

INFORMÁTICA Y
COMUNICACIONES



UF1304. ELABORACIÓN DE PLANTILLAS Y FORMULARIOS

Contenidos basados en los Certificados de profesionalidad



Certia
editorial

UF1304. ELABORACIÓN DE PLANTILLAS Y FORMULARIOS

DATOS DEL AUTOR

Enrique García Casal es ingeniero técnico en Informática de Gestión por la Universidad de Vigo.

Ha ejercido como profesional de la informática desde 2002 desarrollando *software* para grandes multinacionales de seguros y banca durante su estancia en Madrid. Es aquí donde descubre las posibilidades de HTML y comienza su aprendizaje y formación autodidacta en esta materia.

En 2005 se traslada a Galicia, donde sigue desarrollando *software* como analista programador en una conocida consultora.

En 2006 ficha por una empresa de comercio electrónico dedicada al turismo y comienza a experimentar con otras tecnologías de cliente como Flex, que le permiten tener una visión más profunda de lo que se puede conseguir desde el lado del cliente de una aplicación web.

En la actualidad ejerce como *freelance* a la vez que sigue formándose en nuevas tecnologías como AngularJS o NodeJS, y lo combina con el desarrollo de manuales formativos relacionados con certificados de profesionalidad.

FICHA

Elaboración de plantillas y formularios.
Informática y comunicaciones

1ª Edición
Certia Editorial, Pontevedra, 2015

Autor: Enrique García

Formato: 170 x 240 mm • 109 páginas.

ELABORACIÓN DE PLANTILLAS Y FORMULARIOS. INFORMÁTICA Y COMUNICACIONES.

No está permitida la reproducción total o parcial de este libro, ni su tratamiento informático, ni la transmisión de ninguna forma o por cualquier medio, ya sea electrónico, mecánico, por fotocopia, por registro u otros métodos, sin el permiso previo y por escrito de los titulares del Copyright.

Derechos reservados 2015, respecto a la primera edición en español, por Certia Editorial.

ISBN: 978-84-16019-97-7

Depósito legal: PO 718-2015

Impreso en España - Printed in Spain

Certia Editorial ha incorporado en la elaboración de este material didáctico citas y referencias de obras divulgadas y ha cumplido todos los requisitos establecidos por la Ley de Propiedad Intelectual. Por los posibles errores y omisiones, se excusa previamente y está dispuesta a introducir las correcciones pertinentes en próximas ediciones y reimpressiones.

Fuente fotografía portada: MorgueFile, autoriza a copiar, distribuir, comunicar públicamente la obra y adaptar el trabajo.

Familia profesional: INFORMÁTICA Y COMUNICACIONES

Área profesional: Desarrollo

FICHA DE CERTIFICADO DE PROFESIONALIDAD (IFCDO110) CONFECCIÓN Y PUBLICACIÓN DE PÁGINAS WEB (RD 1531/2011, de 31 de octubre modificado por el RD 628/2013, de 2 de agosto)

Correspondencia con el Catálogo Modular de Formación Profesional			
H. Q	Módulos certificados	H. CP	Unidades formativas
210	MF0950_2: Construcción de páginas web	210	UF1302: Creación de páginas web con el lenguaje de marcas
210	MF0951_2: Integración de componentes soft	180	UF1303: Elaboración de hojas de estilo
90	MF0952_2: Publicación de páginas web	90	UF1304: Elaboración de plantillas y formularios
	MF0278: Módulo de prácticas profesionales no laborales	80	UF1305: Programación con lenguajes de guión en páginas web
510	Duración horas totales certificado de profesionalidad	560	UF1306: Pruebas de funcionalidades y optimización de páginas web
			Duración horas módulos formativos
			480

ÍNDICE

• INTRODUCCIÓN.....	11
• UNIDAD DIDÁCTICA 1. Formularios en la construcción de páginas web	13
1.1. Características	15
1.1.1. La interactividad de las páginas web	15
1.1.2. La variabilidad de los datos de la página web	16
1.1.3. El envío de información a servidores.....	16
1.2. Elementos y atributos de formulario.....	21
1.2.1. Descripción y definición de los elementos de un formulario .	22
1.2.2. Utilización de campos y textos	23
1.2.3. Etiquetas de los formularios	25
1.2.4. Tamaños, columnas y filas de los formularios.....	28
1.3. Controles de formulario	29
1.3.1. Descripción de los controles de los formularios.....	29
1.3.2. Utilización de botones de acción.....	39
1.3.3. Utilización de listas desplegables.....	40
1.3.4. Utilización de casillas de verificación.....	42
1.3.5. Utilización de campos de textos	43
1.3.6. Cuadro de contraseña.....	44

1.4. Formularios y eventos. Criterios de accesibilidad y usabilidad en el diseño de formularios.....	45
1.4.1. Agrupación de datos.....	51
1.4.2. Adecuación del tamaño del formulario (división en distintas páginas)	52
1.4.3. Identificación de los campos obligatorios.....	53
1.4.4. Ordenación lógica de la petición de datos	54
1.4.5. Información correcta al usuario	55
1.4.6. Utilización de páginas de error y de confirmación	57
RESUMEN.....	59
EVALUACIÓN.....	61
 • UNIDAD DIDÁCTICA 2. Plantillas en la construcción de páginas web	63
2.1. Funciones y características.....	65
2.1.1. Descripción de una plantilla web.....	66
2.1.2. Elementos de una plantilla web.....	67
2.1.3. Estructura y organización de los elementos de las plantillas ..	68
2.1.4. Especificar las zonas modificables de una plantilla y las partes fijas	71
2.1.5. Utilización de plantillas	77
2.2. Campos editables y no editables.....	77
2.2.1. Definir y crear los campos susceptibles de cambios en una plantilla.....	78

2.2.2. Definir y crear los campos no modificables en una plantilla ..	79
2.3. Aplicar plantillas a páginas web	80
2.3.1. Las plantillas en la web.....	80
2.3.2. Búsqueda de plantillas en la red.....	80
2.3.3. Adaptación de plantillas a páginas web	81
RESUMEN.....	89
EVALUACIÓN.....	91
 • RESUMEN FINAL.....	95
 • EJERCICIOS	97
 • EVALUACIÓN FINAL	99
 • SOLUCIONES	101
 • BIBLIOGRAFÍA/WEBGRAFÍA	105

