

ISSD

I2

Informática 2

Módulo didáctico - 2015

—*Asc*—
Analista de
Sistemas

UNA HERRAMIENTA
ESENCIAL:
LOS FORMULARIOS



Unidad 1 y 2



Clase 7



(U1) HTML

(U2) CSS

En este clase...

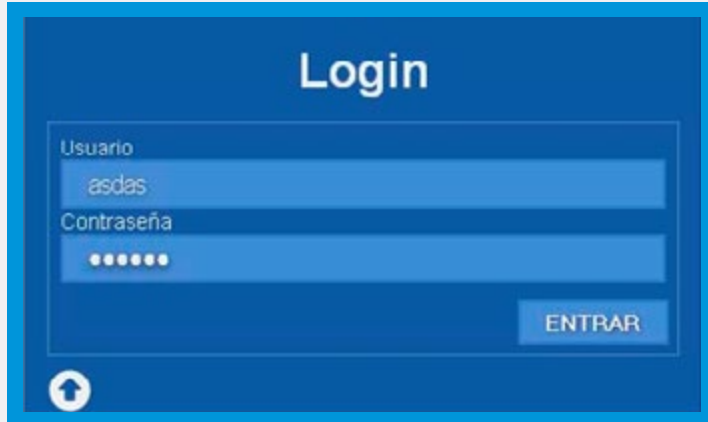
| Comprenderás por qué es tan importante el uso y el desarrollo de formularios en HTML.

| Conocerás los distintos controles de formulario como así también el uso de sus propiedades.

| Entenderás los tópicos dados hasta el momento codificando los distintos ejercicios expuestos y ejemplificados en las clases.



Si sos un asiduo comprador de productos online, o bien alguna vez te tocó consultar homebanking o realizar el pago de alguna cuenta vía débito, habrás visto que siempre debés ingresar tus datos personales o suscribirte al sitio para poder tener acceso al mismo. Te habrás preguntado ¿a dónde van a parar mis datos? ¿qué hacen con ellos? Para responder a todas tus inquietudes te invito a participar de la lectura de esta clase.



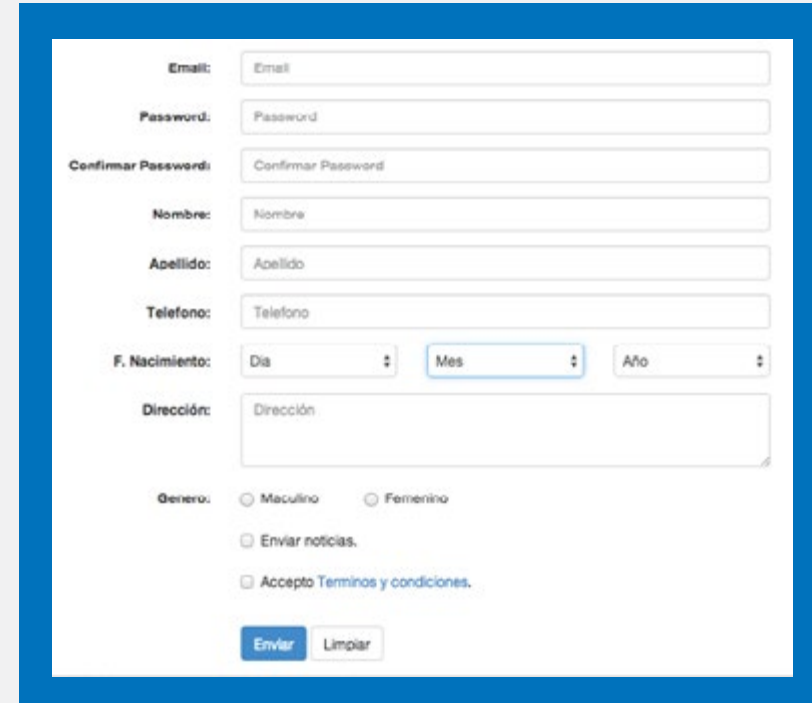
Login

Usuario
asdas

Contraseña
•••••

ENTRAR

⬆



Email:

Password:

Confirmar Password:

Nombre:

Apellido:

Telefono:

F. Nacimiento: Dia Mes Año

Dirección:

Genero: ☐ Masculino ☐ Femenino

☐ Enviar noticias.

☐ Accepto Terminos y condiciones.

Enviar **Limpiar**



Una herramienta esencial: los formularios

¡Bienvenido a una nueva clase! En los módulos anteriores vimos las partes fundamentales de HTML y todo lo referido a la estructura visual, lo que seguramente te sirvió para comenzar a defenderte y poder armar una intro o bien una index. Ahora pasaremos a la siguiente estructura HTML, que son los formularios. Comenzaremos a ver:

| Controles básicos y atributos propios de HTML5.

| Distintos aspectos de validación que poseen los controles para que se ingresen correctamente los datos y no se comentan errores.

Como siempre te digo, no dejes de repasar el código, esta materia sólo se aprende codificando los ejemplos y ejercicios de cada una de las clases, no es necesario que los desarrolles a todos, eso dependerá de ti y del tiempo con el que cuentes para esta actividad.

¡Mucha suerte!



Desempeño de Síntesis

En este desempeño deberás interpretar, analizar y codificar de forma individual los tópicos desarrollados hasta el momento. De esta manera se evaluarán los siguientes ítems:

1 | Conceptos, investigación y relación:

- | Tópicos comprendidos hasta el momento.
- | Incremento del conocimiento y la comprensión de los tópicos investigando o interactuando con compañeros.
- | Relación de los conceptos con otras materias en orden vertical u horizontal.

2 | Aspectos técnicos:

- | Codificación clara, uso de tags, propiedades y atributos correctos, legibilidad en la codificación.
- | Maquetación y uso de herramientas adicionales.



Criterios de valoración

- (10%) Maquetación y estética correcta
- (50%) HTML
- (40%) CSS3

Los formularios

La Web se ha convertido en un arma poderosa para las empresas que se dedican a realizar encuestas o compras, los formularios han sido una de las herramientas que han ayudado a este auge.

¿Para qué sirven los formularios?

Los formularios nos van a permitir, desde dentro de una presentación web, solicitar información al visitante. Estarán compuestos por tantos campos como informaciones queramos obtener. Una vez introducidos los valores en estos campos serán enviados a una URL donde se procesará toda esta información.

Los formularios permiten obtener información de un cliente que visita una Web. Utilizando esta herramienta un usuario puede enviar datos para que luego sean procesados en el servidor.

Un formulario contiene dos tipos de elementos: campos de datos y control. Los primeros permiten al usuario introducir información y los segundos indicar qué debe hacerse con esos datos. Los datos son ingresados por el usuario y el control es un proceso que en ciertas ocasiones se realiza para permitir procesar dicha información.

El formulario no es más que una página escrita en **HTML** como cualquier otra. Lo

que la hace diferente es que permite capturar datos de varias maneras: directamente desde el teclado, escogiendo un valor de una lista desplegable o seleccionando una opción desde botones.

Una vez terminada la captura de datos, estos forman un paquete de información, que es enviado a un servidor http (web) que lo procesará y devolverá una respuesta. Pueden utilizarse también para enviar la información por correo electrónico o como un archivo html.

