

Fuente: <https://lenguajehtml.com>

## Listas

En HTML podemos crear listas de información de una forma muy sencilla. Empecemos por las etiquetas disponibles:

Etiqueta	Atributos	Descripción
<b>&lt;ul&gt;</b>		Define una lista sin orden. Se trata de la etiqueta contenedora.
<b>&lt;ol&gt;</b>	<b>start, reversed, type</b>	Define una lista numerada (con orden). Etiqueta contenedora.
<b>&lt;li&gt;</b>	<b>value</b>	Define un ítem de la lista.

En primer lugar debemos elegir la etiqueta **<ul>** o la etiqueta **<ol>** dependiendo de si queremos una lista donde no importa el orden (*lista de objetos que tengo en el bolsillo*) o una lista en la que sí importa el orden (*lista de canciones preferidas de mayor a menor*). Una vez elegido, comenzamos a añadir ítems, uno por etiqueta **<li>**. La forma más básica de crear una lista sería la siguiente:

```
<ul>
  <li>Primer ítem.</li>
  <li>Segundo ítem.</li>
  <li>Tercer ítem.</li>
</ul>
```

Esto crearía una lista, donde cada uno de los ítems aparecería con un círculo **negro** cada uno. Una particularidad poco conocida es que HTML nos permite omitir las etiquetas de cierre **</li>** si van seguidas de otra etiqueta de apertura **<li>**.

Si reemplazamos la etiqueta **<ul>** por **<ol>** los ítems pasan a ser ítems numerados (por defecto, números enteros a partir de 1). Las etiquetas **<ol>** permiten indicar varios atributos opcionales para modificar la forma en que se numera cada ítem:

Atributo	Valor	Significado
<b>start</b>	<u>[número]</u>	Indica el número del primer ítem de la lista y del que empezará a contar.
<b>reversed</b>	-	Si se indica este atributo, la lista se numera en orden inverso.
<b>type</b>	1   a   A   i   I	Lista con números enteros, letras o números romanos (en minúsculas o mayúsculas).

El siguiente ejemplo muestra una lista numerada con letras minúsculas, que empiezan en **c**, pero irán en orden inverso, por lo que terminarán en **c**:

```
<ol type="a" reversed>
  <li value="1">Primer ítem.</li>
  <li value="2">Segundo ítem.</li>
  <li value="3">Tercer ítem.</li>
</ol>
```

Obsérvese que también se puede establecer el atributo **value** a cada ítem de la lista para asignarle, por ejemplo, un valor numérico o ponderarlos.

Las **tablas** están incluidas en HTML desde sus primeras versiones y son una forma fantástica de mostrar datos claramente. Además, si las construimos de forma semántica y correctamente, es muy sencillo darle estilos desde CSS y cambiar su diseño con unas cuantas propiedades CSS, puesto que mediante las etiquetas que la componen se puede hacer referencia a cada parte de la misma.

## Tablas básicas

Veamos primero las **etiquetas básicas** para crear una tabla de la forma más sencilla posible:

Etiqueta	Descripción
<b>&lt;table&gt;</b>	Etiqueta contenedora que tendrá en su interior toda la tabla.
<b>&lt;tr&gt;</b>	<b>Table Row.</b> Etiqueta contenedora de cada fila de la tabla.
<b>&lt;td&gt;</b>	<b>Table Data.</b> Cada una de las celdas de la tabla.
<b>&lt;th&gt;</b>	<b>Table Header.</b> Cada una de las celdas de cabecera de la tabla.

La etiqueta **<table>** sería el elemento que contendría todos los elementos de la tabla, mientras que **<th>** y **<td>** se utilizarían para cada uno de los campos de la tabla (*cabecera y celda respectivamente*). Cada vez que se quisiera añadir una nueva fila, habría que incluirlo todo dentro de una etiqueta **<tr>**.