INTRODUCCIÓN

Dentro de la construcción y diseño de un sitio web, el desarrollador web ha de tener en cuenta la interactividad que le va a proporcionar. Entendemos la interactividad como la comunicación entre el sitio web y el usuario, esto es, la reacción del sitio web ante una acción del usuario. Por ello, en este manual se abordará, en primer lugar, el diseño de formularios para los sitios web con el objetivo de saber incluir interactividad entre el usuario web y la propia página. También se identificarán los elementos o etiquetas y atributos que se utilizan para la elaboración de los formularios web y se realizarán páginas web que incorporen formularios según el diseño especificado, identificando los controles que se van utilizar en cada formulario, asignando propiedades, aplicando criterios de usabilidad y accesibilidad. Por último, se probará su funcionamiento en varios navegadores y se llevará a cabo el análisis de verificación y corrección de errores.

Otro tema que vamos a tratar en esta unidad formativa será la utilización de plantillas en la construcción de páginas web. Se confeccionarán plantillas para páginas web y se describirán las características que ofrecen, así como las utilidades de las herramientas de edición de páginas para crear plantillas. Además se identificará las partes principales de una plantilla, las regiones editables y no editables, y se insertará los elementos y atributos especificados para cada plantilla. Por último, se aplicará una plantilla a un sitio web y se probará su comportamiento en varios navegadores.

FORMULARIOS EN LA CONSTRUCCIÓN DE PÁGINAS WEB

1

- **CONTENIDO**

- 1.1. Características
- 1.2. Elementos y atributos de formulario
- 1.3. Controles de formulario
- 1.4. Formularios y eventos. Criterios de accesibilidad y usabilidad en el diseño de formularios

- **RESUMEN**

- **EVALUACIÓN**

En el diseño de páginas web es necesario contar con apartados en los cuales el usuario web pueda interactuar con la empresa o el propietario de la página. De ahí la necesidad de utilizar los formularios como método de comunicación. Generalmente, este tipo de interacción consiste en que el usuario rellene los datos, a través de los formularios, los cuales serán enviados por el navegador a un servidor o un correo electrónico por ejemplo. Una vez que se reciban, serán procesados y se utilizarán para el fin para el que se solicitan (reservas, encuestas, etc.).

1.1. Características

La interacción del usuario web con la página visitada se puede realizar a través de formularios, chat, videoconferencia, correo electrónico, etc.

La existencia de formularios web es un hecho casi total en la mayoría de los sitios web que pueblan internet. Todos ellos han sido desarrollados usando lenguajes interpretados por el navegador como HTML o JavaScript, o ejecutados en el servidor web como pueden ser PHP, Net, Java, etc. Sea cual sea la tecnología de programación usada, el uso de cualquier formulario presenta una serie de características comunes que describiremos a continuación.

1.1.1. La interactividad de las páginas web

El desarrollador web ha de procurar que la acción de llenar un formulario sea agradable y fácil. Para ello, se tratará de reducir al máximo el número de campos que se deberán llenar, evitar las listas desplegables, los campos innecesarios, y sobre todo se evitará fragmentar demasiado la información.

Otro de los usos que se suele dar a los formularios es la consulta de información existente en una base de datos. Usando formularios es posible mostrar la información común obtenida de la consulta en diversas formas o formatos.

1.1.2. La variabilidad de los datos de la página web

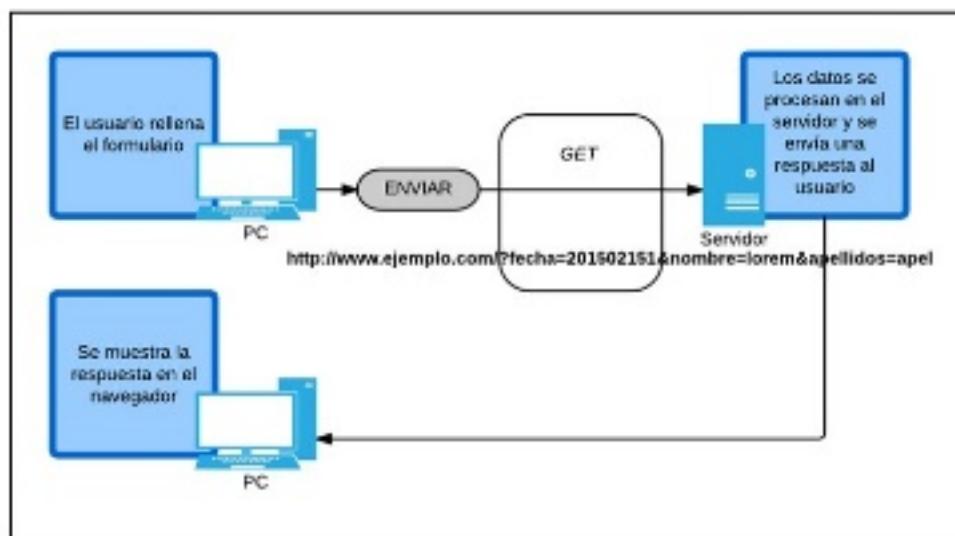
La gran diversidad de contenidos de la web implica que los datos solicitados en los formularios web sean en ocasiones diferentes entre distintas páginas web, e incluso dentro en un mismo sitio web. Además, en un mismo formulario los datos pueden variar en el tiempo o según hayamos seleccionado un valor u otro en campos anteriores. No es lo mismo un formulario de búsqueda de información dentro de un sitio web que uno de registro de clientes.

Debido a los diferentes tipos de datos que se suelen utilizar en un formulario, es aconsejable utilizar los campos adecuados para cada tipo de dato. No tiene sentido utilizar una lista desplegable para introducir el nombre del usuario, ya que es inviable hacer una lista con todos los posibles nombres, pero sí para seleccionar por ejemplo el país.

