



Tabla de contenido

Introducción	2
Etiqueta form	2
Cuadro de texto	2
Password	3
Área de texto	3
Checkbox	3
Radio	4
Select.....	4
Enviar	5
Borrar	5
Image.....	6
Hidden	6
Button (1)	6
Button (2)	7
Label.....	7
File.....	7
Fieldset	8
Otros atributos	8
Orden de tabulación	8
Tecla de acceso rápida.....	8
Desactivar un elemento	9
Hacer obligatorio un elemento.....	9
Indicar un texto de ayuda.....	9
Darle el foco por defecto a un control	9
Autocompletado de datos.....	9
Colocar un elemento fuera de su formulario	9
Contenido editable	9
Nuevos controles de HTML 5.....	9
Datalist	10
Validando los datos de un formulario.....	11
Enviando y recibiendo los datos.....	11
Ejemplo completo	12
Ejercicios	13



Introducción

Muchas páginas HTML necesitan que el usuario introduzca o elija algunos datos en una página web, como usuario y contraseña, introducir su nombre, apellidos, aceptar condiciones, ... Estos datos serán procesados en el lado cliente, mediante JavaScript en el navegador, o en el lado servidor, mediante algún lenguaje como PHP, Java,

Para que el usuario introduzca o elija estos datos, HTML ofrece una serie de etiquetas que veremos a continuación, etiquetas para escribir textos, elegir elementos de una lista, marcar alguna casilla, pulsar algún botón, ...

Etiqueta form

Muchas veces estas etiquetas irán dentro de una etiqueta **form**, para tenerlos agrupados y poder enviarlos en conjunto al servidor de forma automática.

Su sintaxis es la siguiente:

```
<form action="destino" method="post|get" enctype="tipo de codificación">
  <!-- aquí irá el formulario, pudiendo incluir cualquier etiqueta html -->
</form>
```

Donde:

- **action** indica la URL a la que se le enviarán los datos del formulario. Si no se indica nada, se enviará a la misma URL.
- **method** indica la forma como se envían los datos. Lo normal será usar POST y GET, las cuales explicaremos posteriormente.
- **enctype** indica como se codifican los datos para ser enviados, siendo su valor por defecto application/x-www-form-urlencoded

A continuación iremos viendo los controles de formulario que podemos usar, teniendo en cuenta que cada navegador o versión, puede mostrarlos de forma diferente.

Cuadro de texto

Muestra un cuadro de texto de una línea donde el usuario podrá escribir algo.

```
<input type="text" name="nombre" size="número" maxlength="número" value="valor" readonly>
```

- **size**: indica el tamaño en caracteres del cuadro de texto
- **maxlength**: indica el número máximo de caracteres que podrá introducir el usuario
- **minlength**: indica el número mínimo de caracteres que podrá introducir el usuario
- **name**: el nombre que tendrá este elemento y nos será útil a la hora de reconocer la información recibida o procesarlo con JavaScript (explicado más adelante)
- **value**: un texto que queramos que aparezca por defecto en el cuadro de texto

- **readonly**: para hacer que el usuario no pueda escribir en el cuadro de texto.

Password

Es idéntico a un cuadro de texto pero el texto introducido por el usuario aparece como asteriscos o puntos con el fin de introducir contraseñas o datos confidenciales.

```
<input type="password" name="nombre" size="número" maxlength="número" value="valor">
```

Los parámetros son los mismos que para un cuadro de texto.

Área de texto

Usaremos este control cuando queramos que el usuario escriba un texto de varias líneas.

```
<textarea name="nombre" rows="número" cols="número" readonly>texto a mostrar dentro del cuadro de texto</textarea>
```

- **name**: el nombre que tendrá este elemento y nos será útil a la hora de reconocer la información recibida o procesarlo con JavaScript (explicado más adelante)
- **rows**: indica, en filas, el tamaño del control (no el número máximo de filas)
- **cols**: indica el número de caracteres por línea
- **readonly**: para hacer que el usuario no pueda escribir en el cuadro de texto.