Estas cuatro etiquetas serían las etiquetas necesarias para crear una tabla. Un ejemplo muy sencillo de una tabla de **3x2 celdas** (*exceptuando las cabeceras*), sería la siguiente:

```

<table>
  <!-- Cabecera -->
  <tr>
    <th>Título columna 1</th>
    <!-- Celda de cabecera de la columna 1 -->
    <th>Título columna 2</th>
    <!-- Celda de cabecera de la columna 2 -->
    <th>Título columna 3</th>
    <!-- Celda de cabecera de la columna 3 -->
  </tr>
  <!-- Primera fila -->
  <tr>
    <td>Celda 1x1</td>
    <!-- Primera celda de la primera fila -->
    <td>Celda 2x1</td>
    <!-- Segunda celda de la primera fila -->
    <td>Celda 3x1</td>
    <!-- Tercera celda de la primera fila -->
  </tr>
  <!-- Segunda fila -->
  <tr>
    <td>Celda 1x2</td>
    <!-- Primera celda de la segunda fila -->
    <td>Celda 2x2</td>
    <!-- Segunda celda de la segunda fila -->
    <td>Celda 3x2</td>
    <!-- Tercera celda de la segunda fila -->
  </tr>
</table>

```

En este caso de ejemplo tendríamos una sencilla tabla de **3x2** celdas. Por defecto, en la tabla se crea de forma que se adapta al número de celdas que tiene la primera fila. Es decir, si en el primer elemento **<tr>** se indican 3 elementos **<th>**, la tabla esperará siempre 3 elementos en cada fila. Si indicáramos menos, las celdas correspondientes donde deberían estar aparecerían vacías (*sin celda*). Si añadiéramos una de más, se saldría de la tabla.

## Organización de tablas

Por defecto, al crear una tabla, el navegador se encarga de crearla a medida que va leyendo las etiquetas, por lo que la tabla se crea en el orden que se han especificado sus elementos, de arriba a abajo. Sin embargo, podemos utilizar una serie de etiquetas contenedoras que establecerán la **zona de la tabla** donde deben aparecer su contenido:

Etiqueta	Descripción
<b>&lt;thead&gt;</b>	Etiqueta contenedora de la cabecera de la tabla. Parte superior de la tabla.
<b>&lt;tbody&gt;</b>	Etiqueta contenedora del cuerpo de la tabla. Parte central de la tabla.
<b>&lt;tfoot&gt;</b>	Etiqueta contenedora del pie de la tabla. Parte inferior de la tabla.
<b>&lt;caption&gt;</b>	Establece un título de la tabla, independientemente de su posición.

A las etiquetas de tabla ya vistas, podemos añadir estas etiquetas, pudiendo definir la zona donde aparecerán sin que, necesariamente tengan el orden correcto.

Veamos un ejemplo:

```

<table>
  <!-- Table footer: pie de la tabla (tfoot) -->
  <tfoot>
    <tr>
      <td>Pie de tabla 1</td>
      <td>Pie de tabla 2</td>
    </tr>
  </tfoot>
  <!-- Table header: cabecera de la tabla (thead) -->
  <thead>
    <tr>
      <th>Columna 1</th>
      <th>Columna 2</th>
    </tr>
  </thead>
  <!-- Table body: cuerpo de la tabla (tbody) -->
  <tbody>
    <tr>
      <td>Celda 1</td>
      <td>Celda 2</td>
    </tr>
  </tbody>
  <!-- Leyenda o título de la tabla -->
  <caption>
    Título de la tabla
  </caption>
</table>

```

En este ejemplo, a pesar de seguir el orden **tfoot, thead, tbody, caption** definido en el HTML, el navegador lee la tabla y la redistribuye según su significado semántico, de modo que lo organiza dejándolo con el orden **caption, thead, tbody, tfoot**. De esta forma, estas etiquetas nos pueden servir para indicar secciones concretas de la tabla, independientemente del lugar donde estemos escribiendo, algo que puede ser muy útil si estamos creando la tabla de forma dinámica mediante algún lenguaje de programación.

## Video (modo básico)

La etiqueta **<video>** tiene varios atributos a nuestra disposición:

Atributo	Valor	Descripción
<b>src</b>	<u>Dirección URL</u>	Video a reproducir. Obligatoria si actua como etiqueta contenedora.
<b>poster</b>	<u>Dirección URL</u>	Muestra una imagen a modo de presentación.
<b>preload</b>	<b>auto   metadata   none</b>	Indica como realizar la precarga del video.
<b>mediagroup</b>	<u>nombre</u>	Establece un nombre para un grupo de contenidos multimedia.
<b>autoplay</b>		Comienza a reproducir el video automáticamente.
<b>loop</b>		Vuelve a iniciar el video cuando finaliza su reproducción (bucle).
<b>muted</b>		Establece el video sin sonido (silenciado).

