UT2\_9.- Lenguajes de marcas en entorno web. HTML5

|  |
| --- |
| **RESULTADOS DE APRENDIZAJE ASOCIADOS** |
| 1. Reconoce las características de lenguajes de marcas analizando e interpretando fragmentos de código. 2. Utiliza lenguajes de marcas para la transmisión de información a través de la Web analizando la estructura de los documentos e identificando sus elementos 3. Establece mecanismos de validación para documentos XML utilizando métodos para definir su sintaxis y estructura. |
| **CRITERIOS DE EVALUACIÓN** |
| * De RA1 – desde CEA hasta CEK * De RA2 – desde CEA hasta CEH * De RA3 – desde CEA hasta CEH |

UT2\_9.- Lenguajes de marcas en entorno web. HTML5

Índice de contenido

[**1.-** **Etiquetas y atributos HMTL – Formularios** 3](#_Toc147948104)

[**2.- Input** 5](#_Toc147948105)

[**2.1.- Cuadro de texto (type=”text”)** 6](#_Toc147948106)

[**2.2.- Cuadro de contraseña (type=”password”)** 8](#_Toc147948107)

[**2.3.- Checkbox (type=”checkbox”)** 9](#_Toc147948108)

[**2.4.- Radiobutton (type=”radio”)** 10](#_Toc147948109)

[**2.5.- Botón de envío del formulario (type=”submit”)** 11](#_Toc147948110)

[**2.6.- Botón de reseteo del formulario (type=”reset”)** 11](#_Toc147948111)

[**2.7.- Ficheros adjuntos (type=”file”)** 12](#_Toc147948112)

[**2.8.- Campos ocultos (type=”hidden”)** 12](#_Toc147948113)

[**2.9.- Botón de imagen (type=”image”)** 13](#_Toc147948114)

[**2.10.- Botón genérico (type=”button”)** 14](#_Toc147948115)

[**3.- Fieldset <fieldset> y <legend>** 15](#_Toc147948116)

[**4.- Label <label>** 17](#_Toc147948117)

[**5.- Textarea <textarea>** 19](#_Toc147948118)

[**6.- Select <select>** 20](#_Toc147948119)

[**7.- Parámetros** 24](#_Toc147948120)

[**8.- Formularios en HTML5** 26](#_Toc147948121)

[**8.1.- Nuevos tipos de entrada (Input type)** 27](#_Toc147948122)

[**8.2.- Diferencias semánticas** 28](#_Toc147948123)

[**8.3.- Validación de formularios** 28](#_Toc147948124)

# **1.-** **Etiquetas y atributos HMTL – Formularios**

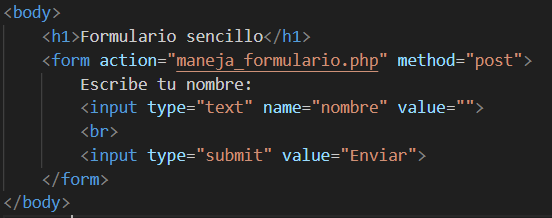
|  |
| --- |
| Manual - Iconos gratis de educaciónManual:  <https://www.w3schools.com/html/html_forms.asp> |

HTML es un lenguaje que tiene como propósito principal estructurar los contenidos de los documentos y páginas web.

HTML también **nos permite crear formularios para que los usuarios puedan interactuar con las aplicaciones web**. Para crear un formulario vamos a necesitar al menos dos etiquetas:

* **<form>**
* **<input>**

Un ejemplo de formulario sencillo:



Cuya salida es:



* La etiqueta **<form>** encierra todos los contenidos del formulario (botones, cuadros de texto, listas desplegables).
* La etiqueta **<input>** permite definir varios tipos diferentes de elementos (botones y cuadros de texto).