En el caso de que el resultado de un formulario sea un correo o un html, es sencillo de manejar. La complicación se produce

cuando la información debe ser procesada en el servidor, porque debe existir un programa que sepa cómo manejar la información de ése formulario. Es decir, se debe hacer un programa para que sirva de puente entre el formulario y el servidor, normalmente estas aplicaciones se denominan **CGI** (Common Gateway Interface). Los programas **CGI** pueden estar escritos en cualquier lenguaje de programación que sea soportado por el sistema operativo del servidor, y pueden estar diseñados para cualquier función: desde una simple captura de datos que serán guardados en un archivo, hasta la más sofisticada consulta a una base de datos.

Formulario Básico

Todo formulario debe estar encerrado entre el par de etiquetas **<FORM>** y **</FORM>** y debe ser ubicado en el cuerpo de cualquier documento **HTML**, es decir, entre el par de etiquetas **<BODY>** y **</BODY>**. Esta etiqueta **<FORM>** presenta tres atributos posibles (1).

Los dos primeros atributos de la tabla son de uso obligatorio para cualquier formulario que generemos, ya que establecen dónde enviar la información y cómo enviarla. Cuando definimos un formulario lo hacemos de la siguiente manera:

```
<form> controles del formulario... </form>
```

Destino de la información

El formulario comienza con esta línea:

```
<FORM ACTION="mailto:alumno@issd.edu.ar" METHOD=POST>
```

Como dijimos anteriormente todo formulario debe comenzar con la etiqueta **<FORM>** y finalizar con **</FORM>**. El parámetro **ACTION** define qué es lo que debemos hacer con la información obtenida, en este caso le indicamos al navegador que nos envíe los datos a nuestra dirección de correo electrónico (en el ejemplo la mandan a issd). La mayoría de los formularios en Internet realizan esta tarea, es decir, recolectan información del usuario y la envían por correo electrónico hacia algún destino.

Comando	Descripción
ACTION	El valor de este parámetro es la URL del programa o guion en el Servidor Web utilizado para procesar la información recolectada.
METHOD	Puede asumir el valor GET o el valor POST, y definen la manera en la cual los datos son transferidos al servidor.

Controles de Formularios

Todos los controles que pueden usarse a la hora de crear un formulario permitirán al navegante insertar información. En algunas oportunidades esta información consistirá en la inserción de un texto, que podrá ser de una sola línea o varias y que podrá estar oculto o no. En otras ocasiones se ofrecerá entre una serie de opciones entre las que simplemente hay que elegir una o varias.

Todo ello se podrá realizar de diferentes formas que ahora mismo comenzaremos a estudiar. Para empezar vamos a ver una serie de controles todos los cuales se crean con la etiqueta **INPUT**. Para diferenciar entre unos controles y otros se hará uso del atributo **TYPE** que puede tomar los siguientes valores: text, password, radio, checkbox, submit, image, reset, file y hidden.

Además del atributo **TYPE**, esta etiqueta consta de algunos atributos más cuya función variará respecto al tipo de control que estemos insertando, es decir, en función del valor del atributo TYPE. Uno común a todos es **NAME** -ya mencionado- con el que damos un nombre al control, y el que te recomendamos usar siempre. Una vez hecho

este comentario, podemos empezar a estudiar cada uno de los tipos de control uno a uno.

Cuadro de texto

Esta misma etiqueta, acompañada de TYPE, será la utilizada para insertar la mayoría de los controles, por ejemplo:

```
<INPUT TYPE="text" SIZE=15  
MAXLENGTH=15 NAME="Nombre">
```

| Size: Determina la anchura de la caja de texto. El valor por defecto de este atributo es 20 caracteres. En ocasiones convendrá decrementar o incrementar este valor por defecto, pero en todo caso siempre será conveniente mantener un valor menor de 50 caracteres de manera que la caja de texto quepa en la mayoría de pantallas

| Maxlength: Con este atributo limitamos el número máximo de caracteres que pueden ser escritos en una caja de texto. El valor de este atributo puede ser mayor o menor que el especificado en SIZE, y es totalmente

¿Cuáles son
los elementos
que contiene un
formulario?



Los campos de datos
y los controles: los
primeros permiten
al usuario introducir
información y los
segundos indican qué
debe hacerse con esos
datos.





```
ejPlaceholder.html
<form id="formulario">
  <label for="user">Usuario:</label>
  <input type="text" placeholder="Solo numeros y letras">
</form>
```


independiente. Si es mayor cuando lleguemos al final de la caja de texto, los nuevos caracteres que insertemos irán desplazando hacia la izquierda a los primeros, que dejarán de estar a la vista.

```
<INPUT TYPE="text" SIZE="20"
MAXLENGTH="20">
<INPUT TYPE="text" SIZE="20"
MAXLENGTH="5">
<INPUT TYPE="text" SIZE="20"
MAXLENGTH="40">
```

Por último queda decir que si no usamos el atributo MAXLENGTH, el número de caracteres que pueden introducirse en la caja de texto no tendrá límite.

| Value: Sirve para especificar un texto que debe aparecer por defecto en la caja de texto, antes de que el usuario escriba nada. Este texto suele ser, en general, o bien instrucciones o bien la respuesta más probable. Veamos un ejemplo que ilustra estos dos casos:

```
<INPUT TYPE="text" VALUE="Introduzca aquí
su nombre">
```

| Name: define el nombre que se le da al control, esto puede servir para que el lenguaje dinámico lo tome como lectura y lo pueda manipular como propio.

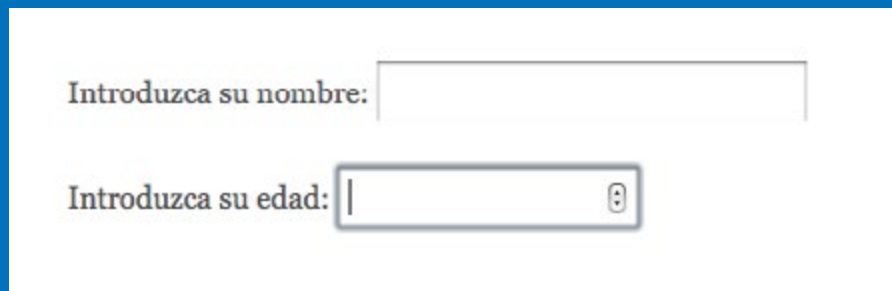
| Placeholder: Un truco común de la usabilidad en formularios web es tener

contenido de relleno en los campos de entrada de texto. Por ejemplo, para dar más datos sobre el tipo esperado de información que queremos que el usuario introduzca, que desaparece cuando el control de formulario adquiere el foco (1).

```
<input type="text"...
placeholder="Tunombres..">
```

| Autofocus: este atributo hace que un campo de formulario adquiera el foco de forma automática cuando se carga una página. Su sintaxis es la siguiente:

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

Introduzca su nombre:

Introduzca su edad:

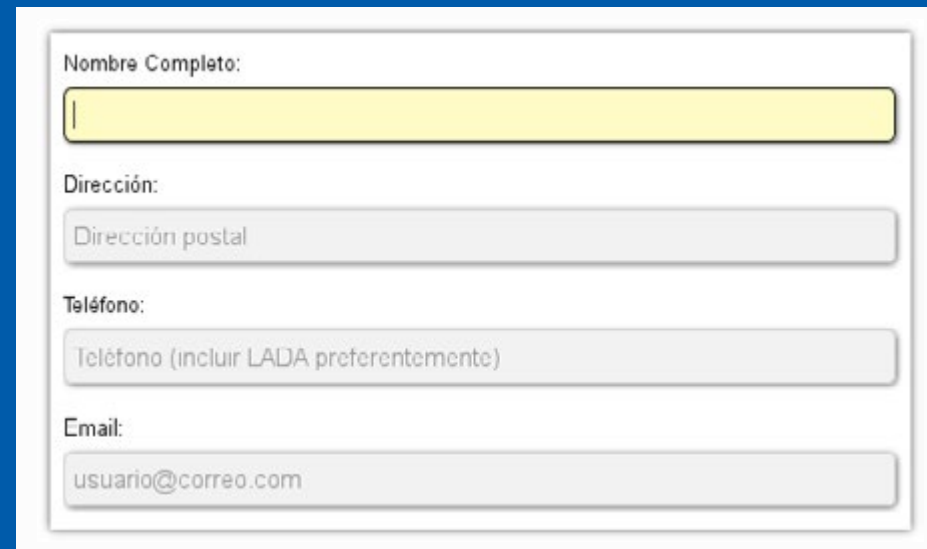
Como vemos en la imagen 1 estamos poniendo el cursor de manera automática en un input perteneciente a la edad.

