



ELABORACIÓN DE PLANTILLAS Y FORMULARIOS (UF1304)

Familia profesional: Informática y Comunicaciones

[Manual de contenidos](#)

Reservados todos los derechos. El contenido de esta obra está protegido por la ley, que establece penas de prisión o multas, además de las correspondientes indemnizaciones por daños y perjuicios, para quienes reprodujeren, plagiaren, distribuyeren o comunicaren públicamente en todo o en parte, una obra literaria, artística o científica, o su transformación, interpretación o ejecución artística fijada en cualquier tipo de soporte o comunicada a través de cualquier medio, sin la preceptiva autorización.

Todos los nombres propios de programas, sistemas operativos, equipos, hardware, programas de afiliación, páginas web, etc. que aparecen en esta publicación son marcas registradas de sus respectivas compañías, organizaciones y propietarios y tan solo se muestran a modo informativo.

Conzepto Comunicación Creativa

ÍNDICE

1. FORMULARIOS EN LA CONSTRUCCIÓN DE PÁGINAS WEB

- 1.1. Características.
 - 1.1.1. La interactividad de las páginas web.
 - 1.1.2. La variabilidad de los datos de la página web.
 - 1.1.3. El envío de información a servidores. cve: BOE-A-2011-19503
- 1.2. Elementos y atributos de formulario.
 - 1.2.1. Atributos básicos aplicados formularios.
 - 1.2.2. Descripción y definición de los elementos de un formulario.
 - 1.2.3. Utilización de campos y textos.
 - 1.2.4. Etiquetas de los formularios.
 - 1.2.5. Tamaños, columnas y filas de los formularios.
- 1.3. Controles de formulario.
 - 1.3.1. Descripción de los controles de los formularios.
 - 1.3.2. Utilización de botones de acción.
 - 1.3.3. Utilización de lista desplegables.
 - 1.3.4. Utilización de casillas de verificación.
 - 1.3.5. Utilización de campos de textos.
 - 1.3.6. Formularios y eventos. Criterios de accesibilidad y usabilidad
- 1.4. Agrupación de datos.
 - 1.4.1. Adecuación del tamaño del formulario (división en distintas páginas).
 - 1.4.2. Identificación de los campos obligatorios.
 - 1.4.3. Ordenación lógica de la petición de datos.
 - 1.4.4. Información correcta al usuario.
 - 1.4.5. Utilización de páginas de error y de confirmación.

2. PLANTILLAS EN LA CONSTRUCCIÓN DE PÁGINAS WEB

- 2.1. Funciones y características.
 - 2.1.1. Descripción de una plantilla web.
 - 2.1.2. Elementos de una plantilla web.
 - 2.1.3. Estructura y organización de los elementos de las plantillas.
 - 2.1.4. Especificar las zonas modificables de una plantilla y las partes fijas.
 - 2.1.5. Utilización de plantillas.
- 2.2. Campos editables y no editables.
 - 2.2.1. Definir y crear los campos susceptibles de cambios en una plantilla.
 - 2.2.2. Definir y crear los campos no modificables en una plantilla.
- 2.3. Aplicar plantillas a páginas web.
 - 2.3.1. Las plantillas en la web.
 - 2.3.2. Búsqueda de plantillas en la red.
 - 2.3.3. Adaptación de plantillas a páginas web

OBJETIVOS

OBJETIVO GENERAL DEL MÓDULO

Crear y publicar páginas web que integren textos, imágenes y otros elementos, utilizando lenguajes de marcas y editores apropiados, según especificaciones y condiciones de "usabilidad" dadas y realizar los procedimientos de instalación y verificación de las mismas en el servidor correspondiente.

OBJETIVOS ESPECÍFICOS

C1: Confeccionar plantillas para las páginas web atendiendo a las especificaciones de diseño recibidas.

CE1.1 Describir las características que ofrecen las plantillas web en la elaboración de páginas con idéntico diseño.

CE1.2 Describir las utilidades que ofrecen las herramientas de edición de páginas web para crear plantillas, teniendo en cuenta las especificaciones técnicas de la herramienta.

CE1.3 Crear plantillas web con herramientas de edición, de acuerdo a un diseño especificado:

- Identificar las regiones editables y no editables que forman la plantilla.
- Insertar los elementos y asignar los atributos especificados.
- Aplicar criterios de «usabilidad» y accesibilidad.
- Aplicar una plantilla creada a una página web.
- Probar la página web con la plantilla asociada utilizando varios navegadores web.
- Documentar la plantilla realizada.

C2: Crear formularios e integrarlos en páginas web para incluir interactividad en las mismas, siguiendo unas especificaciones funcionales recibidas.

CE2.1 Identificar las etiquetas y los atributos que se utilizan en la creación de los formularios, teniendo en cuenta las especificaciones del lenguaje de marcas.

CE2.2 Describir las etiquetas y los atributos que se utilizan para definir los controles que forman los formularios en función de las interacciones a manejar.

CE2.3 Realizar páginas que incorporan formularios para interactuar con el usuario, según un diseño especificado:

- Identificar los controles que hay que crear y colocarlos dentro del formulario.
- Asignar las propiedades especificadas a los controles insertados.
- Asignar las propiedades al formulario (acción, método y tipo de codificación).
- Aplicar criterios de «usabilidad» y accesibilidad.
- Probar la página y el formulario utilizando varios navegadores web.
- Corregir los posibles errores que surjan en la elaboración de la página y el formulario.
- Documentar la página realizada.

CONTENIDOS

1. FORMULARIOS EN LA CONSTRUCCIÓN DE PÁGINAS WEB

Una página web es un “simple” documento que está compuesto de manera que los navegadores web pueden entenderlo y, de ese modo, muestre los elementos de ese documento mediante un monitor de ordenador o cualquier tipo de dispositivo digital.

1.1. Características

Aunque sepa que lo que se muestra en sus pantallas es la página web, no debe olvidar que esos archivos están, realmente, desarrollados y alojados en otro ordenador. Este tipo de archivos están desarrollados, normalmente, en lenguaje HTML junto con otro tipo de lenguajes *scripts* que permiten darle dinamismo. El lenguaje HTML es el que permite que pueda navegar a través de los enlaces que hay en una página.

En el nivel más básico tiene los dispositivos locales, que son los que usan los usuarios para conectarse a Internet. Estos dispositivos pueden encontrarse en una casa, una oficina, etc.

El siguiente nivel lo componen los demás recursos de la red local. Por ejemplo, una impresora que esté conectada a la red, un switch que se encargue de distribuir la señal entre los dispositivos conectados a la red, etc.

En un nivel más elevado, se encuentra el router, que es el intermediario entre su ISP e Internet. Gracias al router puede enviar información de una punta del mundo a la otra en segundo. Esto sucede porque el router es el encargado de enviar y recibir paquetes de datos de Internet. Estos paquetes de datos recorrerán las líneas de todo el mundo en cuestión de segundos hasta llegar a otros.

En el último nivel, se encuentra el servidor que es el encargado de recibir la señal que pasando por todos los dispositivos que se conecten al mismo. Los servidores recibirán datos, lo procesarán y enviarán una respuesta al dispositivo que hizo la petición de esos datos. De esta manera es como mostrará una página web.

Por otra parte, los navegadores web se encargan de coordinar los recursos y elementos de la web que utiliza en la misma dependiendo de cómo estén programados; es decir, si la página contiene *scripts*, hojas de estilo, imágenes, etc.

Una curiosidad sobre el protocolo World Wide Web es que no se empezó a utilizar hasta 1989, aunque se comenzó a utilizar en 1991, sin embargo, el origen de Internet fue en 1969.

En su origen, las páginas web eran totalmente estáticas; es decir, se mostraban idénticamente a cómo eran almacenadas en el sistema.

Sin embargo, al cabo de unos años, se empezaron a crear páginas web más dinámicas, es decir, que detrás de los que se muestra en la página va a haber código script para darle funcionalidad a la estructura primitiva de la propia página.

Debido a la evolución en el desarrollo y diseño web y debido a la necesidad de recoger y manipular los datos que los usuarios introducían para crear más interactividad en la web y bases de datos, se comenzaron a utilizar los formularios web de manera básica en HTML.

Se define formulario web como una página web encargada de recoger los datos que el usuario introduce mediante un formato preestablecido por el propio formulario, y que, posteriormente, envía a un servidor para que los datos sean procesados.



Todos los datos que introduzcan los usuarios en un formulario, obligatoriamente, se tienen que enviar a un servidor donde sean procesados.

TOME NOTA

A los formularios se les denomina así porque se asemejan mucho a los formularios clásicos de papel donde cualquier persona rellena sus datos para solicitar un proceso.

Además, los formularios web se adaptaron de forma que tengan una estética similar a los formularios de toda la vida donde se pueden encontrar cajas de texto y opciones. En el caso de los formularios web, puede encontrar casillas de selección, botones y cajas de texto. Por último, los datos se almacenarán en bases de datos.

Por ejemplo, los formularios web se pueden usar para suscribirse al boletín informativo de una página, contactar con un servicio técnico en la web de un e-commerce, mandar un email, etc.

Al principio, los primeros formularios web estaban programados en HTML, que es el lenguaje base de cualquier página web. Como ya sabe, este lenguaje está basado en el uso de etiquetas para, de este modo, estructurar los elementos en el documento web. Se utilizarán, normalmente, etiquetas de apertura y de cierre.

En cambio, actualmente, puede encontrar los formularios programados en otros tipos de lenguajes como Java, .NET o Perl. También, puede encontrarlo programado en lenguajes *script* como, por ejemplo, JavaScript o, utilizar arquitecturas cerradas como, por ejemplo, SAP.

Más adelante, profundizará en los elementos y características comunes que tienen todos los tipos de formularios, que serán las que programen y administre cuando cree su primer formulario.

En definitiva, los formularios permiten que los usuarios introduzcan sus datos en el servidor para que se procesen, esto permite crear páginas web mucho más interactivas y dinámicas donde un aspecto

muy importante será la recolección de los datos del usuario. Debe tener siempre presentes las características que ha visto hasta ahora de los formularios, ya que son las más importantes.

1.1.1. La interactividad de las páginas web

La interactividad fue el concepto más importante que introdujo el uso de los formularios en las páginas web.

Puede describir la interactividad como la capacidad cambiante que tiene un medio para brindarle a los usuarios una mayor libertad en la selección e interacción con los contenidos, y en la comunicación y expresión.

En resumen, la interactividad es la capacitación que tiene el receptor de un mensaje, el cual se relaciona directamente con el receptor, y donde el emisor del mensaje es el que tiene los límites de esa comunicación.

Es importante saber que, en un principio, la interactividad supuso un gran esfuerzo y trabajo para el desarrollo informático; ya que, se tuvo que planificar una navegación correcta entre los diferentes vínculos, textos e imágenes, de modo que el usuario pudiera sentir que tenía realmente el control de las aplicaciones que utilizaba. Por ello, lo que ve hoy día como algo común, en su día fue un gran reto de diseño e ingenio.

Podría calificar la interacción en varios niveles. Sin embargo, decir que la interactividad, en informática, es la capacidad de interacción que existe entre el usuario y la máquina. Vea a continuación los niveles de interactividad que puede clasificar:

Interactivo	Existe una relación entre los elementos y todos ellos se relacionan entre sí.
Reactivo	Es cuando un elemento solo se relaciona con el elemento anterior a él.
No interactivo	Es cuando un elemento no se relaciona con ningún elemento.

En los inicios del desarrollo web, las páginas eran totalmente estáticas, aunque podría decir que no tenían el nivel completamente de **no interactiva**; puesto que, tenían textos que se relacionaban con otros textos previos. Sin embargo, tampoco eran totalmente **interactivas**, puesto que, solo se relacionaban directamente con un enlace previo.

En ese momento, una página web era, tan solo, una agrupación de información a la que tenía acceso el usuario para solo poder leerla, pero sin poder modificarla o introducir datos.

La revolución principal para solucionar esta situación fue la de integrar formularios a las webs. Sin embargo, esta evolución no solo fue exclusivamente por los formularios, cosa que verá a lo largo de este tema.

Hoy en día la utilización de páginas webs es totalmente dinámica, siendo el entorno de Internet cada vez más interactivo. De hecho, la mayor parte de usuarios diría que hoy día la primera característica que destacarían de Internet es la interactividad.

Ciertamente, no existe hoy por hoy ningún otro medio que ofrezca un nivel de interactividad tan alto como lo hace Internet. Puede decir que los usuarios que antiguamente escuchaban la radio o veían la tele, tenían la opción de poder cambiar de canal. Sin embargo, este nivel de interactividad es tan bajo que sería ridículo compararlo con la interactividad que ofrece Internet.

De facto, las televisiones modernas o, como se suele llamar, Smart TV vienen preparadas para conectarse a Internet y ofrecen servicios interactivos como un buscador propio, plataformas de streaming, juegos, etc. Y, también puede ver películas en las que interactuar con la historia.

La interactividad de la que se habla no está tan condicionada con el desarrollo de las aplicaciones multimedia, sino que está mayormente condicionada por el desarrollo de las herramientas sociales y de la transmisión de los datos y la información entre las webs y los usuarios.

Al principio, los formularios solo eran una herramienta para enviar datos a un servicio. Sin embargo, hoy día esto ha evolucionado y los formularios se han convertido en herramientas que pasan los datos a una base de datos para crear encuestas, comentarios en un blog, compras en un e-commerce, etc.

Hoy por hoy, la interacción en Internet ha llegado a tal punto que ya no solo son los usuarios los que interactúan con las aplicaciones y páginas web, sino que también interactúan entre los propios usuarios.

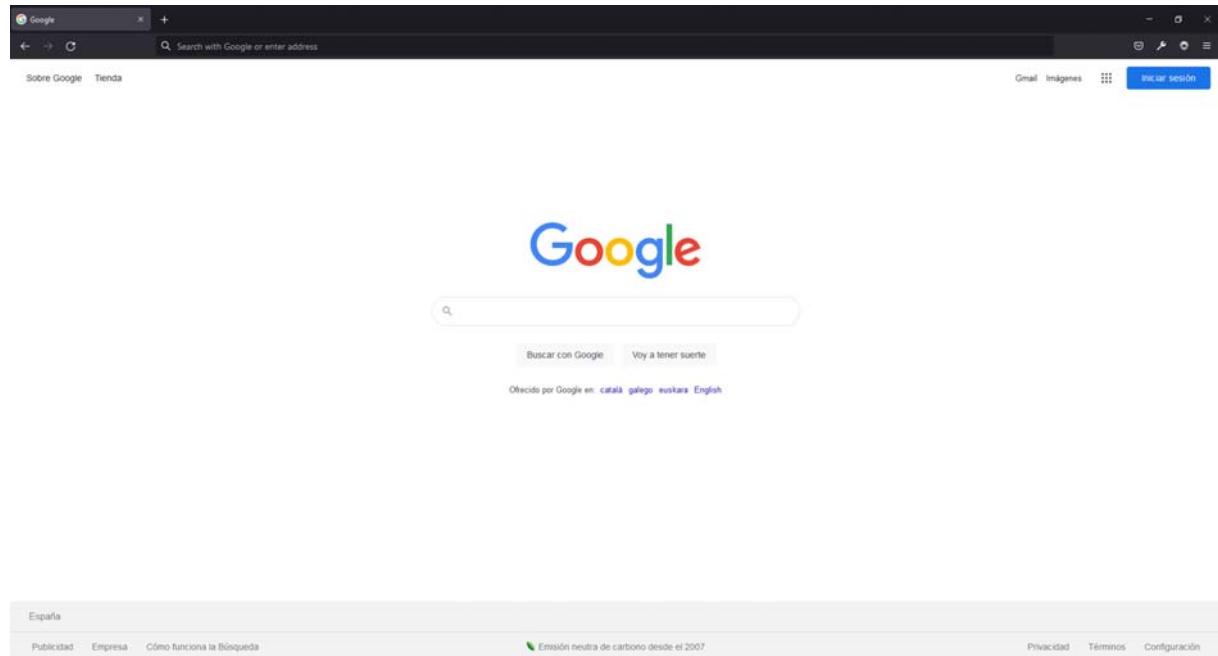
La interactividad que existe hoy con Internet es tan fuerte entre los usuarios y las propias herramientas que puede decir que cada usuario comprende cada día mejor cómo funcionan los buscadores de Internet y, por ello, son capaces de encontrar cierta información de manera rápida y eficaz.

La publicidad siempre ha sido unidireccional antes de que existiera Internet. Los anuncios en los periódicos, carteles publicitarios, anuncios de televisión, etc. En cambio, hoy por hoy, la comunicación es totalmente bidireccional, ya que, consta de un emisor y un receptor funcionando hacia las dos direcciones. Además, también puede destacar que la información puede viajar de un emisor a otro emisor, por lo tanto, esto hace que los intermediarios de la información no sean tan importantes como antes.

1.1.2. La variabilidad de los datos de la página web

En este momento, debe saber ya la importancia que tiene una web dinámica frente a una estática. Además, también debe tener en cuenta la funcionalidad que tiene un formulario en una web. En este epígrafe, se va a adentrar en uno de los puntos más importantes que tienen relación con los formularios, la variabilidad dinámica de los datos que poseen las páginas web.

Para proceder con esto, verá el formulario más conocido de Internet, pero que a la vez es el que pasa más desapercibido. Este formulario lo utilizan cientos de veces al día y es el buscador de Google:

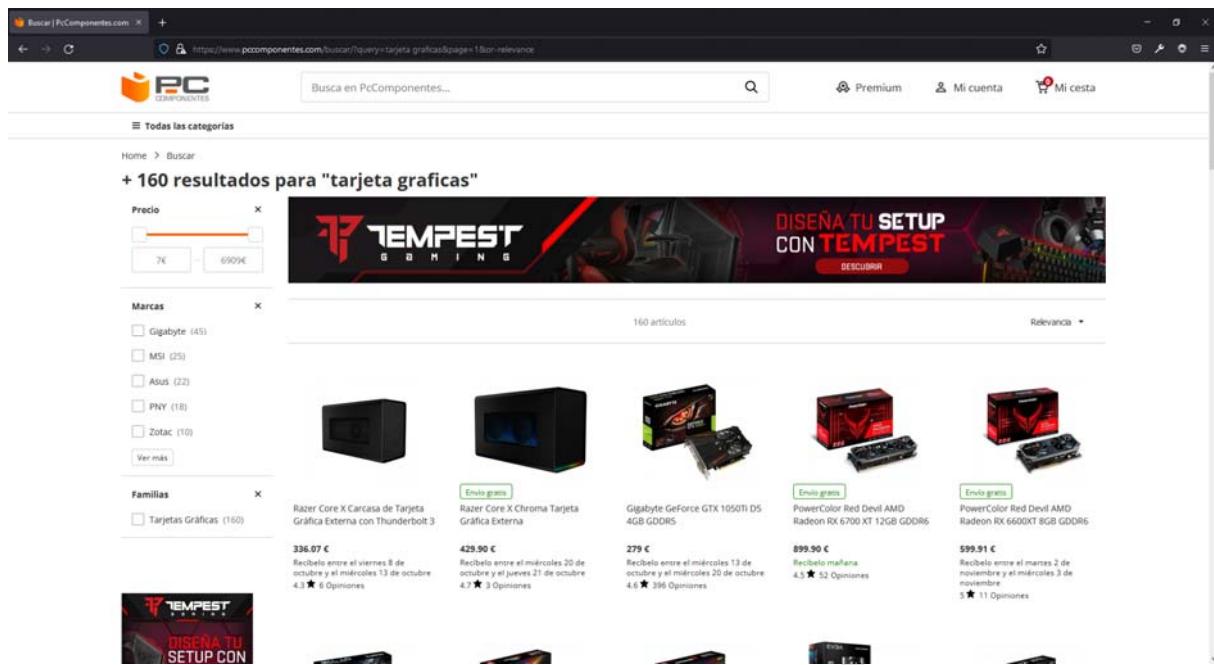


Esta caja de texto unida a los dos botones que aparecen en su parte inferior es un formulario web. Este formulario tiene la diferencia de que, mediante herramientas de programación, su utilidad se centra en la búsqueda de información gracias a un algoritmo que se encarga de ello.

Una curiosidad sobre Google es que tiene dominios registrados con nombres muy parecidos, de manera que, si un usuario escribe mal la palabra Google, el navegador pueda redirigirlo hacia Google. Mediante este pequeño formulario, puede ver la variabilidad de los datos que puede mostrar una página web en relación a los datos introducidos por el usuario.

Un ejemplo de ello son los datos que muestra una web que van a depender de si el usuario está registrado en la web o no. Y cuando registra en una web lo normal es introducir los datos necesarios en el formulario. Una vez haya completado el registro debe llenar otro formulario para iniciar sesión.

Otro de los ejemplos que puede ver es en las webs e-commerce, que dependiendo de cómo vaya utilizando los buscadores, se irán mostrando los diferentes productos. Además, cuando vaya a comenzar con el proceso de compra, también debe llenar datos como, por ejemplo, la dirección de envío, el número de tarjeta, etc.



La importancia de los formularios en el desarrollo de páginas web es total, puesto que, debido a los formularios se alcanza la interactividad y la variabilidad de datos dependiendo de la información que haya introducido el usuario, por lo tanto, los formularios son una herramienta totalmente necesaria en la creación de una web.

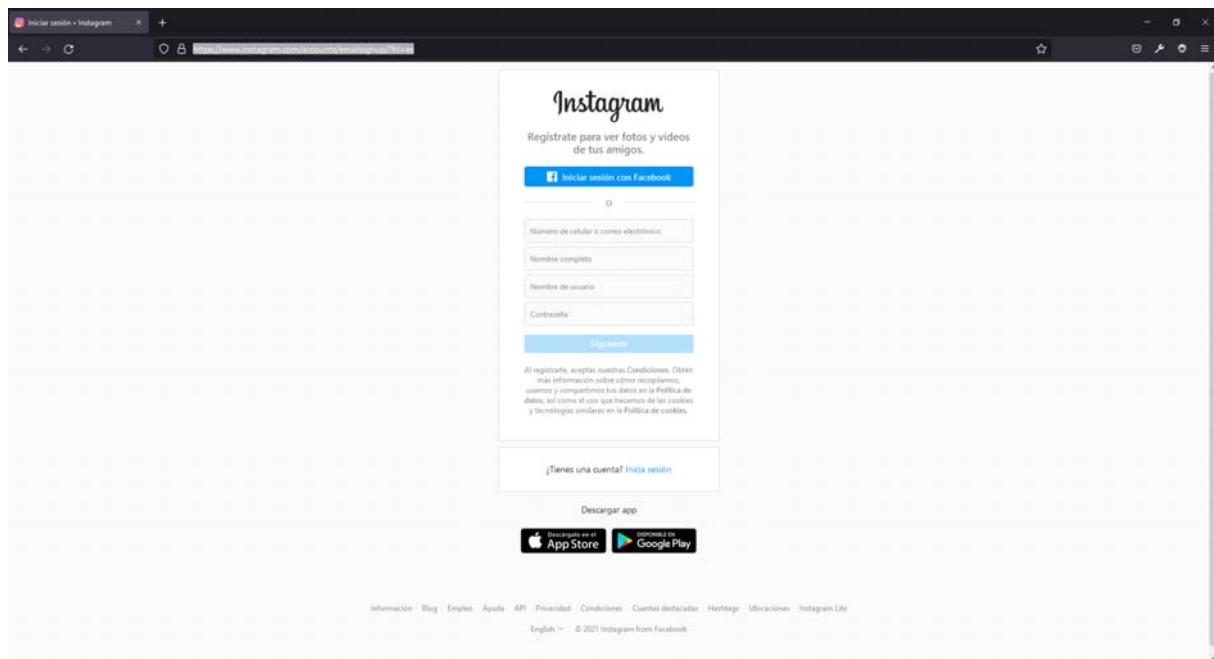


Debe tener en cuenta que los formularios son los encargados de darle a la página web un entorno abierto y cambiante. Para ello, se encargarán de mostrar datos que dependerán de lo que haya introducido previamente el usuario.

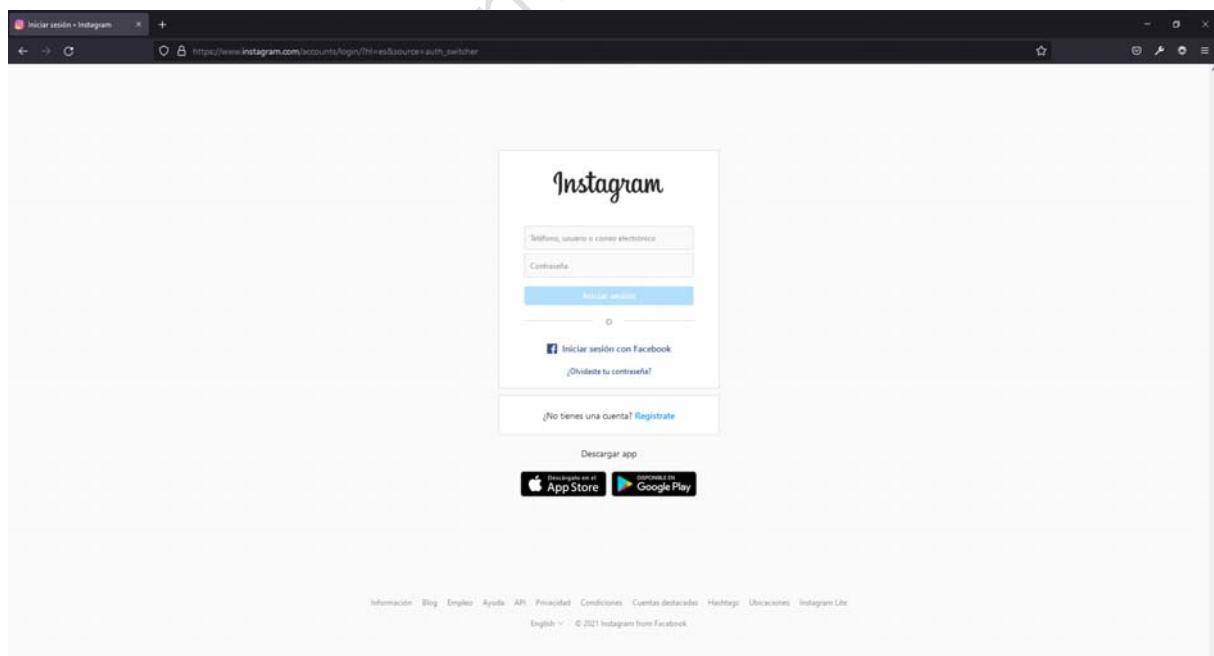
TOME NOTA

A continuación, verá dos ejemplos donde puede determinar la importancia de los formularios web y la variabilidad de los datos:

Vea el formulario de Instagram. Si no existieran los formularios, tampoco existiría esta red social ni ninguna otra. Sin embargo, lo más importante de esto es que imagine cómo sería si no tuviera que llenar ningún formulario en una red social y cualquiera pudiera acceder a sus archivos o a su ordenador mediante el router y hacerse pasar por otros.

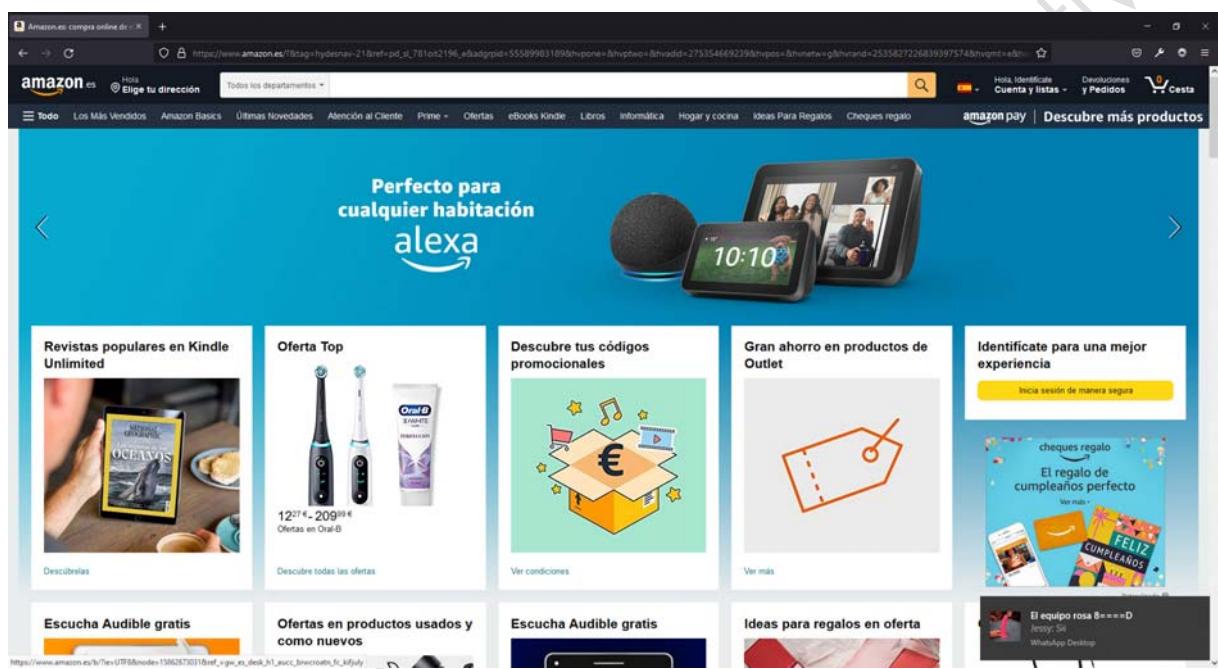


Esta imagen muestra el formulario de registro de Instagram. Como puede observar, tiene las opciones de llenar sus datos o, en caso de que tenga cuenta de Facebook, tomar los datos que tiene en Facebook. También, si pincha en el link de Inicia Sesión, la página redirigirá a otro formulario, pero esta vez de ingreso a Instagram. Por lo tanto, ambos formularios funcionan con algoritmos diferentes, el de registro lo que hará es ingresar los datos que introduzca a una base de datos y el formulario de ingreso, tomará los datos de una base de datos para ver que concuerdan con los datos introducidos. Vea el formulario de ingreso:



Una curiosidad sobre las redes es que una de cada trece personas del planeta tiene cuenta en una red social, por lo tanto, estas personas habrán tenido que introducir sus datos en el formulario de registro.

El segundo ejemplo que verá es el formulario de búsqueda de productos dentro de una página web en este caso Amazon. Si no existieran este tipo de formularios de búsqueda, para comprar un artículo en concreto, debe recorrer las cientos o miles de páginas que tiene Amazon para encontrar el artículo deseado. De hecho, por muy buena organización que tuviera una página web, en un sitio como Amazon sería muy complicado, por no decir imposible, encontrar el artículo concreto que desea, puesto que, hay miles de artículos más.



En esta imagen de la tienda de Amazon, puede ver el formulario de búsqueda en la parte superior de la web. Si esta página no contara con este formulario de búsqueda, sería prácticamente imposible encontrar el artículo concreto que quiera.