Texto

Descripción generada automáticamente

|  |
| --- |
| Información con relleno sólido  La mayoría de los formularios utilizan sólo los atributos *action* y *method*. |

* El atributo ***action*** indica la URL de la aplicación del servidor que se encarga de procesar los datos introducidos por los usuarios. Esta aplicación también se encarga de generar la respuesta que muestra el navegador.
* El atributo ***method*** establece la forma en la que se envían los datos del formulario al servidor. Este atributo hace referencia al método HTTP, por lo que no es algo propio de HTML. Los dos valores que se utilizan en los formularios son ***GET*** y ***POST***. De esta forma, casi todos los formularios incluyen el atributo method="get" o el atributo method="post".

# **2.- Input**

Los elementos de formulario como botones y cuadros de texto también se denominan *"campos de formulario"* y *"controles de formulario"*. La mayoría de los controles se crean con la etiqueta <input>, por lo que su definición formal y su lista de atributos es muy extensa:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

|  |
| --- |
| megáfono1 con relleno sólido  El atributo *type* diferencia los controles que se pueden crear con la etiqueta <input>. |

## **2.1.- Cuadro de texto (type=”text”)**

Se trata del elemento más utilizado en los formularios. En el caso más sencillo, se muestra un cuadro de texto vacío en el que el usuario puede escribir cualquier texto:



La salida sería:



Para los cuadros de texto, **el valor del atributo input es *text***.

* El atributo **name** es el más importante en los campos del formulario.

|  |
| --- |
| Información con relleno sólidoSi un campo no incluye el atributo *name*, sus datos no se envían al servidor. El valor que se indica en el atributo *name* es el nombre que utiliza la aplicación del servidor para obtener el valor de este campo de formulario. |

****

**Pero ¿cómo funciona realmente el envío de información?**

Cuando el usuario pulsa el botón de envío del formulario, el navegador envía los datos a una aplicación del servidor para que procese la información y genere una respuesta adecuada.

En el servidor, la aplicación que procesa los datos debe obtener en primer lugar toda la información introducida por el usuario. Para ello, utiliza el valor del atributo ***name*** para obtener los datos de cada control del formulario.

|  |
| --- |
| Información con relleno sólidoEn el atributo *name* no se deben utilizar caracteres problemáticos en programación (espacios en blanco, acentos y caracteres como ñ o ç) |

* El atributo **value** se emplea para establecer el valor inicial del cuadro de texto.
  + Si se crea un formulario para insertar datos, los cuadros de texto deberían estar vacíos. Por lo tanto, o no se añade el atributo value o se incluye con un valor vacío value="".
  + Si por el contrario se crea un formulario para modificar datos, lo lógico es que se muestren inicialmente los datos guardados en el sistema. En este caso, el atributo ***value*** incluirá el valor que se desea mostrar: <input type="text" name="nombre" value="Juan Pérez" >

Ejemplo:



La salida:



* El atributo **size** permite establecer el tamaño, en caracteres, con el que se muestra el cuadro de texto. Su uso es imprescindible en muchos formularios, en los que algunos campos como la dirección deben mostrar más caracteres de lo normal (<input size="100" ...) y otros campos como el código postal deben mostrar menos caracteres de lo normal (<input size="5"...).

Ejemplo:



Cuya salida es:



Como se puede observar en la imagen anterior, el campo de texto se dibuja más grande.

* El atributo **maxlength** permite establecer el máximo número de caracteres que el usuario puede introducir en un cuadro de texto. Su uso es imprescindible para campos como el código postal, el número de la Seguridad Social y cualquier otro dato con formato predefinido y limitado.
* El atributo **readonly** permite que el usuario pueda ver los contenidos del cuadro de texto, pero no pueda modificarlos.
* El atributo **disabled** deshabilita un cuadro de texto de forma que el usuario no pueda modificarlo y, además, el navegador no envía sus datos al servidor.

## **2.2.- Cuadro de contraseña (type=”password”)**

**La única diferencia entre este control y el cuadro de texto normal es que el texto que el usuario escribe en un cuadro de contraseña no se ve en la pantalla**. En su lugar, los navegadores ocultan el texto utilizando asteriscos o círculos, por lo que es ideal para escribir contraseñas y otros datos sensibles.