Hoy día los navegadores suelen incluir la opción de cambiar el tamaño del Textarea arrastrando por la esquina inferior derecha.

Checkbox

Muestra una casilla de verificación rectangular que el usuario podrá marcar o desmarcar. Esto es útil para pedir datos de tipo sí/no o verdadero/falso.

```
<input type="checkbox" name="nombre" value="valor" checked> ☐
```

- **name**: el nombre que tendrá este elemento y nos será útil a la hora de reconocer la información recibida o procesarlo con JavaScript (explicado más adelante)
- **checked**: con esta opción el control aparecerá marcado




- **value:** es el valor que se nos enviará si la casilla está marcada. Si no se indica, se suele enviar el valor on

Este control no incluye ningún texto, sólo el cuadro para marcarlo.

Radio

Parecido al anterior pero la casilla es circular y se usa, junto con otros controles radio, para que el usuario seleccione una opción entre varias, mientras que los checkbox están pensados para poder marcar una o varias opciones.

```
<input type="radio" name="nombre" value="valor" checked> 
```

- **name:** el nombre que tendrá este elemento y nos será útil a la hora de reconocer la información recibida o procesarlo con JavaScript (explicado más adelante)
- **checked:** con esta opción el control aparecerá marcado
- **value:** es el valor que se nos enviará si la casilla está marcada. Si no se indica, se suele enviar el valor on

Este control no incluye ningún texto, sólo el círculo para marcarlo.

Para que varios elementos radio actúen en conjunto, es decir, que al activar uno se desmarque el que estaba marcado, todos los radio deseados deberán tener el mismo nombre (*name*). De la misma manera, si queremos tener varios grupos de radio, cada grupo deberá tener el mismo nombre y diferente al de los otros grupos.

Select

Sirve para mostrar una lista desplegable de elementos de los cuales el usuario podrá seleccionar uno o varios.

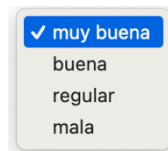
```
<select name="nombre" multiple size="número">
  <option value="valor">texto</option>
  <option value="valor">texto</option>
  <option value="valor" selected>texto</option>
  ...
</select>
```

- **name:** el nombre que tendrá este elemento y nos será útil a la hora de reconocer la información recibida o procesarlo con JavaScript (explicado más adelante)
- **multiple:** se podrá seleccionar más de un elemento
- **option:** tendremos que poner una etiqueta *option* por cada elemento que queramos que aparezca en la lista
- **size:** indica el número de elementos visibles a la vez. Por defecto solo uno, con lo que se comporta como una lista desplegable.

- **texto:** el texto que se muestra en cada elemento de la lista
- **value:** si indicamos esta propiedad, su valor será el que se envíe al servidor. Si no se indica, se enviará el texto de la etiqueta.
- **selected:** podemos indicar el elemento de la lista que aparecerá seleccionado por defecto

Ejemplo:

```
<select name="opinion">
  <option selected>muy buena</option>
  <option>buena</option>
  <option>regular</option>
  <option>mala</option>
</select>
```

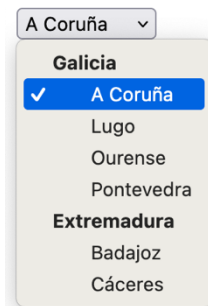


A visual representation of the first example. It shows a dropdown menu with a blue header bar containing a checkmark and the text 'muy buena'. Below the header, the options 'buena', 'regular', and 'mala' are listed in a light gray box.

También podemos crear agrupaciones de elementos de una lista con la etiqueta **<optgroup label="texto">**. El texto de estas agrupaciones aparecerá en la lista pero no se podrá seleccionar.

