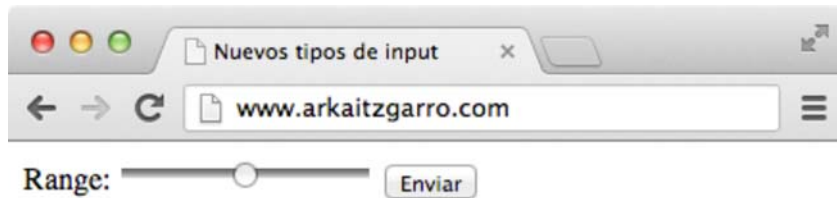


Figura 3.9 Campo de tipo `range` en Google Chrome

3.1.10 TIPO TEL

El nuevo tipo `<input type="tel">`, espera que se proporcione un número de teléfono. No se realiza ninguna validación, ni se obliga a que únicamente se proporcionen caracteres numéricos, ya que un número de teléfono puede representarse de muchas maneras: `+44 (0) 208 123 1234`.

La gran ventaja de este campo es que en dispositivos con un teclado virtual, éste se adaptará para mostrar únicamente los caracteres asociados on números de teléfono.

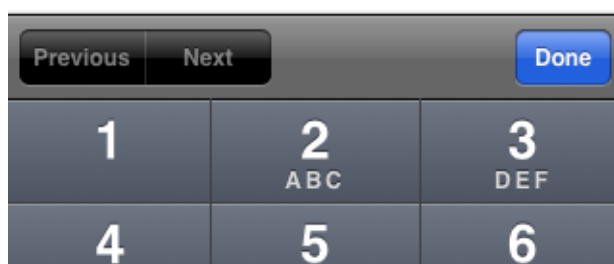
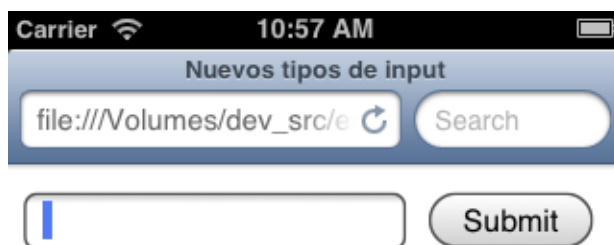




Figura 3.10 Campo de tipo `tel` en un dispositivo iOS

3.1.11 TIPO SEARCH

El nuevo tipo `<input type="search">`, espera que se proporcione un término de búsqueda. La diferencia con un campo de texto normal, es únicamente estética, aunque puede ofrecer alguna funcionalidad extra como un histórico de últimos términos introducidos o una ayuda para el borrado. Por norma general, toma el aspecto de un campo de búsqueda del navegador o sistema operativo.

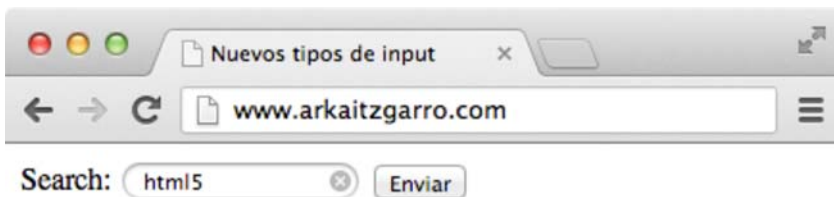


Figura 3.11 Campo de tipo `search` en Google Chrome

3.1.12 TIPO COLOR

El nuevo tipo `<input type="color">`, permite seleccionar un color de una paleta de colores mostrada por el navegador. Esta paleta de colores, coincide, por norma general, con la interfaz de selección de colores del sistema operativo.

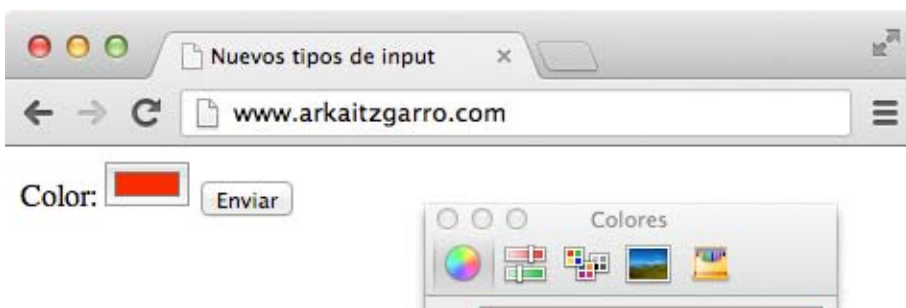




Figura 3.12 Campo de tipo `color` en Google Chrome

3.2 NUEVOS ATRIBUTOS

Al igual que los nuevos tipos de campo, el elemento `input` ha recibido nuevos atributos para definir su comportamiento y restricciones: `autocomplete`, `min`, `max`, `multiple`, `pattern`, `autofocus`, `placeholder`, `required` y `step`. Existe además un nuevo atributo, `list`, que hace referencia a otro elemento, permitiendo crear un nuevo tipo de entrada de datos.