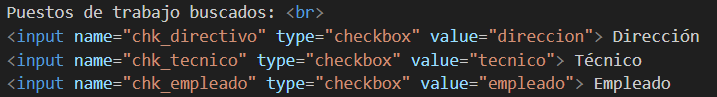


## **2.3.- Checkbox (type=”checkbox”)**

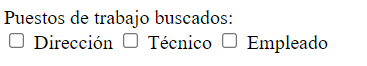
Los checkbox o *"casillas de verificación"* son controles de formulario que permiten al usuario seleccionar y deseleccionar opciones individualmente.

|  |
| --- |
| Información con relleno sólidoAunque en ocasiones se muestran varios *checkbox* juntos, cada uno de ellos es completamente independiente del resto. Por este motivo, se utilizan cuando el usuario puede activar y desactivar varias opciones relacionadas, pero no excluyentes. |

Ejemplo:



Y la salida es:



* El valor del atributo **type** para estos controles de formulario es checkbox. Como se muestra en el ejemplo anterior, el texto que se encuentra al lado de cada *checkbox* no se puede establecer mediante ningún atributo, por lo que es necesario añadirlo manualmente fuera del control del formulario. Si no se añade un texto al lado de la etiqueta <input > del *checkbox*, el usuario sólo ve un pequeño cuadrado sin ninguna información relativa a la finalidad de ese *checkbox*.
* El valor del atributo ***value***, junto con el valor del atributo ***name***, es la información que llega al servidor cuando el usuario envía el formulario.
* Si se quiere mostrar un *checkbox* seleccionado por defecto, se utiliza el atributo ***checked***. Si el valor del atributo es checked, el *checkbox* se muestra seleccionado. En cualquier otro caso, el *checkbox* permanece sin seleccionar. Aunque resulta redundante que el nombre y el valor del atributo sean idénticos, es obligatorio indicarlo de esta forma porque los atributos en XHTML no pueden tener valores vacíos:



Se vería tal que así:

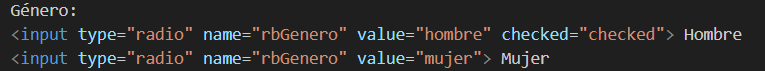


## **2.4.- Radiobutton (type=”radio”)**

Los controles de tipo ***radiobutton*** son similares a los controles de tipo checkbox, pero presentan una diferencia muy importante: **son mutuamente excluyentes**.

|  |
| --- |
| Información con relleno sólidoLos *radiobutton* se utilizan cuando el usuario solamente puede escoger una opción entre las distintas opciones relacionadas que se le presentan. Cada vez que se selecciona una opción, automáticamente se deselecciona la otra opción que estaba seleccionaba. |

Ejemplo:



El resultado sería:



* El valor del atributo ***type*** para estos controles de formulario es radio.
* El atributo ***name*** se emplea para indicar los radiobutton que están relacionados. Por lo tanto, cuando varios *radiobutton* tienen el mismo valor en su atributo *name*, el navegador sabe que están relacionados y puede deseleccionar una opción del grupo de *radiobutton* cuando se seleccione otra opción.

## **2.5.- Botón de envío del formulario (type=”submit”)**

La mayoría de los formularios dispone de un botón para enviar al servidor los datos introducidos por el usuario:

* El valor del atributo ***type*** para este control de formulario es submit. El navegador se encarga de enviar automáticamente los datos cuando el usuario pincha sobre este tipo de botón.
* El valor del atributo ***value*** es el texto que muestra el botón. Si no se establece el atributo value, el navegador muestra el texto predefinido “Enviar consulta”.



La salida es:



## **2.6.- Botón de reseteo del formulario (type=”reset”)**

Se trata de un botón que **borra todos los datos introducidos por el usuario y devuelve el formulario a su estado original**, con los valores que hayamos dado por defecto en los campos value:



La salida es:



* El valor del atributo ***type*** para este control de formulario es reset.