Tené en cuenta que no deberías tener más de un control de formulario **autofocus** en una página. Además, hay que utilizar este tipo de funcionalidad con precaución, en situaciones en que el formulario representa la principal área de interés en una página (2)

Una página de búsqueda es un buen ejemplo: si no hay una gran cantidad de contenido y texto explicativo, tiene sentido dar foco automáticamente a la entrada de texto

del formulario de búsqueda. El atributo "autofocus" es de tipo booleano (o es verdadero o es falso). Hay varias maneras de dirigir el foco hacia un elemento:

- | Designando el elemento con el mouse.
- | Navegando de un elemento a otro con el teclado. El diseñador del formulario puede definir un **orden de tabulación** que especifique el orden en que se dirige el foco hacia los elementos si el usuario navega por el documento. Una vez seleccionado, el elemento puede activarse con alguna otra secuencia de teclas.



Nombre Completo:

Dirección:

Teléfono:

Email:

- | Seleccionar un elemento por medio de una tecla de acceso ("hotkey").

Navegación con tabulador

| **tabindex** = Este atributo especifica la posición del elemento actual dentro del orden de tabulación del documento actual. Este valor debe ser un número entre 0 y 32767.

Los agentes de usuario deberían navegar por los elementos a los que puede dirigirse el foco de acuerdo con las siguientes reglas:

1 | Navegar en primer lugar por aquellos elementos que soporten el atributo **tabindex** y tengan asignado para éste un valor positivo. La navegación se produce desde el elemento con menor valor de **tabindex** hasta el elemento con el valor más alto. Los valores no necesitan ser secuenciales ni deben comenzar por un valor en particular. Si hay elementos que tengan valores idénticos de **tabindex** debería navegarse por ellos según el orden en que aparezcan en el flujo de caracteres.

2 | Los elementos que estén deshabilitados no participan en el orden de tabulación. Los siguientes elementos soportan el atributo **tabindex**: **A**, **AREA**, **BUTTON**, **INPUT**, **OBJECT**, **SELECT** y **TEXTAREA**.

```
<!DOCTYPE HTML >
<HEAD>
<TITLE>Undocumento con FORM</TITLE>
</HEAD>
<BODY>
...texto...
<P>Ir al <A tabindex="10" href="#">Link 1</A>
...mástexto...
<BUTTON type="button" name="bton1"
```

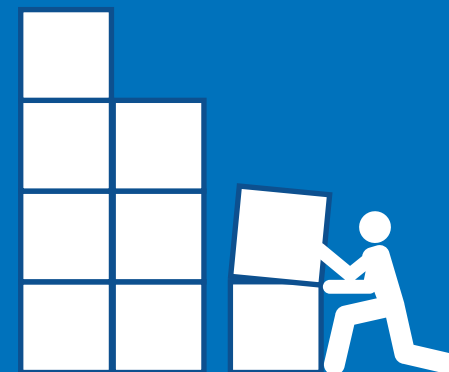
```
tabindex="1" onclick="datos ">
Obtener Datos </BUTTON>
...más texto...
<FORM action="..." method="post">
<P>
<INPUT tabindex="1" type="text"
name="campo1">
<INPUT tabindex="2" type="text"
name="campo2">
<INPUT tabindex="3" type="submit"
name="enviar">
</P>
</FORM>
</BODY></HTML>
```

Cuadro de texto para claves

En ocasiones puede ser conveniente pedir al usuario algún tipo de información confidencial como puede ser una clave de acceso o password.

Debemos tener en cuenta que algunos usuarios pueden estar en algún lugar público o, al menos, con gente alrededor y por tanto no le interesará que lo que escribe sea mostrado en la pantalla.

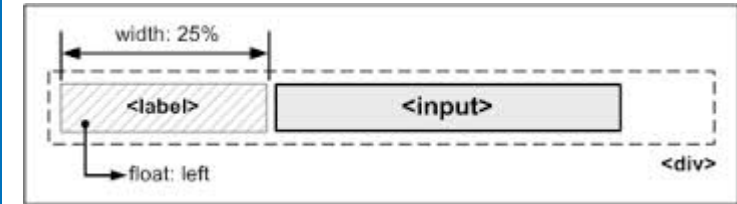
El orden de tabulación define el orden en que el foco se dirige hacia los elementos cuando se navega por medio del teclado. El orden de tabulación puede incluir elementos anidados en otros elementos.



Rellene los campos del siguiente formulario con los valores deseados para generar los archivos `.htaccess` y `.htpasswd`:

Usuario:

Contraseña:



`<H2>Formulario de autenticación</H2>`
`FORM>`

`<P>Introduzca su nombre: <INPUT`
`TYPE="text" NAME="nombre">`

`<P>Introduzca su clave: <INPUT`
`TYPE="password" NAME="clave">`

`</FORM>`

Los cuadros de texto para claves también tienen los atributos **SIZE**, **MAXLENGTH**, **VALUE** y **ALIGN** que vimos en las cajas de texto. Al igual que hicimos con estas, veamos un resumen de todos los atributos de las cajas de texto para claves. Observá el código y luego la imagen 1.

`<INPUT TYPE="password" NAME="clave"`
`SIZE="tamaño" MAXLENGTH="longitud_`
`máxima" VALUE="clave por defecto"`
`ALIGN="alineamiento">`

Etiquetar controles

En el caso de querer incluir una etiqueta a un elemento, utilizamos la etiqueta `<LABEL>... </LABEL>`. Esta no sólo permite darle texto a un elemento de un formulario, sino que comparte el control del mismo. Esto quiere decir que podemos seleccionar no sólo al elemento sino que también a su rótulo. Mirá el código y luego la imagen 2.

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

Botones

Un botón nos permite realizar una acción con el formulario que lo contenga. Existen dos tipos básicos de botones: enviar el formulario (envío o **submit**) y poner todos los campos a sus valores iniciales (restablecimiento o **reset**). Además de estos dos tipos de botones, existe un botón especial que no posee ninguna acción predefinida y que, por tanto, deberá ser especificada con un lenguaje del lado del cliente. Los botones pueden insertarse a través de la etiqueta (tag) HTML

input (botones para enviar y restablecer y botones de imagen) o el elemento HTML button (botones de contenido). A continuación veremos cada uno de ellos

Botón de Envío

Este control envía automáticamente el formulario en el que se encuentra, cuando es presionado. Es decir, cuando se pulsa el botón el navegador nos lleva a la dirección web indicada en el campo **action** y al mismo tiempo envía a esa dirección los datos que contiene el formulario. Para insertar un botón usamos la etiqueta `<input />`. Para indicar que se trata de un botón de envío dentro de esta etiqueta escribiremos **type = "submit" (1)**.