1.1.3. El envío de información a servidores

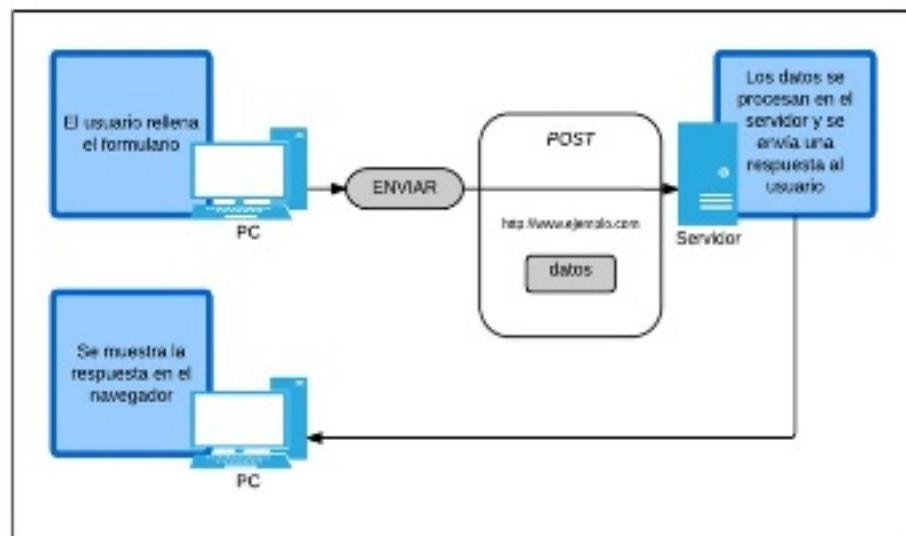
La información recogida por un formulario puede ser enviada al servidor para su posterior tratamiento usando dos tipos de métodos que veremos a continuación.

El primero de ellos es el método «*get*», en el cual los datos del formulario se envían al servidor a través de la URL por medio de parámetros. Este método se suele utilizar cuando enviamos poca información y esta no es de carácter confidencial, ya que se vería en el navegador. Un ejemplo del método «*get*» serían los formularios de búsqueda de artículos dentro de una página web.



El otro método se conoce como método «*post*», y se caracteriza porque los

datos pasan ocultos y encriptados al servidor. Se utiliza para enviar gran cantidad de datos o información confidencial. Un ejemplo sería un formulario de registro de clientes o un acceso privado a una parte de la página con los campos «nombre de usuario» y «contraseña».



A continuación se puede ver un ejemplo de formulario escrito en HTML utilizando el método `get`:

```
<form id="contactar" name="contactar" method="get" action="obtenerdatos.php">
    <label>Nombre:</label>
    <input name="nombre" type="text" /><br/>
    <label>Mail:</label>
    <input name="correo" type="text" /><br/><br/>
    <input type="submit" name="enviar" id="enviar" value="Enviar" />
</form>
```

Esto se visualizaría en el navegador de la siguiente forma:

Formulario de Contacto

Nombre: mi nombre

Mail: mimail@mail.com

Enviar

Como se observa, se crea un formulario con la etiqueta <FORM>, que se verá posteriormente. La etiqueta <LABEL> sirve para destacar los títulos de los campos de tipo texto que se incluyen con la etiqueta <INPUT> y su atributo «*type*» con valor «*text*». El elemento <INPUT>, cuyo atributo «*type*» tiene el valor «*submit*», sirve para que el usuario pueda dar la orden al navegador para que envíe el formulario al servidor.

Los datos son enviados al servidor en la URL, formando lo que se conoce como la *query string* ('cadena de consulta' en castellano). La *query string* que el navegador enviará al servidor será: «obtenerdatos.php?nombre=<DATOS EN CAMPO NOMBRE>&correo=<DATOS EN CAMPO EMAIL>&enviar=Enviar».

En el servidor, los datos se podrían obtener de la siguiente forma (usando el lenguaje de servidor «PHP»):

- Crear un fichero de texto llamado «*obtenerdatos.php*» que se debe almacenar en la misma carpeta que contiene al fichero «.html» con la definición del formulario.
- El contenido del fichero, de manera muy simplificada, podría ser este:

```
<!doctype html>
<html>
<head><meta charset="utf-8">
</head>
<body>
<?php
    $nombre = $_GET['nombre'];
    $correo = $_GET['correo'];

    echo "su nombre es: \"$nombre";
    echo "</br>su e-mail es: \"$correo;

?
</body>
</html>
```

Y su representación en el navegador una vez recogidos los datos, procesados y enviados al navegador será:

su nombre es: mi nombre
su e-mail es: mimail@mail.com

El mismo formulario anterior usando el método «post» remite la información de forma oculta y sin limitaciones de tamaño. El único cambio necesario es el valor del atributo `<method>`, y será el navegador el que realice todo lo necesario para enviar al servidor los datos en el formato correcto.

```
<form id="contactar" name="contactar" method="POST" action="obtenerdatos.php">  
    <label>Nombre:</label>  
    <input name="nombre" type="text" /><br/>  
    <label>Mail:</label>  
    <input name="correo" type="text" /><br/><br/>  
    <input type="submit" name="enviar" id="enviar" value="Enviar" />  
</form>
```

En el servidor, el archivo `«obtenerdatos.php»` contendrá, de una manera muy simplificada, el siguiente código para recuperar los datos enviados por el formulario, generar el HTML necesario para mostrar dichos datos y enviarlos al navegador:

```
<?php  
    $nombre = $_POST['nombre'];  
    $correo = $_POST['correo'];  
  
    echo "su nombre es: ".$nombre;  
    echo "</br>su e-mail es: ".$correo;  
?>
```

Como se puede apreciar, la única diferencia entre los dos casos es el uso de las palabras clave `$_GET` y `$_POST` en el archivo `«obtenerdatos.php»` para

recoger los datos enviados desde el formulario HTML. Es un error muy común usar «get» o «post» en el formulario HTML y posteriormente, al recuperar los datos en el servidor, usar el método contrario con «\$_GET» o con «\$_POST» (usando PHP como lenguaje de servidor, aunque cada lenguaje tiene sus propias palabras clave para representar esta acción).