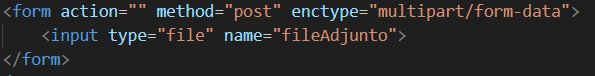
Como es habitual en los botones de formulario, el atributo value permite establecer el texto que muestra el botón. Si no es utiliza este atributo, el navegador muestra el texto predefinido del botón, que en este caso es Restablecer.

## **2.7.- Ficheros adjuntos (type=”file”)**

* El valor del atributo ***type*** para este control de formulario es **file**. El navegador se encarga de mostrar un cuadro de texto donde aparece el nombre del archivo seleccionado y un botón que permite navegar por los directorios y archivos del ordenador del usuario.

|  |
| --- |
| Información con relleno sólido  Si se incluye un control para adjuntar archivos, es obligatorio añadir el atributo enctype en la etiqueta <form> del formulario. El valor del atributo enctype debe ser multipart/form-data |

Ejemplo:



La salida sería:



## **2.8.- Campos ocultos (type=”hidden”)**

Los campos ocultos se emplean para añadir información oculta en el formulario.

El valor del atributo type para este control de formulario es hidden. **Los campos ocultos no se muestran por pantalla**, de forma que el usuario desconoce que el formulario los incluye.

|  |
| --- |
| Ayuda con relleno sólido  Normalmente los campos ocultos se utilizan para incluir información que necesita el servidor pero que no es necesario o no es posible que la establezca el usuario. |

## **2.9.- Botón de imagen (type=”image”)**

El aspecto de los botones de formulario se puede personalizar por completo, ya que incluso es posible utilizar una imagen como botón:

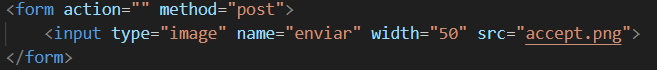
* El valor del atributo **type** para este control de formulario es *image*. El atributo ***src*** indica la URL de la imagen que debe mostrar el navegador en lugar del botón normal.

Su principal **ventaja** es que permite personalizar por completo la estética de los botones y mostrarlos con un aspecto homogéneo en todos los navegadores.



El principal **inconveniente** es que ralentiza la carga del formulario y si se quiere modificar su aspecto, es necesario crear una nueva imagen.

Ejemplo:





## **2.10.- Botón genérico (type=”button”)**

Algunos formularios complejos necesitan botones más avanzados que los de enviar datos (type="submit") y resetear el formulario (type="reset"). Por ese motivo, existe un **botón de tipo genérico.**



La salida es:

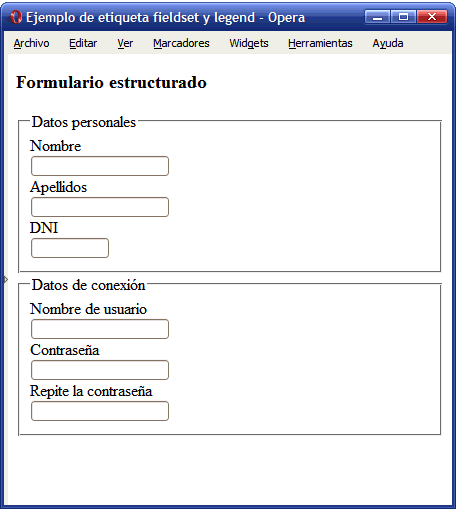


|  |
| --- |
| ACTIVIDAD1 - UT2\_9 |
| Determinar el código HTML necesario para crear el formulario que se muestra en la siguiente imagen:   1. Elegir el método más adecuado para el formulario (GET o POST) y cualquier otro atributo necesario. 2. La aplicación que se encarga de procesar el formulario se encuentra en la raíz del servidor, carpeta "php" y archivo "insertar\_cv.php". 3. El nombre puede tener 30 caracteres como máximo, los apellidos 80 caracteres y la contraseña 10 caracteres como máximo. 4. Asignar los atributos adecuados al campo del DNI. 5. Por defecto, debe estar marcada la casilla de suscripción al boletín de novedades. |
| La salida debe parecerse a lo siguiente: |

# **3.- Fieldset <fieldset> y <legend>**

Utilizando solamente las etiquetas <form> y <input> es posible crear formularios. No obstante, HTML define algunos elementos adicionales para mejorar la estructura de los formularios creados.