<b>controls</b>		Muestra los controles de reproducción. Por defecto no se muestran.
<b>width</b>	<u>tamaño</u>	Indica el tamaño de ancho del video.
<b>height</b>	<u>tamaño</u>	Indica el tamaño de alto del video.

Un primer ejemplo muy básico para colocar un video en nuestra página web:

```
<video src="video.mp4" width="640" height="480"></video>
```

Sin embargo, esto mostrará el primer fotograma del video, con un tamaño de 640x480, pero se verá como una imagen, ya que no muestra los controles del video y tampoco tiene la autoreproducción activada. Podríamos solucionarlo indicando los atributos **controls** o **autoplay**.

Otro ejemplos básicos para colocar videos en nuestra página:

```
<!-- Ejemplo 1 -->
```

```
<video src="video.webm" poster="presentacion.jpg" controls></video>
```

```
<!-- Ejemplo 2 -->
```

```
<video src="video.mp4" autoplay muted loop></video>
```

En este caso cargamos un video, pero que no se mostrará porque se ha indicado que se utilice una imagen de presentación que se mostrará hasta que el usuario pulse en el botón de reproducir de los controles. En el segundo ejemplo, tenemos un video que se reproduce automáticamente al cargar la página, en silencio y en bucle, iniciándose una y otra vez.



## Marcos en línea (iframes)

Antiguamente, en versiones anteriores de HTML, existieron unas etiquetas llamadas marcos, mediante las cuales se permitía dividir la pantalla en varias secciones. Actualmente, ese modo de desarrollo ya no se utiliza ya que acarrea múltiples problemas de accesibilidad, posicionamiento, etc.

Sin embargo, de toda la familia de etiquetas de marcos, han sobrevivido los **<iframe>**, que siguen siendo útiles para incrustar contenido externo. Veamos un ejemplo extraído de servicios como **YouTube** o **SoundCloud**:

```
<iframe src="https://www.youtube.com/embed/lmeq3GeRttw" width="560" height="315"></iframe>
```

```
<a href="http://www.bing.com/" target="marco">Abrir buscador Bing</a>
```

```
<iframe name="marco" width="100%" height="450" src="https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/38289724&visual=true"></iframe>
```

Esta etiqueta permite incrustar contenido externo anidado en nuestro propio documento. Es decir, podemos crear un recuadro que en su interior va a tener otra dirección URL de otra página web, documento o recurso que queramos cargar en él. Generalmente, es la etiqueta más utilizada para todo tipo de contenido externo.

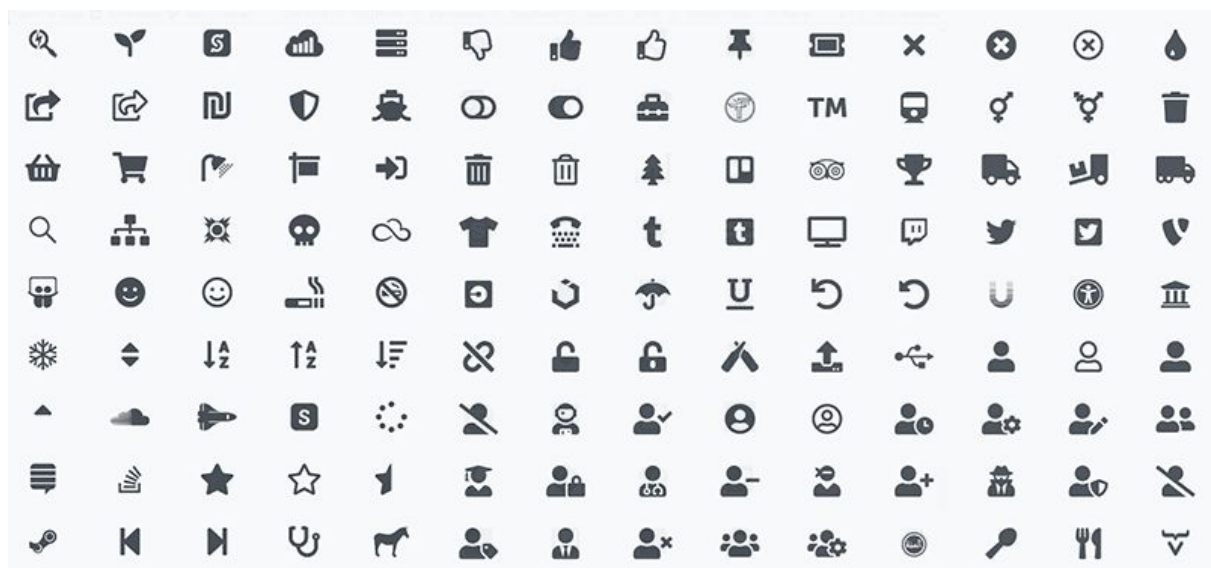
<https://www.google.com.ar/maps/place/Congreso+de+la+Nación+Argentina/@-34.6048686,-58.3969404,15.5z/data=!4m5!3m4!1s0x95bccac2cd859503:0x75cdafaaef2b946!8m2!3d-34.6098199!4d-58.3926076>