Otro ejemplo en el que se explica el intercambio de datos entre el navegador y el servidor es el acto de subir archivos al servidor. Un «upload» de ficheros. A continuación se muestran el código HTML y el código PHP simplificado para enviar y recibir un archivo desde un formulario HTML. Así se mostraría en el navegador:

Seleccione el archivo a subir

Enviar este archivo: Ningún archivo seleccionado

El código HTML necesario será:

```
<body>
<h2>Seleccione el archivo a subir</h2>
<!-- El tipo de codificación de datos, enctype, se DEBE especificar como
a continuación -->
<form enctype="multipart/form-data" action="obtenerdatos.php"
method="POST">
    Enviar este archivo: <input name="fichero" type="file" />
    <input type="submit" value="Enviar" />
</form>
</body>
```

Lo principal de este código es el valor del atributo «enctype». Con «*multipart/form-data*» se indica al servidor que los datos que viajan en esa petición (en este caso «post»), forman parte de un archivo que se intenta subir al servidor.

Una vez seleccionado el archivo con el diálogo que mostrará el navegador, este se envía al servidor donde será recogido por el archivo «*obtenerdatos.php*»,

que contendrá el código PHP necesario para manejar el fichero subido desde la página HTML:

```
<?php  
$carpeta_destino = './uploads/';  
$fichero_subido = $carpeta_destino.basename($_FILES['fichero'][  
['name']);  
  
echo '<pre>';  
if (move_uploaded_file($_FILES['fichero'][['tmp_name']], $fichero_subi-  
do))  
{  
    echo "Subido perfectamente.\n";  
} else {  
    echo "Error!\n";  
}  
print "</pre>";  
?>
```

Una vez procesado, el código PHP genera la salida HTML en la que se informa de si ha ocurrido un error o si se ha subido correctamente. El resultado de la subida debe ser el siguiente:

Subido perfectamente.

1.2. Elementos y atributos de formulario

Un formulario web se construye usando el lenguaje HTML (*Hyper Text Markup Language* o ‘lenguaje de marcas de hipertexto’) el cual será interpretado por un navegador. Es decir, un formulario web será parte de un documento HTML y, como tal, debe usarse dicho lenguaje de marcas para crearlo.

Un formulario web es un elemento HTML definido con las etiquetas <FORM> de apertura y cierre. Este formulario se compondrá de campos de texto principalmente, pero también pueden aparecer otros tipos como son los campos de listas desplegables, los campos numéricos, los campos de verificación, los botones de opción, etc. Además, como en cualquier formulario, debe haber un botón que será el que envíe la información al servidor para su posterior tratamiento.

1.2.1. Descripción y definición de los elementos de un formulario

En un formulario tendremos que definir varios elementos para que este sea operativo. Va a estar formado por una serie de controles, ya sea para la obtención de datos, como los campos de texto, o para el envío de estos datos, como los botones de acción.

Dentro de un formulario podremos definir diferentes tipos de elementos o controles:

- Controles para introducir texto, números o contraseñas.
- Controles para seleccionar una o varias opciones, combos o listas desplegables.
- Controles para verificar una o varias opciones, como las casillas de verificación o los grupos de opciones.
- Controles de acción como los botones de envío, de borrado de datos del formulario o de verificación de datos.
- Etiquetas informativas para que el usuario sepa en cada momento qué información debe introducir en cada campo.

Cada uno de estos controles posee sus propias propiedades o atributos para personalizar su uso y funcionamiento. A alguno de estos controles incluso es posible añadirle funciones de validación que limitarán el rango de valores posibles que se podrán introducir, o el tipo de datos necesarios. En los capítulos sucesivos se irán definiendo los distintos elementos que pueden componer un formulario web.

1.2.2. Utilización de campos y textos

Grasso modo un formulario web se compone de dos tipos de elementos: campos de entrada de información y textos descriptivos de estos campos.

Un *campo de entrada de información*, más comúnmente conocido como *campo de texto*, es la forma que el formulario ofrece al usuario para que pueda interactuar con la página introduciendo o seleccionando la información presentada. Los campos dentro de un formulario vendrán definidos por la información que se quiere obtener del usuario. Así, por ejemplo, si solicitamos al usuario su nombre o correo electrónico utilizaremos un campo tipo texto para que pueda introducirlo. En cambio, cuando queramos que el usuario elija una serie de opciones determinadas utilizaremos campos tipo lista de desplegables para proporcionarle esas opciones.

Los textos descriptivos, por su parte, deben acompañar a los campos de texto, complementándolos y aclarando qué información se solicita.

En la siguiente imagen se observan los posibles tipos de campos existentes en una web. El texto que los acompaña describe el tipo de dato que espera dicho campo de texto.

Tipos de «input» en HTML

Texto:
Contraseña:
Submit:
Radio: Masculino Femenino
 Casilla de verificación (checkbox)
Botón:

Nuevos tipos de «input» en HTML5

Fecha:
Fecha/Hora:
Fecha/Hora local:
Mes:
Hora:
Semana:
Color:
Correo electrónico: Format: example@example.com
Teléfono: Format: 1234567890
Rango:
Lista desplegable:
Número:
Url:

El código usado para generarla ha sido:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin título</title>
</head>

<body>
<h1>Tipos de Input en HTML</h1>
<form>
    <label>Texto:</label><input type="text"><br>
    <label>Contraseña:</label><input type="password"><br>
    <label>Submit:</label><input type="submit" value="enviar"><br>
    <label>Radio:</label>
    <input type="radio" name="sex" value="male" checked>Masculino
    <input type="radio" name="sex" value="female">Femenino
    <br />
    <input type="checkbox" name="vehicle" value="Bike">Casilla de verificación
    (checkbox)<br>
    <label>Botón:<input type="button" onclick="alert('Hello World!')" value="-
    botón">
</form>
<h1>Tipos Nuevos de Input en HTML5</h1>
<form>
    <label>Fecha:</label><input type="date"><br>
    <label>Fecha/Hora:</label><input type="datetime"><br>
    <label>Fecha/Hora local:</label><input type="datetime-local"></
    br>
    <label>Mes:</label> <input type="month"><br>
    <label>Hora:</label> <input type="time"><br>
    <label>Semana:</label> <input type="week"><br>
    <label>Color:</label> <input type="color" value="#c04019"></
    br>
    <label>Email:</label> <input type="email" placeholder="Format:
    example@sixrevisions.com"><br>
```