La siguiente imagen muestra un formulario que agrupa sus elementos y añade etiquetas a cada campo para mejorar su estructura:



* La etiqueta **<fieldset>** agrupa campos del formulario
* La etiqueta **<legend>** asigna un nombre a cada grupo.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Ejemplo:



# **4.- Label <label>**

Todos los controles de formulario salvo los botones presentan una carencia muy importante: no disponen de la opción de establecer el título o texto que se muestra junto al control. En el código HTML del ejemplo anterior, el nombre de cada campo se incluye en forma de texto normal, sin ninguna relación con el campo al que hace referencia.

El lenguaje HTML incluye una etiqueta denominada ***<label>*** y que se utiliza para establecer el título de cada campo del formulario. Su definición formal es la siguiente:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

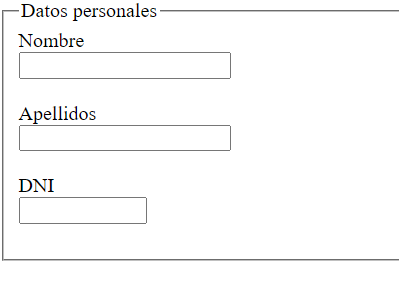
Descripción generada automáticamente

|  |
| --- |
| Información con relleno sólido  El único atributo que suele utilizarse con la etiqueta <label> es *for*, que indica el identificador (atributo id) del campo de formulario para el que esta etiqueta hace de título. |

Ejemplo:

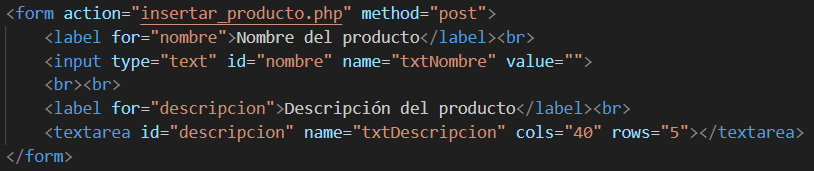


Este código HMTL se vería tal que así:

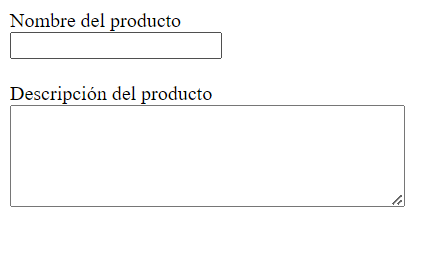


# **5.- Textarea <textarea>**

Las áreas de texto son útiles cuando se debe introducir una gran cantidad de texto, ya que es mucho más cómodo de introducir que en un campo de texto normal:



Se vería así:



La definición formal de la etiqueta ***<textarea>*** es:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

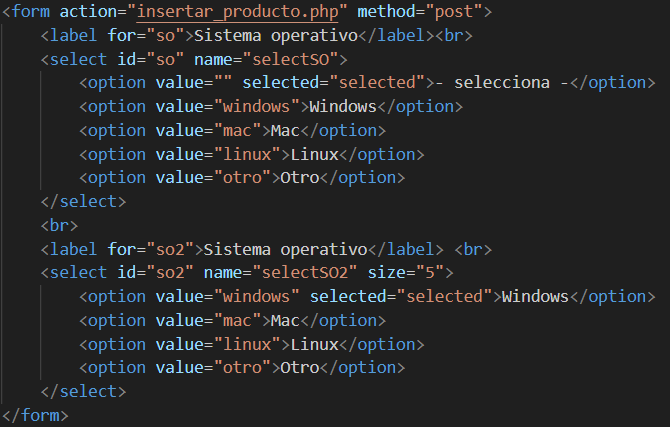
Los atributos más utilizados en las etiquetas <textarea> son los que controlan su anchura y altura.