<https://www.youtube.com/watch?v=rgZfmWXTqas>

Los **iconos de Font Awesome** son un conjunto de cerca de 3.000 iconos vectoriales, que en este momento dispone de una colección de **más de 1.000 iconos gratis** y casi 2000 en su versión Pro.

La ventaja de estos iconos frente a otros, es que puedes añadirlos a tus archivos **«tratándose igual que cualquier otra fuente de texto»**. Con lo que podrás **personalizar tus textos con iconos prefabricados**, aportandoles un estilo similar al de los diseñadores gráficos profesionales.

Si te has estado preguntando cómo se consiguen esos iconos para decorar páginas web o presentaciones, la respuesta es que están creados a partir de **fuentes de iconos** como esta.



## Usar los iconos de Font Awesome

Antes de usar los **iconos de Font Awesome** hay que **instalar la fuente en tu sistema operativo**. Para ello puedes descargarla desde su página oficial **fontawesome.com**.

**Se instala igual que una fuente normal** y se usa de la misma manera. Seleccionando la fuente, por ejemplo en el documento de Word que estés redactando en ese momento.

Una vez instalada la tendrás **disponible para todos tus programas** de tratamiento de textos o de diseño gráfico. Cómo no, podrás usar los iconos también en Photoshop.

### Cómo empezar a usar los iconos en un sitio web

Dentro del **body** utilizar los íconos con la etiqueta **i**. Por ejemplo:

```
<i class="fas fa-camera"></i>
```

Para saber qué íconos hay y como tiene que escribirse la etiqueta entrar a:

<https://fontawesome.com/icons>

<https://fontawesome.com/how-to-use/on-the-web/referencing-icons/basic-use>

### Tamaño



```
<i class="fas fa-camera fa-xs"></i>  
<i class="fas fa-camera fa-sm"></i>  
<i class="fas fa-camera fa-lg"></i>  
<i class="fas fa-camera fa-2x"></i>  
<i class="fas fa-camera fa-3x"></i>  
<i class="fas fa-camera fa-5x"></i>  
<i class="fas fa-camera fa-7x"></i>  
<i class="fas fa-camera fa-10x"></i>
```

## Rotación



```
<div class="fa-4x">  
  <i class="fas fa-snowboarding"></i>  
  <i class="fas fa-snowboarding fa-rotate-90"></i>  
  <i class="fas fa-snowboarding fa-rotate-180"></i>  
  <i class="fas fa-snowboarding fa-rotate-270"></i>  
  <i class="fas fa-snowboarding fa-flip-horizontal"></i>  
  <i class="fas fa-snowboarding fa-flip-vertical"></i>  
  <i class="fas fa-snowboarding fa-flip-both"></i>  
</div>
```

## Animaciones



```
<div class="fa-3x">  
  <i class="fas fa-spinner fa-spin"></i>  
  <i class="fas fa-circle-notch fa-spin"></i>  
  <i class="fas fa-sync fa-spin"></i>  
  <i class="fas fa-cog fa-spin"></i>  
  <i class="fas fa-spinner fa-pulse"></i>  
  <i class="fas fa-stroopwafel fa-spin"></i>  
</div>
```

## Formularios

HTML5 proporciona una serie de etiquetas interesantes para utilizar en formularios. Los **formularios** son una forma sencilla de establecer mecanismos para que el usuario pueda introducir información en una página web de forma sencilla e intuitiva, y el sitio web sea capaz de procesarla correctamente y añadirla en una base de datos de la página, enviarla por email o procesarla para mostrar información final al usuario.