Ejemplo:

```
<select name="provincia">
  <optgroup label="Galicia">
    <option>A Coruña</option>
    <option>Lugo</option>
    <option>Ourense</option>
    <option>Pontevedra</option>
  </optgroup>
  <optgroup label="Extremadura">
    <option>Badajoz</option>
    <option>Cáceres</option>
  </optgroup>
</select>
```



A visual representation of the second example. It shows a dropdown menu with a header 'A Coruña' and a dropdown arrow. The menu is open, showing two groups: 'Galicia' and 'Extremadura'. Under 'Galicia', 'A Coruña' is selected with a checkmark. Other options in the 'Galicia' group are 'Lugo', 'Ourense', and 'Pontevedra'. Under 'Extremadura', the options are 'Badajoz' and 'Cáceres'.

Enviar

Usaremos este control, que se muestra como un botón, para que se envíen los datos introducidos por el usuario a la URL indicada en la propiedad *action* de la etiqueta *form*.

```
<input type="submit" name="nombre" value="texto">
```

- **name:** el nombre que tendrá este elemento y nos será útil a la hora de reconocer la información recibida o procesarlo con JavaScript (explicado más adelante)
- **value:** será el texto que queremos que aparezca en el botón. Si no lo indicamos, cada navegador pone un texto diferente por defecto

Enviar consulta

Borrar

Usaremos este control, que se muestra como un botón, para que se vuelvan a poner todos los datos a sus valores por defecto.

```
<input type="reset" name="nombre" value="texto">
```



HTML - Formularios / Web Forms

- **name:** el nombre que tendrá este elemento y nos será útil a la hora de reconocer la información recibida o procesarlo con JavaScript (explicado más adelante)
- **value:** será el texto que queremos que aparezca en el botón. Si no lo indicamos, cada navegador pone un texto diferente por defecto

Restablecer

Image

Usaremos este control para que actúe como el botón de enviar pero usando una imagen en vez del botón.

```
<input type="image" src="url" name="nombre">
```

- **name:** el nombre que tendrá este elemento y nos será útil a la hora de reconocer la información recibida o procesarlo con JavaScript (explicado más adelante)
- **src:** será la URL de la imagen a mostrar

Se enviarán las coordenadas X e Y de la imagen donde el usuario haya pulsado

Hidden

Usaremos este control para incluir textos que no queremos que se vean en la página web. Útil para enviar datos al servidor que no es necesario que vea el usuario

```
<input type="hidden" value="valor" name="nombre">
```

- **name:** el nombre que tendrá este elemento y nos será útil a la hora de reconocer la información recibida o procesarlo con JavaScript (explicado más adelante)
- **value** será el texto que queremos que se envíe

Button (1)

Usaremos este control para que se muestre un botón. Su utilidad será cuando hagamos páginas que incluyan JavaScript y queremos que se haga algo al pulsar el botón.

```
<input type="button" value="valor" name="nombre">
```

- **name:** el nombre que tendrá este elemento y nos será útil a la hora de reconocer la información recibida o procesarlo con JavaScript (explicado más adelante)
- **value** será el texto que queremos que aparezca en el botón



HTML - Formularios / Web Forms

Button (2)

Existe otra manera de poner botones usando una etiqueta *button* y que admite un texto y/o una imagen a mostrar en el botón. En el caso de un texto se puede acompañar de otras etiquetas como *b*, *i*,...

```
<button type="tipo" name="nombre">texto y/o imagen</button>
```

- **name:** el nombre que tendrá este elemento y nos será útil a la hora de reconocer la información recibida o procesarlo con JavaScript (explicado más adelante)
- **tipo:** puede ser *button*, *reset* o *submit*

Label

Sirve para asociar un texto en aquellos controles que no muestran texto. Su utilidad puede ser aplicarles estilos, que determinados navegadores lean sus contenidos, ... Se puede usar de dos modos:

```
<label for="id">texto</label>
```

En este caso se asocia la etiqueta con un control que tenga un id igual al indicado en el parámetro *for*.

Ejemplo:

```
<label for="nombre">Nombre</label><input type="text" id="nombre" name="nombre">
```

La otra forma es:

```
<label>texto y elemento de formulario</label>
```

En este caso el control va dentro de la etiqueta *label*.