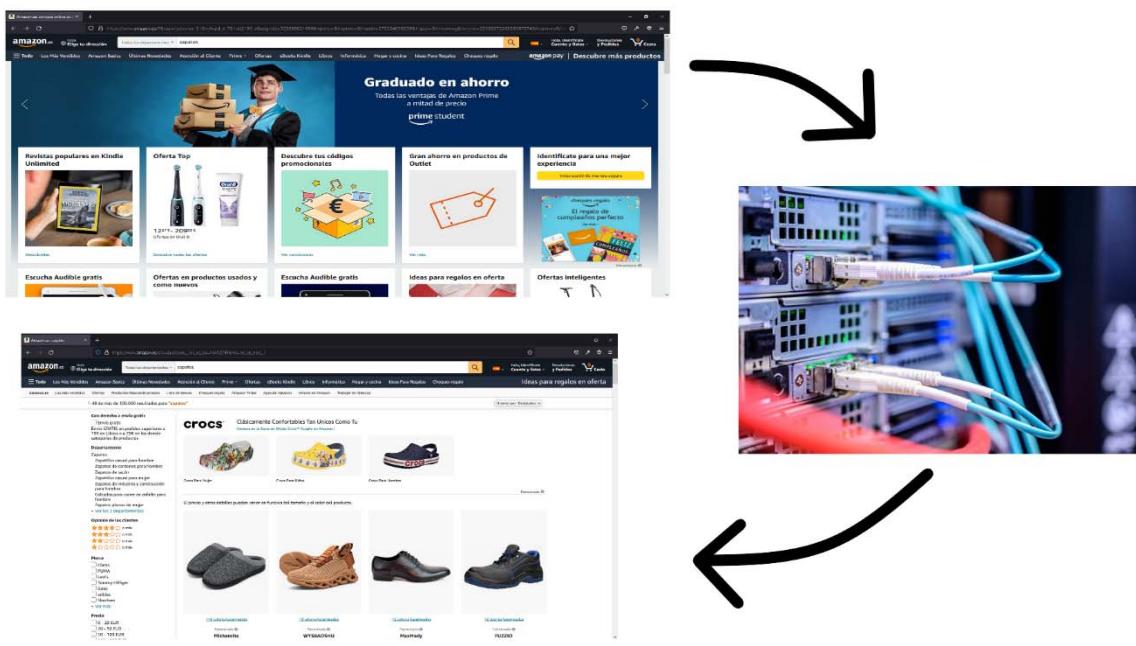
Debe tener en cuenta que páginas como Amazon se venden miles de productos, desde electrodomésticos, filtros, fontanería, etc. Hasta comida. Por lo tanto, sin la ayuda de ese formulario de búsqueda, encontrar un objeto concreto y, sobre todo si quiere compararlo con otros objetos del mismo tipo, sería casi imposible.

1.1.3. El envío de información a servidores

Ha visto anteriormente que cuando un usuario rellena un formulario y lo envía, lo que está haciendo realmente es enviar esos datos desde su dispositivo a un servidor que se encuentra en Internet.

Más adelante, verá el proceso que ocurre cuando el usuario envía sus datos. Sin embargo, ahora, va a ver el flujo que sigue esta información, ya que, siempre va a seguir los mismos pasos:

- Primeramente, el usuario introduce sus datos en el formulario y pulsa el botón enviar, para mandar los datos.
- A continuación, el navegador que el usuario esté utilizando, mandará esos datos al servidor con el que esté conectado el formulario.
- Más adelante, el servidor recibe los datos que se han enviado y los procesa, enviando el resultado del proceso de esos datos al navegador del usuario.
- Por último, el navegador muestra ese resultado al usuario.



En esta imagen puede ver el flujo que sigue la información cuando rellena datos en un formulario y los envía. Primeramente, el usuario introduce los datos, el formulario envía los datos a un servidor, y cuando los procesa el servidor envía, en este caso, la página web donde vienen los artículos asociados con los datos que el usuario ha introducido y el navegador los mostrará al usuario.

Para ser más precisos, lo que realmente envía el navegador al servidor, cuando el usuario introduce sus datos y le da al botón de enviar, son los nombres de cada uno de los elementos que contiene el formulario junto con el valor asociado que ha introducido el usuario.

Como curiosidad, debe saber que el 1981 se creó el primer servidor, que se llamaba IMB VM Machine, y permitió crear grupos de correo electrónico. Además, ya estaba preparado para soportar mensajes spam y trolls de Internet.



TOME NOTA

Normalmente, los datos del formulario suelen ser procesados en el servidor mediante lenguajes que utilizan los servidores. Sin embargo, mediante lenguajes como JavaScript, se pueden modificar de manera dinámica estos datos en el lado el cliente.

Cuando un programador crea un formulario, deberá saber completamente qué función va a tener ese formulario y adónde se enviarán los datos que se introduzcan en el mismo.

Por otra parte, verá más adelante que podrá elegir varias opciones para enviar la información. Estas dos opciones van a depender de diferentes factores:

- Si el formulario trabajará con una base de datos o no.
- El tamaño que tendrá el formulario.
- Si el formulario va a contener información sensible como contraseñas, o datos personales.

Vea este ejemplo de formulario:

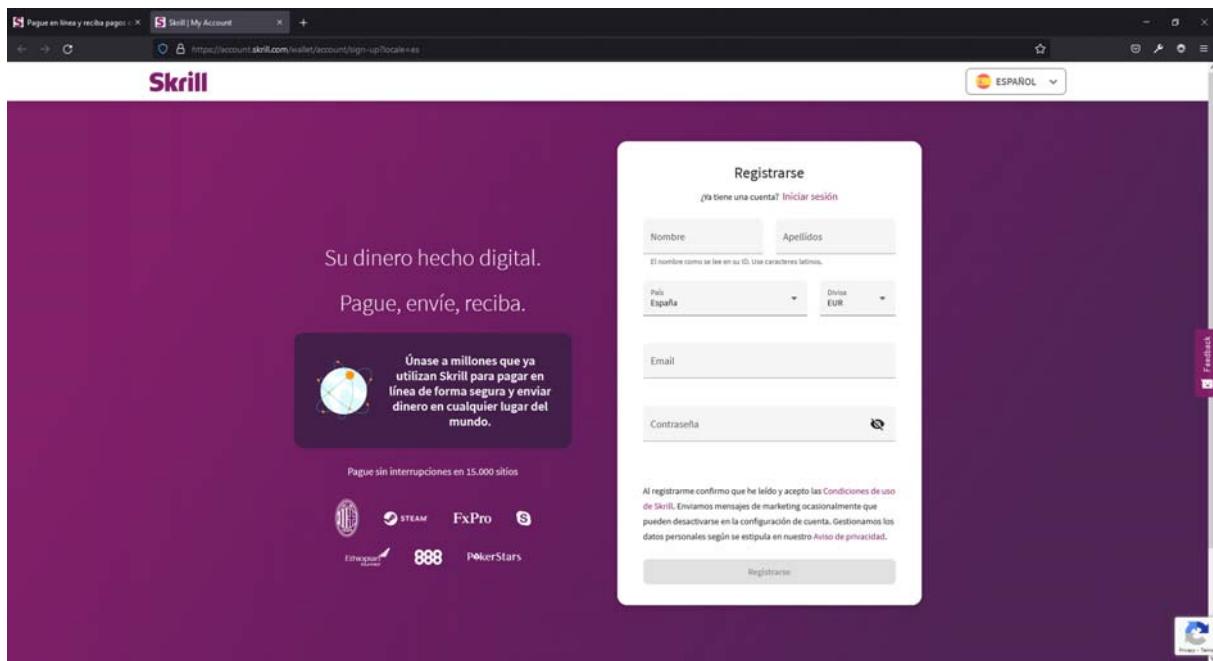
The screenshot shows a forum post titled "Necesito ayuda por favor!" on the "JavaScript - Necesito ayuda por favor!" board. The post was published by "Piezas" and has 21 interventions. The code provided is:

```
1 function filtrarPares(v) {
2     let resultado = [];
3     v.filter((valor, i) => {
4         resultado[i] = v[i] / valor === 5;
5     });
6     return resultado;
7 }
8 let v = [2, 7, 20, 21, 75, 110, 9000, 54];
9 let resultadosFiltrar = filtrarPares(v);
10 console.log(resultadosFiltrar);
```

The sidebar on the right includes a poll titled "GRATIS QUIERO POLLER" and links to other sections such as "Otras secciones relacionadas con JavaScript", "Cursos de JavaScript", "Temas de JavaScript", "Códigos de JavaScript", and "Chat de JavaScript".

Esta captura de pantalla muestra un formulario que se utiliza para responder en un foro de JavaScript. Por lo tanto, este sería un ejemplo de formulario que guarda información no demasiado sensible. Por ello, no se deberá hacer mucho hincapié en ocultar o encriptar este tipo de información a la hora de enviar los datos al servidor.

Por otra parte, tiene el siguiente formulario:



Por otra parte, tiene este formulario de una plataforma de pago digital llamada Skrill, que es parecido a Paypal, y que para registrar debe poner datos personales como, por ejemplo, su contraseña, nombre y apellidos. Además, después, debe introducir datos bancarios para poder operar con su dinero. Por lo tanto, este tipo de formulario deberá enviar la información al servidor de manera totalmente encriptada. Y se guardará esa información también encriptada en la base de datos.

Debe tener en cuenta que, por ejemplo, un formulario de búsqueda como puede ser el buscador de Google, no va a almacenar la información encriptada, ya no solo porque esa información no sea sensible, sino porque ese formulario tiene que ser rápido en el intercambio de información y, además, va a tener mucho tráfico de usuarios.

De todos modos, el método que elija para trabajar con la información debe ser el más adecuado para la seguridad de la información y para el funcionamiento de los lenguajes de programación que se vayan a utilizar para manipular esa información. En lo que más debe enfocarse es en el volumen de información que se va a enviar en el formulario.

Por otro lado, algunos formularios también pueden enviar archivos al servidor. En este tipo de formularios, se deberá especificar qué tipo de codificación se va a utilizar al enviar esos archivos al servidor. Y, además, tendrá que tener presente el ancho de banda que necesita, ya que, es el que va a soportar todo el tráfico del servidor. En caso de que no planifique esto correctamente, el rendimiento que tendrá el formulario será nefasto y la página web era demasiado lenta.

1.2. Elementos y atributos de formulario

Puede crear formularios utilizando, tan solo, código HTML mediante la etiqueta <form>.

Para utilizar esta etiqueta, deberá añadirle diferentes atributos para definir los diferentes valores del propio formulario. Además, deberá tener elementos dentro del formulario que representen los datos que el usuario puede introducir en el mismo. Por otro lado, también especificará el método con el que se enviarán los datos al servidor.

En caso de que quiera definir los elementos dentro de un formulario, se utiliza la etiqueta <input>, que dependerá de los atributos que le asigne permitirá definir diferentes tipos de elementos como, por ejemplo, cuadros de texto, botones, casillas de verificación, etc.



Debe tener en cuenta que las dos etiquetas más importantes para la creación de formularios en HTML son la etiqueta <form> que es la que define el inicio y el final del formulario, y la etiqueta <input> que es la que utiliza para definir los elementos inter del mismo.

TOME NOTA

Vea a continuación el ejemplo de código de un formulario básico. En este ejemplo, podrá ver de manera clara las etiquetas <form> para establecer los atributos del formulario y la etiqueta <input> que utiliza para definir dos elementos en el formulario, un botón de enviar y un cuadro de texto. Vea el ejemplo:

```
<!DOCTYPE html>

<html>

  <head>
    <title>Formulario básico</title>
  </head>
  <body>

    <form action="http://www.ejemplo.com/form.php" method="post">

      Nombre: <input type="text" name="nombre" value="" /><br/>

      <input type="submit" value="Enviar">

    </form>
```

</body>

</html>

A continuación, vea lo que muestra el navegador:

Nombre:

A partir de este punto, va a ver muchos ejemplos con código sobre los formularios y sus funciones. De ese modo, se familiarizará con el código HTML y, sobre todo, con la creación de formularios en HTML. Además, lo más óptimo es que se realicen estos ejemplos otros solos para que, posteriormente, sea más sencillo crear un formulario sin ningún tipo de guía.

Como ya sabe, en la etiqueta `<form>` puede utilizar muchos atributos. Vea a continuación los atributos que pueden utilizar la mayor parte de las etiquetas HTML:

- Atributos internacionales que indican el idioma del elemento.
- Atributos básicos para establecer identificadores en los elementos del código o para establecer estilos en los elementos.

Si la página utiliza JavaScript, atributos de eventos para realizar funciones dinámicas con los elementos.

Atributos de foco para aplicar accesibilidad web.

The screenshot shows a Wikipedia article page in Arabic. At the top, there's a search bar and a navigation menu. Below the title 'مشروع موسوعة فرة سلطان الجميع تحريرها' (Arabic version of the project), there's a section titled 'محتوى مفهومي - المصطلحات - فهرس المحتوى' (Conceptual content - Terminology - Content outline). The main content area displays several examples of HTML code snippets, each with a title and a detailed description. One example is about 'العنوان' (The Title) which discusses the concept of the title in HTML. Another example is about 'العنوان المترافق' (Consistent title) which discusses how the title should be consistent across different parts of the page. The code snippets include attributes like 'name' and 'id'.

Esta imagen muestra una página web que está preparada para soportar todos los idiomas del mundo. Esta página es Wikipedia en árabe, pero como ya sabrá, puede cambiar el idioma de esta página a cualquier idioma del mundo.

Todos los atributos son importantes y en algunos formularios, no deberá dejarlos de lado sin darle la importancia que realmente tienen.

A continuación, va a ver los atributos únicos y más importantes que tiene la etiqueta <form>.

Vea la siguiente tabla:

Atributo	Funcionalidad
Action	Indica la url donde enviar los datos del formulario.
Target	Indica donde muestra la respuesta que se recibe del servidor.
Method	Indica el método http a utilizar para enviar el formulario
Name	Indica el nombre del formulario
Enctype	Indica el tipo de codificación a utilizar para enviar los datos al servidor.
Accept	Especifica los archivos que acepta el servidor.
Autocomplete	Activa o desactiva la funcionalidad de autocompletar el formulario.
Accept-charset	Indica la codificación de caracteres a utilizar para el envío del formulario.
Novalidate	Si el atributo está presente indica que los datos no deben validarse cuando se envían.

Action

Absolutamente todos los formularios deben llevar un atributo action y que esté configurado de manera correcta, puesto que, si esto no sucede así, los datos no llegarán a su destino y, por lo tanto, no serviría para nada. La url que se establece en este atributo puede ser de dos tipos relativa o absoluta. Vea la diferencia:

En caso de que la url sea relativa, quiere decir que el destino donde enviará el formulario, se encuentra ubicado en el mismo espacio donde se ubica la página web. Por lo tanto, solo deberá

insertar la dirección del propio archivo que estará en una carpeta que se sitúe en el mismo espacio. De ese modo, se ahorra tener que buscar toda la dirección del archivo.

Por otro lado, la url absoluta, cuenta con la ventaja de que no va a importar el nivel en el que se encuentre el formulario o que el destino de los datos sea una web diferente, ya que, tan solo se introduce la dirección completa de la ubicación del archivo. Entonces cuando envíe el formulario irá justamente a ese lugar.

Sin embargo, si la url que ha especificado no es válida, el formulario dará un error cuando se pulse el botón enviar.

A continuación, va a ver algunos ejemplos. Empieza por ver un formulario que utiliza la url absoluta:

```
<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>Formulario url absoluta</title>

    </head>

    <body>

        <form action="http://www.ejemplo.com/form.php" method="post">

            Usuario: <input type="text" name="usuario" value="" /><br/>

            Contraseña: <input type="password" name="password" value="" /><br/>

            <input type="submit" value="Enviar">

        </form>

    </body>

</html>
```

A continuación, va a ver cómo muestra el navegador este código:

Usuario:

Contraseña:

Lo siguiente que verá es un ejemplo de un formulario que utiliza una url relativa:

```
<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>Formulario url relativa</title>

    </head>

    <body>

        <form action="form.php" method="post">

            Usuario: <input type="text" name="usuario" value="" /><br/>

            Contraseña: <input type="password" name="password" value="" /><br/>

            <input type="submit" value="Enviar">

        </form>

    </body>

</html>
```

Vea a continuación una imagen de cómo muestra el navegador este ejemplo:

Usuario:

Contraseña:

Como puede ver, el código de ambos ejemplos es prácticamente el mismo, lo único que cambia es la url. Y el cómo se muestra el formulario en el navegador es idéntico. Sin embargo, la funcionalidad que tienen ambos a la hora de enviar los datos, es bastante diferente.

De todas maneras, la funcionalidad del atributo action, esté expresado con url absoluta o relativa es tan solo de diseño funcional y no de diseño visual, ya que, este proceso es transparente para el usuario.

En caso de que esté creando una web donde la velocidad es lo más importante, y los usuarios no cuentan con una buena conexión a Internet, lo mejor es que utilice url relativas.

En cambio, si los usuarios disponen de una buena conexión y tiene una web con mucho ancho de banda que pueda absorber todo el tráfico, puede utilizar url absolutas.

Hoy por hoy, el uso de la url absoluta es el más utilizado, aunque en los formularios más básicos puede utilizar la relativa.

	<p>TOME NOTA</p> <p><i>Como ya sabe, el atributo action es totalmente obligatorio en la etiqueta <form> para saber dónde enviar los datos del formulario. Sin embargo, en HTML5, puede añadir el atributo action en el botón de enviar del formulario, por lo tanto, no sería necesario añadir el atributo action en <form>.</i></p>
--	---

Method

Utilice este atributo para indicar el método http que va a utilizar para enviar el formulario. Este atributo puede tener dos valores: get y post.

Comience con el valor get. Este valor envía los datos como variables al final de la url que se especifica en el atributo action.

Utilizar este método es práctico para formularios pequeños y que no deban estar encriptados como, por ejemplo, un buscador web. Además, es el método más correcto si quiere enviar datos con un formulario donde no deben ser borrados o añadidos en una base de datos.

A continuación, va a ver el código de ejemplo de un formulario que utiliza el atributo method con el valor get:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>

<meta charset="UTF-8">

<title>Formulario method con valor get</title>

</head>

<body>

<form action="form.php" method="get">

    Buscar en la página:

    <input type="text" name="buscar" value="" />

    <br/>

    <input type="submit" value="Buscar" />

</form>

</body>

</html>
```

Vea lo que muestra el navegador:

Buscar en la página:

Si introduce en esa caja de texto una palabra y pulsa el botón Buscar, puede ver como en la url se añade esa palabra al final. De esta manera, se enviarían los datos al servidor. Vea el ejemplo:

ⓘ Archivo | D:/VSCode%20Ejercicios/Ejemplo/form.php?buscar=Ordenador

Por otra parte, va a investigar el valor post. Este valor envía los datos como cabeceras http, por lo tanto, esos datos no aparecen en la url.

Utilice este método para formularios grandes, en caso de que quiera enviar un archivo adjunto junto al formulario, si el formulario contiene datos sensibles y si quiere que esos datos se añadan a una base de datos.

Vea a continuación un ejemplo de un formulario que utiliza el valor post en el atributo method:

```
<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Formulario method con valor post</title>

</head>

<body>

    <form action="form.php" method="post">

        Usuario:

        <input type="text" name="usuario" value="" />

        <br/>

        Contraseña:

        <input type="password" name="pass" value="" />

        <br/>

        <input type="submit" value="Enviar" />

    </form>

</body>

</html>
```

Vea a continuación el ejemplo que muestra el navegador:

Usuario:

Contraseña:

En caso de que introduzca datos en estas dos cajas de texto y pulse el botón Enviar, verá que esos datos no se reflejan en la url, sino que se verá el destino que haya especificado en action que, en este caso es form.php.

En las especificaciones del lenguaje HTML, se define la diferencia entre post y get como el sistema que utiliza el navegador para codificar los datos en una url. Y, como ha visto, esto se traduce en que los datos sean visibles o no en la propia url.

En cambio, en las especificaciones puede ver que se recomienda utilizar get cuando al procesar los formularios varias veces, siempre se obtienen los mismos resultados. Esta propiedad se llama idempotencia.

Para simplificar este ejemplo, puede decir que get siempre se utilizará para recuperar datos, y post se puede utilizar para cualquier finalidad como, por ejemplo, el almacenamiento o modificación de datos, el envío de un correo o la compra de productos. En caso de que el procesamiento del formulario resulte en diferentes resultados como, modificación de datos en una base de datos o suscripción a un boletín de noticias, el método siempre será post.

Sin embargo, las especificaciones del protocolo http no son totalmente claras, por lo tanto, podría decir que los usuarios no podrán ser considerados responsables de los efectos adversos que pudieran suceder en el uso de un método u otro. Esto quiere decir que es el programador el que elegirá qué método es el adecuado para cada situación.

Concretamente, los métodos head y get solo deben utilizarse en caso de que la acción a realizar sea una recuperación de datos, ya que, con cualquier otro tipo de caso como modificar o enviar datos no son métodos seguros.

De todos modos, aunque el programador tome todas las precauciones posibles, todos los métodos utilizados pueden verse afectados por software malicioso. Por esto, el envío e información sensible se debe reducir de manera drástica y solo se deberá enviar la mínima información posible.

Características y uso de los valores del atributo method	
Get	Es más rápido y efectivo para envío de formularios con datos no sensibles.
	Añade los datos enviados al final de la url
	Es útil cuando un usuario quiere visualizar los datos enviados
	No se debe utilizar para enviar datos sensibles como contraseñas
	La longitud de la url se limita a 3000 caracteres
Post	No se puede visualizar los resultados después de enviar el formulario
	Los datos que se envían no aparecen en la url, sino en el código enviado
	No tiene límite de tamaño



En caso de no especificar ningún valor dentro del atributo method, por defecto será get.

TOME NOTA

Enctype

Utilice este atributo para indicar el tipo de codificación que desea utilizar cuando envíe el formulario.
Utilice este método en formularios que permitan adjuntar archivos.

Solo podrá utilizar este atributo, si utiliza el valor post en el atributo method.

Valores que puede tomar enctype	
Text/plain	Se enviarán los espacios como +, en cambio, el resto de caracteres no se codificarán
Application/x-www-form-urlencoded	Todos los caracteres serán codificados antes de que se envíen. Se enviarán los espacios como + y los caracteres codificados en ASCII
Multipart/form-data	No se codifican los caracteres

En caso de que no especifique ningún valor en el atributo enctype, este atributo tendrá por defecto el valor application/x-www-form-urlencoded.

Vea un ejemplo de formulario donde se utiliza el atributo enctype:

```
<!DOCTYPE html>
```

```
<html>

<head>

<meta charset="UTF-8">

<title>Uso del atributo enctype</title>

</head>

<body>

<form action="form.php" method="post" enctype="multipart/form-data">

MAnde el archivo:

<input type="file" name="archivo"/>

<br/>

<input type="submit" value="Enviar archivo"/>

</form>

</body>

</html>
```

Ahora, vea cómo muestra el navegador el resultado de este código:

Mande el archivo: Ningún archivo seleccionado

A continuación, va a ver otro ejemplo donde se utilizará el valor application/x-www-form-urlencoded del atributo enctype:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">
```

```
<title>Uso del atributo enctype</title>

</head>

<body>

<form action="form.php" method="post" enctype="application/x-www-form-urlencoded">

    Usuario:

    <input type="text" size="20" maxlength="20" name="usuario" value="">

    <br/>

    <br/>

    Contraseña:

    <input type="password" size="10" maxlength="10" name="pass">

    <br/>

    Confirme la contraseña:

    <input type="password" size="10" maxlength="10" name="c_pass"/>

    <br/>

    <br/>

    Mande el archivo:

    <input type="file" name="archivo"/>

    <br/>

    <br/>

    <input type="submit" value="Enviar archivo"/>

</form>

</body>

</html>
```

Por último, verá el código con el último valor que le puede dar a enctype, text/plain:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Uso del atributo enctype</title>

</head>

<body>

<form action="form.php" method="post" enctype="text/plain">

    Usuario:

    <input type="text" size="20" maxlength="20" name="usuario" value="" />

    <br/>

    <br/>

    Contraseña:

    <input type="password" size="10" maxlength="10" name="pass" />

    <br/>

    Confirme la contraseña:

    <input type="password" size="10" maxlength="10" name="c_pass" />

    <br/>

    <br/>

    Mande el archivo:

    <input type="file" name="archivo" />

    <br/>
```

```
<br/>

<input type="submit" value="Enviar archivo"/>

</form>

</body>

</html>
```

Accept

Utilice este atributo cuando esté creando un formulario que permita adjuntar archivos. Con este atributo indicará todos los tipos de archivos que acepta el servidor.

Los tipos especificados que utiliza deberán estar en los estándares que puede encontrar con el nombre MYME TYPES. Por ejemplo, si quiere solo permitir que se suban imágenes tipo gif o jpeg, deberá utilizar el siguiente código:

```
<form action="envioIMG.php" accept="image/gif, image/jpeg">
```



TOME NOTA

No deberá utilizar este atributo como si fuera una herramienta de validación de archivos. La función de validación siempre se realizará por parte del servidor.

Vea, a continuación, un ejemplo del uso del atributo accept:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Uso del atributo accept</title>

</head>

<body>
```

```
<form action="form.php" method="post" accept="image/gif, image/jpeg">  
  
Envíe una imagen gif o jpeg:  
  
<input type="file" name="imagen"/>  
  
<br/>  
  
<input type="submit" value="Enviar Imagen"/>  
  
</form>  
  
</body>  
  
</html>
```

Ahora, vea cómo lo muestra el servidor cuando ejecute este código:

Envíe una imagen gif o jpeg: Ningún archivo seleccionado

Name

Utilice este atributo para especificar el nombre de un formulario. Podrá utilizar este nombre para referenciar el formulario con elementos de JavaScript, o para referenciarlo una vez se hayan enviado los datos del mismo.

Podrá utilizar el atributo id en vez de name, puesto que, la función de ambos es muy parecida y este atributo también es utilizado para referenciar elementos del formulario en JavaScript o CSS.

En cambio, el atributo name es específico de la etiqueta form.

Se podrán introducir en este atributo los valores que sean de texto plano que el programador quiera asociar con el formulario.

No se recomienda utilizar el atributo name en XHTML. Para ello, mejor utilice el atributo id.

Vea a continuación el código con un ejemplo del atributo name:

```
<!DOCTYPE html>  
  
<html>  
  
<head>
```

```
<meta charset="UTF-8">

<title>Uso del atributo name</title>

</head>

<body>

<form action="form.php" method="post" name="envioform">

    Nombre:

    <input type="text" name="nombre" value="" />

    <br/>

    <input type="submit" value="Enviar" />

</form>

</body>

</html>
```

Vea la demostración de este código en el navegador:

Nombre:

Target

Utilice este atributo cuando quiera que la web que devuelva el servidor después de haber enviado los datos del formulario, se muestre en una nueva ventana o marco del navegador.

Valores que puede utilizar el atributo target	
<u>blank</u>	El resultado se mostrará en una nueva ventana.
<u>_top</u>	El resultado se mostrará con pantalla completa en la ventana
<u>_self</u>	El resultado se mostrará en la misma ventana que el formulario
<u>_parent</u>	El resultado se mostrará en el marco padre
Nombre del marco	El resultado se mostrará en el marco cuyo nombre haya indicado.

En caso de que no especifique el valor en el atributo target, el valor por defecto será _self.

Por otra parte, los valores _top, _parent y nombre del marco solo podrán utilizarse en lenguajes y webs que utilicen marcos.

Vea un código donde puede ver un ejemplo del uso de target:

```
<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>Uso del atributo target</title>

    </head>

    <body>

        <form action="form.php" method="post" target="_blank">

            Buscar:

            <input type="text" name="buscar" value="" />

            <br/>

            <input type="submit" value="Buscar"/>

        </form>

    </body>

</html>
```

Vea lo que muestra el navegador:

The screenshot shows a simple search interface. The word "Buscar:" is in bold black font above a rectangular input field. Below the input field is a blue rectangular button with the word "Buscar" in white.

Accept-charset

Utilice este atributo para indicar la codificación que van a tener los caracteres del formulario cuando éste se envíe hacia el servidor. Lo mismo que ocurre con el atributo accept, con este atributo podrá utilizar varios valores separados por comas, o con espacios.

El valor por defecto para este atributo es una cadena unknown. Esta cadena indica la codificación con la que se debe enviar el documento es la misma que la que contiene el formulario.

Las codificaciones que más se utilizan son:

ISO-8859-1: Esta es una extensión del código ASCII que contiene los caracteres internacionales, y es más completa que ANSI.

UTF-8: Está desarrollada por el Consorcio Unicode. Este es el estándar de codificación que contiene todos los símbolos de puntuación, símbolos escritos y caracteres de todo el mundo.

Vea un código de ejemplo utilizando el atributo accept-charset:

```
<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>Uso del atributo accept-charset</title>

    </head>

    <body>

        <form action="form.php" method="post" accept-charset="utf-8">

            Buscar:

            <input type="text" name="buscar" value="" />

            <br/>

            <input type="submit" value="Buscar"/>

        </form>

    </body>

</html>
```

Vea el ejemplo cómo se muestra en el navegador:

Buscar:

Autocomplete

Utilice este atributo para permitir que los campos del formulario puedan autocompletarse, aunque para ello, el navegador deberá permitir esta función y deberá estar configurado para esta finalidad.

La funcionalidad de este atributo permite, por ejemplo, si en una web hay diferentes formularios con campos parecidos y el usuario tiene activa la función autocompletar, podrá elegir datos que haya ya rellenado para que se auto rellenen en los nuevos campos.

Los valores que puede tomar este atributo son:

- On: Permite que la función autocompletar esté activa.
- Off: Desactiva la función autocompletar.

En caso de no especificar el valor del atributo autocomplete, tendrá por defecto el valor on.

Por otro lado, es bastante común encontrar formularios que tiene la opción autocompletar activada en algunos campos y en otros no.

Vea a continuación un código que muestra cómo utilizar este atributo:

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="UTF-8">

    <title>Uso del atributo autocomplete</title>

  </head>

  <body>

    <form action="form.php" method="post" autocomplete="on">
```

Buscar:

```
<input type="text" name="buscar" value="" />  
<br/>  
<input type="submit" value="Buscar" />  
</form>  
</body>  
</html>
```

Vea lo que muestra el navegador:



Novalidate

Este atributo es realmente una variable booleana. La función de este atributo es que si se declara en el formulario es como si tuviera un valor true, por lo tanto, los datos no se van a validar hasta que se finalice el formulario.

Este atributo es un atributo vacío. Esto quiere decir que puede escribir simplemente la palabra “novalidate” dentro del código del formulario y, el navegador no lo validará. Sin embargo, en XHTML puede escribir novalidate=novalidate.

Vea a continuación un ejemplo de uso de este atributo:

```
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>Uso del atributo novalidate</title>  
</head>  
<body>  
<form action="form.php" method="post" novalidate>
```

Envíe el archivo:

```
<input type="file" name="adjuntar" value="" />  
<br/>  
<input type="submit" value="Enviar" />  
</form>  
</body>  
</html>
```

El navegador mostrará lo siguiente:

Envíe el archivo: Ningún archivo seleccionado

Debe saber la diferencia entre los atributos novalidate y formnovalidate. Novalidate es un atributo que se aplica al formulario y evita que éste no sea validado. En cambio, el atributo formnovalidate se aplica al botón de enviar. Esto quiere decir que el formulario no se validará si utiliza ese botón.

Ponga el ejemplo donde un usuario guarda los datos de un formulario en vez de enviarlos, estos datos puede que estén incompletos o no sean correctos, pero como no se van a publicar porque el usuario tan solo quiere guardar esos datos, podría añadir al botón de guardado el atributo formnovalidate.

Utilizando estos atributos puede crear casi todos los tipos de formulario en HTML y en otros tipos de lenguajes como PHP.

Vea un ejemplo de todos estos atributos explicados en un formulario:

```
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>Uso de todos los atributos aprendidos</title>
```

```
</head>

<body>

    <form action="form.php" method="post" enctype="application/x-www-form-
urlencoded" accept="image/jpeg" name="pruebaForm" target="_blank" accept-charset="utf-
8" autocomplete="on" novalidate>

        Usuario:
        <input type="text" name="usuario" value="" />
        <br/>

        Contraseña:
        <input type="password" name="pass" value="" />
        <br/>

        Repetir contraseña:
        <input type="password" name="cPass" value="" />
        <br/>

        Foto de perfil:
        <input type="file" name="avatar" />
        <br/>

        <input type="submit" value="Enviar">

    </form>

</body>

</html>
```

Vea lo que muestra el navegador:

Usuario:

Contraseña:

Repetir contraseña:

Foto de perfil: Seleccionar archivo Ningún archivo seleccionado



1.2.1. Atributos básicos aplicados a formularios

Aparte de los atributos más generales que ha visto hasta ahora, también tiene que ser capaces de utilizar los atributos básicos en los formularios que diseñe.

La mayoría de estos atributos se utilizan comúnmente en código HTML y se han explicado ya en otras unidades formativas. Sin embargo, debe aprender a cómo utilizar estos atributos en los formularios.

Vea a continuación los siguientes atributos:

ATRIBUTOS BÁSICOS:

Id

Utilice este atributo cuando quiere identificar un elemento de manera única en el código HTML de una web. Como ya vi anteriormente, puede utilizar también el atributo name para realizar la misma acción. Sin embargo, es mucho más correcto utilizar id.

El atributo id funciona igual que el atributo name. Sin embargo, el atributo id puede ser utilizado tanto en las hojas de estilo, como en JavaScript para manipular el elemento desde estos lenguajes.

Debe tener en cuenta que cuando utiliza este atributo, deben cumplirse tres cosas:

- No puede contener espacios.
- No hay diferencia entre mayúsculas o minúsculas.

- Debe contener como mínimo un carácter.

En caso de que utilice PHP, deberá tener en cuenta que, al contrario que HTML, éste sí diferencia entre mayúsculas y minúsculas, por lo tanto, debe poner especial atención, ya que, si programa primero en HTML, puede que equivoque al pasar al PHP.

Vea un código de ejemplo que utiliza el atributo id:

```
<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>Atributo id</title>

    </head>

    <body>

        <form action="form.php" method="post" id="EnvioNombreUsuario">

            Usuario:

            <input type="text" name="usuario" value="" />

            <br/>

            <input type="submit" value="Enviar" />

        </form>

    </body>

</html>
```

Vea lo que muestra el navegador:

Usuario:

Class

Utilice este atributo para especificar uno o más nombres de clase asociados a uno o más elementos para, posteriormente, utilizar diferentes estilos CSS según la clase. De esta manera, solo tendré que programar el diseño de la clase para que se aplique a todos los elementos que pertenecen a la misma.

Por otra parte, puede utilizar este atributo para que apunte a una hoja de estilos y también, puede utilizarlo desde JavaScript para modificar los elementos de HTML utilizando las clases específicas que ha programado con esta intención. Para crear los nombres de clase debe tener en cuenta lo siguiente:

- HTML no es case sensitive.
- Si quiere definir múltiples cases, las separare por un espacio.
- Después del primer carácter puede utilizar números, guiones o letras.
- El nombre de clase debe comenzar por una letra.

Vea un código que utiliza el atributo class:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Atributo class</title>

<style>

.verde {color: green;}

</style>

</head>

<body>

<form action="form.php" method="post" class="verde">
```

Usuario:

```
    <input type="text" name="usuario" value="" />

    <br/>

    <input type="submit" value="Enviar">
```

```
</form>  
  
</body>  
  
</html>
```

Vea lo que muestra el navegador:

Usuario:
Enviar

Style

En caso de que quiera definir el estilo del formulario de manera directa, utilice el atributo style.