### Contenedor de formulario

Para comenzar con la creación de un formulario, el primer paso es indicar una etiqueta contenedora **<form>**, que incluirá toda la información que se quiere recoger en este formulario:

`<form></form>`

De esta forma, se puede incluso crear varias etiquetas **<form>** por página, pudiendo así crear varios formularios diferentes, como por ejemplo, uno para realizar una búsqueda en la página, otro para dejar un comentario y otro para enviar un email al autor. La etiqueta **<form>** dispone de varios **atributos** para utilizar:

Atributo	Valor	Descripción
<b>action</b>	<u>URL</u>	Dirección URL del back-end donde se enviará la información del formulario.
<b>method</b>	<b>get   post</b>	Método HTTP de envío. <b>GET</b> a través de URL, <b>POST</b> para envío extenso.
<b>name</b>	<u>nombre</u>	Nombre del formulario. Útil para procesar posteriormente.

Así pues, un primer formulario base de ejemplo (*aún vacío, sin campos de entrada de*

*datos, sólo con la etiqueta contenedora*) podría ser el que vemos en el siguiente ejemplo:

```
<form name="formulario" method="post" action="/send.php"></form>
```

Nótese que se definen 3 atributos base en **<form>**:

- En **action** se indica la dirección donde se enviarán los datos cuando el usuario pulse el botón de enviar. En caso de omitirse esta información, se enviarían los datos a la propia página actual, volviéndola a cargar.
- En **method** se define el tipo de método HTTP a utilizar. Utilizaremos el método **GET** cuando nos interese que los datos se envíen en la propia URL. En caso contrario, **POST**.
- En **name** el nombre del formulario para cuando necesitemos hacer referencia a él (*generalmente desde Javascript o desde backend*).

## Tipos de datos

Para que un usuario pueda introducir información en un formulario, requiere que se le indiquen una serie de campos que le permitan introducir la información de la forma más cómoda y rápida posible. Para ello, debemos saber a priori que **tipo de dato** le vamos a pedir al usuario (*texto, numérico, fecha, etc...*), y así saber que tipo de campo de entrada de datos es más conveniente colocar.

Información a obtener	Ejemplos	Etiqueta y atributo a utilizar
Información de texto	Nombres, apellidos, direcciones físicas...	<b>&lt;input&gt;</b>  <b>&lt;textarea&gt;</b>
Números o cantidades	Edades, precios, cuantías...	<b>&lt;input&gt;</b> para números
Fechas u horas	Fecha de nacimiento, inicio de evento...	<b>&lt;input&gt;</b> para fechas
Verdadero/falso	Si/No, Opción A/B, ON/OFF...	<b>&lt;input type="checkbox"&gt;</b>
Opción única	(Elegir una opción de 2 ó más posibles)	<b>&lt;input type="radio"&gt;</b>  <b>&lt;select&gt;</b>
Varias opciones	(Elegir varias opciones de 2 ó más posibles)	<b>&lt;select multiple&gt;</b>  <b>&lt;input type="checkbox"&gt;</b>
Opción única abierta	(Elegir una opción o indicar una propia)	<b>&lt;datalist&gt;</b>
Selección de color	(Escoger un color o tonalidad)	<b>&lt;input type="color"&gt;</b>

Selección de archivo	(Escoger un archivo para enviar)	<b>&lt;input type="file"&gt;</b>
----------------------	----------------------------------	----------------------------------

En HTML5, de forma nativa, el navegador proporciona estos **campos de entrada** en los que el usuario puede introducir información específica. Cada uno de ellos tiene sus propias particularidades y características, además de que hay que tener cuidado con algunos, ya que pueden no estar soportados por completo en todos los navegadores.