Ejemplo:

```
<label>Nombre<input type="text" id="nombre" name="nombre"></label>
```

File

Sirve para enviar ficheros desde el cliente al servidor.

```
<input type="file" name="nombre" multiple accept="tipo mime">
```

- **name:** el nombre que tendrá este elemento y nos será útil a la hora de reconocer la información recibida o procesarlo con JavaScript (explicado más adelante)
- **multiple:** permite seleccionar varios ficheros en lugar de un solo
- **accept:** permite indicar el tipo mime de los archivos que se admitirán. Por ejemplo, "image/png, image/gif, image/jpeg", "image/*", "application/pdf", ... Podemos encontrar una lista de estos tipos en la siguiente URL:

https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Common_types



HTML - Formularios / Web Forms

Los navegadores suelen mostrar un texto con el nombre del archivo seleccionado y un botón para seleccionar el fichero o ficheros.

No se ha seleccionado ningún archivo.

Cuando se envíen ficheros la codificación indicada en el atributo `enctype` del formulario debería ser `multipart/form-data`.

Fieldset

Permite añadir un marco alrededor de una serie de controles (este marco llega hasta el final del contenedor). Usando la etiqueta **legend** en su interior podemos ponerle un texto en la parte superior de ese marco.

```
<fieldset>
  <legend>texto</legend>
  controles y otras etiquetas
</fieldset>
```

Ejemplo:

```
<fieldset>
  <legend>Estado Civil</legend>
  <input type="radio" name="estadoCivil" value="soltero">soltero<br>
  <input type="radio" name="estadoCivil" value="casado">casado<br>
  <input type="radio" name="estadoCivil" value="viudo">viudo<br>
</fieldset>
```

Otros atributos

Orden de tabulación

Usando el atributo **tabindex**=`"número"` podemos cambiar el orden de tabulación dentro de los controles del formulario o enlaces (siendo cero el primero). La principal utilidad es saltarnos elementos como enlaces que estén entre controles de un formulario. No tenemos por qué asignar los números correlativamente, pudiendo ser mejor dejar un "hueco" entre ellos.

Ejemplo:

```
<input type="text" name="nombre" tabindex="1">
<a href="ayuda.htm">ayuda</a><br>
<input type="text" name="edad" tabindex="2">
```

En este ejemplo saltaremos del primer cuadro de texto al segundo sin pararnos antes por el enlace.

Si no queremos que el foco se pare en un control podemos darle un *tabindex* negativo.

En el caso de controles radio con el mismo nombre, el navegador puede considerar que una tabulación se debe hacer al siguiente elemento que haya después de esos controles radio, y no pasar por cada uno de los radios.

Tecla de acceso rápida

Con el atributo **accesskey**=`"tecla"` podemos hacer que pulsando una tecla de acceso y la tecla indicada el control obtenga el foco. En las etiquetas *label* el foco lo recibe su control asociado. Esta



HTML - Formularios / Web Forms

tecla de acceso varía según el navegador, siendo ALT + MAY + tecla en Firefox para Windows, CTRL + ALT + tecla en Firefox para MacOS, ALT + tecla en Chrome para Windows, CTRL + ALT + tecla en Chrome para MacOS, ... También debemos tener en cuenta que cada navegador tiene sus propias teclas de acceso rápido que no deben entrar en conflicto con las nuestras (pues se ignorarán).

Este parámetro también podemos indicarlo en una etiqueta a *href*, con lo que al pulsar la tecla de acceso, se abrirá el vínculo de ese enlace.

Desactivar un elemento

Usando el atributo **disabled** en un elemento hacemos que esté desactivado (normalmente se mostrará grisáceo).

Hacer obligatorio un elemento

Podemos indicar el atributo **required** para indicar que un valor es obligatorio. En un grupo de radio buttons, basta con poner el atributo a un de los radios, aunque podemos hacerlo en todos.

Indicar un texto de ayuda

Podemos usar el atributo **placeholder** para mostrar un texto de ayuda en el control de texto. Normalmente este texto aparece dentro de control de un color gris claro y, cuando el control obtiene el foco o el usuario comienza a escribir en este, desaparece.