Botón de borrado

Este tipo de botones reestablecen los controles en un formulario a sus valores iniciales cuando es presionado. Son insertados con la etiqueta (tag) HTML input con el valor "reset" en su atributo "type".

```
<!DOCTYPE HTML>
<body>
<form method="post" action="enviar.php">
<input type="text" name="texto1" value="Valor
pordefecto" /><br />
<input type="checkbox" name="condiciones"
checked="checked" />
Acepto las condiciones<br />
<input type="reset" value="Limpiar campos" />
</form>
</body>
</html>
```


Con el código que se muestra a continuación veremos el uso de ambos botones (submit y reset)

```
<H2> Formulario de identificación</H2>
<FORM>
<P>Introduzca su nombre:
<INPUT TYPE="text" NAME="nombre"
VALUE="IntroduzcasuNombre" SIZE="30">
<P>Sexo:
<INPUT TYPE="RADIO" NAME="sexo"
VALUE="hombre"
CHECKED>Hombre
<INPUT TYPE="RADIO" NAME="sexo"
VALUE="mujer">Mujer
```



¡Ahora te toca a vos escribir este código y ver cuál es el resultado!

```
<!DOCTYPEHTML>
<body>
<formmethod="post" action="enviar.
php">
<input type="submit"
value="Enviaresteformulario" />
</form>
</body>
</html>
```



Lo que ocurrirá al pulsar sobre el botón 'Enviar este formulario' es que el navegador irá a la página 'enviar.php' y recibirá los datos que pudiera llevar el formulario.

```
<P>Comentarios:
<INPUT TYPE="text" NAME="comentarios"
SIZE="40">
<P>
    <INPUT TYPE="submit"
VALUE="Enviar">
    <INPUT TYPE="reset"
VALUE="Restablecer">
</FORM>
```

Botones de imagen

Estos tipos de botones funcionan exactamente como los botones de envío, salvo que los de imagen son representados visualmente con la imagen especificada en el atributo **"src"**. Además, estos tipos de botones envían las coordenadas donde ha ocurrido el click cuando el formulario fue enviado, tomadas desde la esquina superior izquierda del botón (por ejemplo, para un botón de imagen llamado "boton1" las coordenadas serán enviadas en la forma "boton1.x" y "boton1.y"). Los botones de imagen se insertan con el tag HTML input, usando el valor "image" en su atributo "type".

```
<!DOCTYPE HTML >
<body>
<form method="get" action="enviar.php">
<input name="boton1"
type="image"src="Buscar.png">
</form>
</body>
</html>
```

Botones de contenido o genéricos

Los botones de contenido pueden ser usados como botones de envío o restablecimiento, o bien pueden no tener ninguna acción preestablecida (dependiendo del valor de su atributo "type"). Su característica es que se puede insertar contenido HTML dentro de ellos.

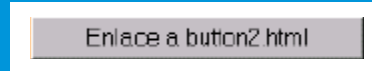
Los botones de contenido se insertan con el tag HTML button, usando el valor "button", "submit" o "reset" en su atributo "type". Un valor button supondrá que el botón no hará nada en respuesta a un click, a no ser que se especifique una acción de otra manera.

Entonces, existen dos tipos básicos de botones: enviar el formulario (envío o submit) y poner todos los campos a sus valores iniciales (restablecimiento o reset). Además de estos dos tipos de botones, existe un botón especial que no posee ninguna acción predefinida y que, por lo tanto, deberá ser especificada con un lenguaje del lado del cliente.

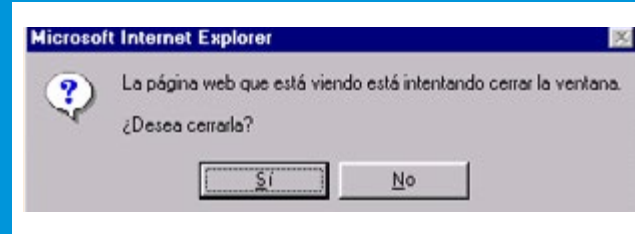
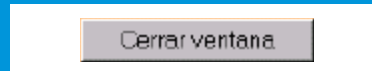


Botones genéricos

Ejemplo 1: Botón para cerrar una ventana



Ejemplo 2: Botón enlace



```
<INPUT TYPE="button"
VALUE="Cargarotraventana"
onClick="window.location.replace
('button2.html');">
```



```
<!DOCTYPE HTML >
<body>
<form method="post" action="enviar.php">
```

```
<button type="submit">
<b>Boton de Tipo Submit</b>
</button>
</form>
</body>
</html>
```

Como podemos observar, hemos introducido contenido HTML dentro del propio botón; en concreto hemos usado la etiqueta `` para poner texto en negrita y la etiqueta `
` para introducir un salto de línea.

A su vez hemos escrito `<buttontype="submit">`, lo que supondrá que el botón funcione como un botón de envío. Si hubiéramos escrito `<buttontype="reset">` el botón funcionaría como un botón de restablecimiento de valores por defecto, y si hubiéramos escrito `<buttontype="button">` el botón no daría lugar a ninguna acción. En caso de no especificarse `type` para el botón,

en general se entenderá que es un botón tipo submit.

Otro ejemplo del uso de estos botones es asignándole código JavaScript, que en este caso permitirá cerrar la ventana del navegador. Observa el código y luego mira la imagen 1.

```
<INPUT TYPE="button"
VALUE="Cerrarventana"
onClick="window.close();">
```

Campo Imagen

El control de tipo imagen inserta una imagen que funciona como un botón (aunque los navegadores no le dan relieve como a los botones). Al hacer clic en un punto de la imagen se envía el formulario (como si se hubiera pulsado un botón submit). Su sintaxis es la siguiente y su aspecto puede verse en la imagen 1:

```
<html>
<head>
<title>Formulario</title>
</head>
<body>
<form action="enviar.php" method="post">
Nombre: <input type="text"
name="nombre"><br>
Contraseña: <input type="password"
name="contra"><br><br>
<input type="image" src="enviar.png">
</form>
</body>
</html>
```

Botones de elección

Estos controles reciben también el nombre de **botones de radio**, como traducción directa de su denominación *inglesa* **radio buttons** y porque en general los navegadores suelen darles una forma circular (imagen 2). Para insertar un botón de elección usaremos de nuevo la etiqueta INPUT, pero esta vez el valor del atributo TYPE será radio:

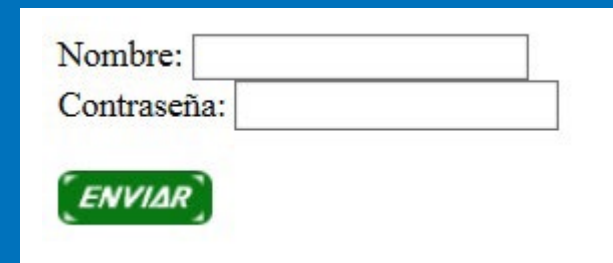
`<INPUT TYPE="radio">`

Este tipo de controles tiene dos estados o posiciones: seleccionado (ON) o no seleccionado (OFF), estando inicialmente todos en la posición **OFF**. También es posible especificar que un botón determinado esté seleccionado inicialmente insertando el atributo **CHECKED** en la etiqueta **INPUT** de la siguiente manera:

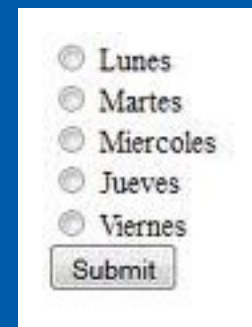
`<INPUT TYPE="radio" CHECKED>`

Los botones de elección suelen ser insertados en los formularios en grupos, dando al usuario la posibilidad de elegir entre una serie de opciones. Cuando pulsamos sobre un botón de radio, lo pasamos a la posición ON y permanecerá en ese estado hasta que

1



2



`<INPUT TYPE="radio">`



`<INPUT TYPE="radio" CHECKED>`



pulsemos en otra opción del mismo grupo. Esto es así porque entre los botones de radio de un mismo grupo sólo uno de ellos puede estar seleccionado, por tanto cuando seleccionamos uno, aquel que estuviese seleccionado previamente dejará de estarlo.