Este atributo se define por aplicar de manera directa un estilo a cualquier elemento de HTML y, de esta manera, se lo aplica directamente al formulario que quiera.

Además, lo que haya dentro de este atributo estará por encima de cualquier otro estilo que se le haya aplicado al mismo documento o al formulario. Por ejemplo, utilizando las etiquetas <style> en la cabecera del documento o con una hoja CSS externa.

En caso de que establezca diferentes estilos, deberá separarlos con comas.

Vea el ejemplo de código:

```
<!DOCTYPE html>  
  
<html>  
  
  <head>  
    <meta charset="UTF-8">  
    <title>Atributo style</title>  
  </head>  
  
  <body>  
    <form action="form.php" method="post" style="color:red">  
  
      Usuario:  
  
      <input type="text" name="usuario" value="" />
```

```
<br/>

<input type="submit" value="Enviar">

</form>

</body>

</html>
```

A continuación, vea el resultado:

Usuario:

Title

Utilizando este atributo podrá añadir información más específica al formulario utilizando, para ello, texto plano.

Esta información va a depender del tipo de formulario que este programando, pero, normalmente, puede ser pistas para los usuarios cuando pasen el puntero del ratón por encima de un elemento del formulario u otro tipo de ayuda que requiera el usuario al usar el mismo.

Deberá tener en cuenta que la información escrita deberá ser legible y útil para el usuario, puesto que, esto no es un comentario para los programadores, sino que es un comentario de ayuda para el usuario final.

Vea un ejemplo:

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="UTF-8">
    <title>Atributo title</title>
  </head>
  <body>
```

```
<form action="form.php" method="post" title="Formulario que envía el nombre de usuario.">  
  Usuario:  
  <input type="text" name="usuario" value="" />  
  <br/>  
  <input type="submit" value="Enviar">  
</form>  
</body>  
</html>
```

Vea el resultado:

Usuario:

Formulario que envía el nombre de usuario.



TOME NOTA

Debe tener en cuenta que este tipo de atributo no es visible a simple vista, sino cuando pasa el puntero del ratón por encima del elemento al que se le aplica el propio atributo.

INTERNACIONALIZACIÓN

Lang

Al programar un formulario no siempre debe de estar en el mismo idioma que la página en sí. De hecho, dependiendo de hacia donde vaya dirigido ese formulario estará en un idioma u otro o si la página se encuentra en varios idiomas.

Cuando encuentra en esta situación, puede que vea problemas en la representación de caracteres o en la autocorrección de algunos navegadores como, por ejemplo, en Google Chrome. Para solucionar

esto, deberá utilizar el atributo Lang. Este atributo le indica al navegador el idioma que se utiliza en el formulario.

El código que utilice en el atributo Lang para indicar el español es: es.

Vea, a continuación, un ejemplo de uso del atributo Lang:

```
<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>Atributo lang</title>

    </head>

    <body>

        <form action="form.php" method="post" lang="es">

            Usuario:

            <input type="text" name="usuario" value="" />

            <br/>

            <input type="submit" value="Enviar" />

        </form>

    </body>

</html>
```

Vea el resultado que muestra el navegador:

Usuario:

Ahora, va a ver los códigos de varios idiomas del mundo para que poda utilizarlos con el atributo Lang:

Lenguaje	Código ISO
Albano	Sq
Alemán	De
Árabe	Ar
Armenio	Hy
Bielorruso	Be
Búlgaro	Bg
Catalán	Ca
Coreano	Ko
Croata	Hr
Checo	Cs
Chino	Zh
Chino (Simplificado)	Zh-Hans
Chino (Tradicional)	Zh-Hant
Danés	Da
Eslovaco	Sk
Esloveno	Sl
Español	Es
Estonio	Et
Euskera	Eu
Finlandés	Fi
Francés	Fr
Gallego	Gl
Griego	El
Hebreo	He, iw
Hindú	Hi
Holandés	Nl
Húngaro	Hu
Indonesio	Id, in
Inglés	En
Irlandés	Ga
Islandés	Is
Italiano	It
Japonés	Ja

Lituano	Lt
Macedonio	Mk
Moldavo	Mo
Mongol	Mn
Nepalí	Ne
Noruego	No
Polaco	Pi
Portugués	Pt
Rumano	Ro
Ruso	Ru
Serbio	Sr
Serbo-croata	Sh
Sueco	Sv
Tailandés	Th
Turco	Tr
Ucraniano	Uk
Vietnamita	vi

xml:Lang

Es cierto que el atributo Lang es el que se utiliza la mayoría de las veces. Sin embargo, hay ocasiones en las que es más adecuado utilizar xml:Lang. Sobre todo, si el contenido del formulario se asocia directamente con un documento tipo XML.

En estos casos, utilice más correctamente el atributo xml:Lang para el idioma de destino del mismo.

El atributo Lang no es incompatible con el atributo xml:Lang, por lo tanto, es totalmente posible encontrarlos de manera conjunta en la misma etiqueta de un formulario.

Los valores que puede utilizar el atributo xml:Lang son los mismo que puede utilizar el atributo Lang, por lo tanto, la lista de los idiomas que ha visto anteriormente podrá usarla también para este atributo.

Vea un pequeño ejemplo del uso de xml:Lang:

```
<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>Atributo xml:lang</title>
```

```
</head>

<body>

<form action="form.php" method="post" xml:lang="es">

    Usuario:

    <input type="text" name="usuario" value="" />

    <br/>

    <input type="submit" value="Enviar" />

</form>

</body>

</html>
```

Ejemplo de lo que muestra el navegador:

Usuario:

Dir

Utilice el atributo dir para especificar la dirección de escritura que tendrá el texto que va a insertar en el formulario.

Utilizar este atributo, en la mayoría de los casos, puede ser poco útil, sin embargo, es importante que conozca y sepa sobre la utilidad del atributo para, de esta manera, poder utilizarlo en formularios específicos que precisen del mismo.

Por otra parte, para utilizar ciertos idiomas, sí es necesario cambiar la dirección del texto por defecto y si utiliza este atributo junto con los atributos xml:Lang y Lang, el atributo dir cuenta de una notable importancia.

Vea a continuación los valores que puede tomar el atributo dir:

- Auto: Deja que el navegador elija la dirección del texto.
- Rtl: El texto va de derecha a izquierda.
- Ltr: El texto va de izquierda a derecha.



TOME NOTA

Como es de esperar, el valor por defecto del atributo dir es ltr, que es el que indica que el texto va de izquierda a derecha.

Vea un ejemplo de código con el atributo dir:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Atributo dir</title>

</head>

<body>

<form action="form.php" method="post" dir="rtl">

    Usuario:

    <input type="text" name="usuario" value="" />

    <br/>

    <input type="submit" value="Enviar" />

</form>

</body>

</html>
```

Vea cómo se muestra:

	:Usuario
	<input type="button" value="Enviar"/>

EVENTOS

onblur

A continuación, va a ver dos atributos que se relacionan entre sí, puesto que, sus funciones están relacionadas.

Por otra parte, cabe destacar que estos dos atributos no son atributos específicos de los formularios, sino que lo son de los elementos que componen el mismo como, por ejemplo, los botones, las cajas de texto, etc.

En cambio, debe conocerlos ahora, puesto que, como estos atributos son generales de varios elementos, deberá dominarlos para poder utilizarlo en los formularios.

Específicamente, el atributo onblur tiene funcionalidad cuando el elemento con el mismo nombre actúa, esto ocurre cuando el elemento del formulario al que le ha asignado el evento, pierde el foco.

Vea un ejemplo a continuación:

```
<!DOCTYPE html>

<html>

  <head>
    <meta charset="UTF-8">
    <title>Atributo onblur</title>
  </head>

  <body>
    <script type="text/javascript">
      function vaciar (control)
      {
        control.value="";
      }
    </script>
  </body>
</html>
```

```
}

function verificarForm(control)

{

if(control.value=="")  
    alert ('El formulario no puede estar vacío')

}

</script>

<form>

    <input type="text" id="nombre" onfocus="vaciar(this)" onblur="verificarForm(this)"
value="nombre">

    <br/>

    <input type="text" id="edad" onfocus="vaciar(this)" onblur="verificarForm(this)"
value="edad">

    <br/>

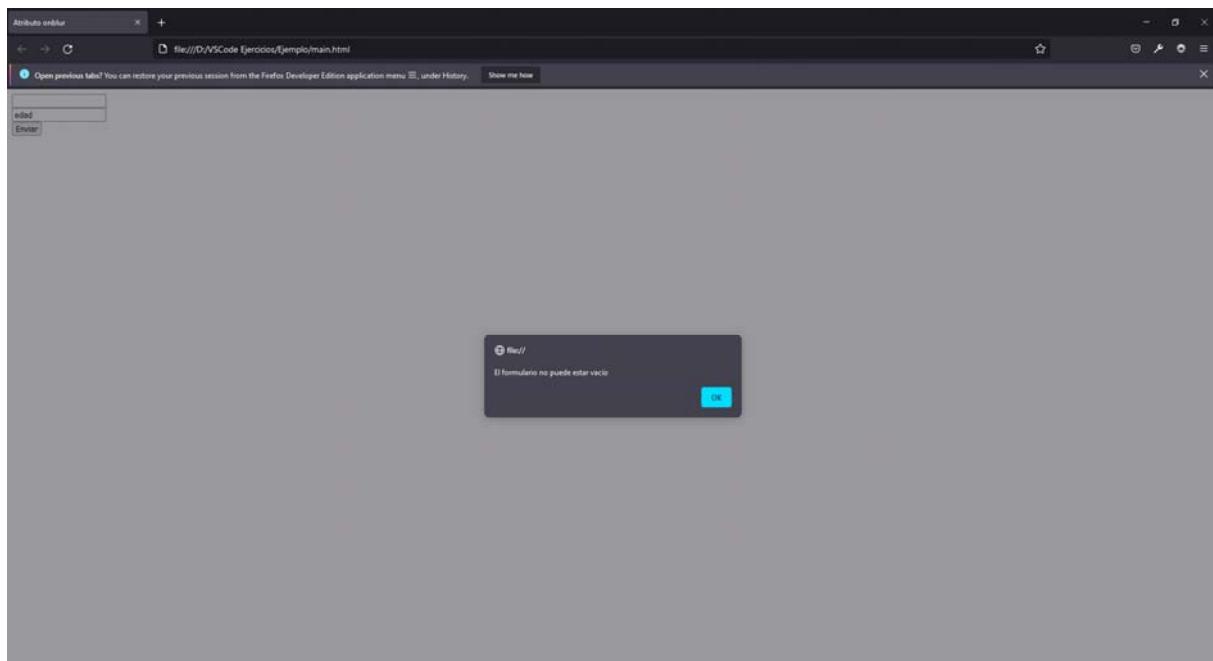
    <input type="submit" value="Enviar">

</form>

</body>

</html>
```

Vea lo que muestra el navegador:



onfocus

Este atributo funciona al revés que el atributo onblur y se ejecuta cuando el elemento tiene el enfoque.

Estos dos atributos funcionan de manera inversa, por lo tanto, se pueden programar de manera independiente, sin embargo, tienen más sentido cuando ambos van de la mano.

En el siguiente código que va a ver, cuando pulsa en una zona distinta a donde están programados los eventos y los elementos pierden el enfoque, en el caso de que estén vacíos saldrá un mensaje de aviso. Sin embargo, cuando los elementos vuelven a tener el foco el atributo onfocus hará que los campos se vacíen.

Vea el ejemplo:

```
<!DOCTYPE html>

<html>
  <head>

    <meta charset="UTF-8">

    <title>Atributo onfocus</title>

  </head>

  <body>
```

```
<script type="text/javascript">

function vaciar (control)

{
    control.value="";
}

function verificarForm(control)

{
    if(control.value=="")
        alert ('El formulario no puede estar vacío')

}
</script>

<form>

    <input type="text" id="nombre" onfocus="vaciar(this)" onblur="verificarForm(this)"
value="nombre">

    <br/>

    <input type="text" id="edad" onfocus="vaciar(this)" onblur="verificarForm(this)"
value="edad">

    <br/>

    <input type="submit" value="Enviar">
</form>

</body>

</html>
```

Vea el resultado:



onchange

Solo podrá aplicar este atributo a algún elemento específico de un formulario, no al propio formulario.

Este atributo actúa como un evento que se ejecuta cuando un elemento del formulario pierde el enfoque y, a la vez, se ha modificado su valor.

Cuando este atributo se ejecuta en una caja de texto, no se activa hasta que haya pasado a la siguiente caja.

Vea un ejemplo del código de onchange:

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="UTF-8">
    <title>Atributo onchange</title>
  </head>
  <body>
    <script type="text/javascript">
```

```
function ModifForm (control)

{
    if (control)

        alert ('Se ha modificado un elemento del formulario.');

    }

</script>

<form>

<input type="text" id="nombre" onchange="ModifForm(this)" value="nombre">

<br/>

<input type="text" id="edad" onchange="ModifForm(this)" value="edad">

<br/>

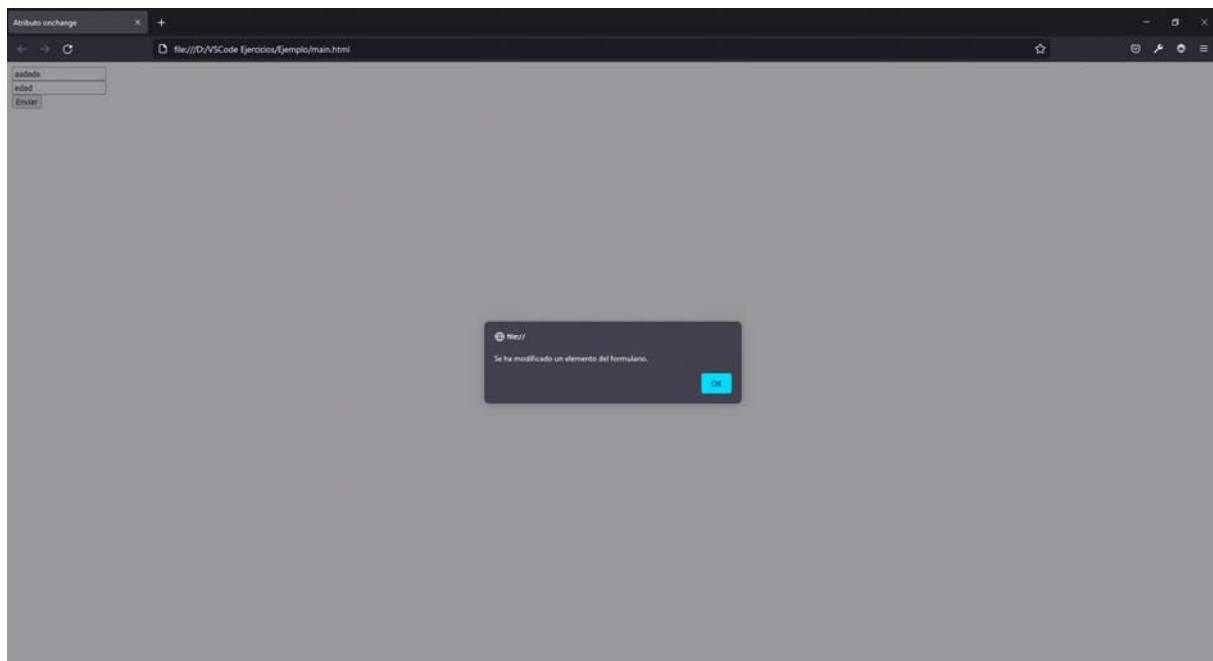
<input type="submit" value="Enviar">

</form>

</body>

</html>
```

Vea el resultado en pantalla:



onclick

El atributo onclick va a estar relacionado con el que conocerá a continuación, puesto que, funcionan de manera similar. La única diferencia que tiene uno de otro es que dependerá de las veces que pulse el ratón.

Este atributo se ejecuta cuando el usuario hace clic con el ratón dentro de los elementos de un formulario donde está asociado el propio atributo. Este clic activará la función de script asociada al evento onclick.

Vea un ejemplo:

```
<!DOCTYPE html>

<html>

  <head>
    <meta charset="UTF-8">
    <title>Atributo onclick</title>
  </head>

  <body>
    <script language=JavaScript>
```

```
var nClicks=0;

function miBoton ()

{

nClicks++;

window.document.formulario.miboton.value='Botón pulsado '+ nClicks +' veces';

}

</script>

<form name="formulario">

    <input type="button" name="miboton" value="Botón pulsado 0 veces"
onclick="miBoton()">

</form>

</body>

</html>
```

Vea el resultado:

Botón pulsado 8 veces

ondblclick

Como ya se dijo anteriormente, este atributo es similar al atributo onclick, es decir, también funciona como un evento que se asocia al clic de ratón. Sin embargo, en este caso en vez de hacer un solo clic, deberá hacer un doble clic en el elemento en el que está asociado este evento.

Vea un ejemplo a continuación:

```
<!DOCTYPE html>

<html>

<head>
```

```
<meta charset="UTF-8">

<title>Atributo ondblclick</title>

</head>

<body>

<script language=JavaScript>

var nClicks=0;

function miBoton ()

{

nClicks++;

window.document.formulario.miboton.value='Doble clic pulsado '+ nClicks +' veces';

}

</script>

<form name="formulario">

    <input type="button" name="miboton" value="Doble clic pulsado 0 veces"
ondblclick="miBoton()">

</form>

</body>

</html>
```

Para que pueda ver la funcionalidad de este evento, deberá hacer doble clic en el botón.

Vea la demostración:

Doble clic pulsado 5 veces

onkeydown

Este atributo va a estar relacionado con los que verá más adelante.

Concretamente, tanto el atributo onkeydown, como en el atributo onkeypress y onkeyup se van a aplicar a los elementos del formulario a los cuales asocie estos eventos.

Este atributo funciona de tal manera que cuando el usuario presiona una tecla, activa el elemento en el cual está asociado el atributo.

Vea un ejemplo:

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="UTF-8">

    <title>Atributo onkeydown</title>

  </head>

  <body>

    <script language=JavaScript>

      function trarEvento (nombre)

      {

        var mensaje= window.document.formulario.textarea2.value;

        mensaje=mensaje + 'Se ha utilizado el evento: '+ nombre;

        window.document.formulario.textarea2.value=mensaje;

      }

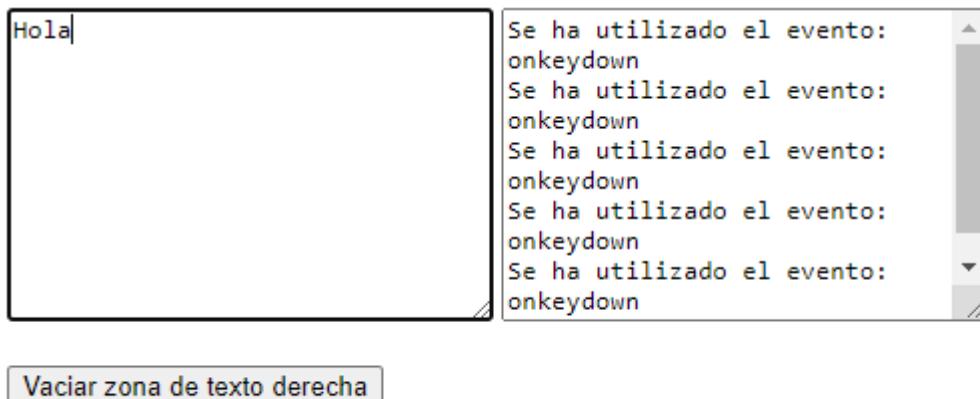
    </script>

    <form name="formulario">
```

```
<textarea rows="10" cols="30" name="textarea1" onkeydown="trarEvento('onkeydown\n');">  
</textarea>  
  
<textarea rows="10" cols="30" name="textarea2">  
</textarea>  
  
<br/>  
  
<br/>  
  
    <input type="button" value="Vaciar zona de texto derecha" name="button1"  
onclick="window.document.formulario.textarea2.value=''">  
  
</form>  
  
</body>  
  
</html>
```

Este código sirve para ver de manera visual cuando se activan los eventos asociados a los atributos especificados.

Vea el resultado:



onkeypress

Este atributo, al igual que el anterior, también se asocia a los elementos del formulario y no al formulario en sí.

La función de este atributo hace que se active el evento cuando el usuario mantiene la tecla pulsada en el elemento donde está asociado el evento.

Vea el ejemplo:

```
<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>Atributo onkeypress</title>

    </head>

    <body>

        <script language=JavaScript>

            function trarEvento (nombre)

            {

                var mensaje= window.document.formulario.textarea2.value;

                mensaje=mensaje + 'Se ha utilizado el evento: '+ nombre;

                window.document.formulario.textarea2.value=mensaje;

            }

        </script>

        <form name="formulario">

            <textarea rows="10" cols="30" name="textarea1" onkeypress="tragEvento('onkeypress\n');">

            </textarea>

            <textarea rows="10" cols="30" name="textarea2">

            </textarea>

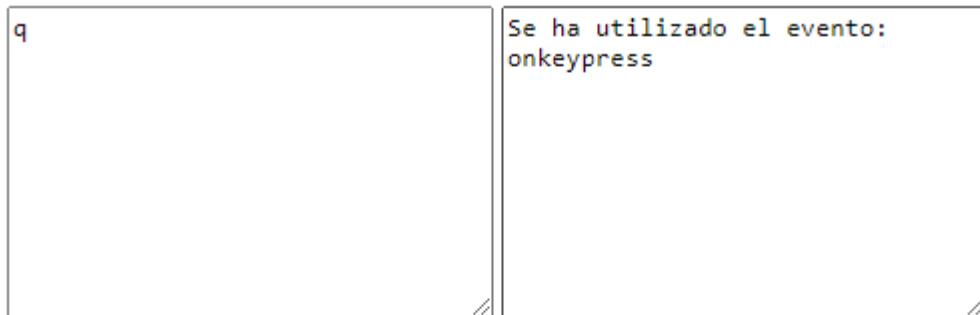
        </form>

    </body>

</html>
```

```
<br/>  
<br/>  
    <input type="button" value="Vaciar zona de texto derecha" name="button1"  
    onclick="window.document.formulario.textarea2.value=''">  
</form>  
  
</body>  
</html>
```

Vea el resultado:



Vaciar zona de texto derecha

onkeyup

Este atributo, al igual que los anteriores, también se aplica a los propios elementos del formulario.

La funcionalidad de este atributo es que cuando el usuario deja de pulsar una tecla dentro del elemento donde está asociado el evento, se activa el evento.

Vea el siguiente ejemplo:

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
<meta charset="UTF-8">
```

```
<title>Atributo onkeyup</title>

</head>

<body>

<script language=JavaScript>

function trarEvento (nombre)

{

    var mensaje= window.document.formulario.textarea2.value;

    mensaje=mensaje + 'Se ha utilizado el evento: '+ nombre;

    window.document.formulario.textarea2.value=mensaje;

}

</script>

<form name="formulario">

<textarea rows="10" cols="30" name="textarea1" onkeyup="tragEvento('onkeyup\n');">

</textarea>

<textarea rows="10" cols="30" name="textarea2">

</textarea>

<br/>

<br/>

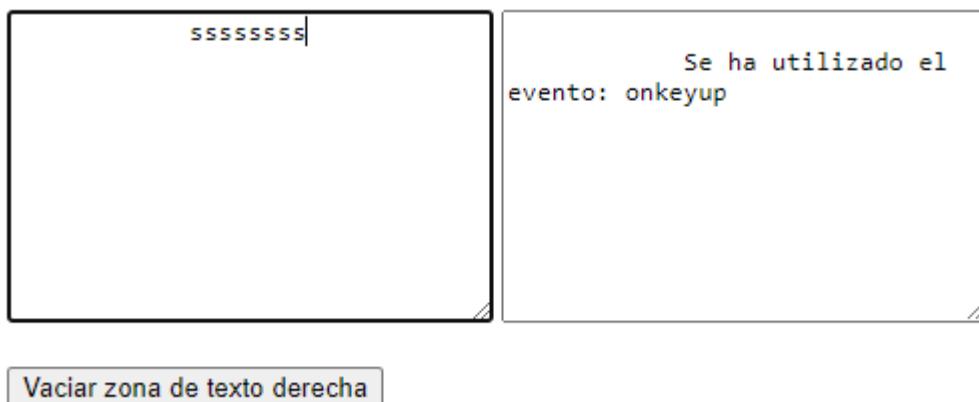
<input type="button" value="Vaciar zona de texto derecha" name="button1" onclick="window.document.formulario.textarea2.value=''">

</form>

</body>
```

```
</html>
```

El resultado es el siguiente:



Vaciar zona de texto derecha

onmousedown

Este atributo que va a ver ahora estará relacionado con los atributos que va a ver después. Los atributos onmousedown, onmousemove y onmouseup se van a aplicar a los elementos del formulario y no al propio formulario.

La funcionalidad del atributo onmousedown es que el elemento del formulario que tiene asociado el atributo sea el responsable de activar el evento onmousedown cuando el usuario pulsa el botón del ratón cuando el puntero está tocando el elemento.

Vea a continuación un ejemplo:

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="UTF-8">

    <title>Atributo onmousedown</title>

  </head>

  <body>

    <script language=JavaScript>

      function botonOnmouseup ()
```

```
{  
    document.formulario.boton.value="El ratón no está pulsado"  
  
}  
  
function botonOnmousedown ()  
  
{  
    document.formulario.boton.value="El ratón está pulsado"  
  
}  
  
function botonOnmousemove ()  
  
{  
    document.formulario.boton.value="El puntero está sobre el botón"  
  
}  
  
</script>  
  
<form name="formulario">  
  
    <input type="button" name="boton" value="El botón del ratón no está pulsado, ni el ratón  
está sobre el botón" onmouseup="botonOnmouseup()" onmousedown="botonOnmousedown()"  
onmousemove="botonOnmousemove()">  
  
</form>  
  
</body>  
</html>
```

En este ejemplo de código, verá como si el ratón se mueve por encima del botón y si se pulsa o no el botón, se van a activar los atributos que ha codificado como onmousemove, onmousedown y onmouseup.

Vea una demostración:

El ratón está pulsado

onmousemove

Este atributo se aplica solo a los elementos del formulario y no al formulario en sí. La funcionalidad de este atributo hace que el elemento donde se ha aplicado el atributo active el evento cuando el usuario mueva el puntero del ratón por encima del elemento.

Vea un ejemplo:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Atributo onmousemove</title>

</head>

<body>

<script language=JavaScript>

function botonOnmouseup ()

{

    document.formulario.boton.value="El ratón no está pulsado"

}

function botonOnmousedown ()

{

    document.formulario.boton.value="El ratón está pulsado"

}

function botonOnmousemove ()
```

```
{  
    document.formulario.botón.value="El puntero está sobre el botón"  
}  
  
</script>  
  
<form name="formulario">  
  
    <input type="button" name="botón" value="El botón del ratón no está pulsado, ni el ratón  
está sobre el botón" onmouseup="botónOnmouseup()" onmousedown="botónOnmousedown()"  
onmousemove="botónOnmousemove()"/>  
  
</form>  
  
</body>  
  
</html>
```

A continuación, vea el resultado:

El puntero está sobre el botón

Onmouseup

Este atributo se aplica solo a los elementos del formulario y no al formulario en sí. La funcionalidad de este atributo hace que el elemento donde se ha aplicado el atributo active el evento cuando el usuario deja de pulsar con el punto del ratón el elemento.

Vea el ejemplo:

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
    <meta charset="UTF-8">  
  
<title>Atributo onmousemove</title>
```

```
</head>

<body>

<script language=JavaScript>

function botonOnmouseup ()

{

    document.formulario.boton.value="El ratón no está pulsado"

}

function botonOnmousedown ()

{

    document.formulario.boton.value="El ratón está pulsado"

}

function botonOnmousemove ()

{

    document.formulario.boton.value="El puntero está sobre el botón"

}

</script>

<form name="formulario">

    <input type="button" name="boton" value="El botón del ratón no está pulsado, ni el ratón
está sobre el botón" onmouseup="botonOnmouseup()" onmousedown="botonOnmousedown()"
onmousemove="botonOnmousemove()">

</form>

</body>

</html>
```

Vea el resultado:

El ratón no está pulsado

Onmouseout

Los dos eventos que va a ver a continuación funcionan de la misma manera que los anteriores, es decir, que se relacionan con los elementos del formulario.

Este atributo, se utiliza cuando quiere controlar lo que va a ocurrir en caso de que el puntero del ratón abandone el formulario, el elemento específico. Si esto ocurre, el evento se activará.

Vea un ejemplo:

```
<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Atributo onmouseout</title>

</head>

<body>

    <script language=JavaScript>

        function colorear (color)

        {

            document.bgColor=color;

        }

    </script>

    <form name="formulario">
```

```
<input type="text" id="nombre" onmouseover="colorear('#BDFF00')"
onmouseout="colorear('#FF0F00')">

<br/>

<input type="button" value="Enviar">

</form>

</body>

</html>
```

Vea el resultado:



Onmouseover

Utilice este atributo cuando quiera controlar lo que sucede cuando el puntero del ratón pasa sobre un formulario o elemento del formulario. De esta manera, activare el evento que ha programado.

En este ejemplo, cuando pase el puntero del ratón por encima del cuadro de texto del formulario, el fondo de la pantalla se pondrá de color verde lima y cuando el puntero sale del cuadro de texto, el color cambiará a rojo.

Vea un ejemplo:

```
<!DOCTYPE html>

<html>

<head>
    <meta charset="UTF-8">
    <title>Atributo onmouseout</title>
</head>

<body>
    <script language=JavaScript>
```

```
function colorear (color)

{
    document.bgColor=color;

}

</script>

<form name="formulario">

    <input type="text" id="nombre" onmouseover="colorear('#BDFF00')"
onmouseout="colorear('#FF0F00')">

    <br/>

    <input type="button" value="Enviar">

</form>

</body>

</html>
```

Ahora, vea el resultado:



1.2.2. Descripción y definición de los elementos de un formulario

Ahora, que ya conoce los atributos que puede utilizar en los formularios dentro de la etiqueta `<form>`, va a proceder a insertar elementos, que como ya vi anteriormente en las características de los formularios, son totalmente obligatorios para que el formulario pueda recoger la información del usuario.