* La anchura del área de texto se controla mediante el atributo **cols**, que indica las *columnas* o número de caracteres que se podrán escribir como máximo en cada fila.
* La altura del área de texto se controla mediante **rows**, que indica directamente las filas de texto que serán visibles.

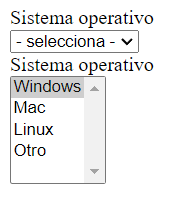
El principal **inconveniente** de los elementos ***<textarea>*** es que el lenguaje HTML no permite limitar el número máximo de caracteres que se pueden introducir. Mientras los elementos <input type="text"> disponen del atributo maxlength, las áreas de texto no disponen de un atributo equivalente, por lo que sólo es posible limitar el número de caracteres mediante su programación con JavaScript.

# **6.- Select <select>**

El otro control disponible para los formularios es el de las listas desplegables:



La salida:



Los tres tipos de listas desplegables se definen con la misma etiqueta <select> y cada elemento de la lista se define mediante la etiqueta <option>:

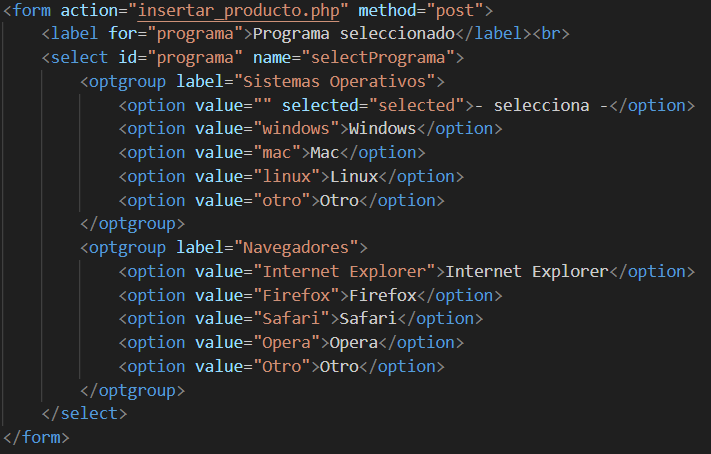
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

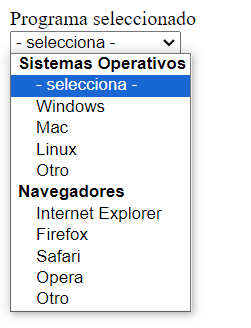
La etiqueta ***<select>*** define la lista y encierra todas las opciones que muestra la lista. Cada una de las opciones de la lista se define mediante una etiqueta <option>.

* El atributo **value** de cada opción es obligatorio, ya que es el dato que se envía al servidor cuando el usuario envía el formulario. Para seleccionar por defecto una opción al mostrar la lista, se añade el atributo selected a la opción deseada.

|  |
| --- |
| Información con relleno sólido  Las listas desplegables permiten agrupar sus opciones de forma que el usuario pueda encontrar fácilmente las opciones cuando la lista es muy larga con la etiqueta <optgroup> |



La salida sería:



Definición de la etiqueta **<optgroup>:**

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

* El único atributo que suele utilizarse con la etiqueta <optgroup> es **label**, que indica el nombre de cada agrupación. Los navegadores muestran de forma destacada el título de cada agrupación, de forma que el usuario pueda localizar más fácilmente la opción deseada.

# **7.- Parámetros**

Para completar el funcionamiento de los elementos de los formularios tenemos los parámetros.

Aunque durante el tema ya hemos visto el funcionamiento de alguno, vamos a volver a ver su funcionamiento y vamos a ver el funcionamiento de alguno nuevo.