## Campos de entrada

Una de las etiquetas que más utilizaremos para obtener información a modo de campo de entrada de datos en un formulario es la etiqueta **<input>**. Su funcionalidad más utilizada es la de servir como campo de texto:

```
<form name="formulario" method="post" action="/send.php">
  <!-- Primer campo de entrada de datos -->
  Nombre de usuario:
  <input type="text" name="nombre" value="" />
  <!-- Botón de envío del formulario -->
  <input type="submit" value="Enviar" />
</form>
```

En este ejemplo tenemos un pequeño formulario donde se le pide al usuario su nombre con la etiqueta **<input type="text">** y, posteriormente, se coloca un botón para enviar el formulario con la etiqueta **<input type="submit">**. Aunque vemos que se trata de la misma etiqueta **<input>**, dependiendo del valor indicado en el atributo **type** realizará una tarea u otra.



## La etiqueta "input"

La etiqueta **<input>** tiene una gran cantidad de atributos (*algunos dependiendo del valor de type, que mostraremos más adelante*). Estos son los atributos que podemos utilizar de forma general para prácticamente cualquier campo de entrada de datos con la etiqueta **<input>**:

Atributo	Valor	Descripción
<b>type</b>	<u>tipo de campo</u>	Indica el tipo de campo del que se trata.
<b>name</b>	<u>nombre del campo</u>	Indica el nombre del campo para hacer referencia más tarde.
<b>value</b>	<u>valor por defecto</u>	Indica el valor inicial que tendrá ese campo de datos.
<b>form</b>	<u>nombre del formulario</u>	Asocia este campo de datos con un formulario específico.
<b>placeholder</b>	<u>sugerencia</u>	Indica una sugerencia al usuario antes de escribir.
<b>size</b>	<u>número</u>	Tamaño visual ( <u>número de caracteres</u> ) del campo de datos.
<b>autocomplete</b>	<b>on   off</b>	Activa o desactiva el autocompletado para este campo.

<b>autofocus</b>	-	Establece el foco ( <i>coloca el cursor</i> ) en este campo al cargar la página.
------------------	---	--

El atributo **type** nos permitirá indicar qué tipo de campo de dato mostrará en el navegador (*text, number, date... los veremos más adelante*). Mediante el atributo **name** le daremos un nombre al campo de texto, así cuando enviemos la información del formulario, podremos manejarla desde javascript o desde el back-end haciendo referencia a dicho nombre.

La etiqueta HTML **<input>** puede tomar varios valores diferentes en su atributo **type** para permitir al usuario introducir **información de texto**, además de otra etiqueta denominada **<textarea>** para cantidades de texto más grandes como varios párrafos.

Según nuestras necesidades, tendremos varias opciones:

<b>Tipo de información a obtener</b>	<b>Etiqueta a utilizar</b>
--------------------------------------	----------------------------

Texto alfanumérico libre (texto corto)	<b>&lt;input type="text"&gt;</b>	
Texto para búsquedas	<b>&lt;input type="search"&gt;</b>	
Número de teléfono o móvil	<b>&lt;input type="tel"&gt;</b>	
Dirección URL	<b>&lt;input type="url"&gt;</b>	

Dirección de correo email	<b>&lt;input type="email"&gt;</b>	
Clave o contraseña	<b>&lt;input type="password"&gt;</b>	
Campo oculto (no mostrar al usuario)	<b>&lt;input type="hidden"&gt;</b>	
Texto alfanumérico libre (extenso)	<b>&lt;textarea&gt;</b>	

### Texto alfanumérico corto

En primer lugar, si lo que queremos es guardar información de texto libremente, la opción más habitual es optar por utilizar el campo de entrada de datos **<input type="text">**. Este atributo es el que se utiliza en las etiquetas **<input>** por defecto si no indicamos ningún atributo **type**. Es un campo de texto ideal para fragmentos cortos de información: nombres, apellidos, direcciones físicas, etc...

En resumen, pequeños fragmentos de texto menores a un par de palabras. Un sencillo ejemplo:

```
<form name="formulario" method="post" action="/send.php">
  <!-- Campo de entrada de datos -->
  <input type="text" name="nombre" placeholder="Introduce tu nombre de usuario" />
</form>
```

Obsérvese que además hemos indicado dos atributos adicionales:

- **Autocompletado desactivado:** No aparecerá información escrita anteriormente en ese campo.
- **Placeholder:** Es el texto que aparecerá a modo de sugerencia/ayuda para el usuario (*en gris, ligeramente menos visible que el texto del campo*). En cuanto el usuario escriba en el mismo, el mensaje desaparecerá. Sólo volvería a aparecer si el usuario decide borrar nuevamente la información que ha escrito en ese campo.