3.2.1 ATRIBUTO `list` Y `<datalist>`

La combinación del atributo `list` y un elemento de tipo `<datalist>` da como resultado un campo de texto, donde el usuario puede introducir cualquier contenido, y las opciones definidas en el `<datalist>` se muestran como una lista desplegable. Hay que tener en cuenta que la lista tiene que estar contenida en un elemento `<datalist>` cuyo `id` coincide con el indicado en el atributo `list`:

```
<input id="form-person-title" type="text" list="mylist">
<datalist id="mylist">
  <option label="Mr" value="Mr">
  <option label="Ms" value="Ms">
  <option label="Prof" value="Mad Professor">
</datalist>
```

En este ejemplo se utiliza un campo de tipo `text`, pero puede ser utilizado igualmente con campos de tipo `url` y `email`.

igualmente con campos de tipo `url` y `email`.

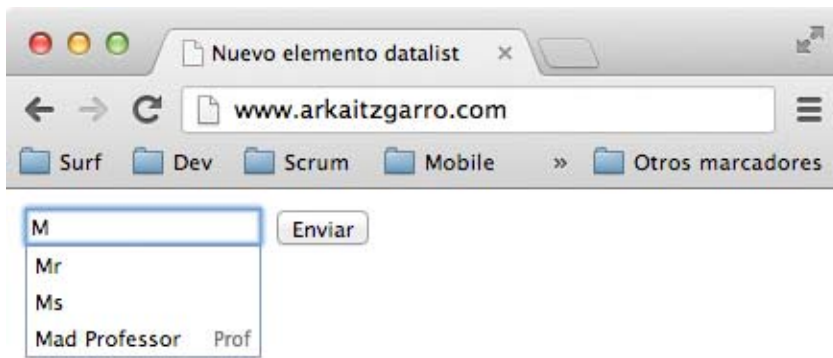


Figura 3.13 Nuevo elemento `datalist`, asociado a un campo de texto

3.2.2 ATRIBUTO AUTOFOCUS

El atributo *booleano* `autofocus` permite definir que control va a tener el foco cuando la página se haya cargado. Hasta ahora, esto se conseguía a través de JavaScript, utilizando el método `.focus()` en un elemento concreto, al cargarse el documento. Ahora es el navegador el encargado de esta tarea, y puede comportarse de manera más inteligente, como no cambiando el foco de un elemento si el usuario ya se encuentra escribiendo en otro campo (éste era un problema común con JavaScript).

Únicamente debe existir un elemento con este atributo definido en el documento. Desde el punto de vista de la usabilidad, hay que utilizar este atributo con cuidado. Hay que utilizarlo únicamente cuando el control que recibe el foco es el elemento principal de la página, como en un buscador, por ejemplo.

3.2.3 ATRIBUTO PLACEHOLDER

Una pequeña mejora en la usabilidad de los formularios, suele ser colocar un pequeño texto de ayuda en algunos campos, de manera *discreta* y que desaparece cuando el usuario introduce algún dato. Como con el resto de elementos, hasta ahora era necesario utilizar JavaScript para realizar esta tarea, pero el atributo `placeholder` resuelve esta tarea.

Es importante recordar que este atributo no sustituye a la etiqueta `<label>`.

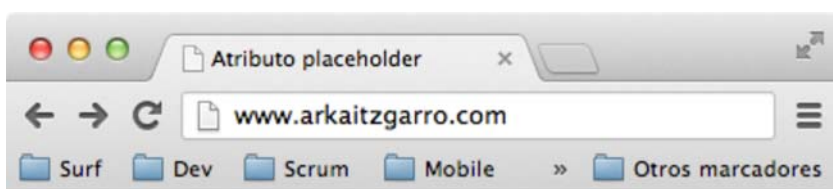




Figura 3.14 Atributo `placeholder` en un campo de búsqueda

3.2.4 ATRIBUTO REQUIRED

Este atributo puede ser utilizado en un `<textarea>` y en la gran mayoría de los elementos `<input>` (excepto en los de tipo `hidden`, `image` o botones como `submit`). Cuando este atributo está presente, el navegador no permite el envío del formulario si el campo en concreto está vacío.