[**autofocus**](https://developer.mozilla.org/es/docs/Web/HTML/Element/input#attr-autofocus)

Este atributo Booleano permite especificar que un control de formulario tenga el cursor cuando la página se carga, a menos que el usuario lo reemplace, por ejemplo, escribiendo en un control diferente. Solo un elemento de formulario en un mismo documento puede tener el atributo autofocus, el cual es Booleano. No puede ser aplicado si el atributo type tiene valor hidden (es decir, no se puede establecer automáticamente el cursor en un control oculto).

[**checked**](https://developer.mozilla.org/es/docs/Web/HTML/Element/input#attr-checked)

Cuando el valor del atributo type es radio o checkbox, la presencia de este atributo Booleano indica que el control está seleccionado de forma predeterminada. De lo contrario, será ignorado.

[**disabled**](https://developer.mozilla.org/es/docs/Web/HTML/Element/input#attr-disabled)

Este atributo Booleano indica que el control no está disponible para interacción. De igual forma, el valor de un control deshabilitado no es enviado con el formulario.

[**maxlength**](https://developer.mozilla.org/es/docs/Web/HTML/Element/input#attr-maxlength)

Si el valor del atributo type es text, email, search, password, tel, o url, este atributo especifica el número máximo de caracteres (en puntos de código Unicode) que el usuario puede introducir.

[**minlength**](https://developer.mozilla.org/es/docs/Web/HTML/Element/input#attr-minlength)

Si el valor del atributo type es text, email, search, password, tel, o url, este atributo especifica la longitud mínima de caracteres (que el usuario puede introducir.

[**readonly**](https://developer.mozilla.org/es/docs/Web/HTML/Element/input#attr-readonly)

Este atributo indica que el usuario no puede modificar el valor del control.

[**required**](https://developer.mozilla.org/es/docs/Web/HTML/Element/input#attr-required)

Este atributo especifica que el usuario debe llenar el control antes de enviarlo al formulario.

[**size**](https://developer.mozilla.org/es/docs/Web/HTML/Element/input#attr-size)

El tamaño inicial del control. Este valor es en píxeles, a menos que el atributo type sea text o password, en cuyo caso será el número entero de caracteres. A partir de HTML5, este atributo aplica sólo cuando el atributo type es text, search, tel, url, email, o password, de otro modo es ignorado. Además, el tamaño debe ser mayor a cero. Si no se especifica un tamaño, se usa un valor predeterminado de 20.

|  |
| --- |
| ACTIVIDAD2 - UT2\_9 |
| Determinar el código HTML necesario para crear el formulario que se muestra en la siguiente imagen: |
| La salida debe parecerse a lo siguiente: |

# **8.- Formularios en HTML5**

Siguen la misma estructura que en HTML 4.01.

* Se encapsulan dentro de la etiqueta <form>
* Mandan los valores de control al servidor cuando el usuario o la aplicación manda la página.
* Todos los controles anteriores persisten y trabajan de la misma forma, con las nuevas características.

El concepto más importante de los formularios HTML5 es que su especificación trabaja con el **comportamiento funcional y la semántica**, no con su apariencia o visualización.

## **8.1.- Nuevos tipos de entrada (Input type)**

|  |  |
| --- | --- |
| Tipo | Función |
| tel | Números de teléfono. |
| email | Direcciones de correo electrónico. |
| url | URL de localización web. |
| search | Término proporcionado a un motor de búsqueda, por ejemplo, la barra de búsqueda superior del navegador. |
| range | Selector numérico dentro de un rango de valores, típicamente visualizado como un deslizador. |
| number | Contiene sólo valores numéricos. |
| color | Selector de color, que puede ser representado por un círculo de color o un selector de colores predefinidos |
| time | Indicador y selector de hora, sin zona horaria. |
| date | Selección en un calendario. |
| week | Selección para una semana en un año dado. |
| month | Selección para un mes en un año dado. |

* Ejemplo con ***“search”***:



La salida es:



Al empezar a escribir en el campo type=“search”, aparece una ‘X’ en el campo de búsqueda que nos permite borrar el texto a buscar.