Darle el foco por defecto a un control

Podemos indicar el atributo **autofocus** en un control para que reciba el foco automáticamente al cargar la página.

Autocompletado de datos

Los navegadores actuales permiten la posibilidad de autocompletar datos de formularios si ya estaban guardados. Usando el atributo **autocomplete** con el valor on u off podemos activarlo o desactivarlo, pudiéndose hacer a nivel de formulario y / o elementos.

Colocar un elemento fuera de su formulario

Se puede indicar el atributo **form** con el nombre (atributo id) del formulario lo cual nos permite tener controles fuera de las etiquetas `<form></form>`, facilitando el diseño de las páginas.

Contenido editable

Usando el atributo **contenteditable** permitimos que el contenido de una etiqueta HTML pueda ser modificado por el usuario como si fuera un cuadro de texto. Según el navegador pondrá un borde a la hora de escribir.

Se admite pulsar la tecla *enter* mientras se escribe con lo que, dependiendo de la etiqueta, se realizará un nuevo salto de línea, como un `br` en un `p`, se añadirá algún elemento, como un `li` en una lista, ...

Nuevos controles de HTML 5

La etiqueta *input* se ha ampliado para añadir nuevos controles. Algunos de los nuevos valores que se pueden indicar en la etiqueta *type* son:



HTML - Formularios / Web Forms

- **number** para que el usuario indique un número
- **email**: para indicar una dirección de correo electrónico. Se puede usar el atributo *multiple* para poder indicar varias separadas por comas
- **url**: para indicar urls
- **search** para indicar una búsqueda
- **color**: para indicar un color
- **datetime**, **date**, **month**, **week**, **time**: para indicar fechas y horas, horas, meses, semanas y horas respectivamente. El valor de la fecha para los atributos *value*, *min* y *max* lo pondremos de la forma año-mes-día
- **tel**: para indicar números de teléfono
- **range**: una barra deslizante
- **pattern**: muestra un cuadro de texto que permite establecer un patrón (expresión regular) para establecer los valores válidos. Estudiaremos las expresiones regulares cuando veamos JavaScript, pero podemos acceder a páginas como <https://regex101.com/>, para crearlas, ver ejemplos y documentación sobre ellas.

Ejemplo:

```
<!-- dos letras seguidas de uno o más números -->  
<input pattern="[a-z]{2}[0-9]{1,}" name="codigo">
```

Notas:

- la manera de representar los controles y de mostrar los errores de validación no está definida y varía entre navegadores
- cuando un navegador no reconoce un control lo convierte en un cuadro de texto
- muchos de estos nuevos tipos son bastante útiles en los navegadores de los smartphones ya que pueden mostrar un teclado virtual u otro dependiendo del tipo
- en los tipos *number*, *range* y los de fecha y hora podemos indicar los atributos *min* y *max* para indicar valores mínimos y/o máximos admitidos

Ejemplo:

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

Datalist

Se pueden crear listas despegables donde el usuario también puede escribir el valor deseado, al contrario que la etiqueta *select*. Para ello se establece el atributo **list** en la etiqueta *input* deseada con el nombre de una lista (su atributo *id*) que se creará con etiquetas **datalist** y *option*.

En cada *option* podemos tener los siguientes atributos:

- **label**: el texto a mostrar. Si no se indica, se usa el valor de *value*
- **value**: es el valor que se envía, si usamos un formulario. Es obligatorio salvo que indiquemos un texto entre la etiqueta *option* de apertura y de cierre

Dependiendo del navegador se mostrará una lista desplegable o los valores irán apareciendo según el usuario vaya escribiendo caracteres.

Ejemplo:

```
<input type="text" list="ciudades">
```

```

<!-- No tienen por qué ir en un formulario y puede haber etiquetas entre ellas -->
<datalist id="ciudades">
  <option label="Vigo" value="Vigo"></option>
  <option label="Baiona" value="Bayona"></option>
  <option label="Cangas" value="Cangas"></option>
</datalist>

```

Validando los datos de un formulario

En los navegadores actuales, si hemos establecido limitaciones en los controles, como un valor mínimo, o requerido, además de si el usuario introduce datos incorrectos en un elemento number, email, ... el navegador no enviará los datos, salvo que modifiquemos este comportamiento con JavaScript, mostrándose además mensajes de error, dependiendo de este.