Vea algunos de los elementos que puede encontrar comúnmente en la mayoría de los formularios:

- Cuadros de texto.

- Casillas de verificación.
- Cuadros ocultos utilizados para contraseñas.
- Botones de envío de información.
- Botones de opciones.
- Botones de reseteo de información.
- Herramientas para adjuntar documentos.
- Botones que usan imágenes.
- Campos ocultos para el usuario.
- Áreas de texto donde se pueden escribir muchos caracteres.
- Botones para otro tipo de funciones.
- Listas de selección.
- Herramientas de agrupación de elementos del formulario.
- Etiquetas para agrupar elementos del formulario.
- Elementos que sirven para renombrar o etiquetas las agrupaciones creadas.
- Elementos que hacen que un formulario se adapte perfectamente a la función que necesita de cara al usuario final.

Cuando programa un elemento de un formulario, debe tener muy presente el objetivo final de ese elemento dentro del formulario. De esta manera, utilice los elementos precisos y adecuados para este formulario. Por ejemplo, si quiere poner un cuadro de edad y otro de fecha en un formulario, podría utilizar un cuadro de texto normal. Sin embargo, si para la edad inserta un cuadro de texto más pequeño y para la fecha un cuadro tipo date, obtendrá elementos mucho más específicos para su tipo de dato y, por lo tanto, un formulario mucho más correcto.

A continuación, va a ver una captura de pantalla del formulario que ofrece Gmail para crear una nueva cuenta. Este ejemplo muestra el perfecto uso de las distintas cajas de texto y los tamaños adecuados para cada uno de los datos que van a albergar los distintos elementos. Vea la imagen:

Google

Crea una cuenta de Google

Ir a Gmail

Nombre _____ Apellidos _____

Nombre de usuario @gmail.com
Puedes utilizar letras, números y puntos

Contraseña Confirmación
Utiliza ocho caracteres como mínimo con una combinación de letras, números y símbolos

Mostrar contraseña

Prefiero iniciar sesión Siguiente



Una cuenta. Todo Google a tu disposición.

Español (España) ▾

Ayuda Privacidad Términos

Si tiene presente estos pequeños detalles no solo tendrá un formulario mejor diseñado, sino que también será mucho más funcional.

Vea para cuántos casos sería útil un cuadro de texto simple:

- Introducir un email.
- Introducir una fecha.
- Introducir texto plano.
- Introducir teléfono.
- Introducir url.

También, podría servir para introducir cualquier dato que tenga una diferencia con los demás, o que tenga pocos caracteres.

Un típico ejemplo de esto es el de los cuadros de texto que piden introducir una edad. Vea que es muy sencillo enumerar los controles que debe implementar en este tipo de cuadro de texto:

- El valor introducido deberá ser mayor que cero.

- Deberá ser un valor corto.
- Solo deberá aceptar números.
- El número no podrá tener más de tres cifras.

Ahora, verá un ejemplo de cómo puede utilizar un cuadro de texto para múltiples funcionalidades dentro del mismo formulario. Para ello, va a ir al formulario de registro de la página de Duolingo.

Como puede ver, primero tiene un cuadrado de texto para la edad que es un cuadro de texto normal con las características que ha visto anteriormente.

Por otra parte, tiene un cuadro de texto para escribir el nombre para que escriba su nombre completo, sin embargo, no es necesario que lo rellene, es algo opcional. Por lo tanto, para programar un cuadro de este tipo, deberá indicar que es opcional y le asignare una capacidad grande para que entre un nombre largo.

Lo que verá a continuación es el correo. Este cuadro de texto se va a caracterizar. Principalmente, porque deberá ser obligatorio para que la web pueda enviar una confirmación de registro al usuario. Además, deberá tener un formato adecuado para email como, por ejemplo, que tenga una arroba y una extensión como puede ser .es o .com.

Por último, encontrará el recuadro para la contraseña que su principal característica será que se introduzcan caracteres ocultos y, además, este tipo de cuadro de texto deberá enviar los datos cifrados a una base de datos. De esta manera, evitirá problemas de seguridad en la web.

Vea una captura de pantalla de este formulario. En este formulario puede ver el uso perfecto de diferentes tipos de cajas de texto y distinta programación en cada una de ellas.

The screenshot shows the 'Crea tu perfil' (Create your profile) page from Duolingo. The page has a light gray background with a white form area. At the top right is a blue 'INGRESAR' (Log In) button. The form itself has a title 'Crea tu perfil'. It contains four input fields: 'Edad' (Age), 'Nombre (opcional)' (Optional name), 'Correo' (Email), and 'Contraseña' (Password). Below these fields is a large blue rectangular button labeled 'CREAR CUENTA' (Create Account). Underneath the password field is a small text input field with the value '0'. At the bottom of the form are two social media login buttons: 'FACEBOOK' and 'GOOGLE'. At the very bottom, in small print, it says 'Al registrarte en Duolingo, aceptas nuestros Términos y' followed by 'Política de privacidad'.

1.2.3. Utilización de campos y textos

En este punto ya debe tener claro que los formularios son elementos compuestos por otros elementos que son los que dan forma al propio formulario.

Un formulario que esté programado solo con las etiquetas básicas será un formulario que no servirá para nada y, además, el aspecto visual del mismo sería totalmente invisible para el usuario, por lo que, tendría un elemento totalmente inútil en su web.

Por lo tanto, de la misma manera que ocurre con los otros elementos de programación, el programador deberá ser consciente de todos los elementos con los que se formará el formulario, para así programar el formulario de una manera eficiente y que el formulario sea una herramienta de utilidad para los usuarios.

Para aplicar todos estos conocimientos, deberá tener presente la metodología de la programación en todo momento. Esta metodología es muy importante para la programación en general, pero en los formularios se vuelve un elemento de importancia vital.

Vea dos ejemplos en los que podrá ver a simple vista que los elementos se identifican mejor en uno que en otro:

Primer ejemplo:

```
<html><head><title>Programación de formulario</title></head><body><form action="enviolImagenes.php" method="post" accept="image/gif, image/jpeg">Envíe una imagen jpeg o gif:<input type="file" name="imagen" /><br/><input type="submit" value="Enviar Imagen"/></form></body></html>
```

Segundo ejemplo:

```
<!DOCTYPE html>

<html>

  <head>
    <meta charset="UTF-8">
    <title>Programación de formulario</title>
  </head>

  <body>
    <form action="enviolImagenes.php" method="post" accept="image/gif, image/jpeg">
```

Envíe una imagen gif o jpeg:

```
<input type="file" name="imagen"/>  
</form>  
  
</body>  
  
</html>
```

Como puede ver, el código de los dos ejemplos es el mismo y el resultado que obtendrá en el navegador, por lo tanto, será el mismo. Sin embargo, para otros será mucho más difícil identificar las partes del código en el primer ejemplo que en el segundo.

Es por esto, que otros como programadores, deberá intentar dividir en diferentes líneas de código los elementos que componen un formulario. Ya no es solo por la facilidad que dará tener el código bien organizado, sino por la comodidad que va a aportar cuando tenga que revisar el código en un futuro, o incluso cuando un compañero deba hacerlo.

Además, otra de las cosas importantes que puede observar en ambos ejemplos, es que en el segundo ejemplo el código está tabulado. Este hecho no supone ningún cambio a la hora de ejecutar el código, sin embargo, sirve para saber de un vistazo rápido qué elementos están dentro de otros y qué elementos están a un nivel superior y cuáles no.

A veces, la programación con etiquetas acarrea algunos problemas y, sobre todo, esto ocurre cuando utiliza la etiqueta `<form>`.

Esto se debe a que esta etiqueta puede contener muchísimos atributos dentro de ella y, puede pasar que complique a la hora de revisar el código.

Cuando esto ocurre, puede recurrir a algunos elementos HTML que ayuda a evitar este tipo de problemas de manera sencilla. Vea el ejemplo:

Para el navegador va a ser lo mismo esto:

```
<form action="formularioalta.php" method="post" enctype="application/x-www-form-urlencoded"  
accept="image/jpeg" name="formulario prueba" target="_blank" accept-charset="UTF-8"  
autocomplete="off" novalidate>
```

Que esto:

```
<FORM ACTION="formularioalta.php" METHOD="post" ENCTYPE="application/x-www-  
form_urlencoded" ACCEPT="image/jpeg" NAME="formulario prueba" TARGET="_blank" ACCEPT-  
CHARSET="UTF-8" AUTOCOMPLETE="off" NOVALIDATE>
```

En caso de que el código no quede claro, como es el caso del primer ejemplo, puede recurrir a este atributo que tiene HTML y es que no diferencia entre mayúsculas y minúsculas.



TOME NOTA

Debe tener cuidado a la hora de programar sin distinguir mayúsculas o minúsculas, ya que, no todos los lenguajes aceptan este atributo. Es por este motivo que, si programa de esta manera de HTML y luego pasa a JavaScript o PHP, tiene que tener muy en cuenta este dato para no cometer errores.

Además, el uso de esta herramienta no solo sirve para facilitar la vida a otros y a todos los que deban revisar el código, sino que también sirve para identificar de una manera clara los campos del formulario.

Cuando se habla de los campos del formulario no solo se refiere a los elementos que son modificables, sino que son todos los elementos que componen el formulario, ya sean elementos visuales o no.

El texto plano es uno de los elementos más importantes de un formulario y el que nadie suele mencionar ni explicar.

Este elemento es el que mayormente se utiliza para dar información entre el programador y el usuario, de manera que el usuario pueda recibir las indicaciones correctas de cómo debe llenar los campos del formulario, seleccionar las opciones o elegir las opciones en un menú emergente.

Sin embargo, hoy día este tipo de información no suele aparecer; en su lugar, la información se da dentro de los cuadros de texto como, por ejemplo, puede encontrar un recuadro que ponga Nombre: y en su interior especifique, escriba su nombre completo.

Como ejemplo, vea el siguiente formulario de registro de Gmail:

Google

Crea una cuenta de Google

Nombre Apellidos

Nombre de usuario @gmail.com

Puedes utilizar letras, números y puntos

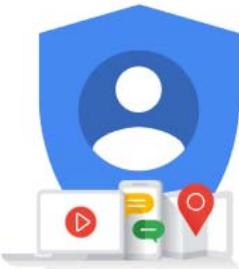
Prefiero usar mi dirección de correo electrónico actual

Contraseña Confirmación

Utiliza ocho caracteres como mínimo con una combinación de letras, números y símbolos

Mostrar contraseña

Prefiero iniciar sesión Siguiente



Una cuenta. Todo Google a tu disposición.

Español (España) ▾

Ayuda Privacidad Términos

Como puede observar, este es un buen ejemplo de lo que se ha mencionado anteriormente. Tiene las distintas cajas de texto, con los distintos tamaños adecuados para cada dato, y en cada una de las cajas aparece el dato que el usuario debe introducir en cada una de las cajas de texto.

A continuación, va a ver un ejemplo de código hecho por otros:

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="UTF-8">
    <title>Texto plano en formulario</title>
  </head>
  <body>
```

```
<form action="form.php" method="post" >

    Usuario:

    <input type="text" name="usuario" value="" /><br/>

    Contraseña:

    <input type="password" name="pass" value="" /><br/>

    <input type="submit" value="Enviar" />

</form>

</body>

</html>
```

Como puede observar, cada elemento que conforma el formulario tiene un texto plano para que el usuario pueda saber qué dato debe introducir en cada una de las cajas de texto.

Puede maquetar el texto plano de múltiples maneras o sin ningún tipo de maquetación.

Aquí es donde HTML ofrece diversas opciones y otros los que escoger la opción más correcta.

Por ejemplo, este código:

```
<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>Maquetación de texto plano en formulario</title>

    </head>

    <body>

        <form action="form.php" method="post" >

            Usuario:

            <input type="text" name="usuario" value="" />
```


Contraseña:

<input type="password" name="pass" value="" />

<input type="submit" value="Enviar" />

</form>

</body>

</html>

Al igual que este otro:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Maquetación de texto plano en formulario</title>

</head>

<body>

<form action="form.php" method="post" >

<p>Usuario:

<input type="text" name="usuario" value="" /></p>

<p>Contraseña:

<input type="password" name="pass" value="" /></p>

```
<br/>

<input type="submit" value="Enviar"/>

<br/>

</form>

</body>

</html>
```

Se tratarán de manera similar. Como en este ejemplo puede encontrar opciones casi infinitas. Por ejemplo, podría utilizar tablas para introducir los elementos en ellas.

1.2.4. Etiquetas de los formularios

Como ya sabe, los formularios se declaran mediante etiquetas al igual que todos los elementos en HTML y, como ya ha visto, estos elementos tienen diferentes atributos que hacen que se comporten de manera diferente.

Las etiquetas más importantes de los formularios son las siguientes:

Form

Esta etiqueta es la que se utiliza para delimitar los formularios, es decir, todo lo que haya dentro de esta etiqueta será parte del formulario.

Esta es la etiqueta más importante para crear formularios en HTML, ya que, sin ella todo lo demás carecería de sentido. Además, para darle más diversidad, esta etiqueta tiene distintos atributos.

Vea un pequeño ejemplo:

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="UTF-8">
    <title>Ejemplo de formulario</title>
  </head>
  <body>
```

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

    Nombre:

    <input type="text" name="nombre" value="" />

    <br/>

    <input type="button" onclick="confirmacion()" value="Enviar" />

</form>

</body>

</html>
```

Input

Esta etiqueta es la más importante para crear los elementos dentro de los formularios, puesto que, con ella se pueden crear una gran diversidad de elementos dependiendo de los atributos que utilice en ella.

Con esta etiqueta, podrá crear:

- Botones de envío
- Campos de texto
- Botones de opciones
- Campos ocultos
- Otros botones

Textarea

Esta etiqueta es específica para los formularios, sin embargo, es utilizada que las dos etiquetas anteriores. Con ella, puede declarar un cuadro de texto más extenso que con input, que solo deja definir un cuadro de texto de una línea.

Más adelante, verá los atributos de esta etiqueta.

1.2.5. Tamaños, columnas y filas de los formularios

Como ya sabe, los formularios pueden ser de distintos tamaños, pueden ir desde el formulario más básico que son los formularios de búsqueda, hasta los más complejos que son los formularios que puede encontrar en la Agencia Tributaria.

Por todo ello, la presentación de los diferentes formularios es muy importante, no solo por la cuestión de apariencia, sino por cuestión de practicidad.

Si crea un formulario con demasiada poca información podría resultar totalmente inútil para la funcionalidad de la página web donde se albergue el mismo.

De la misma manera, un formulario con demasiada información podría llegar a agotar a los usuarios. Incluso, si los datos están muy juntos u de otros pueden hacer que la mayoría de los usuarios ni siquiera completen el formulario por falta de paciencia.

Por lo tanto, el tamaño del formulario y la manera en la que estén ordenados los datos en él es una cuestión de vital importancia.

Llegados a este punto, ha visto formularios simples donde se han aplicado las soluciones más fáciles de manera aparente. Sin embargo, para crear un formulario, de la misma forma que ocurre con otros aspectos de la programación en HTML, puede presentar los elementos de manera diferente como, por ejemplo:

- Utilizando párrafos para la separación entre los elementos.
- Utilizando saltos de línea normales.
- Utilizando una etiqueta para que el navegador muestre los datos de la misma manera en la que están programados.
- Utilizando tablas para añadir los elementos.
- Agrupando los elementos mediante etiquetas.

Deberá siempre elegir la opción más recomendada para la comodidad del programador y la coherencia con el resultado que necesita el usuario junto con la cantidad de elementos y datos que tenga el formulario.

Como ejemplo, va a ver el formulario de satisfacción que ofrece la Junta de Andalucía en su página web. Va a encontrar el siguiente formulario:

Encuesta de satisfacción de usuarios

La Agencia Tributaria de Andalucía realiza un estudio para conocer la opinión que tienen los ciudadanos de los servicios que presta. Para ello se están realizando entrevistas a personas que hayan utilizado sus servicios y que voluntariamente decidan relatarlas. La entrevista es completamente anónima. No existen respuestas buenas o malas. No interesa este todo conocer su opinión.

Otras gracias por tu colaboración

Ha completado el 0% de esta encuesta
99%
100%

Agencia Oficina

* Precio
Selecione una de las siguientes opciones

Selección, por favor

* La razón por la que acudió a este centro de la Agencia Tributaria de Andalucía la realiza

Selección, una de las siguientes opciones

• A título particular

• Como profesional

Otros
Selecione una de las siguientes opciones

• Mujer

• Hombre

• Sin respuesta

* En el último año, ¿cuántas gestiones ha realizado en esta Oficina?

Selección, una de las siguientes opciones

• 0-1-2

• 3-4-5

• 6-10

• >10

	1	2	3	4	5	Sin respuesta
Instalaciones del centro al que acudió	<input type="radio"/>	<input checked="" type="radio"/> Sin respuesta				
Horarios de atención al público de la Agencia	<input type="radio"/>	<input checked="" type="radio"/> Sin respuesta				
Identificación de los servicios que se ofrecen en el centro	<input type="radio"/>	<input checked="" type="radio"/> Sin respuesta				
Sistema de atención al público y cita previa	<input type="radio"/>	<input checked="" type="radio"/> Sin respuesta				

Como puede observar, en este formulario puede encontrar elementos que ya conoce, sin embargo, se muestra un formulario bastante antiguo, aunque es útil. Los elementos de este formulario parecen programados todos en el mismo párrafo, aunque están relacionados entre sí.

Lo que se muestra al usuario es una programación totalmente invisible para otros, ya que, si no puede ver el código fuente de la página no podrá realmente saber cómo está programado este formulario, si tiene saltos de línea, diferentes párrafos, etc.

Maquetar formulario utilizando br

Este es el elemento más simple que puede utilizar para maquetar los datos de un formulario. Como ya sabe, aunque otros implemente saltos de línea en el código, el navegador no va a ser capaz de interpretarlos, es por esto que deberá utilizar la etiqueta
.

Por lo tanto, cuando introduce saltos de línea en el código es tan solo para organizar el código de manera que sea más fácil visualizarlo.

Vea un ejemplo de uso de br:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Maquetación con br en formulario</title>
```

```
</head>
```

```
<body>

<form action="form.php" method="post" >

    Usuario:
    <input type="text" name="usuario" value="" />
    <br/>

    Contraseña:
    <input type="password" name="pass" value="" />

    Repetir Contraseña:
    <input type="password" name="reppass" value="" />
    <br/>

    Subir foto:
    <input type="file" name="foto" />
    <br/>

    <input type="submit" value="Enviar" />
    <br/>

</form>

</body>

</html>
```

Vea cómo se muestra en el navegador:

Usuario:

Contraseña: Repetir Contraseña:

Subir foto: Seleccionar archivo Ningún archivo seleccionado

Maquetar formulario utilizando p

Como ha visto anteriormente, maquetar los datos con br es el método más simple, pero el más elegante de todos.

Aunque esto es totalmente opaco para el usuario, siempre se recomienda seguir u estándares de calidad y, por lo tanto, utilizar el método más correcto.

Para ello, puede utilizar la etiqueta p que hará que se distribuyan los elementos en párrafos y, además, evitará algunos problemas que pueden ocurrir dependiendo del navegador que utilice el usuario.

Vea un ejemplo:

```
<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Maquetación con p en formulario</title>

</head>

<body>

    <form action="form.php" method="post" >

        <p>
            Usuario:
            <input type="text" name="usuario" value="" />
        </p>

        <p>
            Contraseña:
            <input type="password" name="pass" value="" />
        </p>
    </form>
</body>
</html>
```

```
<p>  
Repetir Contraseña:  
<input type="password" name="reppass" value="" />  
</p>  
  
<p>  
Subir foto:  
<input type="file" name="foto"/>  
</p>  
  
<input type="submit" value="Enviar"/>  
</form>  
  
</body>  
</html>
```

Vea el resultado:

Usuario:

Contraseña:

Repetir Contraseña:

Subir foto: Seleccionar archivo Ningún archivo seleccionado

Maquetar formulario utilizando pre

Con esta etiqueta puede solucionar el problema de que cuando añade saltos de línea en el código de manera visual.

Sin embargo, esto es un arma de doble filo, por lo tanto, debe tener mucho cuidado cuando tabula o utiliza saltos de línea para mantener la claridad del código, puesto que, al utilizar esta etiqueta el navegador interpretaría todos lo que introduzca por teclado.

Vea un ejemplo:

```
<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>Maquetación con pre en formulario</title>

    </head>

    <body>

        <form action="form.php" method="post" >

            <pre>

                Usuario:

                <input type="text" name="usuario" value="" />

            </pre>

            <pre>

                Contraseña:

                <input type="password" name="pass" value="" />

            </pre>

            <pre>

                Repetir Contraseña:

                <input type="password" name="reppass" value="" />

            </pre>

            <pre>

                Subir foto:


```

```
<input type="file" name="foto"/>  
</pre>  
  
<input type="submit" value="Enviar"/>  
  
</form>  
  
</body>  
  
</html>
```

Vea lo que muestra el navegador:

Usuario:

Contraseña:

Repetir Contraseña:

Subir foto:
Seleccionar archivo Ningún archivo seleccionado

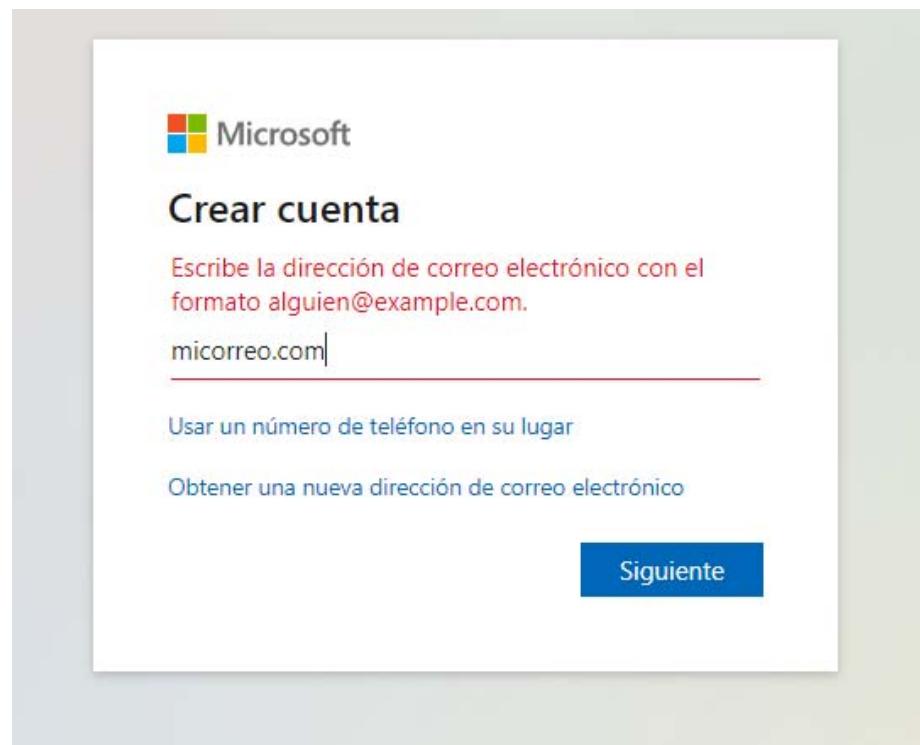
1.3. Controles de formulario

Los controles de un formulario son elementos que se programan utilizando, para ello, etiquetas propias. Estos elementos estarán albergados dentro de la etiqueta de formulario y van a servir para que dote al formulario de las herramientas adecuadas para su utilidad.

Se denomina a estos elementos del formulario controles, puesto que, se utilizan para controlar el flujo de la información y, aunque a simple vista estos elementos parezcan que quedan relevados al usuario que los rellene, el programador es realmente quien tiene el control de los datos que el usuario puede introducir.

Por ejemplo, cuando se encuentra en un formulario una caja de texto que sirve para enviar un correo electrónico, no tendría sentido que no llevara una arroba.

Vea un ejemplo:



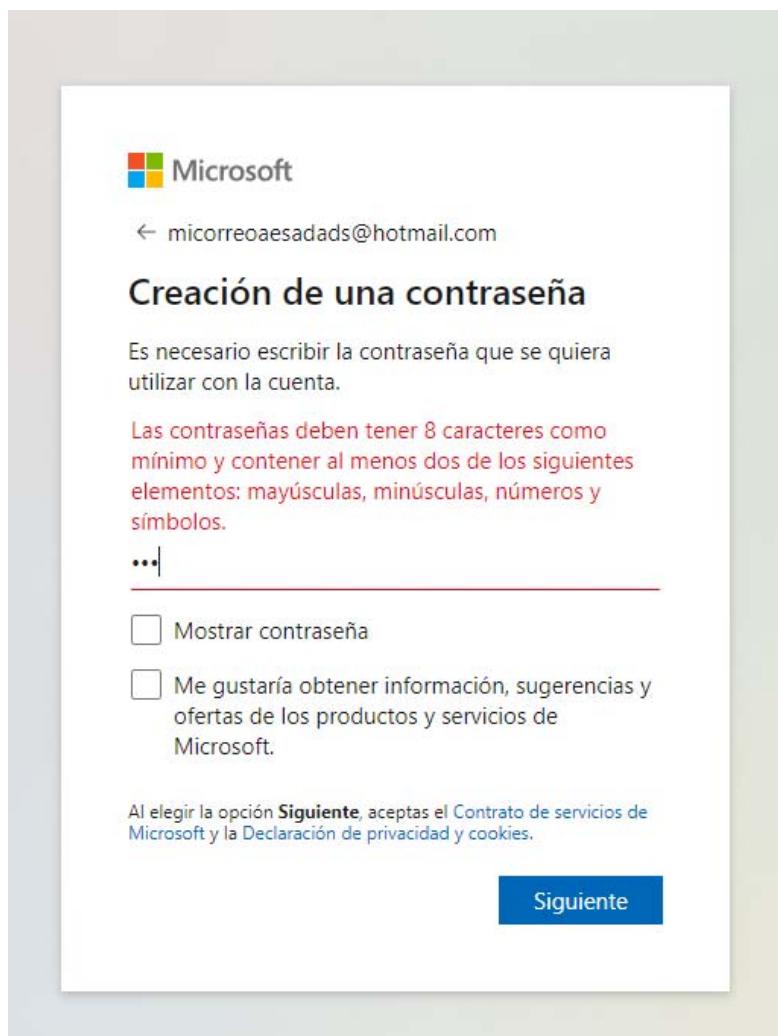
Como puede ver, si intenta registrar en Outlook e introduce un correo sin la arroba, lo primero que va a saltar es un error en el formulario. Esto ocurre porque el programador ha establecido el control del flujo de la información.

Otro de los ejemplos que puede ver es los caracteres máximos que pueden contener las cajas de texto según se programen.

Por ejemplo, en el campo de la edad, lo más lógico es establecer un máximo de tres caracteres porque no hay nadie que pueda tener una edad de mil años.

Este tipo de control de flujo donde se suele utilizar más comúnmente es en los campos de contraseña y de usuario. Esto se debe a que según la base de datos y a cómo esté establecida, podrá aceptar un número de caracteres mayor o menor.

Por ejemplo, siguiendo en la plataforma de Outlook puede ver que la contraseña tiene un mínimo de 8 caracteres; es decir, que no puede establecer una contraseña de siete caracteres. Vea un ejemplo:



Debe tener muy presente que establecer controles no supone que los propios controles sean restricciones del flujo de información, sino que son los propios elementos del formulario que permiten controlar el flujo de la información.



TOME NOTA

Decir elementos, campos o controles de un formulario tiene el mismo significado.

Vea un ejemplo de formulario:

```
<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>Formulario simple</title>

    </head>

    <body>

        <form action="form.php" method="post" >

            <pre>

                Nombre:

                <input type="text" name="nombre" value="" />

            </pre>

            <input type="submit" value="Enviar"/>

        </form>

    </body>

</html>
```

En este ejemplo de formulario, puede diferenciar dos controles, más una línea de texto plano.

En el primero de los controles, se indica que el formulario va a enviar la información que haya introducido el usuario en el cuadro de texto.

Además, el formulario le indica al usuario que debe introducir su nombre; sin embargo, ese recuadro no tiene ninguna restricción, por lo que el usuario podría introducir cualquier cosa.

El segundo control es un botón de tipo submit, es decir, que cuando el usuario pulse ese botón, se enviará la información que contenga el formulario.

En este ejemplo es muy fácil aislar los elementos que hay en el formulario y el flujo de la información. En cambio, no siempre es así de fácil aislar estos elementos; ya que, dependerá de la complejidad del formulario y de lo bien organizado que esté el código.

1.3.1. Descripción de los controles de los formularios

Como todos los elementos de HTML, los controles se definen también utilizando etiquetas en las que puede establecer diferentes atributos, que podrán ser declarados o no, y además, podrán contener otros elementos dentro de ellos.

Cada elemento se creará de una manera determinada, sin embargo, la mayoría se van a crear utilizando la etiqueta `<input>`, que en inglés significa entrada de datos.



TOME NOTA

La entrada y modificación de datos de la información es un punto muy importante de la informática. De hecho, es el más importante de la informática.

Utilizando la etiqueta `input` puede declarar:

- Campos de contraseña.
- Campos de texto.
- Listas de botones de radio.
- Lista de cajas.
- Botones de borrar.
- Botones de enviar.
- Botones para subir archivos.
- Otros botones.
- Campos ocultos.
- Imágenes.

Por otra parte, también puede establecer otro tipo de controles mediante otras etiquetas como:

- Unir elementos del mismo formulario con `<label>`
- Cuadros de texto con diversas líneas con `<textarea>`
- Nombrar agrupaciones de elementos con `<legend>`
- Agrupar elementos para separarlos de manera visual con `<fieldset>`
- Crear lista desplegable con `<select>`
- Definir opciones en las listas desplegables `<option>`
- Crear grupos dentro de las listas desplegables `<optgroup>`
- Crear botones con la etiqueta `<button>`

- Crear varias opciones en un campo de texto con <datalist>
- trar datos de salida con <output>

De este modo, podrá utilizar cualquiera de estos elementos cuando sea necesario.

Vea un ejemplo de formulario que utiliza estos controles:

```
<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Formulario poniendo todos los controles en uso</title>

</head>

<body>

    <form action="form.php" method="post" >

        <fieldset>

            <legend>

                Controles de formulario

            </legend>

            <label>

                <input type="text" name="campoTexto" value="Esto es un campo de texto"/><br/>

                <input type="password" name="campoPass" value="Pass"/><br/>

                <input type="checkbox" name="opcionesCaja" value="Opción con botones tipo caja"/><br/>

                <input type="radio" name="opcionesRadio" value="Opción con botones tipo radio"/><br/>

                <input type="submit" name="bontonEnvio" value="Botón Enviar"/><br/>

                <input type="reset" name="botonBorrar" value="Botón de borrar"/><br/>

            </label>
        </fieldset>
    </form>
</body>

```

```
<input type="file" name="subirArchivo" value="Subir archivo"/><br/>

<input type="hidden" name="campoOculto" value="Este campo está oculto"/><br/>

<input type="image" name="imagen" src="imagen.jpg"/><br/>

<textarea name="areaTexto" rows="2" cols="55"></textarea><br/>

<select name="opciones">

    <optgroup label="Grupo de opciones 1">

        <option value="opcion1.1">Opción 1.1</option>

        <option value="opcion1.2">Opción 1.2</option>

    </optgroup>

    <optgroup label="Grupo de opciones 2">

        <option value="opcion2.1">Opción 2.1</option>

        <option value="opcion2.2">Opción 2.2</option>

    </optgroup>

</select><br/>

    <button name="botonDesabilitado" type="button" value="Botón desabilitado" disabled></button><br/>

<input list="opciones" name="opciones">

<datalist id="opciones">

    <option value="Opción 1"></option>

    <option value="Opción 2"></option>

</datalist>

</label>

</fieldset>

</form>
```

</body>

</html>

Vea lo que muestra el navegador:



Como ya puede haber intuido, este código es tan solo una recopilación de los controles que ha visto, por lo tanto, está desprovisto de cualquier sentido técnico o funcional.

RECURSO MULTIMEDIA



1.3.2. Utilización de botones de acción

Como ya sabe, los formularios web se crearon a partir de los formularios reales clásicos. Ambos tienen la misma función, recoger información.