## Textos cortos específicos

Por otro lado, en HTML5 se añaden nuevos tipos para etiquetas **<input>** como **search**, **tel**, **url** o **email**. Aunque la idea es la misma que un campo de texto y en principio no notaremos ningún cambio, estos campos específicos tienen como objetivo hacer más sencillo el proceso de introducción de datos por parte del usuario.

Si utilizamos **tel** en vez de **text**, al hacer clic en el campo del formulario desde un smartphone/tablet, se nos desplegará el teclado numérico en lugar del teclado habitual, lo que nos facilitará en gran medida escribir el dispositivo móvil, ya que la intención es utilizar sólo caracteres para números. Lo mismo ocurre con los campos de tipo **email** o **url**, ya que nos mostrará un teclado con botones especiales con **@**, **.com** u otros atajos rápidos:

```
<form name="formulario" method="post" action="/send.php">
  <!-- Campo de entrada de teléfono -->
  Teléfono:
  <input type="tel" name="telefono" placeholder="+XX XXX XXXXXX" />
  <!-- Campo de entrada de email -->
  Email:
  <input type="email" name="correo" placeholder="nombre@dominio.com" />
  <!-- Campo de entrada de url -->
  Página web:
  <input type="url" name="web" placeholder="https://pagina.com/" />
</form>
```

El atributo **type** también permite el valor **search**, que es muy similar a los anteriores. Se utiliza para campos orientados a la búsqueda de palabras o textos en un sitio web, mostrando desde dispositivos móviles una **x** para borrar la palabra escrita con un sólo clic.

## **Campos para contraseñas**

La etiqueta **<input>** tiene el valor **password** en su atributo **type** para crear campos de texto orientados a almacenar información sensible como podrían ser contraseñas, claves o pins numéricos. Esto no lo convertiría en un campo más seguro técnicamente, sino que simplemente muestra puntos o asteriscos en lugar del texto visualmente, protegiéndolo visualmente de ojos ajenos.

Por otro lado también tenemos el tipo **hidden**, que se puede utilizar cuando queremos enviar información en el formulario, pero que no queremos que el navegador lo muestre visualmente en el formulario, como por ejemplo un código numérico o una id informativa:

```
<form name="formulario" method="post" action="/send.php">
  <!-- Campo de entrada de usuario -->
  Usuario:
  <input type="text" name="nombre" placeholder="Por ejemplo, Juan" />
  <!-- Campo de entrada de password -->
  Password:
  <input type="password" name="pass" placeholder="Contraseña" />
  <!-- Campo oculto: Este campo no se verá en el navegador -->
  <input type="hidden" name="informacion" value="72625" />
</form>
```

## Campos extensos de texto

Por último, si queremos indicar un campo para escribir textos más extensos, lo más apropiado es utilizar la etiqueta **<textarea>**, que tiene también algunos atributos adicionales:

Atributo	Valor	Descripción
<b>cols</b>	<u>número de columnas</u>	Número de caracteres que caben en horizontal ( <u>columnas</u> ).
<b>rows</b>	<u>número de filas</u>	Número de caracteres que caben en vertical ( <u>filas</u> ).
<b>wrap</b>	<b>soft   hard</b>	Indica si el texto debe enviarse recolocado o como se ve visualmente.

Esta etiqueta nos permite añadir gran cantidad de texto, ya que las etiquetas **<input type="text">** sólo nos permiten escribir una línea de texto y es muy incómodo si nuestra intención es escribir grandes cantidades de texto. Sus atributos **cols** y **rows** nos permiten aumentar el tamaño de ancho y alto, así como indicar si queremos que se modifique la apariencia visual del párrafo al enviar al formulario con el atributo **wrap**.