```
<label>Teléfono:</label> <input type="tel" placeholder="Format:  
1234567890" pattern="[0-9]{10}"><br>  
<label>Rango:</label> <input type="range" min="1"  
max="10"><br>  
<label>Lista desplegable:</label> <input type="search" list="onli-  
ne-stores"><br>  
<datalist id="online-stores" style="display:hidden;"><br>  
<option value="Amazon.com">  
<option value="Zappos">  
<option value="Buy.com">  
<option value="Apple Store">  
<option value="Threadless">  
</datalist>  
<label>Número:</label> <input type="number"><br>  
<label>Url:</label> <input type="url"><br>  
</form>  
</body>  
</html>
```

1.2.3. Etiquetas de los formularios

En la creación de un formulario web intervienen una serie de elementos HTML definidos por sus etiquetas y relacionados entre sí. Todos juntos hacen que el usuario pueda introducir información en una página web y que esta sea enviada y procesada en un servidor web, y este a su vez pueda actuar en función de los datos recibidos.

La etiqueta usada para la creación de formularios es la etiqueta **<FORM>**.

La etiqueta **<FORM>** es el elemento principal en la definición de un formulario web. Es el elemento padre dentro del cual se insertarán todos los demás elementos que formarán el formulario.

En el siguiente ejemplo se muestra una página web que contiene un formulario compuesto por un campo de texto y un botón «Enviar»:

```

<!DOCTYPE html>
<html>
<head><title>Ejemplo de formulario</title></head>
<body>
<h3>Formulario muy sencillo</h3>
<form action="obtenerdatos.php" method="post">
    Nombre:
    <input type="text" name="nombre" value="" />

    <br/>

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

```

Este código se mostrará de la siguiente forma en un navegador:

Nombre:	<input type="text"/>
<input type="button" value="Enviar"/>	

A la etiqueta <FORM> se le pueden aplicar los atributos que se enumeran en la siguiente tabla:

Atributo	Descripción
<i>accept-charset</i>	Especifica el conjunto de caracteres usado en el formulario. Por defecto usa el definido en la página web
<i>action</i>	Especifica la URL a la que se enviará el formulario. Si no se especifica, por defecto lo enviará a la misma página web cargada
<i>autocomplete</i>	Señala si el navegador debería usar la función « <i>autocomplete</i> ». Por defecto tiene el valor « <i>true</i> »

<i>enctype</i>	Indica la codificación de los datos enviados en el formulario. El valor por defecto es «url-encoded»
<i>method</i>	Especifica el método HTTP usado cuando se envía el formulario. Por defecto es «get»
<i>name</i>	Especifica un nombre usado para identificar el formulario
<i>novalidate</i>	Indica al navegador que no valide el formulario antes de su envío
<i>target</i>	Especifica si la respuesta de la dirección especificada en el atributo «action» se abrirá en la propia página o en otro destino. Por defecto tiene el valor «_self»

Con estos atributos se personaliza el funcionamiento del formulario web. Por ejemplo, un formulario cuyos datos deben ser procesados por la página PHP «obtenerdatos.php» usando el método «get», cuyo resultado deba ser mostrado en una nueva página, sin validación y sin autocompletado automático de campos, se definirá como sigue:

```
<form action="obtenerdatos.php" method="GET" target="_blank"
accept-charset="UTF-8"
enctype="application/x-www-form-urlencoded" autocomplete="off"
novalidate>
    elementos del formulario
</form>
```

Para enviar un correo usando el cliente de correo que tenga el usuario definido por defecto en su sistema operativo, se debe especificar en el atributo «action» la dirección *e-mail* de destino precedida de la palabra «mailto»:

```
<form name="formulario" action="mailto:sucorreo@dominio.com"
method="post" enctype="text/plain">
```

También es posible, por último, utilizar un formulario para subir ficheros al servidor web. Para ello es necesario especificar el atributo «*enctype*» y darle el valor «*multipart/form-data*»:

```
<form enctype="multipart/form-data" action="obtenerdatos.php"
method="POST">
    Enviar este archivo: <input name="fichero" type="file" />
    <input type="submit" value="Enviar" />
</form>
```

1.2.4. Tamaños, columnas y filas de los formularios

A los controles de un formulario se les pueden aplicar atributos de anchura y altura, se puede configurar la cantidad de caracteres que se podrán introducir en cada campo, así como el número de líneas en los campos «*textarea*». Los atributos que se podrán utilizar son los siguientes:

- «*Cols*»: indica el número de caracteres que puede contener una línea.
- «*Rows*»: indica el número de líneas.
- «*Size*»: indica el número de líneas de la lista en controles «*textarea*» o el ancho del campo en controles tipo texto.
- «*maxlength*»: indica el número de caracteres en los controles tipo texto.

Así, por ejemplo, si se desea un campo de texto tipo «*textarea*» con 5 columnas de ancho y 5 filas de alto se definiría el HTML como sigue:

```
<textarea rows="5" cols="5"> </textarea>
```

Si se desea que el campo «*textarea*» tenga como máximo una longitud de 120 caracteres (independientemente de los que se muestren en el navegador, especificados como se ha visto con «*rows*» y «*cols*») se utilizará la propiedad «*maxlength*» como sigue:

```
<textarea rows="5" cols="5" maxlength="5"> </textarea>
```

De la misma forma se usaría para los campos de texto tipo «*text*».

1.3. Controles de formulario

Como ya se ha comentado anteriormente, un formulario HTML se compone de controles de entrada que, dependiendo del atributo que se les aplique, se comportarán como campos de texto, casillas de verificación, botones de radio, botones de envío, etc. También puede contener listas de selección o cuadros para texto grande.

Los controles no dejan de ser elementos HTML que se representan por su etiqueta y se caracterizan por sus propiedades o atributos.

1.3.1. Descripción de los controles de los formularios

Un formulario web se compone de diferentes controles para la manipulación de datos de entrada. Estos controles se añaden en el HTML dentro de la etiqueta <FORM>, y se opta por uno u otro en función del tipo de dato que se maneje y la representación de este.

Dentro de los controles de un formulario se puede diferenciar:

- Controles de texto.
- Casillas de verificación.
- Radios o grupos de opciones.
- Etiquetas.
- Menús de selección.
- Campos ocultos.
- Botones.
- Subir archivos.

a. Controles de texto

Permiten introducir texto en el formulario. Es posible introducir un valor concreto o un texto simple. Los elementos HTML usados son los siguientes:

- `<INPUT>`: es el control más utilizado en los formularios. Se utiliza para introducir texto, introducir contraseñas, verificar datos, seleccionar entre varias opciones, seleccionar archivos para enviar e incluir botones para enviar el formulario, borrar los datos o ejecutar código JavaScript. Se define de la siguiente forma:

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

Los atributos posibles para este elemento en HTML5 se enumeran en la siguiente tabla¹:

Atributo	Valor	Descripción
<code>accept</code>	<code>audio/*, video/*, image/*, tipo de medio</code>	Especifica el tipo de ficheros que el servidor acepta. Se usa solo cuando se cumple que <code>type="file"</code>
<code>alt</code>	<code>texto</code>	Especifica un texto alternativo para las imágenes. Solo es aplicable para elementos <code><input></code> con <code>type="image"</code>
<code>autocomplete</code>	<code>on, off</code>	Especifica si un elemento <code><input></code> usa el autocompletado o no
<code>autofocus</code>	<code>autofocus</code>	Especifica que un elemento <code><input></code> debería solicitar el foco cuando se cargue la página
<code>checked</code>	<code>checked</code>	Especifica si un elemento <code><input></code> debería estar preseleccionado cuando se cargue la página. Solo es aplicable para elementos <code><input></code> con <code>type="checkbox"</code> o <code>type="radio"</code>

¹ Tabla tomada de la siguiente fuente: W3SCHOOLS, Tag Input [Internet]. Disponible en: http://www.w3schools.com/tags/tag_input.asp [Fecha de acceso: 05/06/2015].

<i>disabled</i>	<i>disabled</i>	Especifica si un elemento <input> debería estar deshabilitado
<i>form</i>	<i>form_id</i>	Especifica los formularios a los que pertenece el elemento <input>
<i>formaction</i>	<i>URL</i>	Especifica la URL del fichero que procesará el control <input> cuando el formulario se envíe. Solo es aplicable para elementos <input> con <i>type</i> =”submit” o <i>type</i> =”image”
<i>formenctype</i>	<i>application/x-www-form-urlencoded, multipart/form-data, text/plain</i>	Especifica cómo deberían ser codificados los datos del formulario cuando se envíe al servidor. Solo es aplicable para elementos <input> con <i>type</i> =”submit” o <i>type</i> =”image”
<i>formmethod</i>	<i>get, post</i>	Especifica el método HTTP usado para enviar los datos a la URL del «action». Solo es aplicable para elementos <input> <i>type</i> =”submit” o <i>type</i> =”image”
<i>formnovalidate</i>	<i>formnovalidate</i>	Especifica que los elementos del formulario no deberían validarse cuando se envía
<i>formtarget</i>	<i>_blank, _self, _parent, _top, framename</i>	Especifica dónde mostrar la respuesta recibida después de enviar el formulario. Solo es aplicable para elementos <input> <i>type</i> =”submit” o <i>type</i> =”image”
<i>height</i>	<i>pixels</i>	Especifica la altura de un elemento <input>. Solo es aplicable para elementos <i>input type</i> =”image”
<i>list</i>	<i>datalist_id</i>	Hace referencia a un elemento <datalist> que contiene opciones predefinidas para un elemento <input>

<i>max</i>	<i>number, date</i>	Especifica el valor máximo para un elemento <input>
<i>maxlength</i>	<i>number</i>	Especifica el máximo número de caracteres permitidos en un elemento <input>
<i>min</i>	<i>number, date</i>	Especifica el valor mínimo para un elemento <input>
<i>multiple</i>	<i>multiple</i>	Especifica si un usuario puede introducir más de un valor en un elemento <input>
<i>name</i>	<i>text</i>	Especifica el nombre de un elemento <input>
<i>pattern</i>	<i>regexp</i>	Especifica una expresión regular sobre la que es validado un elemento <input>
<i>placeholder</i>	<i>text</i>	Especifica un texto que describe el valor esperado de un elemento <input>
<i>readonly</i>	<i>readonly</i>	Especifica si un elemento <input> es de solo lectura
<i>required</i>	<i>required</i>	Especifica si un campo <input> debe llenarse antes de enviar el formulario
<i>size</i>	<i>number</i>	Especifica el ancho, en caracteres, de un elemento <input>
<i>src</i>	<i>URL</i>	Especifica la URL de la imagen para usar como botón de envío. Solo es aplicable para elementos <input> type="image"
<i>step</i>	<i>number</i>	Especifica el intervalo de suma para un elemento <input>

<i>type</i>	<i>button, checkbox, color, date, datetime, datetime-local, email, file, hidden, image, month, number, password, radio, range, reset, search, submit, tel, text, time, url, week</i>	Especifica el tipo de elemento <code><input></code> que se va a mostrar
<i>value</i>	<i>text</i>	Especifica el valor de un elemento <code><input></code>
<i>width</i>	<i>pixels</i>	Especifica el ancho de un elemento <code><input></code> . Solo es aplicable para elementos <code><input></code> type="image"

- `<TEXTAREA>`: usado para escribir texto más extenso empleando varias líneas. Se define en HTML de la siguiente forma:

```
<textarea cols="5" rows="6"></textarea>
```

Los atributos posibles para este elemento en HTML5 son los que mostramos en la siguiente tabla²:

Atributo	Valor	Descripción
<i>autofocus</i>	<i>autofocus</i>	Especifica que un área de texto de forma automática debe recibir todo el enfoque cuando se carga la página
<i>cols</i>	<i>number</i>	Especifica el ancho visible de un área de texto
<i>disabled</i>	<i>disabled</i>	Especifica que un área de texto se debe desactivar

²Tabla tomada de la siguiente fuente: W3SCHOOLS, Tag Textarea [Internet]. Disponible en: http://www.w3schools.com/tags/tag_textarea.asp [Fecha de acceso: 05/06/2015].

<i>form</i>	<i>form_id</i>	Especifica uno o más formularios a los que el área de texto pertenece
<i>maxlength</i>	<i>number</i>	Especifica el número máximo de caracteres permitido en el área de texto
<i>name</i>	<i>text</i>	Especifica un nombre para un área de texto
<i>placeholder</i>	<i>text</i>	Especifica un texto corto que describe el valor esperado de un área de texto
<i>readonly</i>	<i>readonly</i>	Especifica que un área de texto debe ser de solo lectura
<i>required</i>	<i>required</i>	Especifica que un área de texto debe ser llenada, ya que es requerido
<i>rows</i>	<i>number</i>	Especifica el número visible de líneas de un área de texto
<i>wrap</i>	<i>hard soft</i>	Especifica cómo será envuelto el texto de un área de texto cuando se presente en un formulario

b. Casillas de verificación

Permiten seleccionar opciones de entre una lista de ellas. Es común llamarlas con su término en inglés: «checkbox». Como dato importante, cabe recalcar que solo se enviarán al servidor aquellas opciones seleccionadas y su valor por defecto será «on». Se definen de la siguiente forma:

```
<input type="checkbox" name="vehiculo" value="bici"> Tengo una bici
<input type="checkbox" name="vehiculo" value="coche"> Tengo un coche
```

<input type="checkbox"/>	Tengo una bici
<input type="checkbox"/>	Tengo un coche

c. Radios o grupos de acciones

Con este tipo de controles el usuario puede seleccionar una y solo una de las opciones presentadas. Se crean de la siguiente forma, teniendo en cuenta que deben tener el mismo valor para el atributo «*name*» para que pertenezcan al mismo grupo:

```
<input type="radio" name="genero" value="hombre" /> Hombre  
<input type="radio" name="genero" value="mujer" /> Mujer
```

Hombre Mujer

No se pueden aplicar atributos específicos además de los genéricos ya comentados. Explicaremos su uso en los formularios en el apartado 2.3.4.

d. Etiquetas

Sirven para asociar un texto a un control. En *radios* o *checkbox* se suelen utilizar para seleccionar la opción al hacer clic sobre ellos.

- <*LABEL*>: se utiliza para establecer un nombre para un campo. No tiene efecto sobre el formulario, y se usa para añadir estilos a los controles del formulario y como herramienta de accesibilidad. Se aplica junto al parámetro «*for*» del siguiente modo:

```
<label for="correo">Mail</label>  
<input id="correo" name="correo" type="text">
```

Su valor debe ser el mismo que el valor «*id*» del campo, por lo que nos obliga a añadir un valor «*id*» a cada elemento.

Los atributos posibles para este elemento en HTML5 se enumeran en la siguiente tabla³:

³Tabla obtenida de la siguiente fuente: W3SCHOOLS, Tag Textarea [Internet]. Disponible en: http://www.w3schools.com/tags/tag_textarea.asp [Fecha de acceso: 05/06/2015].

Atributo	Valor	Descripción
<i>for</i>	<i>element_id</i>	Especifica a qué elemento del formulario está enlazado el elemento <label>
<i>form</i>	<i>form_id</i>	Especifica los formularios a los que pertenece el elemento <textarea>

- <*FIELDSET*>: es utilizado para agrupar secciones de un mismo tipo dentro un formulario cuando este es demasiado largo. Mostraría estas secciones con una línea alrededor. Se define de la siguiente forma:

```
<fieldset><input type="text"></fieldset>
```

Los atributos posibles para este elemento en HTML5 se enumeran en la siguiente tabla⁴:

Atributo	Valor	Descripción
<i>disabled</i>	<i>disabled</i>	Especifica si un grupo de elementos de un formulario debería estar deshabilitado
<i>form</i>	<i>form_id</i>	Especifica los formularios a los que pertenece el elemento <fieldset>
<i>name</i>	<i>text</i>	Especifica un nombre para el elemento

- <*LEGEND*>: permite etiquetar con un texto cada una de las secciones delimitadas por el control <*FIELDSET*>. No existe ningún posible atributo que se pueda aplicar.

A continuación se muestra un ejemplo de la utilidad de <*FIELDSET*> y de <*LEGEND*>:

The screenshot shows a simple HTML form. At the top, there is a label "Nombre del Usuario" followed by a text input field. Below this, there is a legend box with the text "Nombre:" and a text input field next to it. At the bottom of the legend box is a button labeled "Enviar".

⁴Tabla obtenida de la siguiente fuente: W3SCHOOLS, Tag Fieldset [Internet]. Disponible en: http://www.w3schools.com/tags/tag_fieldset.asp [Fecha de acceso: 05/06/2015].

Y el código HTML usado para generarla:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin título</title>
</head>
<body>
<fieldset>
<legend>Nombre del Usuario</legend>
<form action="obtenerdatos.php" method="post">
    Nombre:
    <input type="text" name="nombre" value="" />
    <br/>
    <input type="submit" value="Enviar" />
</form>
</fieldset>
</body>
</html>
```

e. Menús de selección

Son usados para seleccionar uno o varios valores de una lista desplegada.

- **<SELECT>**: se aplica para seleccionar uno o varios valores de una lista predefinida. Se define de la forma expuesta a continuación, donde cada elemento **<OPTION>** es una opción seleccionable:

```
<select> <option value="valor">Opción 1</option></select>
```

Los atributos posibles para este elemento en HTML5 se enumeran en la siguiente tabla⁵:

⁵ Tabla tomada de la siguiente fuente: W3SCHOOLS, Tag Select [Internet]. Disponible en: http://www.w3schools.com/tags/tag_select.asp [Fecha de acceso: 05/06/2015].

Atributo	Valor	Descripción
<i>autofocus</i>	<i>autofocus</i>	Especifica si la lista desplegable debería obtener el foco automáticamente cuando se carga la página
<i>disabled</i>	<i>disabled</i>	Especifica si la lista desplegable debería estar deshabilitada
<i>form</i>	<i>form_id</i>	Especifica los formularios a los que pertenece el elemento <select>
<i>multiple</i>	<i>multiple</i>	Especifica si se pueden seleccionar varias opciones al mismo tiempo
<i>name</i>	<i>name</i>	Define un nombre para la lista desplegable
<i>required</i>	<i>required</i>	Especifica si un elemento <input> debe llenarse antes de enviar el formulario
<i>size</i>	<i>number</i>	Especifica el ancho, en caracteres, de un elemento <select>

Su uso en una página web se detallará en el apartado 1.3.3.

f. Botones de acción

Los botones se usan para realizar algún tipo de acción que puede estar ya definida por el propio HTML, como el envío de un formulario o el borrado de todos sus campos, o puede definirse con ayuda de funciones de *script* usando, por ejemplo JavaScript.