Sin embargo, el formulario web se diferencia del real en que no debe dárselo a alguien específico, sino que será el servidor el encargado de guardar, revisar y utilizar ese formulario cuando sea necesario.

Por lo tanto, cuando un usuario web termina de llenar un formulario, deberá tener alguna manera de enviarlo, pues físicamente le será imposible hacerlo. Por este motivo, utiliza los botones del formulario.

Los botones que puede encontrar en un formulario son:

- Botones creados con la etiqueta <button>
- Botones creados con la etiqueta <input> y el atributo reset.
- Botones creados con la etiqueta <input> y el atributo submit.

El atributo submit es el que cumple la función de enviar el formulario. Sin embargo, los otros dos tipos de botones tienen otro tipo de funcionalidades.

Por ejemplo, si esta llenando un formulario y comete algún error en el proceso, o necesita limpiar el formulario por completo, utilice el botón reset, que es el que permitirá resetear el formulario por completo.

Por otra parte, con la etiqueta button puede crear los siguientes tipos de botones:

- Un botón con un texto largo.
- Un botón con una imagen.
- Un botón que envíe parte de una información.
- Un botón fuera de un formulario, que esté ligado a él.
- Un botón para eliminar parte de la información.
- Un botón que lleve a una url específica.
- Un botón que sirva para marcar o desmarcar la validez de un formulario.

Vea un ejemplo en el que puede encontrar todos los tipos de botones:

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="UTF-8">

    <title>Ejemplo de botones</title>

  </head>

  <body>

    <form action="form.php" method="post" id="formulario" enctype="multipart/form-data">

      Usuario:

      <input type="text" name="usuario" value="" />

      <br/>
```

Contraseña:

```
<input type="text" name="pass" value="" />
```

```
<br/>
```

```
<input type="submit" value="Enviar" />
```

```
<br/>
```

```
<input type="reset" value="Eliminar todo" />
```

```
<br/>
```

```
<button type="button" name="boton" autofocus>Tengo el foco</button>
```

```
<br/>
```

```
<button type="button" name="boton" disabled>Estoy inactivo</button>
```

```
<br/>
```

```
<button type="submit" name="boton" form="formulario" value="Enviar formulario">Enviar formulario</button>
```

```
<br/>
```

```
<button type="button" name="boton" formaction="admin.php">Enviar los datos al administrador de la web</button>
```

```
<br/>
```

```
<button type="button" name="boton" formenctype="text/plain">Enviar como texto plano</button>
```

```
<br/>
```

```
<button type="submit" name="boton" formnovalidate>Enviar sin validar</button>
```

```
<br/>
```

```
<button type="submit" name="boton" formmethod="get">Enviar con get</button>
```

```
<br/>
```

```
<button type="submit" name="boton" formtarget="_blank">Mostrar resultado en otra  
página</button>  
  
</form>  
  
</body>  
  
</html>
```

A continuación, verá lo que muestra el navegador:

Usuario:

Contraseña:

1.3.3. Utilización de listas desplegables

Las listas desplegables son listas que puede encontrar en los formularios. Estas listas, como su propio nombre indica, se pueden desplegar y elegir una de las opciones que haya en la lista. Por ejemplo, estas listas se pueden encontrar en las tiendas online para seleccionar su país o ciudad.

Por ejemplo, puede ver la siguiente imagen donde sale una lista desplegable para indicar su dirección de envío de una tienda online:

The screenshot shows a web form for address input. On the left, there are several input fields with placeholder text: 'Alias' (Placeholder: 'Optional'), 'Nombre' (Placeholder: 'eqweqweqwe'), 'Apellidos' (Placeholder: 'qweqweqweqwe'), 'Teléfono móvil' (Placeholder: 'Optional'), 'Empresa' (Placeholder: 'Optional'), 'Número de IVA' (Placeholder: 'Optional'), 'País' (Placeholder: 'España'), 'Estado' (Placeholder: 'por favor escoge--'), 'Dirección' (Placeholder: 'A Coruña'), 'Dirección Complementaria' (Placeholder: 'Optional'), 'Código postal/Zip' (Placeholder: 'Optional'), 'Ciudad' (Placeholder: 'Optional'), and a checked checkbox 'Usar también este formulario'.

A dropdown menu is open for the 'Estado' field, listing Spanish provinces. The province 'Alava' is selected and highlighted in blue. Other options in the list include: Albacete, Alicante, Almería, Asturias, Ávila, Badajoz, Baleares, Barcelona, Burgos, Cáceres, Cádiz, Cantabria, Castellón, Ciudad Real, Córdoba, Cuenca, and Girona.

A continuación, vea un ejemplo de listas desplegables con todos los atributos que puede utilizar en ellas:

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="UTF-8">
    <title>Ejemplo de listas desplegables</title>
  </head>
  <body>
    <form action="form.php" method="post" id="formulario">
      Nombre: <input type="text" name="usuario" value="" />
      <br/>
    </form>
  </body>
</html>
```

Edad:

```
<select autofocus name="edad" required>

    <option value="menor">Me de 18</option>

    <option value="adulto">De 18 a 65</option>

    <option value="jubilado">Más de 65</option>

</select>
```

```
<br/>
```

Idiomas:

```
<select name="idiomas" size="3" multiple>

    <option value="ingles">Inglés</option>

    <option value="frances">Francés</option>

    <option value="aleman">Alemán</option>

</select>
```

```
<br/>
```

```
<input type="submit" value="Enviar"/>
```

```
</form>
```

```
<br/>
```

Opinión del formulario:

```
<select form="formulario" disabled>

    <option value="buena">Buena</option>

    <option value="regular">Regular</option>

    <option value="mala">Mala</option>
```

```
</select>  
  
</body>  
  
</html>
```

Lo que muestra el navegador será lo siguiente:

Nombre:

Edad: Menos de 18

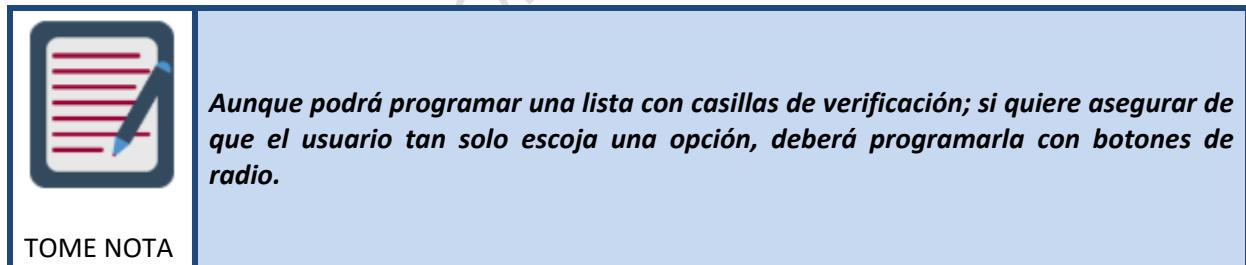
Idiomas:

Opinión del formulario: Buena

1.3.4. Utilización de casillas de verificación

Las casillas de verificación son elementos del formulario que permiten que el usuario seleccione opciones de manera individual. Por lo que el usuario podrá escoger tantas opciones como quiera.

Sin embargo, si quiere que las opciones sean excluyentes entre sí, utilice los llamados botones de radio, por lo que, el usuario tan solo podrá escoger una opción.



Vea a continuación un ejemplo en el que puede ver todos los tipos de casillas de verificación y de opciones de radio:

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
<meta charset="UTF-8">
```

```
<title>Ejemplo de casillas de verificación y botones de radio</title>

</head>

<body>

<form action="form.php" method="post" id="formulario">

    Encuesta de opinión

    <br/>

    <input type="radio" name="opinion" value="buena" tabindex="2" align="left">Me ha parecido
    buena en general.

    <br/>

    <input type="radio" name="opinion" value="regular" tabindex="1" align="left">No creo que
    haya sido ni buena ni mala.

    <br/>

    <input type="radio" name="opinion" value="mala" tabindex="3" align="left">Me ha parecido
    una mala encuesta.

    <br/><br/>

    ¿Cómo ha conocido su encuesta?

    <br/>

    <input type="checkbox" name="fuente" value="internet" tabindex="3" align="left" checked> A
    través de anuncios de Internet

    <br/>

    <input type="checkbox" name="fuente" value="tv" tabindex="2" align="left"> A través de
    anuncios en televisión

    <br/>

    <input type="checkbox" name="fuente" value="prensa" tabindex="1" align="left"> A través de
    anuncios en la prensa

    <br/><br/>
```

```
<input type="submit" value="Enviar"/>  
</body>  
</html>
```

Vea lo que muestra el navegador:

Encuesta de opinión

- Me ha parecido buena en general.
- No creo que haya sido ni buena ni mala.
- Me ha parecido una mala encuesta.

¿Cómo ha conocido nuestra encuesta?

- A través de anuncios de Internet
- A través de anuncios en televisión
- A través de anuncios en la prensa

1.3.5. Utilización de campos de textos

Puede decir que los campos de texto son elementos que permiten que los usuarios introduzcan texto. Puede ser un texto largo o corto dependiendo de lo que el campo requiera, o también puede encontrar texto cifrado.

Este tipo de elemento es el más común de los formularios, puesto que es el elemento que permite que el usuario interactúe de manera total como, por ejemplo, puede ver en los formularios de los buscadores, en los que se utilizan para llenar datos de registro o en los que encuentra en las tiendas online.

Principalmente, existen tres tipos de campo de texto:

- Las áreas de texto.
- Los cuadros de texto.
- Los cuadros de contraseña.

Los elementos más utilizados en el formulario son los cuadros de texto; ya que, aporta sencillez de cara al usuario, sin embargo, no al programador.

Para añadir elementos de texto al formulario, utilice la etiqueta input con el tipo text. Estos elementos, además, tienen atributos importantes:

- Size
- Maxlength
- Name
- Value
- Tabindex
- Align

Ahora, verá todos estos atributos aplicados a un código:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Ejemplo de campos de texto y sus atributos</title>

</head>

<body>

<form action="form.php" method="post" id="formulario">

    Usuario:

    <input type="text" size="30" maxlength="30" name="usuario" value="Usuario" align="left"
    tabindex="1"/>

    <br/>

    Contraseña:

    <input type="password" size="10" maxlength="10" name="pass" value="12345" align="left"
    tabindex="2"/>

    <br/>

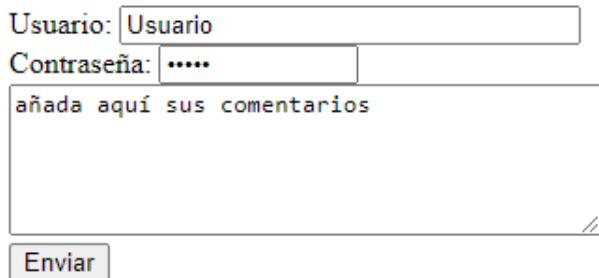
    <textarea cols="40" rows="5" name="comentarios" tabindex="3" wrap="virtual">añada aquí
    sus comentarios</textarea>

    <br/>

    <input type="submit" value="Enviar"/>
```

```
</form>  
  
</body>  
  
</html>
```

Ahora, verá lo que muestra el navegador:



Usuario:
Contraseña:
añada aquí sus comentarios

1.4. Formularios y eventos. Criterios de accesibilidad y usabilidad en el diseño de formularios

Puede aportarle más interactividad a los formularios mediante el uso de eventos que se ejecutan cuando el código del formulario se cargue en el cliente.

Ya ha visto algunos eventos, pero vea la lista completa de ellos:

- **Onload:** Este evento se utiliza para activar algo concreto cuando el usuario termina de cargar la página.
- **Onunload:** Este evento se activará cuando el usuario cierre o salga del formulario.
- **Onclick:** Este evento se activará cuando el usuario clique sobre el elemento concreto.
- **Ondblclick:** Este evento de activará cuando el usuario haga doble clic sobre el elemento concreto.
- **Onmousedown:** Este elemento se activará cuando presione el ratón sobre el elemento.
- **Onmouseover:** Este elemento se activará cuando pase el ratón por encima del elemento.
- **Onmouseup:** Este evento se activará cuando el puntero suelte el elemento.
- **Onmousemove:** Este evento se activará cuando move el puntero por encima del elemento.
- **Onmouseout:** Este evento se activará cuando el puntero salga del elemento.
- **Onfocus:** Este evento se activará cuando el elemento reciba el foco.
- **Onblur:** Este evento se activará cuando el elemento pierda el foco.
- **Onkeydown:** Este elemento se activará cuando se presione una tecla mientras el elemento tiene el foco.
- **Onkeypress:** Este evento se activará cuando se presione una tecla y se suelte cuando el elemento tiene el foco.

- **Onkeyup:** Este evento se activará cuando se suelte una tecla mientras el elemento tenga el foco.
- **Onsubmit:** Este evento se activará cuando el formulario se haya enviado.
- **Onselect:** Este evento se activará cuando el usuario seleccione el texto de un campo de texto.
- **Onreset:** Este evento se activará cuando el usuario reestablezca los valores por defecto del formulario.
- **Onchange:** Este evento se activará cuando el elemento pierde el foco y se ha modificado su valor.

Si utiliza todos estos elementos y, además, añade el uso de JavaScript, tendrá posibilidades casi infinitas para personalizar los formularios.

Sin embargo, si utiliza estos elementos junto con JavaScript, deberá tener en cuenta otros criterios de presentación.

La primera herramienta que verá es la que permite agrupar datos dentro de los formularios con el fin de diferenciar los múltiples grupos de datos que el usuario debe llenar dentro del propio formulario.

Por ejemplo, puede ver en el formulario de búsqueda avanzada de Google cómo están agrupados los diferentes grupos de datos. Esto se debe a que en este formulario hay muchos datos para llenar:

The screenshot shows the Google Advanced Search interface. At the top, there is a red link labeled "Búsqueda avanzada de vídeos". Below it, there are two main sections: "Buscar videos con..." and "Haz lo siguiente en el cuadro de búsqueda".

Buscar videos con...

- todas estas palabras: (description: Escribe las palabras importantes: técnicas de enseñanza)
- esta palabra o frase exactas: (description: Escribe las palabras exactas entre comillas: "lectura rápida")
- cualquiera de estas palabras: (description: Escribe OR entre todas las palabras que quieras: niños OR adultos)
- ninguna de estas palabras: (description: Añade un signo menos delante de las palabras que no quieras que aparezcan -método silábico)

A continuación, limitar los resultados por...

idioma:	<input type="text" value="cualquier idioma"/>	(description: Busca páginas en el idioma que seleccionas.)
Duración	<input type="text" value="cualquier duración"/>	(description: Especifica la duración que quieras.)
Fecha	<input type="text" value="en cualquier momento"/>	(description: Busca páginas actualizadas en el periodo de tiempo especificado.)
Calidad	<input type="text" value="cualquier calidad"/>	(description: Solo se muestran videos de alta calidad.)
sitio o dominio:	<input type="text"/>	(description: Busca un sitio (como youtube.com) o limita los resultados a un dominio como, por ejemplo, .edu, .org o .gov.)
subtítulos:	<input type="text" value="Cualquier tipo"/>	(description: Busca videos con subtítulos.)
Búsqueda Segura:	<input type="text" value="Mostrar resultados explícitos"/>	(description: Indica a Búsqueda Segura si quieres que filtre contenido sexualmente explícito.)

Búsqueda avanzada

The header of a presentation slide features a dark red bar at the top with a small icon and the text 'Más Info'. Below it is a blue bar with the title 'RECURSO MULTIMEDIA'. Underneath the title is a row of seven colorful icons representing different multimedia concepts: a computer monitor with a graduation cap, a smartphone with an apple, a keyboard with a book, a desktop computer with documents, a laptop with letters, a monitor with a play button, and a monitor with a hand cursor.

1.4.1. Agrupación de datos

Como ya ha visto, agrupar los datos en un formulario facilitará el ingreso de los datos por parte del usuario para que, de ese modo, pueda identificar los tipos de datos que tiene que añadir de manera rápida.

Para realizar esta agrupación, utilice la etiqueta `fieldset`. Esta etiqueta se compone de una etiqueta de apertura y de cierre, y todo lo que contenga en su interior serán elementos agrupados.

Los elementos que se agrupan mediante la etiqueta `fieldset`, estarán dentro de cajas diferentes para que puedan diferenciarse entre sí.

Los atributos más importantes y que más se utilizan son los siguientes:

- Name.
- Form.
- Disabled.

Vea un código con todos estos elementos:

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="UTF-8">
    <title>Ejemplo de fieldset y legend</title>
  </head>
  <body>
```

```
<form action="form.php" method="post" id="formulario">

<fieldset name="datosPersonales">

    Nombre:

    <input type="text" name="nombre" value="" />

    <br/>

    Apellidos:

    <input type="text" name="apellidos" value="" />

    <br/>

</fieldset>

<fieldset disabled>

    Usuario:

    <input type="text" name="usuario" value="" />

    <br/>

    <input type="submit" name="enviar" value="Enviar" />

</fieldset>

</form>

<br/>

<fieldset form="formulario">

    <legend align="right">

        Comentarios:

    </legend>

    <textarea></textarea>

</fieldset>
```

```
</body>
```

```
</html>
```

Vea el resultado en el navegador:

The screenshot shows a web page with a form. At the top, there are three input fields: 'Nombre' (Name), 'Apellidos' (Last Name), and 'Usuario' (User). Below these is a blue 'Enviar' (Send) button. To the right of the 'Usuario' field, the word 'Comentarios:' is displayed above a large, empty text area for comments.

1.4.2. Adecuación del tamaño del formulario (división en distintas páginas)

Lo más normal es que encuentre formularios con pocos elementos y que no sean muy extensos. Esto es así porque están creados para cumplir una función sin molestar al usuario con extensos textos ni muchos elementos.

Sin embargo, puede encontrar que en algunas ocasiones necesite formularios que tengan varias páginas. El motivo puede ser desde que haya muchos datos para introducir, hasta que se necesite algún paso previo antes de poder llenar ese formulario.

Debido a esto, es algo esencial, existe una división en distintas páginas.

De todas formas, deberá ser consciente de que este diseño será algo peculiar y solo lo utilice en las siguientes situaciones:

- Se necesita pasar por un formulario previo antes de poder llenar el formulario final.
- El formulario es muy extenso y tiene muchos campos.

Un ejemplo de esto es cuando loguea en una red social y tiene que llenar el formulario del log in, y posteriormente tiene un formulario de búsqueda donde puede insertar el elemento a buscar.

A continuación, vea un ejemplo:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">

<title>Ejemplo formulario</title>

</head>

<body>

<form action="form.php" method="post" id="formulario">

    Usuario:

    <input type="text" size="30" maxlength="30" name="usuario" value="Usuario"/>

    <br/>

    Contraseña:

    <input type="password" size="10" maxlength="10" name="pass"/>

    <br/>

    <input type="submit" value="Entrar"/>

</form>

</body>

</html>
```

Vea lo que muestra el navegador:

Usuario:

Contraseña:

En este caso, si el servidor comprueba que los datos del login son correctos, el usuario podrá acceder al buscador que se encuentra en la segunda página:

```
<!DOCTYPE html>

<html>
```

```
<head>

    <meta charset="UTF-8">

    <title>Ejemplo buscador</title>

</head>

<body>

    <form action="form.php" method="post" id="formulario">

        Introducir datos a buscar:

        <br/>

        <input type="text" name="buscador"/>

        <br/>

        <br/>

        <input type="submit" value="Buscar"/>

    </form>

</body>

</html>
```

Veálo en el navegador:

Introducir datos a buscar:

1.4.3. Identificación de los campos obligatorios

En todos los tipos de formularios, ya sean los clásicos o los virtuales, cuando se añade un campo es porque quiere que el usuario final lo complete, puesto que, no tendría sentido añadir un campo que no sirviera para nada.

En cambio, hay veces que puede encontrar con formularios que tienen campos que son realmente vitales para recoger cierta información, pero otros no lo son tanto.

Cuando topa con esta situación, puede programar los campos realmente importantes como campos obligatorios y los que no lo son tanto, como camposopcionales.

Por ejemplo, cuando se encuentra con un formulario de registro, lo normal es que el nombre de usuario, la contraseña y el correo electrónico sean como mínimo obligatorios. Sin embargo, otros datos como la edad, la ciudad o la fecha de nacimiento pueden ser opcionales.

Para identificar estos campos como obligatorios, se suelen utilizar asteriscos que acompañan a estos campos. Esto sucede porque el usuario debe saber de manera rápida cuáles son los campos obligatorios y cuáles no.

Vea a continuación un ejemplo de esto:

The screenshot shows a registration form with the following fields:

- Correo electrónico***: A text input field with placeholder text "o introduce:".
- Contraseña***: A text input field with placeholder text "FORTALEZA" and a password strength meter consisting of five bars.
- Repetir contraseña***: A text input field.
- Fecha de Nacimiento***: A date input field with dropdown menus for Día, Mes, and Año.
- Conditions**: A note stating "Esta información nos permitirá ofrecerte contenidos y servicios más ajustados a tu perfil".
- Terms and Conditions**: A checkbox followed by the text "He leído y entiendo las Condiciones de uso y Protección de datos ? y mis Preferencias de privacidad +".
- CONTINUAR**: A blue button at the bottom of the form.

Por otra parte, además de indicarle al usuario de manera visual qué campos son obligatorios, también puede hacer que los usuarios no puedan enviar el formulario hasta que no hayan llenado estos campos obligatorios. Para ello tiene dos opciones:

- Programar de cara al servidor que estén llenos ciertos campos para que el formulario pueda ser aceptado.
- Programar en el cliente utilizando etiquetas de campos obligatorios y un atributo que indique que esos campos son obligatorios.

El atributo que utilice será required. Este atributo indicará que en el campo en donde esté añadido será obligatorio y se debe llenar. En caso de que el usuario pulse el botón enviar y los campos obligatorios con este atributo no estén llenos, el formulario no se enviará.

Vea a continuación un ejemplo de formulario que recoge campos obligatorios que deberán ser llenados por el usuario:

```
<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>Ejemplo campos obligatorios</title>

    </head>

    <body>

        <form action="form.php" method="post" id="formulario">

            Usuario:

            <input type="text" size="30" maxlength="30" name="usuario" align="left"/>
            <br/>

            Contraseña:

            <input type="password" size="10" maxlength="10" name="pass" value="12345" align="left" tabindex="2"/>
            <br/>

            Comentario en el libro de visitas:
```

```
<br/>

<textarea cols="40" rows="5" name="comentarios" tabindex="3" required>Añada aquí sus
comentarios:</textarea>

<br/>

<input type="submit" value="Enviar"/>

</form>

</body>

</html>
```

Esto es lo que muestra el navegador:

Usuario:

Contraseña:

Comentario en el libro de visitas:

! Completa este campo

1.4.4. Ordenación lógica de la petición de datos

Para que los elementos sean lo más agradables posible para la vista de los usuarios debe utilizar las siguientes técnicas:

- Crear los elementos de manera que simulen en la medida de lo posible a un formulario real. Por ejemplo, poner los apellidos después del nombre.
- Los datos más importantes estarán los primeros y estarán destacados.
- Los campos que tengan tamaños adecuados como para ponerlos al lado de otro, deberán colocarse juntos. Siempre y cuando no tengan conflicto con los dos puntos anteriores.
- Cuando usa listas desplegables no deberán tapar otros elementos que sean vitales para que el usuario escoja la opción que debe escoger. Por ejemplo, si la lista desplegable tapa el texto explicativo de qué debe escoger el usuario, no habrán cumplido con la funcionalidad total que debería tener el formulario.
- Los elementos que tengan relación entre sí deberán estar agrupados. Por ejemplo, un formulario mal diseñado podría contener datos personales, después otros datos y más tarde,

podría pedir más datos personales. Si hace algo así podría confundir al usuario y, además, no se vería como un formulario profesional.

- Los botones deberán siempre estar en el lugar más adecuado, que suele ser, al final del formulario. No tiene ningún sentido que el botón de enviar el formulario esté al principio o a mitad del mismo. Otra cosa importante es que el botón enviar siempre esté a la derecha del botón borrar y que se diferencien claramente. Esto es así porque sería muy frustrante para un usuario llenar toda la información y darle por equivocación al botón de borrar.
- Tener en cuenta que la manera en la que introduce los datos en los formularios web es prácticamente igual que la de los formularios clásicos, ya que, los formularios webs están sacados de los formularios clásicos. En cambio, muchos de los elementos que puede encontrar en los formularios web, no existen en los formularios físicos como, por ejemplo, las listas desplegables o los botones de opción. Por último, debe recordar que siempre deberá añadir un botón al final del formulario para enviar los datos al servidor.

En conclusión, los formularios más típicos y que tienen una ordenación de datos lo más correctamente posible.

A continuación, vea un ejemplo:

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="UTF-8">

    <title>Ejemplo de formulario</title>

  </head>

  <body>

    <form action="form.php" method="post" id="formulario">

      <table>

        <tr>

          <td>Nombre: </td><td><input type="text" name="nombre" required/></td>

        </tr>

        <tr>

          <td>Apellidos: </td><td><input type="text" name="apellidos" required/></td>

        </tr>

      </table>

    </form>

  </body>

</html>
```

```
</tr>

<tr>

    <td>DNI: </td><td><input type="text" name="dni" required/></td>

</tr>

<tr>

    <td>Domicilio: </td><td><input type="text" name="domicilio"/></td>

</tr>

<tr>

    <td>Población: </td><td><input type="text" name="poblacion"/></td>

</tr>

<tr>

    <td>Código Postal: </td><td><input type="text" name="cp"/></td>

</tr>

<tr>

    <td>Email: </td><td><input type="text" name="email"/></td>

</tr>

<tr>

    <td>Teléfono: </td><td><input type="text" name="telefono"/></td>

</tr>

<tr>

    <td></td><td><input type="submit" value="Enviar"/></td>

</tr>

</table>
```

```
</form>  
  
</body>  
  
</html>
```

Vea lo que muestra el navegador:

Nombre:

Apellidos:

DNI:

Domicilio:

Población:

Código Postal:

Email:

Teléfono:

1.4.5. Información correcta al usuario

Cuando los usuarios rellenan los campos del formulario está utilizando una herramienta muy poderosa de interactividad con el código.

Sin embargo, si el usuario no introduce los datos de manera correcta, la funcionalidad de formulario desaparece completamente.

Es por ello que, deberá, además, de ordenar los datos de manera correcta, hacer que el usuario no se confunda y que el usuario rellene los campos correctamente. Para ello, debe partir siempre de la idea de que el usuario no sabe nada sobre informática y, por lo tanto, no sabe lo que debe llenar en cada campo.

Siempre que cree un formulario, debe ser conscientes de que va a programarlo con la idea de que ese formulario puede rellenarlo, de entre todos los usuarios, el que me conocimientos tenga y el que sea de edad avanzada. De esta manera, asegurare de adecuar los campos de manera que cualquier persona sea capaz de entender lo que tiene que llenar y cómo.

Por lo tanto, cuando tenga que identificar un campo, debe dejar claro lo que pretende que rellene el usuario final.

Por ejemplo, en caso de que un formulario de registro en una tienda online, donde el usuario deberá llenar datos como dirección, método de pago y datos personales, es importante que se asocien de manera adecuada cada uno de los campos a los textos relacionados con su contenido.

Otro ejemplo es el de los botones de reestablecer el contenido del formulario donde si llama a este botón borrar todo el formulario, el usuario sabrá para lo que sirve el botón. En cambio, si llama al botón reestablecer, puede que el usuario no entienda lo que significa y, por lo tanto, para lo que sirve ese botón. Debido a la incomprendición del usuario hacia el botón, podría suceder que el usuario pulse ese botón justo después de haber llenado todos los datos del formulario y puede suponer la frustración del usuario al ver que todos sus datos se han borrado.

Vea a continuación un ejemplo de petición de datos utilizando, para ello, la claridad en los textos:

```
<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>Ejemplo de formulario</title>

    </head>

    <body>

        <form action="form.php" method="post" id="formulario">

            <table>

                <tr>

                    <td><label for="nombre">Nombre: </label></td><td><input type="text" name="nombre" id="nombre" required/></td>

                </tr>

                <tr>

                    <td><label for="apellidos">Apellidos: </label></td><td><input type="text" name="apellidos" id="apellidos" required/></td>

                </tr>

                <tr>
```

```
<td><label for="dni">DNI: </label></td><td><input type="text" name="dni" id="dni" required/></td>

</tr>

<tr>

    <td><label for="domicilio">Domicilio: </label> </td><td><input type="text" name="domicilio" id="domicilio" /></td>

</tr>

<tr>

    <td><label for="poblacion">Población: </label> </td><td><input type="text" name="poblacion" id="poblacion" /></td>

</tr>

<tr>

    <td><label for="cp">Código Postal: </label> </td><td><input type="text" name="cp" id="cp" /></td>

</tr>

<tr>

    <td><label for="email">Email: </label></td><td><input type="text" name="email" id="email" /></td>

</tr>

<tr>

    <td><label for="telefono">Teléfono: </label></td><td><input type="text" name="telefono" id="telefono"/></td>

</tr>

<tr>

    <td></td><td><input type="submit" value="Enviar"/></td>

</tr>
```

```
</table>  
</form>  
</body>  
</html>
```

Vea lo que muestra el navegador:

Nombre:	<input type="text"/>
Apellidos:	<input type="text"/>
DNI:	<input type="text"/>
Domicilio:	<input type="text"/>
Población:	<input type="text"/>
Código Postal:	<input type="text"/>
Email:	<input type="text"/>
Teléfono:	<input type="text"/>
<input type="button" value="Enviar"/>	

1.4.6. Utilización páginas de error y confirmación

Hoy por hoy, los usuarios están acostumbrados a navegar por Internet a alta velocidad, por lo tanto, nadie se detiene a consultar la mayoría de acciones que hace.

A veces, esto resulta un problema para los formularios, puesto que, los usuarios pueden pulsar el botón Enviar sin realmente haber terminado de llenar el formulario y arrepintiéndose de ello.

Cuando sucede esto en un formulario normal, el usuario tiene que volver a reestablecer el formulario, ya sea volviendo atrás en el navegador o cargando el formulario de nuevo teniendo que completar otra vez toda la información.

Existen múltiples herramientas que evitarán que esto ocurra. Por ejemplo, crear una ventana emergente que confirme el envío del formulario o una página que haga de puente entre el formulario y la página final.

Vea cómo puede hacerlo:

- Programando un *script* en el lado del cliente para que cuando el usuario pulse el botón de enviar, se le pregunte si quiere enviarlo o no.
- Programando en el lado del servidor una página que recoja los datos que el usuario ha enviado y, a continuación, se le pediría una confirmación sobre si realmente quiere enviarlo.

En caso de que sea afirmativo, los datos que el usuario ha introducido se enviarán al servidor, en caso contrario, se enviarán al formulario para que el usuario pueda proceder a finalizar el formulario.

Sin embargo, el uso de páginas en la mayoría de los formularios es contraproducente, puesto que, entorpecerían la navegación y cansarán al usuario. Por ejemplo, si cada vez que fuese a hacer una búsqueda en Google, apareciera una ventana de confirmación sería muy engorroso para otros y haría que tardara mucho más tiempo en buscar cualquier cosa.

Si el formulario envía información incorrecta, a veces, el navegador lanzará un mensaje de error. Podrá evitar esto si utiliza una correcta programación de los elementos, por ejemplo, si utiliza el atributo required para que no se envíe el formulario en caso de que haya campos con datos incorrectos en él.

Además, el navegador puede hacer una llamada a una página que no existe en ese momento en el servidor. Es por esto, que se debe programar en el mismo servidor dejando las páginas de error por defecto o enlazando a una página de error por defecto que se trará al usuario cuando ocurra un error al intentar cargar una página en el propio servidor.

Vea un ejemplo de error de servidor:



No se puede acceder a este sitio web

Comprueba si hay un error de escritura en werwerwerwerwer.com.

Si está escrito correctamente, [prueba a ejecutar el diagnóstico de red de Windows.](#)

DNS_PROBE_FINISHED_NXDOMAIN

[Volver a cargar](#)

Para programar una página de error, lo hará de la misma manera en la que programa las páginas normales. Por otra parte, deberá enlazarlas a los errores que puedan encontrarse los usuarios en la

web. Esto se configurará en el servidor en el fichero .htcaccess en la línea ErrorDocument 404. Si no configura esto, simplemente aparecerá la página de error por defecto.

Vea a continuación un ejemplo de JavaScript que puede añadir a su web para pedir confirmación al usuario cuando envía un formulario:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Ejemplo de formulario</title>

<script language="JavaScript">

function confirmacion(){

    if (confirm ('Pulse aceptar para enviar el formulario'))

    {

        document.formulario.submit()

    }

}

</script>

</head>

<body>

<form action="form.php" method="post" name="formulario" id="formulario">

Nombre:

<input type="text" name="nombre" value="" />

<br/>

<input type="button" onclick="confirmacion()" value="Enviar" />

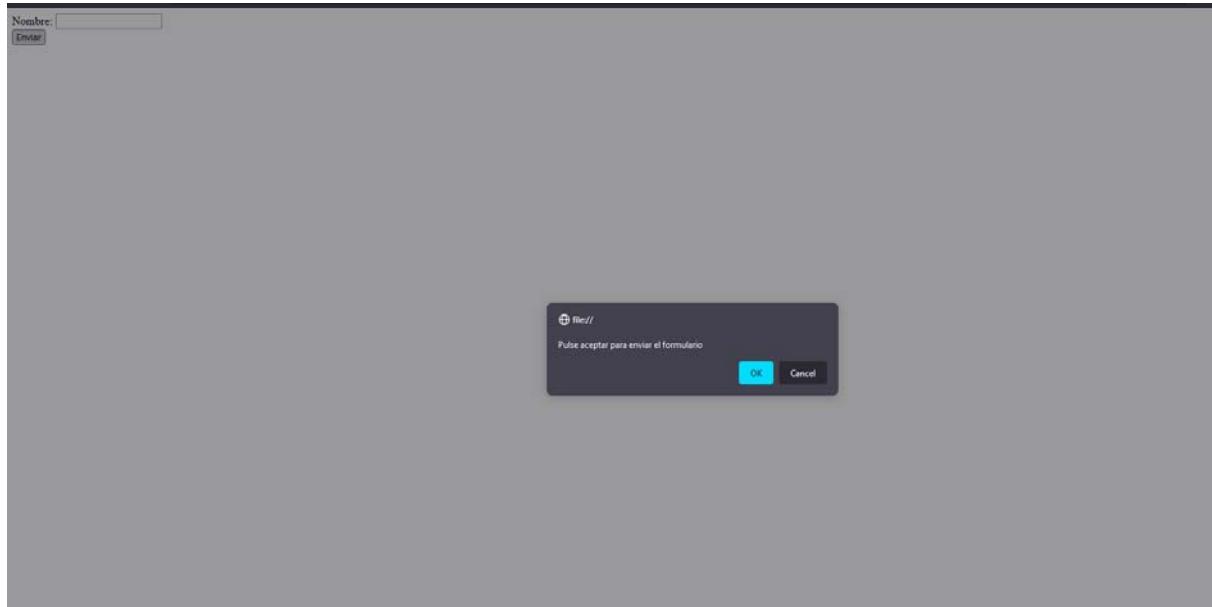

```

</form>

</body>

</html>

Vea lo que muestra el navegador:



Conzepto Comuni...