En Firefox

En Chrome

A los elementos que no sean válidos les podemos aplicar el selector **:invalid** de CSS para decorarlos; a los válidos el selector **:valid**. La validación se va produciendo mientras introducimos los datos, con lo que, si por ejemplo, ponemos el color de fondo cuando el valor no sea válido y un mínimo de 3 caracteres, cuando el usuario comience a escribir ya será no válido hasta que escriba ese mínimo.

Usando el atributo **novalidate** en un elemento *form* deshabilitamos esta validación de sus elementos.

Enviando y recibiendo los datos

Como ya hemos comentado, con el atributo *action* podemos indicar la URL a la que enviaremos los datos, pudiendo ser incluso una dirección de correo si usamos <mailto:emailDeseado>.

Los datos se pueden enviar principalmente de dos modos, los cuales se indican en el atributo *method* del formulario

- GET: los datos van en la misma URL en lo que se conoce como una *querystring* (cadena de consulta). Su formato es el siguiente:

```
urlDestino?nombre1=valor1&nombre2=valor2&....
```

Donde:

- la cadena va separada de la URL por un ?
 - nombre1, nombre2, ... son los nombres que hemos dados a los elementos en su propiedad name y valor1, valor2, ... los valores elegidos o escritos por el usuario
 - cada par de nombre=valor va separado por un &
- POST:
 - los datos van también de la forma nombre=valor1&nombre=valor2... pero en el cuerpo de la petición, no siendo vistos directamente por el usuario.

Luego veremos un ejemplo de ambos.

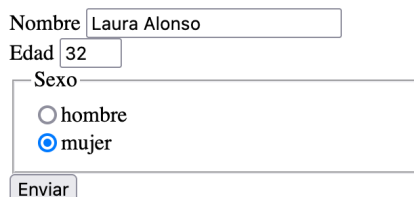
En cuanto a lo que envía cada elemento, tenemos:

- en elementos text, password, textarea, number, date, color,:
nombre=valor si el usuario introduce o elige algo
nombre= si el usuario no introduce o elige nada
- en elementos Checkbox o radio:
nombre=valor indicado en el parámetro value, si se indicó y se marcó/seleccionó la casilla
nombre=on si se marcó/seleccionó la casilla y no se indicó el parámetro value (en algunos navegadores puede devolver algo diferente a on)
nada si el usuario no marcó/seleccionó la casilla
- para un elemento select:
nombre=texto del elemento seleccionado o texto del parámetro value (si se indicó). Si permitimos la selección de más de un elemento recibiremos esto mismo para cada elemento seleccionado. Si no hemos seleccionado uno, siempre está seleccionado por defecto el primero.
- para un elemento image:
nombre.x=valorX y nombre.y=valorY con las coordenadas de la imagen donde el usuario hizo clic, siendo 0, 0 la esquina superior izquierda

Ejemplo completo

Deberemos usar CSS y otras etiquetas para conseguir un diseño más atractivo:

```
<form action="" method="get">
  <label for="nombre">Nombre</label> <input type="text" name="nombre"><br>
  <label for="edad">Edad</label> <input type="text" name="edad" size="3" maxlength="3"><br>
  <fieldset>
    <legend>Sexo</legend>
    <input type="radio" name="sexo" value="hombre">hombre<br>
    <input type="radio" name="sexo" value="mujer">mujer
  </fieldset>
  <input type="submit" value="Enviar">
</form>
```