* Ejemplo con ***“range”***:

Selector numérico dentro de un rango de valores, típicamente visualizado como un deslizador.



La salida es:



***min*** es el mínimo valor que podemos alcanzar, ***max*** es el valor máximo y ***value*** el valor por defecto.

* Ejemplo con ***“number”***:



La salida será:



## **8.2.- Diferencias semánticas**

Los nuevos campos de formularios en HTML5, también aportan un significado semántico, con lo que consiguen adaptar los teclados virtuales a las necesidades del usuario.

Por ejemplo, en el navegador de *Iphone*:

* El usuario al seleccionar el campo Email, al tener un significado semántico, el dispositivo sabe que es un email y modifica el teclado virtual añadiendo la ‘@’.
* Sucede lo mismo con otros campos, como type=”url”, el teclado virtual se adapta a las necesidades de cada campo.

## **8.3.- Validación de formularios**

HTML 5, aporta un gran avance a la hora de validar formularios en cliente.

* No es obligatorio JavaScript para realizar validaciones, aunque con JavaScript tendremos más posibilidades.
* Consigue una validación mucho más cómoda y sencilla.

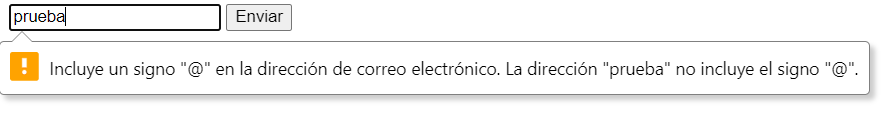
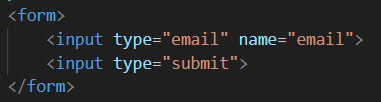
Los distintos tipos de campos de formulario automáticamente se validan en función del tipo que sean.

Cada navegador hará la validación de una forma predeterminada y mostrará mensajes propios de cada navegador al usuario.

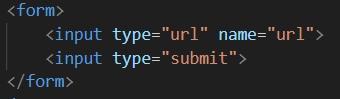
Como hemos visto anteriormente podemos marcar campos obligatorios con el atributo ‘required’.

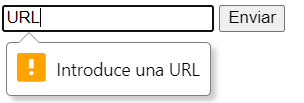
La entrada no tiene el formato correcto y automáticamente salta la validación.

Por ejemplo:



Otro ejemplo:





|  |
| --- |
| ACTIVIDAD3 - UT2\_9 |
|  |
| Crea una página web que contenga un formulario con los siguientes campos de información:   * Nombre, con un control de tipo texto obligatorio. * Correo electrónico, con un control de tipo email obligatorio. * URL, con un control de tipo url que muestre la ayuda "Escribe la URL de tu página web personal". * Fecha, con un control de tipo date. * Tiempo, con un control de tipo time. * Fecha y hora, con un control de tipo datetime. * Mes, con un control de tipo month. * Semana, con un control de tipo week. * Número, con un control de tipo number que limite la entrada a un valor entre -10 y 10. * Teléfono, con un control de tipo tel. * Término de búsqueda, con un control de tipo search. * Color favorito, con un control de tipo color. * Un botón de envío.   Además, debes tener en cuenta los siguientes requisitos:   * El título de la página debe ser “Formulario de prueba de HTML5”. * El método de envío del formulario debe ser GET. * El destino del envío del formulario debe ser "".   ****Nota****: si al validar el código HTML de tu página web con el validador del W3C se muestran mensajes de advertencia como The date input type is not supported in all browsers. Please be sure to test, and consider using a polyfill. no te preocupes, el validador te está avisando de que lleves cuidado al usar una característica de HTML5 que no está admitida por todos los navegadores actuales.  ****Nota****: por lo anterior, cuando realices esta actividad y compruebes cómo se visualiza tu página web en distintos navegadores es muy probable que obtengas diferentes resultados.  Deberás crear una web que tenga el aspecto siguiente: |