2. PLANTILLAS EN LA CONSTRUCCIÓN DE PÁGINAS WEB

En este tema, va a aprender lo que es una plantilla y cómo utilizarla para, de ese modo, facilitar el trabajo de desarrollo web. Para ello, comenzare aprendiendo sobre las funciones y características de las plantillas, los elementos que tiene una plantilla, cuál es su estructura y cómo se organiza, también, aprenderá a crear zonas modificables de manera específica y partes fijas.

Además, aprenderá a crear campos que sean editables o no editables dependiendo lo que convenga en cada circunstancia.

Por último, aprenderá a aplicar estas plantillas a su página web e, incluso, a buscar plantillas predeterminadas y adaptarlas a su web para que sean funcionales.

2.1. Funciones y características

Las plantillas son herramientas que permiten agilizar el desarrollo de una web, puesto que, son una copia de determinados elementos puestos de manera coherente y ordenada.

Por ejemplo, en la vida cotidiana utiliza plantillas sin dar cuenta como, por ejemplo, al crear un CV, al crear invitaciones a un evento o escribir una carta de presentación.

En la antigüedad hubo personas que se dieron cuenta de que, en algunas circunstancias relacionadas con la construcción o el diseño, tener un modelo en el que fijarse hacía que el trabajo artesanal fuera más fácil y eficaz.

Es por ello que con el pasar de los años, esos modelos se fueron perfeccionando y utilizando hasta que se popularizó la informática, en donde se aplicaron las faas plantillas de programación.

En el origen de la programación, había una clara diferenciación entre los datos que manejaba un programa y cómo se manipulaban esos datos. Se puede decir que los datos eran de tipos muy simples y que los algoritmos se diseñaban para desarrollar funciones capaces de trabajar con los mismos.

Al trabajar de esta manera, los programadores se dieron cuenta de que la mayoría del código que se utilizaba para programar tareas específicas, se repetía muy a menudo en los diferentes programas.

De hecho, algunos programas que realizaban ciertas funciones específicas eran muy parecidos entre sí, por este motivo era más rápido utilizar el código ya escrito que volver a escribirlo de nuevo. Sin embargo, la reutilización de código presenta dos grandes problemas:

- Puede pasar que se copien cosas que no se deberían utilizar, de esta manera crearía código redundante e innecesario.
- Quien reutilice un código, deberá tener un conocimiento del mismo. En caso de que esto no sea así, debería estudiar el funcionamiento del mismo antes de reutilizarlo, por ello, la reutilización del código pierde utilidad si el programador no es el propio creador del código.

Para darle solución a estos problemas se optó por poner el código que se iba a reutilizar en unas plantillas, que son fragmentos de código donde se especifica la utilidad del mismo, y que incluso tienen una función específica global para la función especificada.

Además, en las plantillas se podían encontrar unas variables indeterminadas que se podían modificar al gusto del programador. También, había cantidad de comentarios que indicaban a los programadores que accedían por primera vez a la plantilla, la utilidad que tenía cada uno de los elementos de la misma.

Posteriormente, apareció la programación orientada a objetos. Este tipo de programación aportó nuevas opciones a la hora de programar, además de ofrecer muchas más facilidades a la hora de hacer plantillas, puesto que, al poderse crear tipos más complejos, se podían crear plantillas de los mis de manera más sencilla y eficaz.

En este tipo de programación más avanzada, se pudieron crear plantillas para definir clases o funciones. Haciendo posible que, al crear funciones o clases genéricas, utilizar plantillas pasase de ser algo recomendado a ser algo habitual en el mundo de la programación.

Últimamente, con la popularización de Internet, e han creado diversas plataformas que ofrecen plantillas para todos los lenguajes de programación. De esta manera, la vida de los programadores se ha hecho mucho más sencilla, siempre que se sepa lo que necesita y dónde encontrarlo.

Vea un ejemplo donde puede descargar plantillas en la web de Oracle:

The screenshot shows a web page from the Oracle Help Center. The top navigation bar includes the Oracle logo, 'Help Center', and a 'Sign in' button. Below the header, the URL 'Home / EPM Cloud Planning Translated Books' is visible. The main content area has a red header 'Administración de Planning'. To the left is a sidebar with a tree view of topics. The central content area is titled 'Descarga de las plantillas de aplicación' and lists two related topics with links. At the bottom, there are navigation links for 'Previous Page' and 'Next Page', and a page number 'Page 641 of 688'.

El extenso uso que tiene Internet y el hecho de que los lenguajes de programación estén en una evolución constante, hace que el uso de plantillas para la web sea algo casi obligado para el desarrollo web y la programación. Ya no solo para ahorrar tiempo, sino para facilitar el trabajo.

Esto se puede utilizar para todos los lenguajes de programación más utilizados en la actualidad:

- HTML
- SQL
- PHP
- Python
- Java
- C
- C++
- JavaScript
- XML
- Ruby
- AJAX
- Perl

Por último, debe saber que la creación de plantillas se ha convertido en una oportunidad de negocio y que mucha gente se dedica de manera exclusiva a crear plantillas para venderlas en páginas web. Esto ocurre con las plantillas más avanzadas para utilizar en los diversos CMS como Prestashop, Magento o WordPress.

2.1.1. Descripción de una plantilla web

Las plantillas web no tienen ningún tipo de diferencia con las plantillas de programación normales, tan solo se puede decir que se diferenciarían en el tipo de lenguaje de programación que utilizan.

Se puede decir que las plantillas para JavaScript, HTML o PHP son fragmentos de código que tienen diferentes utilidades, las cuales contienen variables estándar y comentarios para que los programadores sepan cómo utilizar las variables en su web.

Una plantilla web sale de un programador que crea y personaliza una web teniendo la intención de crear una plantilla y, por lo tanto, guarda el sitio web como plantilla. Se deberán almacenar las plantillas de manera que sean localizables fácilmente, por ejemplo, podría guardarlas en un nivel superior al que se encuentra el sitio web, de manera que esté disponible a la hora de utilizarla.

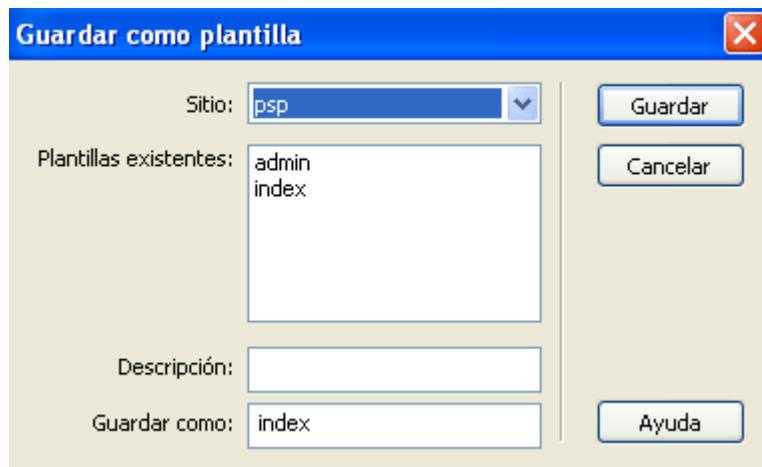
Se recomienda guardar las plantillas web con una extensión diferente a la del archivo original, de este modo, no se confundirá a la hora de saber si un documento es una plantilla u otro tipo de documento web. En caso de que cree y almacene plantillas con Dreamweaver, las plantillas tendrán la extensión dwt, sin embargo, podrá elegir cualquier extensión que sea útil para ello y que no cause confusión con los archivos originales.



TOME NOTA

Debe saber que existen aplicaciones para la construcción, edición y diseño de páginas web. Por ejemplo: Dreamweaver, Blue Griffon y Microsoft Expression Web. Estos programas, facilitarán el trabajo, puesto que, de manera visual, podrá ver el proceso de creación del código.

Vea cómo guardar una plantilla en Dreamweaver:



Siempre deberá almacenar las plantillas en sitios web para que pueda acceder a sus elementos desde cualquier sitio.

Vea a continuación el proceso que deberá seguir para crear una plantilla web:

- Primero, deberá identificar el elemento que quiere crear en la plantilla.
- A continuación, creare el código del elemento.
- Después, deberá crear variables lo más genéricas posible.
- Deberá añadir comentarios para indicar la parte de código que forma parte de la plantilla y qué elementos deberá insertar el programador.
- Por último, deberá guardar la plantilla con una extensión diferente a la del archivo original y, además, deberá guardarla por encima de la raíz del sitio web.

Debe destacar que no es lo mismo una plantilla para un elemento general de HTML como, por ejemplo, un botón que tenga una función muy sencilla, que una plantilla que se encarga de modificar todo el diseño de un sitio web. Puede utilizar este tipo de plantillas sobre páginas que ya se hayan empezado a diseñar, ya que, estas plantillas no sobrescriben el código original, sino que, simplemente, modifica el código que ya hay escrito para mejorar, de ese modo, la web. En caso de que tenga una web creada utilizando algún CMS como WordPress podría escoger como plantilla básica la siguiente:

PopularFX

Lightweight theme to make beautiful websites with Pagelayer.

Parent Page

About Sub-page

Worth A Thousand Words

Posted on October 17, 2008 by Theme Admin



Boat

Boat.

Posted in Uncategorized Tagged boat, lake

Sin embargo, puede modificar esta plantilla utilizando una plantilla y un tema para cambiar este aspecto por el siguiente:

PopularFX

Home Blog About Contact

HELLO. NAMASTE | HOLA

Unleash Your Potential !



Vea un ejemplo de una plantilla para HTML para crear un login de una página web:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title><!-- Introducir título de la página--></title>

</head>

<body>

<form action="--Web de destino--" method="post">

    Usuario:

        <input type="text" size="30" maxlength="30" name="usuario"
value="Usuario"/>

        <br/>

    Contraseña:

        <input type="password" size="10" maxlength="10" name="pass"/>

        <br/>

        <input type="submit" value="Entrar"/>

    </form>

</body>

</html>
```

2.1.2. Elementos de una plantilla web

Antes de diseñar una plantilla web debe preguntarse lo siguiente:

- Qué lenguaje va a utilizar para programar la plantilla.
- Para qué va a utilizar la plantilla.
- En cuántas ocasiones se va a utilizar la plantilla.
- Si necesitare una plantilla para el elemento específico o no.
- Si ganare tiempo al diseñar la plantilla o no.

	<p><i>Es muy importante saber si va a ahorrar tiempo al crear una plantilla y si el elemento o elementos que hay en la plantilla van a ser reutilizados, ya que, si la plantilla no va a ahorrar tiempo, carece de sentido.</i></p>
TOME NOTA	

Una vez haya respondido a esas preguntas, será el momento de realizar un boceto del código de lo que quiere que haga su plantilla. Después, creare los elementos que sean necesarios para crear una plantilla funcional.

Para ello, deberá dividir los elementos en dos tipos:

- Campos no modificables.
- Campos modificables.

Más adelante, verá de manera más específica estos dos tipos de elementos. Sin embargo, es importante que sepa que existe esta división de los elementos. Por otra parte, en las plantillas HTML, deberá haber un conjunto de elementos que serán fijos y se utilizarán a la hora de reutilizar el código. Véalo a continuación:

- Las etiquetas de apertura y cierre del cuerpo del documento, que es donde irá todo el contenido de la plantilla.
- Las etiquetas de apertura y cierre de la cabecera, donde irá también la información de la plantilla.
- Las etiquetas de apertura y cierre propias del código HTML.

De todas maneras, en todos los lenguajes de programación es importante que indique de manera correcta con el uso de comentarios, la funcionalidad de cada elemento de la plantilla.

Añadir los comentarios en el código de la siguiente manera:

- En HTML mediante <!--Comentario-->
- En JavaScript y PHP, mediante /*Comentario*/

2.1.3. Estructura y organización de los elementos de las plantillas

Los elementos de los que ha hablado en el punto anterior son:

- Campos modificables.
- Campos no modificables.
- Código que tiene toda la plantilla.

Deberá de estructurarlos de manera correcta. Es decir, deberá crear un código de la manera más correcta y profesional posible, tabulado correctamente y con sus cierres y aperturas indicados de manera clara y precisa.

Cuando haya finalizado de hacer estos elementos, deberá seguir siempre la misma manera de crear los comentarios, de manera que quede correcta y sea útil para otro programador, para que, a la hora de consultar estos comentarios, podrá encontrar lo que busca fácilmente.



TOME NOTA

Debe nombrar todos los elementos modificables de una plantilla de manera que sepa exactamente qué función cumplen. Por ejemplo, si tiene un elemento es el encargado de enviar la información del formulario, le insertare el valor de paginaDestino.php.

Por ejemplo, a continuación, verá una plantilla que puede utilizar si quiere añadir un buscador a una página web:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title><!-- Ejemplo buscador--></title>

</head>

<body>

<form action="<!--Web de destino donde se enviarán los datos cuando se pulse submit-->" method="post">

    Introducir criterio de búsqueda:

    <br/>

    <input type="text" name="buscador"/>

    <br/><br/>

    <input type="submit" value="Buscar"/>

</form>

</body>

</html>
```

En este vídeo, de cinco minutos de duración, podrá saber para qué utilizar plantillas si sabe crear páginas web desde cero:

RECURSO MULTIMEDIA



2.1.4. Especificar las zonas modificables de una plantilla y las partes fijas

En cualquier plantilla web, podrá encontrar partes que deberán ser modificadas y partes que no. De hecho, podrá encontrar plantillas las cuales no deban ser modificadas nunca.

Las razones por las que puede encontrar plantillas que no se modifiquen son las siguientes:

- Se ha creado un efecto estético con JavaScript y, por lo tanto, el código es fijo.
- Desconoce los efectos del código porque lo ha creado otro programador. Por lo tanto, no se recomienda tocar el código.

En estas dos circunstancias, lo más recomendable es dejar comentario para que si en un futuro otra persona u otros mis revisa el código, no haya ninguna confusión.

	<p>Normalmente se recomienda no utilizar código de terceros, sin embargo, esto carece de sentido, puesto que, existen sitios en Internet que ofrecen plantillas de manera gratuita. Sin embargo, antes de utilizarlas deberá aprender cómo interactúa con el funcionamiento de la web.</p>
TOME NOTA	

Siempre que una zona de la plantilla sea modificable, se debe indicar por el programador que ha creado esa plantilla, y el programador final deberá modificar estos trozos de código en su beneficio.

Cuando el programador final termine de modificar el código deberá proceder a eliminar esos comentarios para que no den lugar a equivocaciones futuras.

Vea un ejemplo:

```
<form action="paginaDestino.php<!--Web de destino donde se enviarán los datos cuando se pulse submit-->" method="post">
```

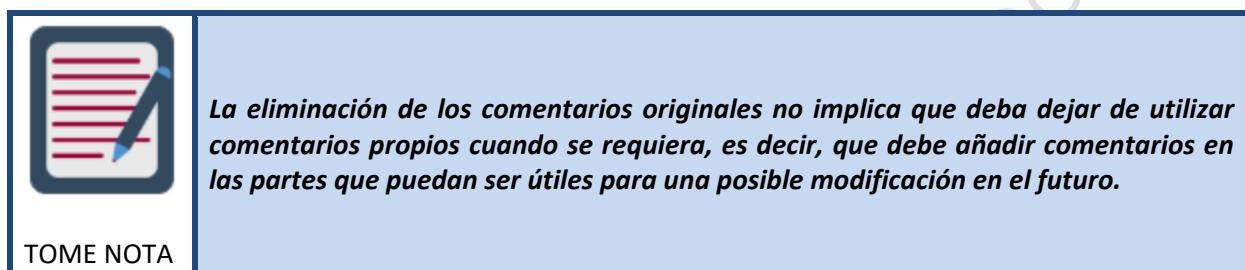
En caso de que utilice esta plantilla para su web y la modifique para que funcione en su página, deberá hacer lo siguiente:

- Cambiar paginaDestino.php por la página real donde se van a enviar los datos.
- Eliminar el comentario <!--Web de destino donde se enviarán los datos cuando se pulse submit-->.

Por lo tanto, obtendría el siguiente resultado:

```
<form action="paginaDestinoReal.php" method="post">
```

Como puede ver, esta línea de código es más corta y más fácil de visualizar que la anterior.



A continuación, verá un ejemplo de una plantilla que utiliza JavaScript con zonas modificables y no modificables:

```
<!-- Comienzo de la plantilla-->

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title><!-- Ejemplo plantilla JavaScript--></title>

<script language="JavaScript">

function abrirVentana (pag){

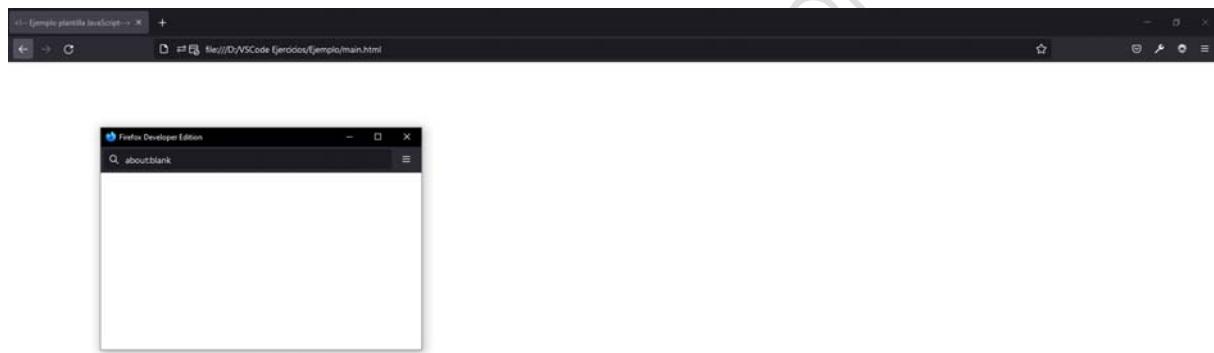
    var opciones="toolbar=no, location=no,directories=no, status=no, menubar=no,
scrollbars=no, resizable=yes, width=508, height=365, tops=85, left=140"

    window.open(pag,"",opciones);

}
```

```
    }  
  
</script>  
  
</head>  
  
<body onload="abrirVentana('ventanaEmergente.html<!-- Este elemento se debe sustituir por la  
url que quiera que se muestre en la ventana emergente que se abrirá cuando cargue la página--  
'>">  
  
</body>  
  
</html>  
  
<!--Fin de la plantilla-->
```

Vea lo que muestra el navegador:



Como puede comprobar, no debe hacer ningún cambio en la mayoría del código, tan solo, deberá cambiar la página que debe cargarse como emergente y el comentario añadido.

En cambio, un programador que conozca en profundidad JavaScript, sí que podría modificar el código en su totalidad y manejarlo a su gusto.

De todos modos, es importante que se identifique de manera sencilla qué elementos son modificables y cuáles no.

2.1.5. Utilización de plantillas

Las plantillas pueden llegar a ser una herramienta muy poderosa si sabe cómo utilizarlas de manera correcta, puesto que, además de permitir ahorrar mucho tiempo y maximizar su productividad, también permiten crear un diseño que sea útil para múltiples trabajos.

Sin embargo, no siempre se recomienda utilizar una plantilla y, por lo tanto, debe saber cuándo es necesario utilizarla y cuándo no.

Antes de construir o utilizar una plantilla, deberá tener en cuenta los siguientes factores:

- Si es un trabajo lo suficientemente complejo como para que la inversión de tiempo en producir una plantilla se vea recompensado.
- Si va a ganar tiempo utilizando la plantilla o si el usarla va suponer una pérdida de tiempo en comparación con programar esos elementos.
- Si utilizar una plantilla sería la solución más eficiente y correcta ante el problema que debe solucionar o sería más conveniente utilizar código propio.
- Si utiliza una plantilla de terceros que hay que pagar, si podrá asumir el coste de la misma.



TOME NOTA

Este grupo de factores son fáciles de responder si tiene la suficiente experiencia como programadores. En cambio, cuando se tiene mucha experiencia se recomienda responder a estas preguntas tranquilamente para poder llegar a discernir en si necesita utilizar o no la plantilla.

El mayor problema con el que se suele encontrar los programadores es el tener que enfrentarse a las páginas totalmente en blanco.

Cuando un programador trabaja durante un tiempo programando un programa debe trabajar con cientos o miles de líneas de código. Sin embargo, cuando termina con ese programa concreto y tiene que comenzar con otro proyecto, supone un reto el tener que enfrentarse, de nuevo, a un documento en blanco.

Por ello, las plantillas son una gran ayuda a cualquier tipo de programador.

Como ejemplo, se puede mencionar que, si un programador trabaja con documento HTML, una plantilla que aporte la estructura básica del HTML podría ser de gran ayuda y, de esa manera, no tener que enfrentarse ante el documento totalmente en blanco.

Además, también ayudaría en caso de tener que crear una tabla, crear la estructura básica para los párrafos, crear listas, enlaces o imágenes, crear formulario o clases propias en CSS, etc. Todo ello se puede crear desde una plantilla como verá más adelante.

De hecho, muchos de estos elementos pueden terminar no siendo útiles para el diseño de la página web. Sin embargo, están pensados para servir de inspiración al programador para, al me, saber dónde ubicar cada uno de los elementos que finalmente cree en esa página web.

A continuación, va a ver el ejemplo de una plantilla con HTML:

```
<!-- Comienzo de la plantilla-->

<!DOCTYPE html>

<html>

  <head>

    <meta charset="UTF-8">

    <title><!-- Ejemplo plantilla HTML--></title>

  </head>

  <body>

    <h1>Esta es la cabecera principal</h1>

    <p>Este párrafo será la introducción de la página. En caso de ser muy larga, se podría dividir en subcabeceras</p>

    <h2>Esta es una subcabecera</h2>

    <p>Contenido de esta subcabecera</p>

    <h2>Esta es otra subcabecera</h2>

    <p>Y este será el otro contenido de la subcabecera</p>

  </body>

</html>

<!--Fin de la plantilla-->
```

Vea lo que muestra el navegador:

Esta es la cabecera principal

Este párrafo será la introducción de la página. En caso de ser muy larga, se podría dividir en subcabeceras

Esta es una subcabecera

Contenido de esta subcabecera

Esta es otra subcabecera

Y este será el otro contenido de la subcabecera

Cuando crea una plantilla, puede que ni utilice esa estructura de párrafos, sin embargo, aunque no utilice esta disposición de los elementos o este código, será capaz de ubicar de un vistazo rápido dónde colocar los elementos principales.

Por otra parte, la ventaja de utilizar este tipo de plantillas es que no necesitaré añadir muchos comentarios para indicar qué es cada elemento, puesto que, el propio código lo indica.

Por ejemplo, cuando programa con HTML y quiere crear una estructura de párrafos, existen unas etiquetas específicas para crearlos.

Todos los programadores de HTML saben qué significa esta etiqueta, por lo tanto, carece de sentido crear plantillas para este tipo de elementos.

Sin embargo, cuando el programador es novel en este lenguaje, sería útil tener una plantilla de este tipo para ahorrarle al programador tener que consultar cómo crear párrafos en un documento HTML, por lo que, le ahorraría tiempo.

A continuación, va a ver un ejemplo de una estructura de párrafos en HTML:

```
<!-- Comienzo de la plantilla-->

<!DOCTYPE html>

<html>

  <head>

    <meta charset="UTF-8">

    <title><!-- Ejemplo plantilla HTML estructura de párrafos--></title>

  </head>
```

```
<body>

    <!--Estructura de párrafos-->

    <h1>Cabecera principal</h1>

    <h2>cabecera nivel 2</h2>

    <h3>Cabecera nivel 3</h3>

    <h4>Cabecera nivel 4</h4>

    <h5>Cabecera nivel 5</h5>

    <h6>Cabecera nivel 6</h6>

    <p>Este es el contenido de los párrafos</p>

    <!--Ejemplo de utilidad de los párrafos en un texto-->

    <h1>Título del libro</h1>

    <h2>Capítulo 1</h2>

    <p>Esto es un texto normal, <i>en cursiva,</i><b>en negrita</b> y <u>subrayado</u>. </p>

</body>

</html>

<!--Fin de la plantilla-->
```

El navegador muestra lo siguiente:

Cabecera principal

cabecera nivel 2

Cabecera nivel 3

Cabecera nivel 4

Cabecera nivel 5

Cabecera nivel 6

Este es el contenido de los párrafos

Título del libro

Capítulo 1

Esto es un texto normal, *en cursiva*, **en negrita** y subrayado.

En ocasiones, cuando crea una página web debe ordenar los diferentes elementos que aparecen en ella. Para ello, utilice las listas que proporciona el lenguaje HTML.

En cambio, debe puntualizar que estos elementos se utilizan poco, por lo que, hasta los programadores más profesionales pueden tener dificultades para recordar la estructura de este tipo de elementos.

Es por ello que puede utilizar una plantilla para aplicar estos elementos. Lo siguiente que verá es una plantilla donde se aplican estos elementos de listas a un documento HTML:

```
<!-- Comienzo de la plantilla-->

<!DOCTYPE html>

<html>

  <head>

    <meta charset="UTF-8">

    <title>Ejemplo plantilla HTML listas</title>

  </head>
```

```
<body>

    <h1>Listas en HTML</h1>

    <!--Listas no ordenadas-->

    <h2>Listas no ordenadas</h2>

    <ul>

        <li>Elemento 1</li>
        <li>Elemento 2</li>
        <li>Elemento 3</li>

    </ul>

    <!--Listas ordenadas-->

    <h2>Listas ordenadas</h2>

    <ol>

        <li>Elemento 1</li>
        <li>Elemento 2</li>
        <li>Elemento 3</li>

    </ol>

</body>

</html>

<!--Fin de la plantilla-->
```

Vea lo que muestra el navegador:

Listas en HTML

Listas no ordenadas

- Elemento 1
- Elemento 2
- Elemento 3

Listas ordenadas

1. Elemento 1
2. Elemento 2
3. Elemento 3

Cuando se habla de programación uno de los elementos más importantes son los enlaces a otras páginas web. Es debido a los enlaces por lo que los usuarios pueden navegar de un sitio a otro y, por lo tanto, es lo que le da sentido y funcionalidad a Internet.

Incluir enlaces en un documento HTML es muy sencillo, sin embargo, la mayoría de herramientas no incluyen estos enlaces por defecto en una web. Además, existen cierto tipo de enlaces que no se suelen utilizar, por ejemplo, el enlace que le pide al usuario final un correo.

Una plantilla que aporta los enlaces necesarios para una web sería la siguiente:

```
<!-- Comienzo de la plantilla-->

<!DOCTYPE html>

<html>

  <head>
    <meta charset="UTF-8">
    <title>Ejemplo plantilla HTML de urls</title>
  </head>

  <body>
    <h1 id="top">Estructura de enlaces en HTML</h1>
    <h2>Enlace normal con texto</h2>
```

```
<p>

<!--Estructura para crear un enlace-->

<a href="urlcontexto.com">Enlace url</a>

</p>

<h2>Enlace para enviar correo</h2>

<p>

<!--Estructura de enlace para enviar correo-->

<a href="mailto:ejemplo@correo.com">Enviar correo</a>

</p>

<h2>Enlace para ir a la parte superior de la página</h2>

<p>

<!--Creación de enlace de ancla para ir hacia arriba-->

<a href="#top">Ir a la parte superior</a>

</p>

</body>

</html>

<!--Fin de la plantilla-->
```

Estructura de enlaces en HTML

Enlace normal con texto

[Enlace url](#)

Enlace para enviar correo

[Enviar correo](#)

Enlace para ir a la parte superior de la página

[Ir a la parte superior](#)

Por otra parte, otro de los elementos importantes en las páginas web son las imágenes que son las que hacen más amena la página web para los usuarios.

Añadir imágenes a una página web también es algo muy sencillo de hacer. De hecho, hoy por hoy todas las páginas web incluyen imágenes y cada día es un elemento más representativo de una web. Sin embargo, el código para incluir estas imágenes puede que no lo tenga siempre tan claro.

Por lo tanto, lo que haremos para solucionar este problema es crear una plantilla con la estructura siguiente:

```
<!-- Comienzo de la plantilla-->

<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>Ejemplo plantilla HTML para imágenes</title>

    </head>

    <body>

        <h1>

            <!--Enlace a una imagen normal-->

            <!--El valor de alt trará un texto alternativo en caso de no trarse la imagen-->

        </h1>

    </body>

</html>
```

<!--Los valores height y width representarán la altura y anchura de la imagen-->

```

```

```
</h1>
```

<!--Enlace a una imagen encuadrada-->

```
<figure>
```

```

```

```
<p>
```

<!--El contenido de figcaption se trará en el pie de la imagen-->

```
</p>
```

```
</figure>
```

```
</body>
```

```
</html>
```

<!--Fin de la plantilla-->

Vea lo que muestra el navegador:



Puede utilizar muchas opciones para organizar y distribuir los elementos en una página creada con HTML.

Por ejemplo, en caso de que quiera poner valores de una estadística o números listados puede utilizar una tabla.

Para facilitar la tarea de crear tablas en HTML, puede crear una plantilla que aporte esas tablas a su web.