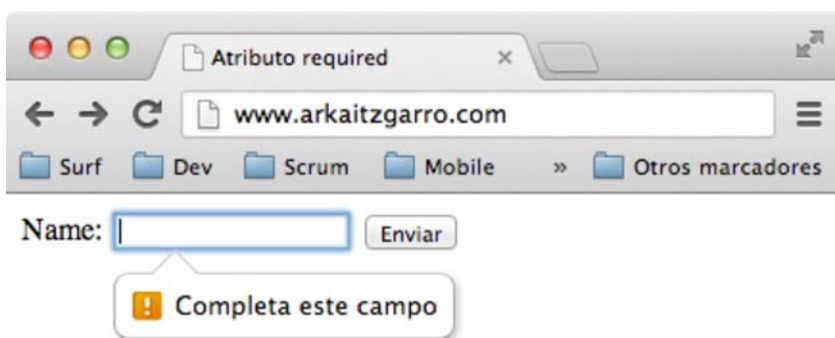


Figura 3.15 Atributo `required` en un campo de texto

3.2.5 ATRIBUTO MULTIPLE

Este atributo permite definir que un campo puede admitir varios valores, como URLs o emails. Un uso muy interesante de este atributo es utilizarlo en conjunto con el campo `<input type="file">`, ya que de esta manera nos permite seleccionar varios ficheros que podemos enviar al servidor al mismo tiempo.

```
<input type="file" multiple="multiple">
```

3.2.6 ATRIBUTO AUTOCOMPLETE

3.2.6 ATRIBUTO AUTOCOMPLETE

Algunos navegadores suelen incluir alguna funcionalidad de autocompletado en algunos campos de formulario. A pesar de haber sido introducido recientemente en el estándar de HTML5, es una característica que lleva mucho tiempo siendo utilizada, concretamente desde la versión 5 de Internet Explorer.

Este atributo permite controlar el comportamiento del autocompletado en los campos de texto del formulario (que por defecto está activado).

3.2.7 ATRIBUTOS MIN Y MAX

Como hemos visto en el campo `<input type="number">`, estos atributos restringen los valores que pueden ser introducidos; no es posible enviar el formulario con un valor menor que `min` o un valor mayor que `max`. También es posible utilizarlo en otro tipo de campos como `date`, para especificar fechas mínimas o máximas.

```
<input type="date" min="2010-01-01" max="2010-12-31">
```



Figura 3.16 Atributos `min` y `max` en un campo numérico

3.2.8 ATRIBUTO STEP

El atributo `step` controla los *pasos* por los que un campo aumenta o disminuye su valor. Si un usuario quiere introducir un porcentaje, pero queremos que sea en múltiplos de 5, lo haríamos de la siguiente manera:

```
<input type="range" min="0" max="100" step="5">
```

3.2.9 ATRIBUTO PATTERN

Algunos de los tipos de `input` que hemos visto anteriormente (`email`, `number`, `url` ...), son realmente expresiones regulares que el navegador

Algunos de los tipos de `input` que hemos visto anteriormente (`email`, `number`, `url`...), son realmente expresiones regulares que el navegador evalúa cuando se introducen datos. El atributo `pattern` nos permite definir una expresión regular que el valor del campo debe cumplir. Por ejemplo, si el usuario debe introducir un número seguido de tres letras mayúsculas, podríamos definir esta expresión regular:

```
<input pattern="[0-9][A-Z]{3}" name="part"
      title="A part number is a digit followed by three uppercase letters.">
```

La especificación indica que la sintaxis de la expresión regular que utilicemos debe coincidir con la utilizada en JavaScript.

3.2.10 ATRIBUTO FORM

Tradicionalmente, los campos de un formulario van incluidos dentro de la correspondiente etiqueta `<form>`. Si por la razón que fuese (principalmente diseño) un elemento tuviese que mostrarse apartado del resto de elementos del formulario, se hacía casi necesario incluir toda la página dentro de una etiqueta `<form>`.