Para indicar que una serie de botones de elección pertenecen a un mismo grupo debemos incluir el mismo valor en el atributo **NAME** en todos ellos. Además debemos usar el atributo **VALUE** para dar un nombre distinto a cada uno de los botones. Veamos un ejemplo: (1)

Indique el tipo de música que más le guste:

<FORM>

<P><INPUT TYPE="radio" NAME="musica" VALUE="Jazz">Jazz

<P><INPUT TYPE="radio" NAME="musica" VALUE="Pop">Pop

<P><INPUT TYPE="radio" NAME="musica" VALUE="Rock">Rock

<P><INPUT TYPE="radio" NAME="musica" VALUE="Country">Country

</FORM>

Cuadro de selección

Las cajas de selección guardan ciertos parecidos con los botones de radio, pero además permitirán seleccionar varias opciones en una lista, tal como lo muestra la imagen 2. Al igual que dichos botones, tienen dos posiciones: seleccionados o no seleccionados, estando en esta última posición inicialmente (a no ser que hayamos usado el atributo CHECKED). Cada caja de selección es independiente del resto, y por lo tanto, el valor del atributo NAME debe ser diferente en cada una. Para insertar una caja de selección debemos usar de nuevo la etiqueta INPUT, pero esta vez con TYPE="checkbox". Veamos un ejemplo de uso de este tipo de controles:

Indique su profesión (escoja todas las que procedan):

<FORM>

<P><INPUT TYPE="checkbox" NAME="medico">Médico

<P><INPUT TYPE="checkbox" NAME="programador">

- ☐ Jazz
- ☐ Pop
- ☐ Rock
- ☐ Country

2

- ☐ Médico
- ☒ Programador
- ☐ Abogado
- ☐ Ingeniero



```
<form>
  <input type="number"
    min="0"
    max="10"
    step="2"
    value="6">
```

* ! #



¿Para qué usaría el par de atributos min y max?

Por ejemplo, en un formulario de reserva de viajes podrías limitar el selector de fechas para que sólo permita al usuario seleccionar fechas futuras.



CHECKED>Programador

<P><INPUT TYPE="checkbox"

NAME="abogado">Abogado

<P><INPUT TYPE="checkbox"

NAME="ingeniero">Ingeniero

<CENTER><INPUT TYPE="submit"

VALUE="Enviar"></CENTER>

</FORM>

Pulsando sobre cada caja de selección se cambia de seleccionado a no seleccionado, independientemente de la posición del resto. En el código podemos apreciar que es posible indicar que una caja de selección esté seleccionada por defecto.

Datos numéricos

<input type="number" ...>

El código anterior crea un tipo especial de campo de entrada para el ingreso de números. En la mayoría de navegadores que lo soportan se presenta como un campo de entrada de texto con un control que permite incrementar y disminuir su valor.

| min y max: Como su propio nombre indica, este par de atributos permiten establecer un límite inferior y superior para los valores que se pueden introducir en un campo de formulario numérico, como los tipos de entrada numéricos, de rango, fecha y hora

(sí, hasta se pueden utilizar para establecer los límites superior e inferior para las fechas). Para entradas **range**, de hecho **min** y **max** son necesarios para definir los valores que se devuelven cuando se envía el formulario.

<input type="number" ... min="1" max="10">

El siguiente ejemplo admite valores pares comprendidos entre 2 y 12. Su aspecto es el que te mostramos en la imagen 1

<input type="number" min="2"max="12" step="2" />

Y eso no era todo... ¡más sobre formularios!

Sliders

Una de las novedades que tenemos dentro de los formularios HTML5 son los **sliders**, veamos cómo es esto:

```
<input type="range">
```

El código anterior permite crear un control deslizante que nos da la posibilidad de elegir entre una gama de valores. Tené en cuenta que, por defecto, esta entrada no suele mostrar el valor seleccionado, ni siquiera el rango de valores que cubre. A continuación vemos otro ejemplo, su aspecto será el de la imagen 1.

```
<input type="range" name="edad" min="18"
max="99" step="1" value="30">
```

- | **máx** y **min** para establecer el rango de valores máximo y mínimo.
- | **value** para especificar un valor por defecto.
- | **step** para establecer el valor numérico de avance del slider o barra de desplazamiento.

¿Cuáles son los atributos válidos para este control?

- | **name**: Nombre del input (name="nombre del input").
- | **disabled**: Atributo que determina si está desactivado el input para la introducción de datos (disable="disabled"), por defecto activado.

1

HTML5 range input

Ejemplo1

Introduce tu edad:

| **autocomplete**: Con autocomplete activado el input se autorrellenará con contenidos anteriormente enviados por él mismo (autocomplete="off", autocomplete="on").

| **autofocus**: Este atributo hace que el cursor se sitúe en el input (autofocus="autofocus").

| **list**: Asocia mediante id una lista de valores predefinidos (list="nombre lista").

| **min** y **max**: Fijan los valores máximos y mínimos que puede tener el **range input** (max="5" y/o min="0").

| **step**: Valor numérico con el cual varías el intervalo del input (step="2", sumarías o restarías de 2 en 2 en el slider).

| **value**: Valor por defecto del input.

Fecha y Hora

HTML5 tiene diferentes tipos de entrada para la creación de selectores complejos de fecha y hora.

Veremos los siguientes tags:

| **datetime**: permite elegir una fecha y una hora.

| **month**: permite elegir un mes, almacenado internamente como un número entre 1 y 12, aunque los diferentes navegadores pueden dar mecanismos más elaborados de selección, como listas de desplazamiento con los nombres de los meses.

| **week**: permite elegir una semana, almacenada internamente con el formato 2010-W37 (semana 37 del año 2010), mediante un selector de fechas.

¡A tener en cuenta para estos controles!

| El **formato de fecha** devuelto por defecto es AAAA-MM-DDTHH:MM.

| Se puede configurar un mínimo de fecha a seleccionar con el atributo '**min**':min="2013-10-08T23:59:60Z".

| Se puede configurar un máximo de fecha a seleccionar con el atributo '**max**':min="2014-10-08T23:59:60Z".

| Con el atributo '**value**' se puede poner el calendario con una fecha concreta por defecto:value="2013-10-08T23:59:60Z".

| Con el atributo '**step**' podés establecer el avance o retroceso de días con las flechas del calendario.

Podemos encontrar ejemplos de selectores de fecha en casi todos los sitios de reservas de vuelos o trenes.