Un ejemplo básico de esta etiqueta **<textarea>**:

```
<form name="formulario" method="post" action="/send.php">
  <!-- Area de texto extensa -->
  <textarea name="texto" cols="80" rows="10" placeholder="Escribe aquí el texto...">
    Este es el valor por defecto
  </textarea>
</form>
```

## Casillas de verificación y botones de opción

Si queremos definir opciones que el usuario debe elegir o escoger, en muchos casos lo más apropiado suele ser utilizar **casillas de verificación** o **botones de opción** (*botones de radio*). La diferencia principal entre estos dos tipos de botones es la siguiente:

- Las **casillas de verificación** se marcan para establecer una opción como **activada** (*verdadero*).
- Los botones de radio se marcan para elegir **sólo una opción de varias posibles**.

Tipo de información a obtener	Etiqueta a utilizar	Ejemplo
Casilla de verificación (activado o desactivado)	<code>&lt;input type="checkbox"&gt;</code>	
Botón radio (casilla de opción única)	<code>&lt;input type="radio"&gt;</code>	

## Casillas de verificación

Las casillas de verificación permiten mostrar al usuario la posibilidad de marcar una opción como activada, verdadera o afirmativa, o dejarla sin marcar, lo que representa una opción desactivada, falsa o negativa.

Se puede añadir el atributo **checked** para forzar que esa casilla esté activada inicialmente.

## Botones de radio

Los botones de radio se suelen utilizar cuando el usuario debe elegir **sólo una opción específica** de varias disponibles. Si las opciones son demasiadas, se suele optar por una lista de selección.

Para tener varios botones radio en un mismo grupo, y que sólo se active uno de ellos, el truco está en colocarle el mismo nombre en el atributo **name** a cada opción. Luego, en el atributo **value** le colocamos el valor que habrá seleccionado el usuario de haber marcado esa opción.

Un ejemplo de ambos tipos de botones:

```
<form name="formulario" method="post" action="/send.php">
  <!-- Botones de opción -->
  <p>
    Sexo:
    <input type="radio" name="sexo" value="H" /> Hombre
    <input type="radio" name="sexo " value="M" /> Mujer
  </p>
  <!-- Casillas de verificación (marcada por defecto) -->
  <input type="checkbox" name="empresa" checked />
  ¿Eres autónomo/empresa?
</form>
```

De la misma forma que en las casillas de verificación, si queremos que un botón de radio se encuentre marcado por defecto, le añadimos el atributo **checked**.

Si las casillas de verificación se nos quedan cortas o necesitamos mostrar una lista más extensa de datos, quizás sería conveniente utilizar una **lista de selección**, también llamada frecuentemente **combo** o **lista desplegable**. Estas listas nos permiten mostrar al usuario varias opciones disponibles para que se decanten por una.

## Listas seleccionables



Su forma más básica se compone de una etiqueta contenedora **<select>** que en su interior incluirá varias etiquetas **<option>**, una por opción posible a elegir. Si queremos que una opción de la lista esté marcada por defecto, incluiremos el atributo **selected**:

```
<form name="formulario" method="post" action="http://pagina.com/send.php">
  <!-- Lista de selección -->
  Selecciona la opción deseada:
  <select name="combo">
    <!-- Opciones de la lista -->
    <option value="1">Opción 1</option>
    <option value="2" selected>Opción 2</option> <!-- Opción por defecto -->
    <option value="3">Opción 3</option>
  </select>
</form>
```

## Relación texto-campo

Se podría modificar un aspecto minúsculo de los formularios que podría mejorar su usabilidad. Por ejemplo, si un usuario quiere pulsar sobre uno de los botones de radio para elegir el sexo y pulsa sobre el texto (*Hombre o Mujer*), no ocurrirá nada. El usuario debe pulsar exactamente sobre el botón de radio a elegir.

Sin embargo, si utilizamos la etiqueta contenedora **<label>**, podemos establecer una relación semántica de un texto con un campo de entrada de datos.

```
<form name="formulario" method="post" action="/send.php">
  <label>Nombre de usuario:
    <input type="text" name="nombre" placeholder="Por ej: John Smith" />
  </label>
  <label>
    Edad:
    <input type="number" name="edad" min="18" max="100" placeholder="18" />
  </label>
  Sexo:
  <!-- Relación del texto con el primer botón -->
  <label> <input type="radio" name="sexo" value="H" /> Hombre </label>
  <!-- Relación del texto con el segundo botón -->
  <label> <input type="radio" name="sexo" value="M" /> Mujer </label>
</form>
```