Vea el ejemplo:

```
<!-- Comienzo de la plantilla-->

<!DOCTYPE html>

<html>
  <head>
    <meta charset="UTF-8">
    <title>Ejemplo plantilla HTML para tablas</title>
  </head>
  <body>
```

```
<table>

    <thead>

        <tr><th></th><th scope="col">Cabecera columna 1</th><th scope="col">Cabecera
columna 2</th></tr>

    </thead>

    <tbody>

        <tr><th scope="row">Cabecera línea 1</th><td>Casilla línea 1 columna 1</td><td>Casilla
línea 1 columna 2</td></tr>

        <tr><th scope="row">Cabecera línea 2</th><td>Casilla línea 2 columna 1</td><td>Casilla
línea 2 columna 1</td></tr>

    </tbody>

    <tfoot>

        <tr><td></td><td colspan="2">Pie de tabla</td></tr>

    </tfoot>

    </table>

</body>

</html>
```

<!--Fin de la plantilla-->

Vea lo que muestra el navegador:

Cabecera columna 1	Cabecera columna 2
Cabecera línea 1	Casilla línea 1 columna 1
Cabecera línea 2	Casilla línea 2 columna 1
Pie de tabla	

2.2. Campos editables y no editables

Como ya ha visto anteriormente, cuando programa necesita incluir campos editables y no editables por el usuario, pero que le aportan información.

Vea los ejemplos que puede encontrar:

- Texto plano que indica o informa a los usuarios.
- Contenido de tablas de elementos.
- Listas no numeradas y numeradas.
- Imágenes que se muestran en la página.

Que los campos sean editables o no, supone una pequeña diferencia entre los permisos de edición y administración y los permisos de un usuario normal el cual no tiene porqué conocer el código de dicho documento web y que, por lo tanto, para el usuario será algo transparente.

Sin embargo, esto cambia cuando el programador accede al código de un sitio web, puesto que, el poder acceder a dicho código hace que adquiera todos los permisos para hacer y deshacer todo lo que quiera.

En cambio, cuando se plantea una plantilla es importante que, como programadores propios o programadores futuros, sepa qué campos son los que debe editar y cuáles no debe tocar.

Es por ello que se debe diferenciar entre campos editables y no editables.

Como ya se ha mencionado, un programador es un administrador del código que es capaz de acceder al mismo sin ninguna restricción. Entonces, para crear campos no editables para los programadores, simplemente, deberá añadir comentarios indicando qué es lo que se puede o no se puede editar.



TOME NOTA

Debe recordar que los comentarios se crean de diferente manera en los lenguajes HTML, PHP y JavaScript y que, por lo tanto, debe tener cuidado para no equivocar con la sintaxis.

En el momento en que haya terminado de crear una plantilla es cuando deberá añadir los comentarios de arriba hacia abajo, indicando los campos que se pueden editar y los que no.

Además de indicar si es editable o no, en los campos editables deberá indicar qué es lo que debería incluir en ese campo.

Cabe destacar que un código con comentarios en los campos no editables es mucho más correcto y, por lo tanto, superaría u estándares de calidad de un programador experto.

En cambio, no es totalmente obligatorio comentar los campos no editables.

De hecho, un uso exagerado de los comentarios podría volver al código ilegible y difícil de visualizar, es por ello que es importante utilizar los comentarios con moderación.

A continuación, va a ver un ejemplo de código de una plantilla que muestra campos editables y no editables con sus respectivos comentarios. Esta plantilla es un formulario que pide al usuario sus datos personales:

```
<!-- Comienzo de la plantilla-->

<!--Inicio de datos editables. Colocar tras "action" la web de destino del formulario-->

<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title><!-- Ejemplo plantilla formulario--></title>

    </head>

    <body>

        <form action="datosPersonales.php" method="post">

            <!--Fin de datos editables.-->

            <!--Inicio de datos no editables.-->

            <table>

                <tr>

                    <td><label for="nombre">Nombre: </label></td><td><input type="text" name="nombre" id="nombre" required/></td>

                </tr>

                <tr>
```

```
<td><label for="apellidos">Apellidos: </label></td><td><input type="text" name="apellidos" id="apellidos" required/></td>

</tr>

<tr>

    <td><label for="dni">DNI: </label></td><td><input type="text" name="dni" id="dni" required/></td>

</tr>

<tr>

    <td><label for="domicilio">Domicilio: </label></td><td><input type="text" name="domicilio" id="domicilio"/></td>

</tr>

<tr>

    <td><label for="poblacion">Población: </label></td><td><input type="text" name="poblacion" id="poblacion"/></td>

</tr>

<tr>

    <td><label for="cp">Código postal: </label></td><td><input type="text" name="cp" id="cp"/></td>

</tr>

<tr>

    <td><label for="email">Email: </label></td><td><input type="text" name="email" id="email"/></td>

</tr>

<tr>

    <td><label for="telefono">Teléfono: </label></td><td><input type="text" name="telefono" id="telefono"/></td>

</tr>
```

```
<tr>

<td></td><td><input type="submit" value="Enviar"/></td>

</tr>

</table>

</form>

<!--Fin de los datos no editables-->

</body>

</html>

<!--Fin de la plantilla-->
```

Ve lo que muestra el navegador:

Nombre:	<input type="text"/>
Apellidos:	<input type="text"/>
DNI:	<input type="text"/>
Domicilio:	<input type="text"/>
Población:	<input type="text"/>
Código postal:	<input type="text"/>
Email:	<input type="text"/>
Teléfono:	<input type="text"/>
<input type="button" value="Enviar"/>	

2.2.1. Definir y crear los campos susceptibles de cambios de una plantilla

Ahora, que ya sabe lo que son los campos modificables, debe aprender cuándo debe añadirlos a las plantillas. Cuando crea una plantilla deberá, como programadores, tener clara dos cosas:

- La utilidad de la plantilla.
- Los elementos que debe tener la plantilla.

Cuando tenga claros ambos puntos es cuando debe comenzar a programar el código de la plantilla. En cambio, todas las plantillas pueden tener campos modificables y modifiables, por lo tanto, si la plantilla tiene campos modificables deberá seguir el protocolo:

- Codificar el campo modifiable.
- Crear variables genéricas e identificativas que deba llevar el campo modifiable.

- Separar el campo modificable del resto de campos no modificables.
- Añadir un comentario al comienzo del campo modificable para indicar su función y el valor que se debe incluir en el campo.
- Añadir un comentario al final del campo modificable de manera que sea un comentario informativo para el programador que consulte el código en un futuro.

Al finalizar este protocolo, deberá seguir creando la plantilla y, en caso de que sea necesario, volver a repetir el protocolo en caso de que cree otro campo modificable.



TOME NOTA

Puede topar con plantillas que no tenga ningún campo modificable, esto quiere decir que para crear una plantilla no es obligatorio añadir campos modificables. Sin embargo, es importante que se añadan este tipo de campos siempre que lo necesite.

Si el código de una plantilla está creado correctamente, se debería poder identificar de un vistazo rápido los campos modificables, tan solo, al leer los comentarios.

Por último, en caso de utilizar una plantilla con campos modificables, es importante que además de cambiar los valores de estos campos de manera correcta, deberá también eliminar los comentarios que hagan referencia a la plantilla.

2.2.2. Definir y crear los campos no modificables en una plantilla

En este apartado va a ver cuándo debe añadir los campos no modificables a la plantilla y cómo hacer que ese campo no se tenga que modificar.

De la misma manera que cuando se crea una plantilla con campos modificables, cuando crea la plantilla con campos no modificables deberá tener claras dos cosas importantes:

- La utilidad de la plantilla.
- Los elementos que deben incluirse en la plantilla.

Estos campos, como puede observar, en apariencia se ven igual que los campos modificables, y es que estos dos tipos de campos están relacionados entre sí, puesto que, normalmente ambos aparecen en casi todas las plantillas.

Como ya sabe, en una plantilla puede haber tanto campos modificables como campos no modificables. Sin embargo, los campos no modificables son los únicos que deberán estar siempre presentes en todas las plantillas.



TOME NOTA

Recordar que los campos no modificables estarán siempre presentes en las plantillas, cosa que no pasa con los campos modificables.

El protocolo que deberá seguir para crear campos no modificables es el siguiente:

- Codificar el campo no modifiable.
- Identificar, en caso de que estén presentes, los campos modificables para que no tenga ninguna equivocación frente a los no modificables.

Cuando haya terminado de dar estos pasos, deberá seguir creando la plantilla e ir siguiendo estrictamente el protocolo, puesto que, jamás debe confundir los campos modificables con los no modificables.

Lo mismo que ocurre con los campos modificables, si el código de la plantilla está creado correctamente, se debería poder identificar de un vistazo rápido cuáles son los campos no modificables, tan solo, al leer los comentarios.



TOME NOTA

Recordar que los campos no modificables no tienen por qué tener comentarios que indiquen lo que debe hacer el programador con ellos, en cambio, hay veces en los que sí tiene comentarios. Si se encuentra con este caso, deberá eliminar este tipo de comentarios cuando utilice la plantilla.

Por último, deberá eliminar los comentarios de comienzo y final de la plantilla.

2.3. Aplicar plantillas a páginas web

Cuando tiene un sitio web ya creado, aplicar una plantilla puede presentar diferentes niveles de complejidad dependiendo el tipo de plantilla que quiera incluir.

En caso de que tenga un sitio web programado HTML y quiere incluir una plantilla, deberá incluir el código de esa plantilla anidado en su código, y la complejidad que se encuentre estará relacionada con el elemento que quiere añadir y el conocimiento que tenga como programadores.

Vea un ejemplo donde tiene una página con un formulario de login:

```
<!DOCTYPE html>
```

```
<html>

<head>

<meta charset="utf-8" />

<title>Formulario de ingreso</title>

</head>

<body>

<form action="login.php" method="post">

    Usuario:

    <input type="text" size="30" maxlength="30" name="usuario" value="Usuario"/>

    <br/>

    Contraseña:

    <input type="password" size="10" maxlength="10" name="pass"/>

    <br/>

    <input type="submit" value="Entrar"/>

</form>

</body>

</html>
```

Vea lo que muestra el navegador:

Usuario:

Contraseña:

Ponga un ejemplo en el que quiere añadir un mensaje emergente al cargar la página.

Para ello, debe añadir un código en JavaScript que puede que no conozca, o incluso no sepa programar con JavaScript, por lo que, programar esta acción llevaría muchísimo tiempo.

En cambio, si accede a una plantilla JavaScript que ofrezca este elemento para añadir a su página web, la solución es más rápida y eficiente, en tiempo y complejidad.

Por lo tanto, en caso de que haya creado una plantilla como esta anteriormente, o la haya tomado de terceros, puede insertarla en su sitio web:

```
<!-- Comienzo de la plantilla-->

<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>Ejemplo plantilla JavaScript</title>

        <script language="Javascript">

            function mensaje(){

                <!--Comienzo de la zona editable, cambiar mensaje por el texto que quiera trar-->

                window.alert("Mensaje")

                <!--Fin de la zona editable-->

            }

        </script>

    </head>

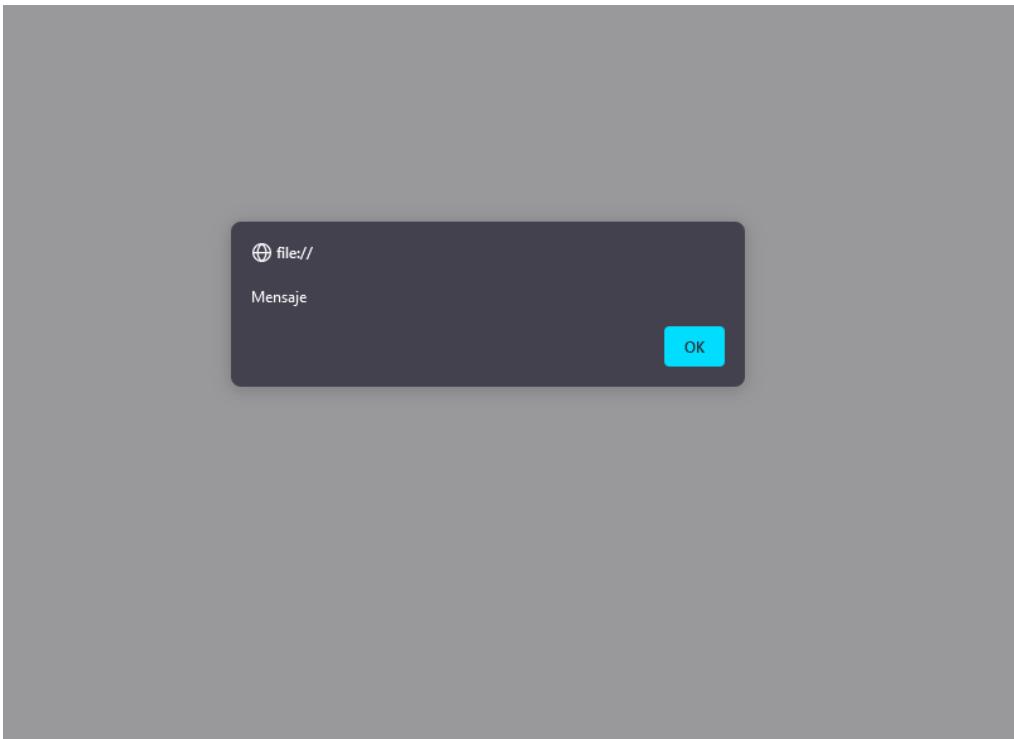
    <body onload="mensaje()">

    </body>

</html>

<!--Fin de la plantilla-->
```

Esto es lo que aparecerá en el navegador:



Esta plantilla aporta la función JavaScript que necesita añadir a su documento HTML.

Por lo tanto, ya solo quedaría incluir el código de esta plantilla a su documento web original donde tendrá que cambiar los elementos modificables, que en este ejemplo solo es uno.

Por ejemplo, si el mensaje que quiere que aparezca en su web es Bienvenidos a mi web, el código quedaría de la siguiente manera:

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8" />
    <title>Ejemplo formulario de ingreso con mensaje emergente</title>
    <script language="JavaScript">

      function mensaje ()
      {
        window.alert("Bienvenidos a mi web")
      }
    </script>
  </head>
  <body>
    <form>
      <input type="text" name="username" placeholder="Nombre de usuario" />
      <input type="password" name="password" placeholder="Contraseña" />
      <input type="submit" value="Iniciar sesión" />
    </form>
  </body>
</html>
```

```
}
```

```
</script>
```

```
</head>
```

```
<body onload="mensaje()">
```

```
    <form action="login.php" method="post">
```

```
        Usuario:
```

```
        <input type="text" size="30" maxlength="30" name="usuario" value="Usuario"/>
```

```
        <br/>
```

```
        Contraseña:
```

```
        <input type="password" size="10" maxlength="10" name="pass"/>
```

```
        <br/>
```

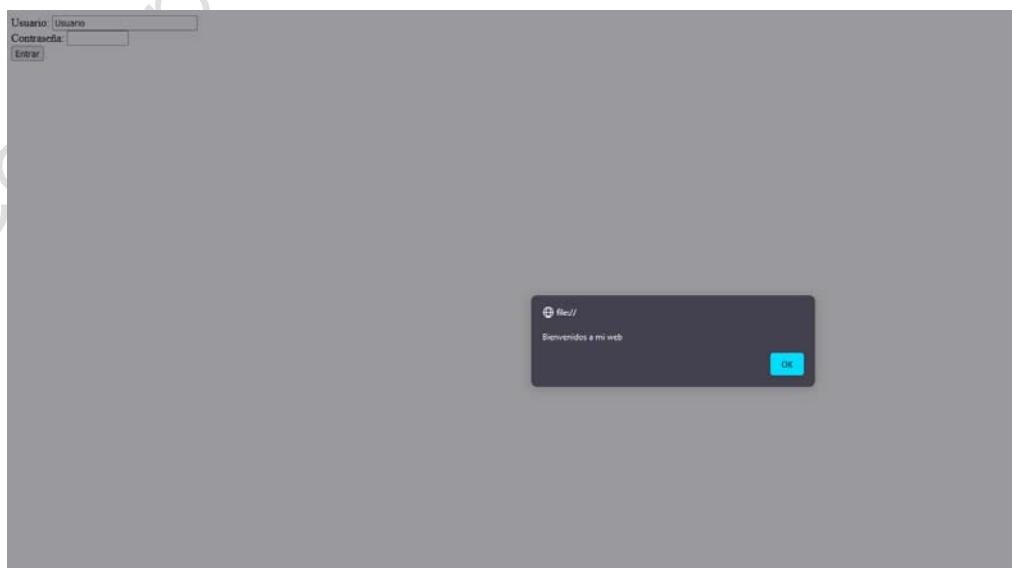
```
        <input type="submit" value="Entrar"/>
```

```
</form>
```

```
</body>
```

```
</html>
```

Vea lo que muestra el navegador.



Como puede ver, tan solo con añadir el código de la plantilla y eliminar los comentarios ha podido añadir un mensaje emergente de bienvenida en la web.

Este sería el protocolo que deberá utilizar para incluir plantillas en su web.

Sin embargo, esta no es la única manera de añadir plantillas en una web hoy por hoy, ya que, los CMS están muy popularizados y extendidos.

Antes que nada, debe saber que los CMS son programas que permiten la creación de páginas web muy estéticas de manera sencilla. Estos programas tienen una base de datos donde se aloja el sitio web y permiten controlar de manera independiente el diseño y el contenido.

Vea los CMS más conocidos y populares por su uso:

- WordPress
- Joomla
- Drupal
- Prestashop
- Magenta

Aunque existen muchos más como, por ejemplo, Moodle, Mediawiki o Serendipity.

Cada uno de los CMS está creado para encargarse de unas necesidades específicas que tiene cada sitio web.

Por ejemplo, el CMS de WordPress está enfocado para crear blogs o páginas muy básicas como una landing page. Además de poder crear una tienda online con pocos productos.

Por otra parte, Joomla está diseñado para crear sitios web interactivos y dinámicos y se utiliza mucho para temas de cursos online.

Drupal es un CMS libre y está diseñado para crear un diseño modular que permite crear una amplia diversidad de herramientas web como, por ejemplo, foros, blogs, encuestas o páginas web. Pero, sobre todo se utiliza para crear y gestionar comunidades en la web.

Por otro lado, Prestashop es un CMS especializado en tiendas online que puede ser de pocos o muchos productos. Este CMS ofrece un panel de control donde cambiar el diseño del sitio y añadir productos con diversas opciones.

Magento, por otra parte, es muy parecido a Prestashop, pero es un CMS mucho más eficiente en tiendas donde se alberguen miles de productos. Además, es una plataforma más cara de Prestashop en cuanto a la compra de *plug-ins*.

Puede decir que los CMS cada vez son más populares y que se puede decir que tienen una funcionalidad parecida a la de las plantillas facilitando el trabajo como programadores web.

Sin embargo, si no es capaz de cambiar un poco las plantillas de los CMS, puede quedar una página muy parecida a otras muchas miles, ya que, es una herramienta muy utilizada.

Es en este punto donde emplearía las plantillas creadas para los CMS, las cuales debe instalar en el servidor donde esté el sitio web.

Como ha visto anteriormente, WordPress es un CMS enfocado a la creación de blogs de manera sencilla y rápida.

Si entra a su web puede bajar la última versión de este CMS de manera gratuita e instalarla en su servidor de manera sencilla creando una base de datos.



Impagable y, al mismo tiempo, gratuito

Descarga WordPress y úsallo en tu sitio.

Descargar WordPress 5.8.2

[Descargar .tar.gz](#)

Instalación

Con nuestra famosa instalación en 5 minutos, configurar WordPress por primera vez es sencillo. Hemos creado una [sencilla guía](#) para ayudarte en el proceso de instalación.

Avisos de actualizaciones

¿Quieres recibir avisos sobre las actualizaciones de WordPress? Únete a la [lista por correo de novedades de WordPress](#) y te enviaremos un mensaje amistoso cuando haya una nueva versión estable.

Cuando instala este CMS, tendrá diversas opciones a la hora de escoger un tema para el sitio web y para modificar ese tema, en cambio, las opciones son más limitadas que si lo hiciera por otros mis.

En caso de utilizar esta herramienta, lo mejor que puede hacer es instalar una plantilla que se adecúe al diseño que está buscando y que necesita su página. Para ello tiene dos opciones:

- Comprar una plantilla de Internet.
- Descargar una plantilla gratuita.

De igual manera, en estos dos casos un programador externo será el que haya programado la plantilla de manera independiente a su proyecto, añadiendo cada elemento y diseño con miras de que otras personas puedan utilizarlas para sus sitios web.

Por otra parte, también puede escoger la opción de que otros mis como programadores la plantilla para WordPress, sin embargo, esto es algo complejo, ya que, para ello necesitará disponer de

conocimientos profundos de la herramienta, además de diferentes lenguajes de programación como SQL, JavaScript o PHP, y también de herramientas como Adobe Photoshop o Gimp.



TOME NOTA

Debe recordar que WordPress es un CMS que está diseñado específicamente para crear blogs. Sin embargo, es una herramienta tan potente que si utiliza una buena plantilla podrá crear una web con un diseño muy actualizado.

Si quiere instalar una plantilla descargada de Internet deberá seguir los siguientes pasos:

- Situados en Panel de control deberá acceder a la sección Apariencia.
- Entrare en el apartado Temas.
- Pulsare en la pestaña Instalar Temas.
- Pulsare a Seleccionar Archivo y seleccionare el Tema que ha descargado anteriormente.
- Seleccionare Instalar.
- Por último, deberá seleccionar entre las opciones que aparezcan.



TOME NOTA

Debe instalar los temas o plantillas antes de añadir los contenidos a la página web, siempre y cuando sea posible. Esto ocurre porque hay plantillas que no serán compatibles con su contenido.

Ahora va a hablar de Joomla, como ya ha visto anteriormente, esta herramienta es un CMS que está diseñada específicamente para la creación de sitios webs interactivos y dinámicos. Por ello, con esta herramienta podrá crear y eliminar contenido a través de un sencillo panel de control, sin tener que estar editando el código de la web.

Puede descargar esta herramienta en su página web oficial la cual podrá instalar en su servidor web y configurar de manera sencilla, tan solo, asignándole una base de datos. Vea la web:



Cuando instale Joomla, al igual que pasaba con WordPress, tendrá varias opciones, como las plantillas que ofrece la plataforma para modificar el diseño de la página, sin embargo, las opciones también son un tanto limitadas.

Lo mismo que ocurría con WordPress, lo mejor que puede hacer es instalar una plantilla que se adecúe al diseño que necesita.

Para obtener una plantilla tiene dos opciones:

- Descargar una plantilla gratuita de Internet.
- Comprar y descargar una plantilla de pago.

Lo mismo que ocurría con WordPress, en ambos casos un programador externo es el que habrá creado la plantilla de manera independiente con cada elemento y diseño para que sea usada por terceros.

Por otro lado, también tiene la opción de programar por sí mismo la plantilla para Joomla, aunque es más fácil de crear una plantilla en Joomla que en WordPress, puesto que, Joomla es un CMS mucho más ameno con el usuario, sigue siendo igualmente complicado para otros, puesto que, debe aprender a utilizar varios lenguajes de programación como HTML, SQL, PHP y JavaScript, además de tener que utilizar herramientas de diseño como Gimp o Photoshop.

Para instalar Joomla, deberá seguir los siguientes pasos:

- Cuando haya descargado la plantilla en su ordenador, deberá entrar en el backend de Joomla.
- Dentro del menú posicionarse en Instaladores, Plantillas y Sitio.
- Elegir una de las opciones que más se correspondan a su diseño y seleccionar el archivo para el paquete a instalar.

- Cuando se haya instalado el paquete, dirigirse al menú Sitio, posteriormente ir a Administrar Plantillas donde seleccionare la plantilla que ha instalado.
- Cuando le de a la opción asignar, en la siguiente pantalla aparecerán los fragmentos de la web donde quiere asignar la plantilla.
- Por último, pulsar guardar y ya habrá instalado la plantilla.



Lo mismo que pasaba con WordPress, deberá instalar los temas, siempre que poda, antes de que añada el contenido de la web, ya que, en caso contrario podría no verse el contenido como espera.

TOME NOTA

Por otro lado, **Drupal** es un CMS totalmente libre, es decir, está diseñado de manera modular de modo que permite realizar múltiples herramientas web como, por ejemplo, foros, blogs, encuestas o páginas web. Sin embargo, se utiliza especialmente para crear comunidades web.

Por otra parte, Drupal se divide en tres módulos a los que se les denomina las 3 C. Estos módulos son el Core o núcleo, el Contributes o contribuciones y el Custom o personalizado, en este último módulo es donde se almacenará el código creado por el desarrollador del sitio web.

Puede descargar Drupal en su página web oficial donde puede descargar su última versión e instalarla fácilmente, asignando una base de datos con la que la herramienta CMS pueda trabajar.

Vea a continuación la página oficial de Drupal donde podrá ver las múltiples opciones que ofrece este CMS:

The screenshot shows the official Drupal website. At the top, there's a dark header with a cookie consent message: "Can we use first and third party cookies and web beacons to understand our audience, and to tailor promotions you see?". Below it are two buttons: "Yes, please!" and "No, do not track me". The main navigation menu includes links for "Why Drupal?", "Build", "Solutions", "Services", "Community", "Resources", "Give", "Events", and "Try Drupal". There's also a search icon and a user profile icon. A prominent green banner features the "Drupal Steward" logo and the text: "Don't let security updates keep you up at night. Stay ahead of potential threats with an added layer of Drupal security." Below the banner, a green button says "Learn more about Drupal Steward". The main content area has a section titled "Download Drupal" with the subtext: "Drupal allows you to create a unique space in a world of cookie-cutter solutions." It provides instructions for using the Drupal Quickstart command, showing terminal commands like "composer create-project drupal/recommended-project drupal" and "php -d memory_limit=256M web/core/scripts/drupal quick-start demo_umami". It also links to "Looking for Drupal hosting?" with options to "Try a hosted demo" or "Explore Hosting".

Cuando instala esta herramienta, traerá una plantilla por defecto que puede cambiar por otros temas que ya vienen instalados.

Lo mejor que puede hacer como programadores es instalar una plantilla que concuerde con el diseño que quiere darle a la página web.

Recuerde que puede crear sitios como: Foros, portales sociales, sitios web corporativos, aplicaciones, blogs o sitios personales, aplicaciones de e-commerce, directorio con recursos y plataformas de redes sociales.

- Para poder utilizar una plantilla diferente a la que trae por defecto puede:
- Descargar una plantilla gratuita de Internet.

Comprar una plantilla de pago.

Al igual que ha visto en los dos CMS anteriores, son terceros los que crean la plantilla de manera independiente a su proyecto, utilizando elementos y diseños adecuados para que otras personas puedan utilizarlas.

Para instalar una de las plantillas descargadas, seguir los siguientes pasos:

- Copiar el tema que ha descargado en la plantilla themes, que se ubica donde esté el servidor de Drupal.
- En panel de control, ir a la sección Administrar.
- Entrar en el apartado apariencia.
- Una vez ahí. Seleccionare el tema para instalar.

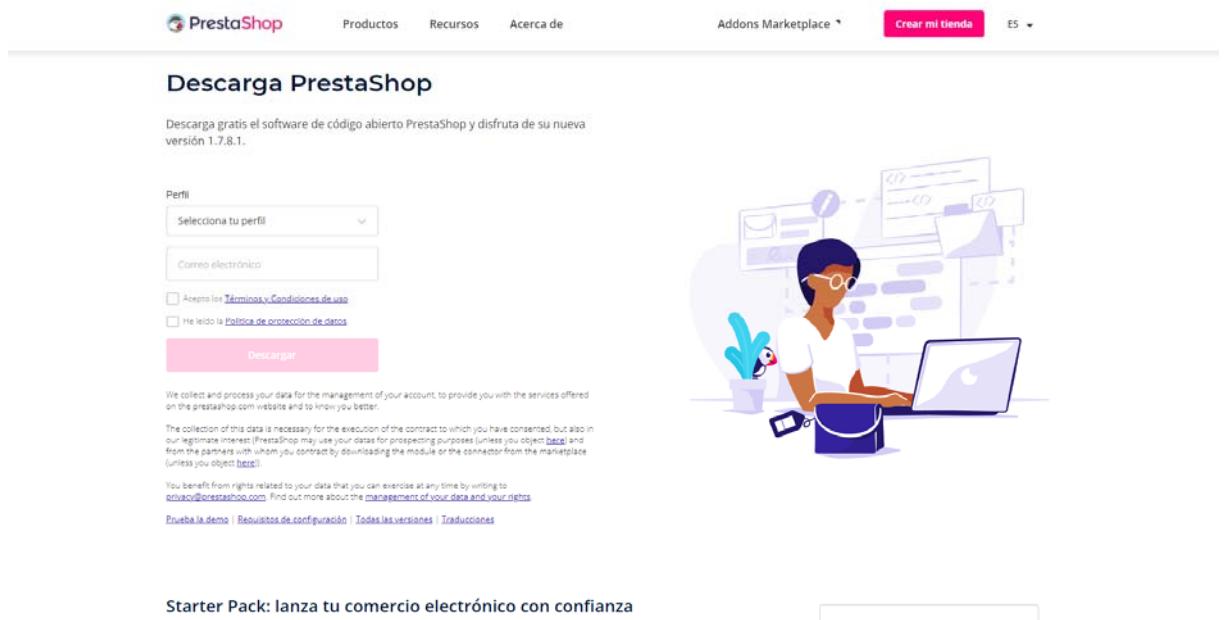


Tal y como hacia con los anteriores CMS, se debe intentar instalar los temas antes de añadir los contenidos a la página, ya que, puede haber algún conflicto con el contenido añadido anteriormente a la plantilla.

TOME NOTA

Ahora, va a hablar del CMS de **Prestashop** que está específicamente diseñado para crear tiendas online. Este CMS ofrece un panel de control mediante el cual puede añadir productos, métodos de pago, entre otras muchas cosas.

Para descargar este CMS, tan solo, tendrá que dirigir a la página web oficial de Prestashop y descargar su última versión que instalare en su servidor fácilmente. Vea una imagen de la página web oficial de Prestashop:



Starter Pack: lanza tu comercio electrónico con confianza

Al instalar esta herramienta, encuentra que ofrecerá diversas opciones como plantillas de diseño para su tienda. En cambio, estas plantillas son limitadas, sobre todo la que viene por defecto.

Debe recordar que este CMS, en concreto, está diseñado especialmente para la creación de tiendas online, lo que lo diferencia completamente de los dos CMS anteriormente vistos, los cuales podía utilizar para cualquier fin.

Sin embargo, recuerde que al igual que ocurría con los CMS anteriores, lo mejor que puede hacer como desarrolladores web es instalar una plantilla que se adecúe a su diseño.

Tiene dos opciones para hacer con una plantilla:

- Descargar una plantilla gratuita de Internet.
- Comprar una plantilla de pago.

En ambas opciones, la plantilla que elija se habrá creado por terceros y tendrá todos los elementos y diseños necesarios para que sea utilizada por otras personas.

En caso de que quiera instalar una plantilla deberá seguir los pasos siguientes:

- Descomprimir el archivo de la plantilla que ha descargado.
- Localizar la plantilla en los archivos que ha descomprimido, normalmente se ubica en la carpeta themes/nombre de plantilla, sin embargo, hay veces en que la plantilla se ubicará directamente en una carpeta con el nombre de la propia plantilla.
- Mover la carpeta hasta el lugar donde se ubica la carpeta de plantillas en el servidor.
- Posteriormente, en el panel de administración de Prestashop, seleccionare Preferencias y, a continuación, Apariencia. Por último, seleccionare Aplicar nombre del tema.

- Verificar en el panel de control de Prestashop lo que la plantilla está instalada con todos sus módulos.



TOME NOTA

Existen plantillas en Prestashop que traen una carpeta que sirve para sobrescribir los módulos que trae Prestashop por defecto.

2.3.1. Las plantillas en la web

Para utilizar plantillas en su web, no solo existe el método de tener que crearlas otros, sino que también puede descargarlas de Internet.

Puede encontrar dos tipos de plantillas:

- Plantillas gratuitas subidas por diseñadores web de manera voluntaria.
- Plantillas de pago que puede encontrar en plataformas especializadas en vender plantillas donde los programadores las suben y permiten que las use quienes paguen por ella.

Hay una regla para que encontrar variedad de plantillas para un sistema y es que cuanto más bajo sea el nivel del sistema que utiliza, más plantillas gratuitas encontrare.

Es decir, es mucho más sencillo encontrar plantillas gratuitas para HTML o JavaScript, que son lenguajes que necesitan de un programador con conocimientos en estos lenguajes para ser utilizados, que encontrar plantillas gratuitas para un CMS como Prestashop o WordPress.

Sin embargo, también es cierto que es más fácil crear webs estéticamente más avanzadas con un CMS, puesto que, la programación más compleja ya las trae por defecto y, además, muchas veces esas plantillas aparte de haber sido programadas por programadores profesionales, han sido diseñadas por diseñadores también profesionales.

De todos modos, siempre que utilice una plantilla de terceros debe tener en mente varias cosas:

- Es importante reconocer el trabajo que ha hecho un programador web creando esa plantilla. Para ello, puede poner un enlace en su pie de página para enlazar el sitio web donde ha descargado esa plantilla. También, puede poner un comentario en el código.
- Nunca debe descargar una plantilla pirata, puesto que, si los programadores web están tirando piedras en su propio tejado.
- Antes de instalar una plantilla web, deberá leer las instrucciones que traiga la misma y que encontraré en el propio sitio de la descarga o entre los archivos descargados.
- Deberá tener cuidado con utilizar determinadas plantillas en su web, puesto que, no son lo que espera y puede resultar difícil revertir el proceso. Por ello, es importante que antes de

instalar una plantilla haga una copia de seguridad. Esta es la regla de oro cuando utiliza plantillas web, puesto que, si no guarda una copia de seguridad y resulta que el cambio que ha hecho en su web es irreversible, habrá tirado por la borda horas y horas de trabajo.

A continuación, va a ver un ejemplo de plantilla de formularios en HTML:

```
<!-- Comienzo de la plantilla-->

<!DOCTYPE html>

<html>

  <head>

    <meta charset="UTF-8">

    <title>Ejemplo plantilla HTML</title>

  </head>

  <body>

    <form name="formhtml" method="post" action="htmlSend.php">

      <table width="450px">

        </table>

      </form>

      <tr>

        <td valign="top"><label for="nombre">Nombre *</label></td>

        <td valign="top"><input type="text" name="nombre" maxlength="50" size="30"></td>

      </tr>

      <tr>

        <td valign="top"><label for="apellidos">Apellidos *</label></td>

        <td valign="top"><input type="text" name="apellidos" maxlength="50" size="30"></td>

      </tr>

    </form>
```

```
<tr>

    <td valign="top"><label for="email">Email *</label></td>

    <td valign="top"><input type="text" name="email" maxlength="80" size="30"></td>

</tr>

<tr>

    <td valign="top"><label for="telefono">Número de teléfono</label></td>

    <td valign="top"><input type="text" name="telefono" maxlength="30" size="30"></td>

</tr>

<tr>

    <td valign="top"><label for="comentarios">Comentarios *</label></td>

    <td valign="top"><textarea name="comentarios" maxlength="1000" cols="25" rows="6"></textarea></td>

</tr>

<tr>

    <td colspan="2" style="text-align: center;">

        <!--Le agradece que mantenga este link en su web. Gracias.-->

        <input type="submit" value="Enviar">(<a href="http://freecontactform.com/htmlForm.php">Formulario HTML</a>

    </td>

</tr>

</table>

</form>

</body>

</html>
```

<!--Fin de la plantilla-->

Como puede observar, en este ejemplo no hay comentarios que indiquen el sitio del que se ha descargado el formulario, en cambio, puede encontrar el enlace del sitio del que se ha descargado la plantilla en el comentario que se encuentra al lado del botón submit.

A continuación, vea un ejemplo de plantilla de formularios en HTML:



2.3.2. Búsqueda de plantillas en la red

Buscar plantillas en la web puede ser desde un trabajo muy rápido hasta un trabajo que la búsqueda requiera de un programador.

De todas maneras, las herramientas principales para realizar estas búsquedas son, como no, los buscadores web como DuckDuckGo, Google o Bing.

Para realizar este tipo de búsquedas con éxito deberá realizar lo siguiente:

- Especificar el lenguaje de programación en el que necesita su plantilla.
- Especificar la función que quiere que realice la plantilla.
- Especificar en qué parte concreta del código o de la web quiere que la plantilla haga efecto.

Por ejemplo, si quiere encontrar una plantilla programada en JavaScript para poner un mensaje emergente en su web, hará lo siguiente:

The screenshot shows a Google search results page. The search query is "plantilla javascript para mensaje emergente". The top result is a link to "Como crear un mensaje de alerta o ventana emergente en ...". Below it, there's a section titled "Otras preguntas de los usuarios" (Other user questions) with four expandable dropdowns: "¿Cómo hacer un mensaje emergente con JavaScript?", "¿Cómo mostrar un mensaje de error en JavaScript?", "¿Cuáles son los tipos de ventanas emergentes en JavaScript?", and "¿Cómo poner un mensaje en JavaScript?". At the bottom right of this section is a "Enviar comentarios" button.

<https://www.wextensible.com/emergente-javascript> ▾
Mensajes emergentes con JavaScript para una página web
25 ago 2010 — Aquí creamos una ventana o **mensaje emergente** mediante objetos de **Javascript** y HTML dinámico. Al buscar un nombre adecuado para este objeto pensé ...

Como puede ver, entre los resultados que ha obtenido deberá encontrar la plantilla que está buscando.

En esta ocasión, puede encontrar con el código que busca en la tercera entrada que aparece en el buscador:

The screenshot shows a CodePen editor interface. The title of the codepen is "Spanish (Español) translation by Carlos (you can also view the original English article)". The code itself is a script for creating modals with JavaScript. It defines functions to launch first and second modals with different slide-in animations. The editor has tabs for HTML, CSS, and JS, with the JS tab selected. On the right side, there's a sidebar with a "CODEPEN" button, a "Subscribe" message, and a "Update me weekly" button.

1. Inicia con el marcado de la página

Como puede ver, esta página indica el código que necesita para crear una ventana emergente en su web utilizando JavaScript, pero además da también el código HTML y CSS a incluir.

En esta ocasión, puede ver que la ventana emergente se ejecuta cuando pulsa un botón. En caso de que quisiera una plantilla donde aparezca la ventana emergente cuando se cargue la página, deberá especificarlo en la búsqueda de la plantilla o programar esta función otros mis.

De todas maneras, en Internet hay muchísimas páginas donde puede encontrar información que ayude a programar cualquier función en cualquier lenguaje. Lo más complejo de este proceso es diferenciar la información útil de la que no lo es y encontrar plantillas que se adecúen a su web y lo que otros espera.

Es por ello que debe especificar al máximo en su búsqueda lo que quiere encontrar.

También debe utilizar las herramientas que ofrecen los navegadores para buscar cadenas concretas de datos.

Y utilizar la opción que también ofrecen los navegadores de buscar en sitios web concretos de donde haya podido encontrar plantillas satisfactorias anteriormente.

Sin embargo, cuando busca plantillas para CMS la búsqueda se va a volver más compleja si quiere utilizar los buscadores convencionales.

Es por ello que las diferentes plataformas CMS suelen tener sus propias páginas oficiales donde ofrecen multitud de plantillas gratuitas y de pago.

En cambio, hoy por hoy existe una gran diversidad de páginas especializadas en ofrecer plantillas para CMS mediante previo pago.

Este tipo de webs son como un escaparate para los programadores que se dedican a diseñar plantillas para los diferentes CMS y que incluso ofrecen una demo de sus plantillas antes de comprarlas para que quien vaya a comprarla sepa qué puede esperar de la plantilla.

Vea como ejemplo la siguiente web donde puede encontrar diversas plantillas para los CMS más conocidos:

The screenshot shows the Envato Elements homepage. At the top, there's a navigation bar with links for 'Videos de stock', 'Plantillas de video', 'Música', 'Efectos de sonido', 'Plantillas de gráficos', 'Gráficos', 'Plantillas de presentación', 'Fotos', 'Fuentes', 'Extensiones', and 'Más categorías'. A search bar is also present. Below the header, a large section titled 'La suscripción creativa ilimitada definitiva' is displayed, stating 'Descargas ilimitadas de más de 57 millones de recursos creativos. Desde solo 14,50 €/mes.' Below this, there are eight categories with images and counts: 'Plantillas de video' (43.000+), 'Audio' (680.000+), 'Plantillas de presentación' (68.000+), 'Plantillas de gráficos' (170.000+), 'Fuentes' (21.000+), 'Fotos' (53M+), 'WordPress' (3900+), and '3D' (100.000+).

Además, en esta web también podrá encontrar las plantillas más actualizadas, en cuanto a su diseño visual en la red:

The screenshot shows the Themeforest website. At the top, there's a search bar with the query 'wordpress'. Below the search bar, a banner says 'Productos digitales con descuento' and 'Es hora de comprar plantillas premium a un precio razonable'. There are several filters at the top: 'Todos los productos', 'Plantillas de sitios web', 'WordPress', 'WooCommerce', 'Shopify', 'PrestaShop', 'OpenCart', 'HTML', 'Presentaciones', 'Gráficos', 'Complementos', 'Audio', 'Video', and '3D'. The main content area displays five WordPress and WooCommerce themes with their respective prices and ratings: 'Monster CMS - Lifetime Plan' (for \$479), 'Techno' (49 €), 'Hazel' (69 €), 'Woostroid2' (86 €), and 'Alma' (63 €). Each theme has a small image, its name, a 'Black Friday SALE' badge, a price, a rating, and the number of sales.

Sin embargo, puede encontrar múltiples sitios web que se parecen a este y donde puede encontrar plantillas a la venta publicadas por los propios diseñadores web.

De todas maneras, este tipo de webs siempre tienen en común dos elementos:

1. Una barra de menú en la parte superior donde puede indicar el CMS para el cual quiere encontrar la plantilla.
2. Un buscador donde puede especificar el tipo de plantilla que quiere encontrar seleccionando un grupo de opciones que permite ajustar la búsqueda.

Una vez que haya encontrado la plantilla perfecta para su sitio, tan solo, deberá pagarla, descargarla e instalarla en su servidor web.

Como ha podido observar, la mayoría de herramientas que ofrecen las mejores opciones para encontrar plantillas están en inglés, es por eso que es importante que conozca lo suficiente de este idioma para poder dedicar al desarrollo web.

De hecho, la mayoría de las veces que predispone a buscar una determinada plantilla en español, la búsqueda va a ser en vano, por el contrario, si realiza la misma búsqueda en inglés, obtendrá muchos más resultados y puede que encuentre lo que busca al encontrar soluciones que no puede encontrar en español. Por ello, debe hacer hincapié en el aprendizaje de este idioma.

2.3.3. Adaptación de plantillas a páginas web

La mayoría de las veces lo más difícil no es encontrar una plantilla, sino utilizarlas en su página web, puesto que el código suele ser bastante denso, o no encaja a priori fácilmente con el código que ha creado otros para su web. Esto se debe a que puede que tenga que modificar ciertos elementos.

Si se encuentra en este caso, es importante que como programadores tenga:

- Un alto dominio de los lenguajes de programación que utiliza la plantilla y de los elementos que la componen.
- Ser flexibilidad para adaptar su código y hacer encajar la plantilla en su sitio web.
- Saber diferenciar las opciones que tiene y escoger la mejor en vez de quedar solo con la primera.

De todas maneras, cuando tenga la plantilla descargada deberá saber en qué parte del código debe ubicarla. Por ejemplo, el código JavaScript que ha visto anteriormente se debía ubicar dentro de la cabecera de la página como puede ver a continuación:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Ejemplo formulario</title>

<><script language="javascript">

    //Visualización de mensaje de bienvenida

    alert ("Bienvenidos a mi web");
```

```
</script>

</head>

<body>

<form action="login.php" method="post">

    Nombre:

    <input type="text" size="30" maxlength="30" name="nombre" value="Nombre"/>

    <br/>

    <input type="submit" value="Entrar"/>

</form>

</body>

</html>
```



TOME NOTA

En este ejemplo ha utilizado esta solución, sin embargo, no tiene porqué ser la mejor solución o la que debiéra utilizar obligadamente.

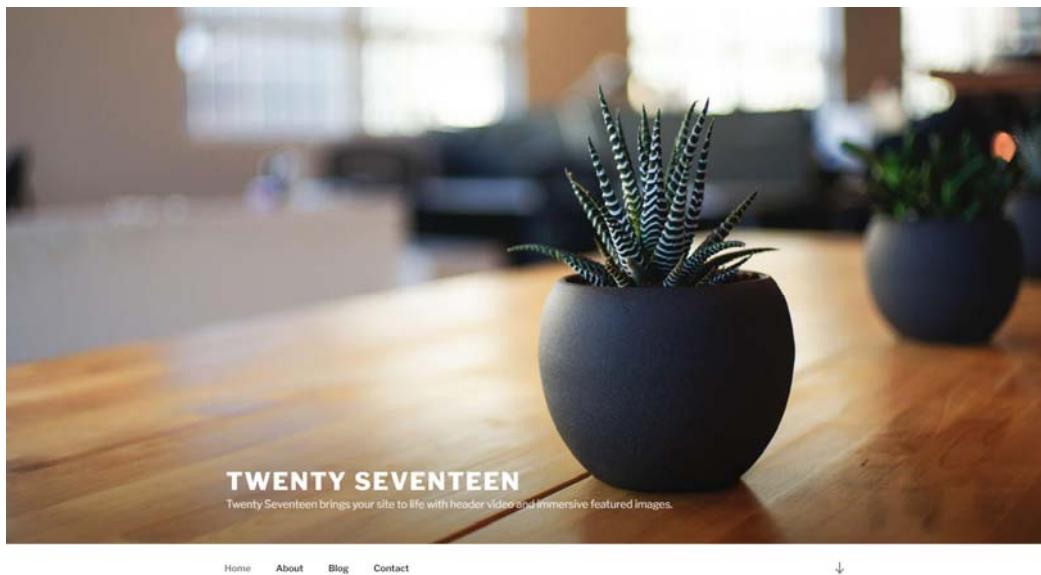
Por otra parte, si quiere utilizar una plantilla en un CMS deberá ser mucho más cuidadoso.

Ponga un ejemplo de una página web creada con WordPress que, como ya sabe, es un CMS diseñado de manera específica para la creación de blogs, sin embargo, puede utilizar este CMS para una gran diversidad de tareas, ofreciendo de manera muy sencilla una apariencia muy profesional. En cambio, esto puede ser complicado implementar en un sitio web creado anteriormente.

Por ejemplo, en caso de que haya creado una web con una apariencia de blog utilizando WordPress y le aplica una plantilla para modificar su apariencia con un slider principal y diferentes ventanas que muestren la información, podría ocurrir que todo el trabajo ya realizado desapareciera con esta nueva apariencia.

De hecho, podría encontrar con el caso de que la información de la nueva plantilla se pudiera introducir mediante widgets, por ejemplo, por entradas de blog. Esto significaría que el trabajo de programación de varias semanas tendría que ser modificado.

Por todo ello, en los CMS la mayoría de las veces deberá ser cuidadoso a la hora de aplicar las distintas plantillas. Vea un ejemplo a continuación de cómo puede cambiar una página en WordPress cuando utiliza una plantilla:



Esta plantilla puede encontrarla de manera gratuita en la web oficial de WordPress donde ha sido subida por un programador.

Como es lógico, como programadores, también deberá tener cuidado a la hora de eliminar una plantilla web de su diseño, puesto que, podría suponer en el proceso una pérdida de información de su web.

En caso de que instalara una plantilla de venta de productos en WordPress, añadiéra productos durante un largo tiempo y llegara un día en que borráse esa plantilla, podría perder todo ese trabajo realizado.



TOME NOTA

Debe hacer especial hincapié en guardar copias de seguridad de cualquier diseño antes de predisponer a realizar cualquier modificación en una web.

Para instalar una plantilla en una web que ya está terminada, debe hacer lo mismo que para instalar una plantilla en una web creada recientemente:

Situados en el panel de control dentro de la sección apariencia, debe:

- Entrar en el apartado temas.
- Seleccionar la pestaña instalar temas.

- Seleccionar el archivo y, posteriormente, el tema que quiera instalar.
- Seleccionar instalar ahora.
- Seleccionar las opciones que saldrán a continuación.

De todas maneras, siempre deberá comprobar el resultado antes de avanzar con el diseño de su web.

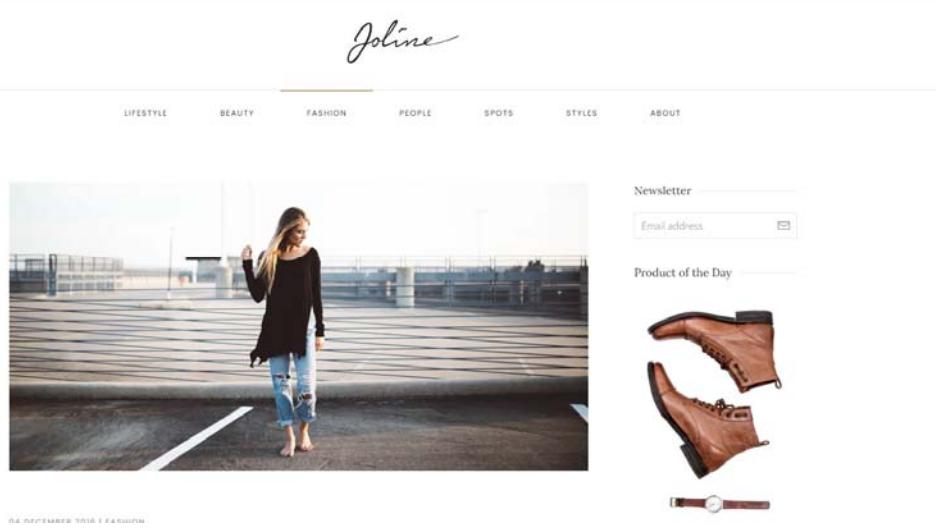
Por otra parte, si crea una web con Joomla, pasará lo mismo que con las creadas con WordPress, debe ser cuidadosos en cuanto a la manera de aplicar la plantilla. Esto es así porque, como ya sabe, Joomla está diseñado para crear sitios web interactivos y dinámicos. Por ello, puede crear o eliminar contenido a través de un panel de control en vez de tener que modificar el código web.

Esto se resume en que, al hacer modificaciones drásticas mediante una plantilla en su web, puede perder el trabajo de diseño que tanto esfuerzo ha costado construir.

Por ejemplo, ponga el caso de que diseña una web con un estilo clásico, que tenga un menú a la izquierda con múltiples opciones y subopciones, en la parte derecha, varias aplicaciones diferentes como enlaces, fecha y hora. Y, posteriormente, en medio tiene contenido textual.

Imagine que, como programadores, quiere cambiar ese diseño, siendo por su propia voluntad o porque el cliente lo haya pedido.

Por ello, puede entrar a una página de plantilla para Joomla y decidirse, por ejemplo, por la siguiente:



Cuando la haya instalado tendrá dos cosas:

- Los elementos que se han creado hasta ahora no servirán para nada.
- Todo el código que se haya introducido, deberá cambiarlo de aspecto y lugar para que concuerde con la plantilla nueva.

Como es lógico, este resultado va a ser un desastre para otros, ya que, ha puesto tiempo y esfuerzo para no obtener ningún resultado.



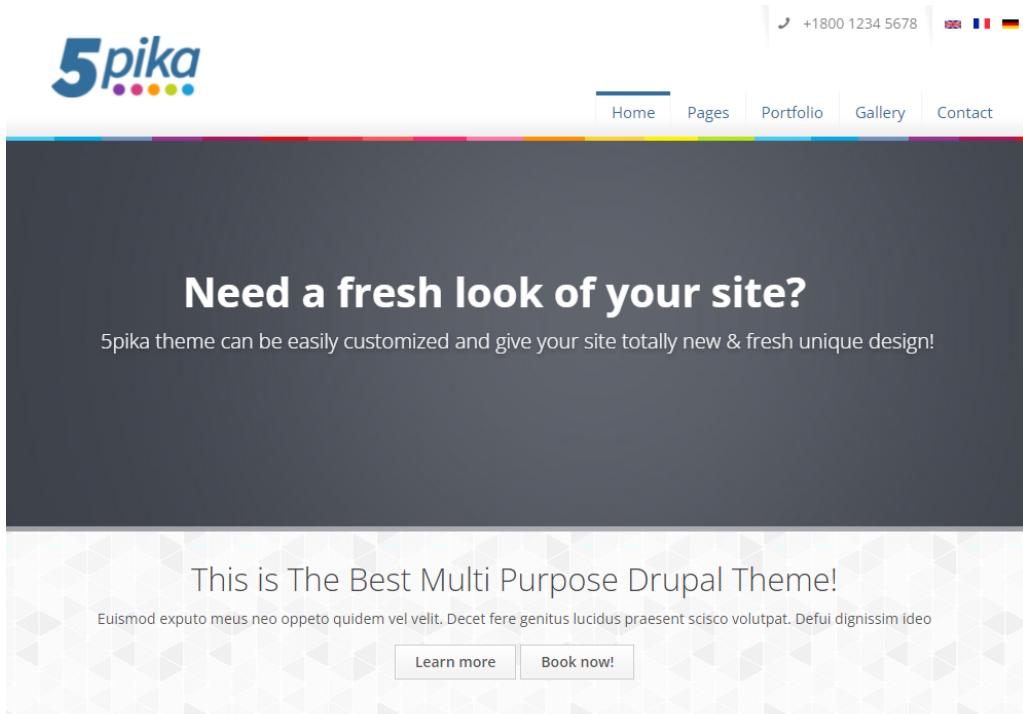
Cuando acepte el desarrollo de una web para un cliente, deberá dejar muy claro el diseño final que tendrá la misma, para evitar situaciones como esta y que su trabajo no haya servido para nada.

TOME NOTA

Lo mismo deberá hacer para instalar una plantilla en una web creada recientemente que en una web ya avanzada, deberá seguir estos pasos:

- Cuando tenga la plantilla descargada en el ordenador, deberá entrar en la administración de la página.
- Dentro de este menú, dirigirse a instaladores, plantillas y, por último, sitio.
- Elegir las opciones que más se adecúen al diseño y seleccionar el archivo para el paquete a instalar.
- Cuando se haya instalado el paquete ir al menú sitio, administrar plantillas y, posteriormente, seleccionar la plantilla que ha instalado.
- Cuando seleccione la opción asignar en la pantalla siguiente aparecerán todas las secciones de la web donde podrá aplicar la plantilla.
- Guardar y ya habrá finalizado de instalar la plantilla.

Por otra parte, con el CMS Drupal se encontrará con el mismo problema que tiene con el resto de CMS que ya ha visto. Esto sucede porque al ser un CMS libre diseñado para que sea modular y que permita realizar diversas herramientas como, por ejemplo, foros, páginas web, encuestas, etc. De hecho, es tal la diversidad de opciones que ofrece este CMS, que el cambio entre opciones puede variar el resultado en 180 grados. Vea a continuación una plantilla de Drupal puesta a la venta en envato market por un diseñador web:



Podría decir que el trabajo que ha realizado el programador es casi en su totalidad desecharable, puesto que, al encontrar ante un CMS que ofrece tantas opciones, el diseño final de su sitio web será tan diferente que parecerá como si hubiese utilizado otro CMS para programarlo.

Es importante que sepa que en los sistemas CMS como, por ejemplo, Drupal es muy importante saber qué es exactamente lo que quiere obtener como resultado final. Esto se debe a que la multitud de resultados que ofrece es también un arma de doble filo, puesto que, puede parecer que lo que esta creando puede dar un resultado que no es el que busca.

De todas maneras, si domina el campo de la programación y sabe cómo buscar plantillas para Drupal, tendrá una multitud de opciones y alternativas de diseño que el utilizar Drupal será mucho más ventajoso que utilizar cualquier otro CMS. Esto sucede porque, además, Drupal es un CMS con licencia GNU/GPL.

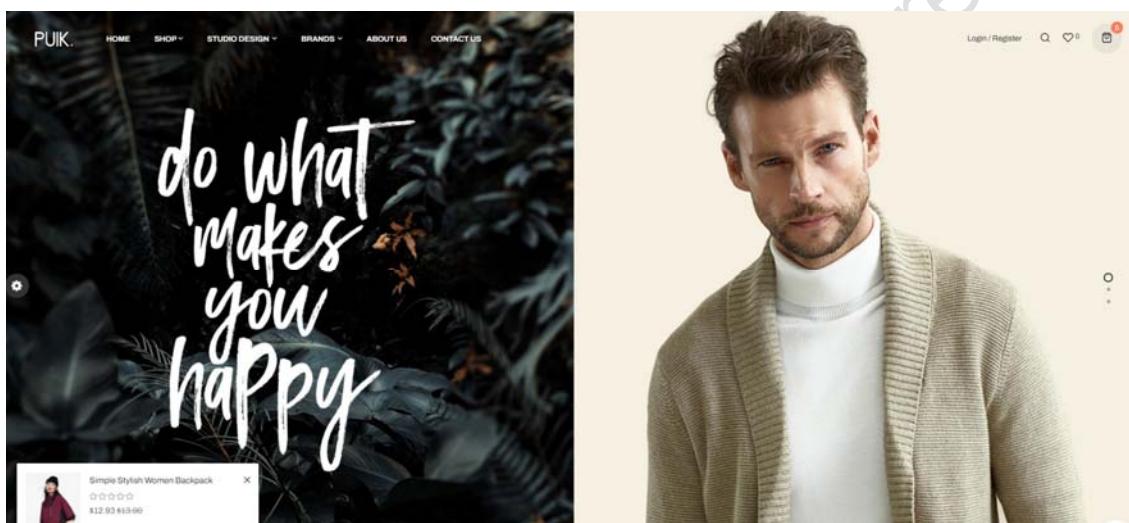
En caso de que quiera instalar una plantilla con Drupal, deberá seguir los siguientes pasos:

- Deberá copiar la plantilla descargada en la carpeta themes, que estará donde haya instalado Drupal.
- Posteriormente, en el panel de control accedere a Administrar.
- Entrar en el apartado apariencia.
- Seleccionar el tema que quiere instalar.

Por último, debe saber que no todos los CMS son iguales y en caso en que encuentre delante de una web ya creada a la que quiere aplicarle una plantilla de Prestashop con mucha programación.

Como ya sabe, este CMS está especializado para diseñar webs para comercio electrónico. Todo ello se hace mediante el panel de control donde tendrá las opciones de añadir productos, añadir diferentes métodos de pago, fotos para los productos, y gestionar las compras de los productos con Paypal. Debe saber que este CMS no se debe utilizar para otros fines que no sean crear un comercio electrónico, al contrario que ocurre con los demás CMS que ha visto hasta hora.

Es por todo esto que, si quiere aplicar una plantilla a una página web diseñada con Prestashop, tendrá muchos problemas que con los demás CMS. Ponga el ejemplo en que un programador ha creado una tienda online con la plantilla por defecto que trae Prestashop y al tiempo, quiere modificar esa plantilla, como por ejemplo la siguiente:



En este caso, todos los productos que haya introducido con anterioridad a la aplicación de esta plantilla se mantendrán en su sitio sin que tenga que modificarlos. Además, todos los métodos de pago, productos destacados, ventas a proveedores, ventas a los clientes, datos personales, opciones de administración, *plug-ins*, etc. se mantendrán exactamente iguales.



Cuando acepte el desarrollo de una web para un cliente, deberá dejar muy claro el diseño final que tendrá la misma, para evitar situaciones como esta y que su trabajo no haya servido para nada.

TOME NOTA

En conclusión, cuando aplica una plantilla en este CMS no es tan importante realizar esta aplicación antes de que la página esté creada en su totalidad. Aunque, es cierto, que siempre se recomienda hacerlo al inicio.

Por último, si quiere instalar una plantilla de este tipo en su Prestashop, deberá seguir los siguientes pasos:

- Descomprimir el archivo de la plantilla que ha elegido.
- Localizar los archivos descomprimidos, que normalmente encontrare en themes/nombrePlantilla, aunque, otras veces puede encontrarlos en una carpeta con el nombre de la plantilla.
- Mover la carpeta de la plantilla a donde esté la carpeta themes del servidor.
- Posteriormente, en el Back Office de su Prestashop, ir a preferencias, apariencia y aplicar nombre del tema y la activare.
- Por último, verificar en el Back Office, modules, que todos los módulos están instalados.

RESUMEN

En esta Unidad Formativa ha aprendido las características que tienen los formularios y la interactividad que puede aportar a las páginas web. Además, ha visto los tipos de interactividad que existen dependiendo de los elementos cómo se relacionen en la página. Otra de las cosas que ha visto es cómo varían los datos en una página web, elementos que son parte de la propia interactividad y el envío de información a los servidores.

Ha repasado los atributos y elementos que puede encontrar en un formulario, con los que puede hacer uso de los diferentes campos y textos. Puede añadir diferentes etiquetas y atributos para que las posibilidades de creación de formularios sean totalmente infinitas.

Por otro lado, ha aprendido lo que son los controles de formulario y cómo utilizarlos para enriquecer y darle más funcionalidad a los mismos. Dentro de los controles están las diversas opciones de botones que ofrece HTML, las listas desplegables, las casillas de verificación, los botones de radio y los campos de texto.

Por último, en el primer tema, se encuentran los eventos de formulario y los criterios de accesibilidad y usabilidad donde ha aprendido a agrupar los datos de un formulario, adecuar el tamaño del formulario, introducir campos obligatorios, ordenar de manera lógica los datos, entre otras cosas.

Durante el tema 2, ha aprendido las funciones y características que tienen las plantillas web y la facilidad que aporta a la hora del desarrollo web. Por otro lado, ha aprendido los elementos que puede utilizar con las plantillas y qué es en sí una plantilla. Además, ha visto cómo organizar y estructurar los diferentes elementos en las plantillas, así como, añadir zonas modificables y zonas fijas. Y cómo utilizar y sacarles provecho a estas plantillas.

También, ha aprendido a crear y definir campos editables y no editables, cosa que es muy útil cuando crea un formulario. Por último, ha visto dónde buscar plantillas ya predefinidas en Internet y cómo utilizarlas y adaptarlas para su web.

GLOSARIO

Atributo: Son valores adicionales que sirven para configurar los elementos en programación.

Campo modifiable: Este tipo de campos se encuentran en las plantillas y son campos que pueden modificar escogiendo un valor que corresponda al tipo de campo.

Campo no modifiable: Este tipo de campos se encuentran en las plantillas y son campos que no se deben modificar nunca.

Campo: Son los bloques con los que se construyen los formularios y donde el usuario debe llenar la información indicada.

CMS: Es un sistema de gestión de contenido que ayuda en el desarrollo de una web aportando sencillez e intuitividad.

Código ISO: Es el estándar internacional de códigos alfanuméricos cortos y únicos para cada país.

Drupal: Es un CMS muy potente y flexible. Con él podrá crear todo tipo de aplicaciones web.

Etiqueta: Son fragmentos del código que permiten crear los diversos elementos en HTML.

Evento: Es una acción que es detectada por un programa. Este programa, a su vez, interactúa con el evento dando un resultado a este.

Formulario: Es una parte de la página que permite al usuario introducir sus datos para obtener un resultado de ellos.

GNU/GPL: Es una licencia pública general de GNU, es decir, un software con esta licencia es un software libre.

Interactividad en la web: Es la capacidad cambiante que tiene un medio para ofrecerle a los usuarios una mayor libertad.

Interactividad web: Es lo que define que el usuario pueda interactuar con los elementos de la web más allá del desplazamiento o los clics.

ISP: Al ISP también se le conoce como proveedor de servicio de Internet, es decir, es cualquier compañía que traiga Internet a casa.

Joomla: Es un software libre y gratuito que se utiliza sobre todo para crear plataformas de cursos online.

Magento: Es un CMS muy parecido a Prestashop, pero mucho más potente. Se utiliza para comercios online muy grandes.

Plantilla: Es una página prediseñada donde se puede modificar los elementos para que tenga el diseño que busca.

Streaming: Es una tecnología que permite consumir la información de un archivo sin tener que descargar sus datos. Por ejemplo, Netflix es una plataforma de streaming.

Web dinámica: Es una página donde los contenidos varían en función de lo que hace el usuario en ella.

XML: Es un lenguaje de marcado que se utiliza para contener información de cualquier tipo.