Fecha y hora actual:

```
<input type="datetime" name="fechahora"
step="1" min="2014-01-01T00">
```

Los valores usados en este ejemplo son:

| **'min'**: He puesto que la fecha mínima sea el 1 de enero 2014 a las 00:00.

| **'max'**: He puesto que la fecha máxima sea el 31 de diciembre 2014 a las 12 am.

| **'value'**: Fecha por defecto el 1 de enero 2014 a las 12.

| **'step'**: Que avance en intervalos de 1 en 1

```
<input type="date" ... >
```

```
<input type="time" ... >
```

Estos atributos crean un selector de fecha plenamente funcional y una entrada de texto que contiene un separador para horas, minutos y segundos que sólo permite introducir un valor de tiempo.

```
<input type="time" ...step="1800">
```

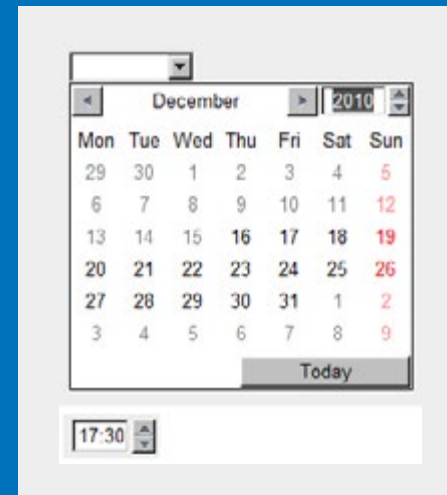
Step

El atributo step se puede utilizar con un valor de entrada numérico para dictar la granularidad de los valores que se pueden introducir. Por ejemplo, es posible que desees que los usuarios introduzcan una hora determinada, pero sólo en incrementos de 30 minutos. En este caso, podemos usar el atributo step, teniendo en cuenta que para entradas time el valor del atributo está en segundos, tal como lo muestra la imagen 1.

Color

Este tipo de entrada muestra un selector de color (2). La implementación de Opera permite al usuario elegir entre una variedad de colores, introducir valores hexadecimales directamente en un campo de texto o invocar el selector de color nativo del sistema operativo.

```
<input type="color">
```



Búsqueda



`<input type="search">`

Podría discutirse que el tipo de entrada de búsqueda no es más que una entrada con un estilo diferente. Por ejemplo, un navegador podría ofrecer al usuario una opción para crear automáticamente una búsqueda personalizada para un sitio específico (1).

Teléfono

A la hora de completar un input de tipo tel, tanto el smartphone como el iPhone convierte su teclado a números de teléfono. Observa el código a continuación y su aspecto en la imagen 2.

`<input type="tel" name="telefono">`

Mail

El nuevo tipo `<input type="email">` indica al navegador que no debe permitir que se envíe el formulario si el usuario no ha introducido una dirección de email **válida**, pero no comprueba si la dirección existe o no, sólo si el formato es válido (3). Como ocurre con el resto de campos de entrada, puede enviar este campo vacío a menos que se indique que es obligatorio.

El atributo **multiple** indica que el valor de este campo puede ser una lista de emails válidos separados por comas.

`<input type="email" name="correo">`

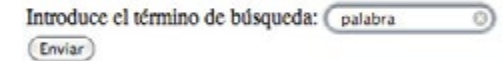
Esta vez, el teclado del smartphone es qwerty pero también tenemos la tecla "@" como podemos ver en la figura 4.

Url

El nuevo tipo `<input type="url">` indica al navegador que no debe permitir que se envíe el formulario si el usuario no ha introducido una URL correcta (mira el ejemplo en la

1

HTML5 search input



2



3



4

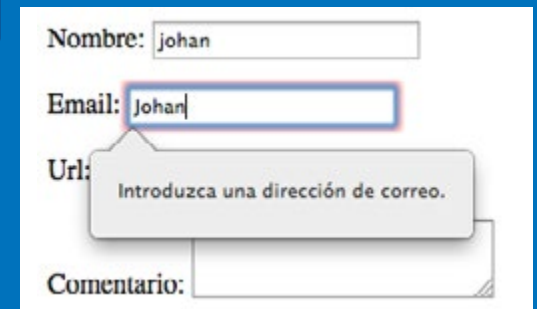


imagen 1 de la siguiente página). Algunos navegadores ofrecen ayudas al usuario, como Opera que añade el prefijo **http://** a la URL si el usuario no lo ha introducido.

```
<input type="url" name="url">
```

En este campo, el teclado del iPhone es qwerty pero en modo "url", ya que ofrece teclas fundamentales para escribir una dirección web como son el punto, la barra "/" o la tecla ".com". Podés mirar el teclado que describimos en la imagen 2

El atributo autocomplete

El atributo **autocomplete** especifica que el formulario o el campo de entrada de formulario debe tener una función de autocompletado. Funciona para las etiquetas `<form>`, y con los siguientes tipos de `<input>`: **text**, **search**, **url**, **telephone**, **email**, **password**, **datepickers**, **range** y **color**.

Cuando el usuario empieza a escribir en un campo de autocompletar, el navegador debe mostrar las opciones para rellenar el campo:

Por ejemplo:

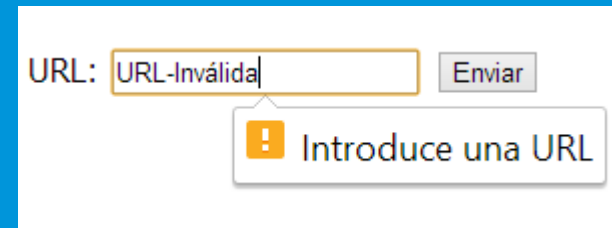
```
<form action="enviar.asp" method="post"
autocomplete="on">
Nombre: <input type="text" name="txtnom"
/><br />
Apellido: <input type="text" name="txtape"
/><br />
E-mail: <input type="email" name="email"
autocomplete="off" /><br />
<input type="submit" />
</form>
```

Campos ocultos

Los campos ocultos son un tipo de control que únicamente sirve para incluir información adicional entre los datos del formulario. Este tipo de control es invisible, y por lo tanto, el navegador no muestra ningún tipo de gráfico para advertir de su existencia.

¿Para qué sirve entonces un control oculto?

Si no aparece en la página y el navegante no puede hacer nada con él ¿cuál es su función? En general, los campos ocultos se usan para



enviar información específica al programa del servidor Web que procesará los datos enviados, por ejemplo si este programa es capaz de realizar varias funciones distintas con estos datos, podemos usar un campo oculto para pasarle la información de que debe hacer con los datos de este formulario concreto (debemos tener en cuenta que este mismo programa puede procesar la información de otros formularios también). Aun así ésta problemática queda fuera de lo que pretendemos abarcar en esta sección dedicada al lenguaje HTML.

Los campos ocultos se crean usando la etiqueta INPUT con TYPE="hidden". Además debemos usar los atributos VALUE y NAME, para indicar los datos y su nombre:

```
<INPUT TYPE="hidden" NAME="nombre"
VALUE="datos">
```

Un ejemplo de esto es:

```
<INPUT TYPE="hidden" NAME="funcion"
VALUE="enviar_mail">
```

Texto multilinea

Al contrario de lo que sucedía con las cajas de texto convencionales (**<INPUT TYPE="text">**), en esta ocasión será posible escribir varias líneas de texto, lo que es muy útil para campos en los que se requiere una respuesta extensa. Por ejemplo, si deseamos que el usuario pueda escribir el texto de un **mail**, o escribir una serie de comentarios largos, las áreas de texto serán tremendamente útiles. La etiqueta usada para insertar este nuevo control es **TEXTAREA**, que consta de los siguientes atributos:

| NAME: El nombre que queremos asignarle al control. Como siempre, este nombre será enviado junto con los datos del área de texto al mandar el formulario.

| ROWS: El número de líneas de la caja de texto.

| COLS: El número de caracteres visibles de cada línea. Este atributo es similar al atributo **SIZE** que vimos en las cajas de texto convencionales.