Un botón de acción es un elemento <INPUT> cuyo valor del atributo «*type*» puede ser «*submit*», «*reset*» o «*button*». En el capítulo 1.3.2 explicaremos más detalladamente su uso.

g. Subir archivos

Si en un formulario se necesita subir archivos al servidor, existe un tipo de <INPUT> para ello. Se define de la siguiente forma:

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

Lo que mostrará es un botón que, al pulsar, generará un diálogo de selección para elegir el o los archivos que deseemos subir. A lo largo de este manual se muestran ejemplos de esta funcionalidad.

1.3.2. Utilización de botones de acción

Los botones de acción son controles tipo <INPUT> donde el atributo «*type*» puede tener el valor «*submit*», «*reset*» o «*button*». Según este atributo realizarán una determinada acción.

En un formulario destaca principalmente el botón de envío, que es un elemento <INPUT> cuyo atributo «*type*» posee el valor «*submit*», que por lo general es el encargado de enviar los datos recogidos en el formulario. Además puede tener otros valores que explicaremos a continuación:

- *Acción «submit»:* un campo de tipo *submit* generará un botón que, al ser pulsado, hará que se aplique la acción indicada en el formulario, es decir, se aplicará el atributo *action* del elemento <FORM> en el que está definido dicho campo *submit*.

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

Esto se podría aplicar en un típico formulario de contacto:

Formulario de Contacto

Nombre:

Email:

Telefono:

Mensaje:

The diagram shows a contact form with four input fields for Name, Email, Phone, and Message. Below the fields is a button labeled "Enviar". The "Enviar" button is highlighted with a red border, indicating it is the focus of the discussion.

- *Acción «reset»:* un campo de tipo «*reset*» eliminará todo el contenido introducido en un formulario dejándolo limpio de nuevo. No se realiza ninguna petición al servidor, ya que la limpieza de campos se lleva a cabo únicamente en el navegador del usuario.

```
<input type="reset" value="Borrar formulario" />
```

- *Ejecución de acciones:* un campo tipo «*button*» permite ejecutar acciones definidas en otros lenguajes de *script* como JavaScript.

En el siguiente ejemplo, «*confirmarBorrado(this)*» es una función definida en JavaScript cuyo objeto está fuera del alcance de esta unidad formativa.

```
<input name="eliminar" id="eliminar" type="button" value="Eliminar" onClick="confirmBorrado(this)">
```

1.3.3. Utilización de listas desplegables

Las listas desplegables ya las hemos visto anteriormente cuando hablamos de los *menús de selección*. En este punto explicaremos más pormenorizadamente su uso, ya que se trata de un elemento muy importante en la construcción del formularios, y de cierta complejidad.

El control <SELECT> permite seleccionar una o varias opciones predefinidas de una lista desplegable.

```
<label>Habitaciones:</label><br>
<select name="habitaciones">
    <option value="indiferente">indiferente</option>
    <option value="">Todos</option>
    <option value="1">1 Habitaciones</option>
    <option value="2">2 Habitaciones</option>
    <option value="3">3 Habitaciones</option>
</select><br>
<input type="submit" value="Buscar" />
```

Su apariencia en el navegador será:

Habitaciones:
<input type="text" value="indiferente"/>
Precio máximo:
<input type="text" value="indiferente"/>
<input type="button" value="Buscar"/>

Con el atributo «*multiple="multiple"*» se puede seleccionar más de una opción de las mostradas pulsando la tecla «Ctrl».

Si se quiere que de forma predeterminada el valor seleccionado sea diferente del primero, podemos añadir el parámetro «*selected*» al elemento de la siguiente forma:

```
<option value="2" selected>2 Habitaciones</option>
```

La etiqueta «*optgroup*» proporciona una variante de los campos de tipo «*option*», pero con la diferencia de que se pueden agrupar las opciones con subtítulos. Por ejemplo:

```
<label>Povincia:</label>
<select>
<optgroup label="Galicia">
<option value="1">A coruña</option>
<option value="2">Lugo</option>
<option value="3" selected >Ourense</option>
<option value="4">Pontevedra</option>
</optgroup>
<optgroup label="Extremadura">
<option value="5">Cáceres</option>
<option value="6">Badajoz</option>
</optgroup>
</select>
```

Este código HTML se mostrará en el navegador de la siguiente forma:

A screenshot of a dropdown menu from a web browser. The menu is titled "Povincia:" and lists the following options:

- Cáceres
- Galicia
 - A coruña
 - Lugo
 - Ourense
 - Pontevedra
- Extremadura
 - Cáceres
 - Badajoz

The option "Cáceres" under "Extremadura" is highlighted with a blue background.

Como se ha visto, el uso del elemento <SELECT> permite personalizar la apariencia del control para facilitar su interacción por parte del usuario. Independientemente de su apariencia, al servidor solo se enviará el valor del <OPTION> seleccionado.

1.3.4. Utilización de casillas de verificación

Se trata de un control tipo <input> que le da al atributo «*type*» el valor «checkbox». Permite marcar con el ratón una o varias opciones dentro de un grupo, y para que las diferentes opciones formen parte de un grupo deberán tener el mismo nombre en el atributo «*name*».

Por ejemplo, el siguiente extracto de código:

```
<label>Selecciona tus bebidas:</label>
<input type="checkbox" name="bebida" value="Cocacola" /> Cocacola <br />
<input type="checkbox" name="bebida" value="Agua" /> Agua <br />
<input type="checkbox" name="bebida" value="Cerveza" /> Cerveza <br />
```

Mostrará lo siguiente en un navegador:

A screenshot of a checkbox group from a web browser. The text "Selecciona tus bebidas:" is displayed above three checkboxes:

- CocaCola
- Agua
- Cerveza