Con HTML5, los elementos que hasta ahora era obligatorio que estuviesen contenidos dentro del elemento `<form>`, pueden colocarse en cualquier lugar de la página, siempre que los relacionemos con el formulario concreto a través del atributo `form` y el `id` de dicho formulario.

```
<form id="foo">
  <input type="text">
  ...
</form>
<textarea form="foo"></textarea>
```

En este caso, el elemento `<textarea>` se encuentra fuera del formulario, pero realmente pertenece a él ya que hemos definido el atributo `form` con el identificador del formulario.

3.3 NUEVOS ELEMENTOS

Además de los nuevos tipos de `input` incluidos en la especificación, se han añadido nuevos elementos de formulario, entre los que se incluyen los siguientes:

3.3.1 <PROGRESS>

El elemento `<progress>` es utilizado para representar un *avance* o *progreso* en la ejecución de una tarea, como puede ser la descarga de un fichero o la ejecución de una tarea compleja. Define los siguientes

progreso en la ejecución de una tarea, como puede ser la descarga de un fichero o la ejecución de una tarea compleja. Define los siguientes atributos:

- `max`: define el trabajo total a realizar por la tarea, la duración de un vídeo... Su valor por defecto es 1.0.
- `value`: el valor actual (en coma flotante) o estado del progreso. Su valor debe ser mayor o igual a 0.0 y menor o igual a 1.0 o el valor especificado en `max`.
- `position`: atributo de sólo lectura que representa la posición actual del elemento `<progress>`. Este valor es igual a `value/max` y -1 si no se puede determinar la posición.

Las unidades son arbitrarias, y no se especifican.

```
<progress value="5" max="20">5</progress>
```

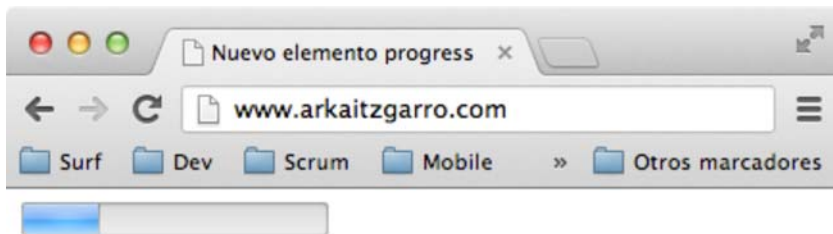


Figura 3.17 Nuevo elemento progress

3.3.2 <METER>

El elemento `<meter>` es muy similar a `<progress>` (de hecho, se discute la necesidad de disponer de dos elementos tan similares). Esta nueva etiqueta se usa para representar escalas de medidas conocidas, como la longitud, masa, peso, uso de disco, entre otras. Define los siguientes atributos:

- `value`: representa el valor actual. Si no se especifica, se toma como valor el primer número que aparece en el contenido del elemento. Su valor por defecto es 0.
- `min`: el mínimo valor permitido. El valor por defecto es 0.
- `max`: el mayor valor permitido. Si no se especifica, su valor por defecto es 1, a menos que el valor mínimo definido sea mayor que 1.

- `max`: el mayor valor permitido. Si no se especifica, su valor por defecto es 1, a menos que el valor mínimo definido sea mayor que 1, en cuyo caso el valor de `max` será igual a `min`.
- `low`: es considerado el límite inferior del rango de valores.
- `max`: es considerado el límite superior del rango de valores.
- `optimum`: representa el valor óptimo del elemento, y se encuentra entre `min` y `max`.

```
<p>Your score is:  
  <meter value="91" min="0" max="100"  
    low="40" high="90" optimum="100">A+</meter>  
</p>
```

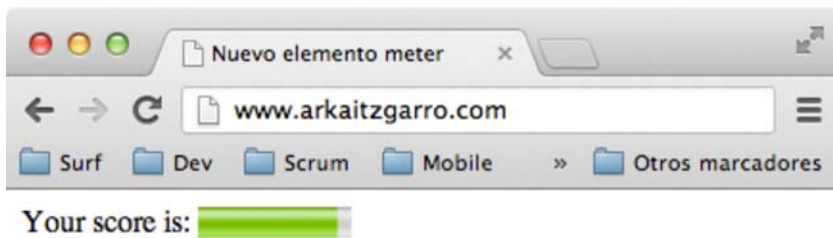


Figura 3.18 Nuevo elemento meter