La etiqueta **TEXTAREA** está compuesta por una instrucción de inicio y una instrucción

* ! #



Si el Control Oculto no aparece en la página y el navegante no puede hacer nada con él ¿para qué sirve?

Sirve para enviar información específica al programa del servidor Web. Te doy un ejemplo, si este programa es capaz de realizar varias funciones distintas con estos datos, podemos usar un Campo Oculto para pasarle la información de qué debe hacer con los datos de este formulario concreto.



de fin, siendo obligatorio el uso de ambas. Entre ellas únicamente podrá insertarse texto llano (sin ninguna etiqueta HTML) y éste será mostrado como contenido inicial del área de texto. Veamos un ejemplo:

Por favor haga sus comentarios sobre esta página.

```
<TEXTAREA NAME="texto" ROWS="10"
COLS="50"> Es la mejor que he visto nunca.
</TEXTAREA>
```

Lo que en aspecto se ve como en la imagen 1.

Si no pulsamos ENTER y llegamos al final de la línea, el navegador debe decidir qué hacer.

Cuadros de selección

Su función será dar a elegir entre una serie de opciones de manera que el usuario pueda elegir una o varias de ellas. Existen dos formas de mostrar estas opciones, como una lista con desplazamiento, o mediante una lista desplegable. Para insertar un cuadro de selección usaremos la etiqueta SELECT,

que consta de una instrucción de inicio y de una instrucción de fin, entre las cuales introduciremos las diferentes opciones para el usuario. Para insertar estas opciones usaremos una nueva etiqueta, OPTION, que consta de una única instrucción. Veamos un ejemplo que ilustra el uso de uso de este tipo de control.

```
<FORM METHOD="POST" ACTION="
mailto:alumno@issd.edu.ar">
<INPUT TYPE="SUBMIT" VALUE="Enviar">
<INPUT TYPE="RESET" VALUE="Borrar">
<SELECT NAME="lista" MULTIPLE SIZE=3>
  <OPTION>Valor 1
  <OPTION SELECTED>Valor 2
  <OPTION>Valor 3
  <OPTION value="Valor 4">Valor 4
especial
  <OPTION>Valor 5
  <OPTION>Valor 6
  <OPTION>Valor 7
</SELECT>
</FORM>
```

Otro ejemplo es el que te mostramos en el siguiente código, cuyo aspecto podés ver en la imagen 2.

Áreas de texto

Por favor haga sus comentarios sobre esta página.

Es la mejor que he visto nunca.

Cuadros de selección

Elija un color:

Verde
Verde
Negro
Rojo
Azul

Enviar elección

```

<FORM>
Elija un color:<BR>
<SELECT NAME="color">
<OPTION>Verde
<OPTION>Negro
<OPTION>Rojo
<OPTION>Azul
</SELECT>
<INPUT TYPE="submit" VALUE="Enviar
elección">
</FORM>

```

Cuando se pulse el botón 'Enviar elección' se mandará al servidor el texto que sigue a la etiqueta **OPTION**, que se haya escogido (Por ejemplo '**color=Rojo**').

Existe un atributo, **SELECTED**, que nos permitirá elegir la opción que debe estar seleccionada por defecto. Por ejemplo, si en el código anterior queremos que el color negro aparezca seleccionado inicialmente tendremos que escribir:

```

<SELECT NAME="color">
<OPTION>Verde
<OPTION SELECTED>Negro
<OPTION>Rojo
<OPTION>Azul
</SELECT>

```

Al principio hemos comentado que los cuadros de selección pueden mostrarse como lista desplegable o bien como listas con desplazamiento. Hasta ahora sólo hemos visto el primer caso en el que únicamente la opción seleccionada es visible.

En ocasiones puede ser conveniente que todas, o al menos algunas de las posibilidades, sean mostradas directamente sin tener que pulsar un botón. Para conseguir esto el lenguaje HTML ofrece el atributo **SIZE**, que determina el número de opciones que pueden ser vistas simultáneamente (1). Si este valor es 1 (o no se usa el atributo **SIZE**), se mostrará una persiana desplegable, como hemos visto hasta ahora. Si es mayor, se mostrará una lista, que podrá tener barras de desplazamiento si no caben todas las opciones disponibles. Veamos un ejemplo:

Escoja una materia: Matemáticas

Obligatorias

Matemáticas

Lenguaje

Optativas

Corte y confección

Astronomía

Selecciona los días de la semana

☐ Lunes

☐ Martes

☐ Miércoles

☐ Jueves

<FORM>

Lista de la compra:


```

<SELECT NAME="compra"
SIZE="6">

```

```

<OPTION VALUE="Fruta"> Fruta

```

```

<OPTION VALUE="Verdura">Verdura

```

```

<OPTION VALUE="Azucar">Azucar

```

```

<OPTION

```

```

VALUE="Mantequilla">Mantequilla

```

```

<OPTION VALUE="Yerba">Yerba

```

Cuadros de selección

Lista de la compra:

Fruta
Verdura
Temera
Mantequilla
Salchichas
Pasta

NOTA (usuarios de Windows): Para seleccionar varias opciones mantengan pulsada la tecla **Control** y seleccionenlas con el ratón

Enviar lista de la compra

En este caso el número de opciones es igual al tamaño que le hemos dado al control, por lo que no han sido necesarias barras de desplazamiento. Si no indicamos lo contrario, los cuadros de selección actúan como los botones de radio, es decir, sólo una opción puede ser elegida simultáneamente. Afortunadamente, es posible cambiar este comportamiento y permitir la selección de varias opciones usando el atributo **MULTIPLE** en el interior de la etiqueta **SELECT**. En el ejemplo de la lista de la compra, ésta posibilidad es de gran utilidad, añadiendo este atributo al código anterior resulta:

A este código le corresponde la imagen 1

Agrupando Elementos

Dentro de un formulario podemos agrupar los elementos por tema, permitiendo un mejor aspecto. Para ello se utiliza la etiqueta **<FIELDSET>... </FIELDSET>**, que permite agrupar elementos de un formulario, creando

LEGEND

INFORMACIÓN

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris feugiat, nisl sed aliquet pretium, justo velit pellentesque arcu, non eleifend magna sem quis odio. Nulla sapien lacus, rutrum vitae vehicula eget, interdum sed dolor. Quisque condimentum fermentum lorem eget commodo. Vestibulum in sem pellentesque mi auctor posuere non id elit. Curabitur lacus arcu, cursus at lobortis a, adipiscing sed ligula.

FIELDSET

una caja alrededor de los mismos. La etiqueta **<LEGEND>... </LEGEND>**, nos permite colocar un título dentro del grupo, este título puede colocarse arriba (top), abajo (bottom), izquierda (left) o derecha (right). Mirá la imagen 2 para acasar estos conceptos.

¡Prestá mucha atención al siguiente ejemplo!

```
<LABEL DISABLED>Texto:
  <INPUT TYPE=TEXT DISABLED>
</LABEL>
```

Texto:

En este ejemplo podemos observar un formulario que se envía por mail en donde tenemos dos grupos: "Datos del Cliente" y "Forma de Pago"; en el segundo grupo tenemos dos botones con su LABEL que nos permite no sólo seleccionar al botón sino también a la leyenda.

```
<FIELDSET>
<LEGEND>Tusdatos</LEGEND>
<LABEL>
Nombre:  <INPUT TYPE="text"
name="nombre">
<br>Edad:  <INPUT TYPE="text"
name="edad">
</LABEL>
</FIELDSET>
```

Todos los controles de un formulario se pueden desactivar, impidiendo así al usuario que los sobrescriba. Se seguirán mostrando en pantalla, aunque con un aspecto distinto para indicar que no son editables. Para ello sólo tenemos que indicarle el parámetro **DISABLED**, tal como se observa en la imagen 1.