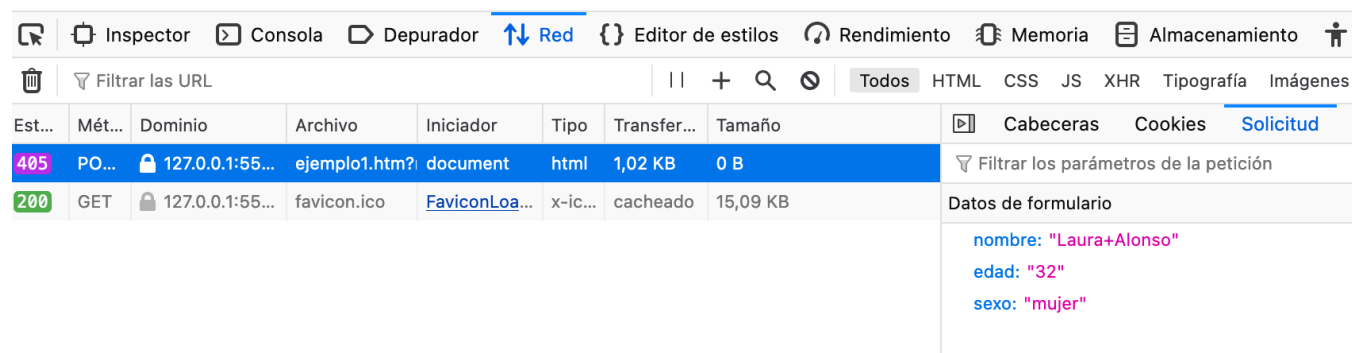
Usando el método GET, los datos anteriores se enviarán de la forma:

urlDestino?nombre=Laura+Alonso&edad=32&sexo=mujer

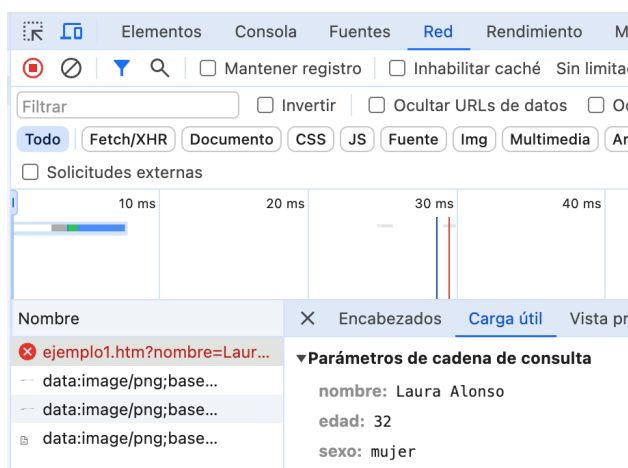


HTML - Formularios / Web Forms

Usando POST debemos verlo en las herramientas de desarrollo del navegador. En Firefox en la pestaña RED y luego Solicitud (también en Cabeceras con más datos) en la parte derecha. En Chrome en la pestaña RED y luego Carga Útil.



Firefox



Chrome

¿Y de dónde sale ese + entre el nombre y el apellido? Los espacios, acentos, eñes, &, ... se codifican en formato application/x-www-form-urlencoded, que es el valor por defecto del parámetro enctype de la etiqueta form.

Ejercicios

Crear los siguientes formularios:

Registro de Usuario

Nombre:

Correo electrónico:



Encuesta de Satisfacción

Nombre:

Nivel de Satisfacción:

Los valores de satisfacción posibles serán: excelente, bueno, regular o malo.

Orden de Compra

Producto 1: ☐

Producto 2: ☐

Producto 3: ☐

Feedback

Nombre:

Comentarios:

¿Recomendarías nuestro servicio?

☐ Sí ☐ No

Formulario variado

Teléfono:

Valor:

Fecha de Nacimiento:

Color Favorito:

El rango debe estar entre 0 y 100 y la fecha entre el 1/1/2024 y el 31/12/2024.



Registro de Usuario

Nombre:

Apellidos:

Edad:

Sexo:

Correo Electrónico:

Contraseña:

Confirmar Contraseña:

Registrarse

Si los datos no son válidos, debe ponerse el color del texto en rojo y un asterisco rojo a la derecha del elemento.