Teclas de acceso

Los **shortcuts** -o atajos de teclado- son accesos rápidos a distintas partes de nuestra web, con sólo tocar una combinación de teclas. Su implementación es sumamente sencilla, y basta con agregar `accesskey="[caracter de acceso rápido]"` dentro de una etiqueta, de la siguiente manera:

```
<input type="submit" accesskey="b"
value="Buscar">
```

Los siguientes elementos soportan el atributo **accesskey**: **A, AREA, BUTTON, INPUT, LABEL, LEGEND y TEXTAREA**.

Este ejemplo asigna la tecla de acceso "U" a un rótulo asociado con un control input. Al pulsar la tecla de acceso, el foco se dirige hacia el rótulo, el cual a su vez lo dirige al control asociado. El usuario puede entonces introducir texto en el área input.

```
<FORM action="..." method="post">
<P>
<LABEL for="nombre-usuario"
accesskey="U">
```



La ventaja de la etiqueta **Datalist** es que reconoce los primeros caracteres ingresados por el usuario, para “autocompletar” la información.

1

Escribe un color (Azul, rojo, amarillo, negro, verde):

verde

2

Enter Name:

Juan Manuel
Jean Pool
Luis Alberto

Nombre de Usuario:
</LABEL>
<INPUT type="text" name="usuario"
id="nombre-usuario">
</P>
</FORM>

Datalist

La nueva etiqueta <datalist> permite automatizar el ingreso de datos. Genera una lista desplegable de opciones para que el usuario realice su selección, tal como te lo mostramos en las imágenes **1** y **2**.

```
<label>Selecciona tu país:</label>  
<input type="text" name="pais" id="pais"
```

```
list="países"/>  
<datalist id="países">  
<option value="España" />  
<optionvalue="México" />  
<optionvalue="Argentina" />  
<optionvalue="Perú" />  
<optionvalue="Colombia" />  
<optionvalue="Otro país" />  
</datalist>
```

Si el navegador no soporta esta nueva característica mostrará un casillero de **input** sencillo sin desplegar las opciones.

```
<input list="listacolores">  
  
<datalist id="listacolores">  
<option value="Azul">
```

```
<option value="Rojo">  
<option value="Amarillo">  
<option value="Negro">  
<optionvalue="Blanco">  
</datalist>
```

Keygen

Esta etiqueta genera dos claves, una pública y una privada. Cuando se envía el formulario, la clave privada se almacena localmente y sólo la clave pública es enviada al servidor. Este es un elemento de seguridad útil para instituciones que deben validar el ingreso de usuarios a información privada desde el exterior de la red.


```
<form method="get">
<label>Usuario:</label><input type="text"
name="usuario">
<label>Clave encriptada:</label>
<keygenname="clavepub" challenge="serie">
<input type="submit" name="security"
value="Enviar">
</form>
```

Los atributos propios de esta etiqueta son:

| **Challenge:** genera una serie de caracteres de validación que se envían junto con la clave.

| **Keytype:** especifica el tipo de algoritmo de seguridad utilizado por la clave. Si no se utiliza, se asume RSA.

| **Name:** el nombre asociado al elemento **<keygen>**.

| **Autofocus:** este elemento **<keygen>** debe tener el foco una vez cargada la página.

| **Form:** nombre del formulario al cual pertenece esta etiqueta.

| **Disabled:** este atributo booleano indica, si su valor es **true**, que la etiqueta está deshabilitada.

Output

La etiqueta **<output>** representa el resultado de un cálculo realizado a través de un **script**.

```
<form onsubmit="return false" oninput="res.
value =parseInt(op1.value) * parseInt(op2.
value)">
<input name="op1" type="number"
step="any"> x
<input name="op2" type="number"
step="any"> =
<output name="res" for "op1 op2"></output>
</form>
```

Los atributos específicos de esta etiqueta son:

| **Name:** asigna un nombre al elemento output.

| **Form:** asocia este elemento con el formulario en el cual está incluido.

| **For:** establece una relación entre el resultado del cálculo y los elementos del mismo.

Recordá que la etiqueta Keygen es un elemento de seguridad útil para instituciones que deben validar el ingreso de usuarios a información privada desde el exterior de la red.



Validación de Formularios

Los tipos y atributos de **input** de HTML5 han sido pensados también para la validación de formularios del lado del cliente, y hacen innecesario recurrir a otros lenguajes o **scripts** (al menos, en los navegadores que actualmente soportan esta funcionalidad).

Atributo required

El atributo **required** controla que el campo de entrada tenga contenido. Si no es así, emite un mensaje o señal de error e impide enviar el formulario, como podés observar en la imagen 1. Al incluir el atributo **required** dentro de un elemento `<input>`, automáticamente se hace obligatorio su llenado y al ser un atributo

tipo booleano, sólo se requiere su presencia nada más.

```
<input type="text" name="nombre" required/>
```

En el momento en que se encuentra un error de validación se ejecuta la acción por defecto del navegador. Normalmente aparece un pequeño globo emergente (tooltip) conteniendo un texto de advertencia. Añadiendo el atributo `title` en la etiqueta `<input>`, se puede extender la información mostrada. Observá esto en la imagen 2.

```
<input title="Se necesita un nombre" type="text" name="nombre" required/>
```

1

Nombre:


! Completa este campo

2


Nombre:

! Completa este campo
Se necesita un nombre

Email:

 Utiliza un formato que coincida con el solicitado
mail@example.com

Email:

 Introduce una dirección de correo electrónico

El atributo pattern

Pattern busca un patrón/expresión regular dentro de un campo de texto y determina si este lo cumple o no. Por tanto, su función es validar la información introducida por los usuarios. Utilizando el atributo **pattern**, conjuntamente con **required** se logra que se verifique no sólo la presencia de un valor, sino que este valor debe contener un formato, una longitud o un tipo de dato específico, como podés ver en la imagen 1. Esto último se logra definiendo un patrón con expresiones regulares.

```
<input type="text" pattern="^[a-zA-Z0-9.!#$%&'*/+=?^_`{}~.-]+@[a-zA-Z0-9-]+>
```

Con el patrón anterior se valida un formato correcto de correo electrónico (mail@example.com), pero en la práctica esto resulta un tanto absurdo, ya que definiendo un tipo email en la etiqueta `<input>`, el navegador por si mismo hace la validación del tipo en cuestión, tal como se muestra en la figura 2.

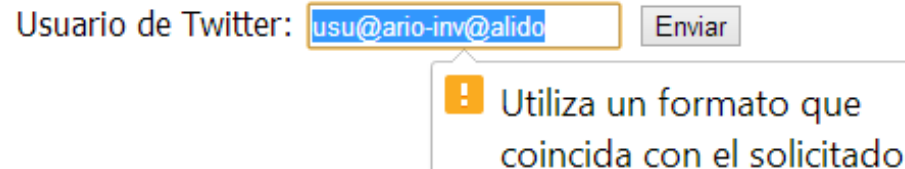
```
<input type="email" name="email" required/>
```

Supongamos que se necesita que el usuario ingrese una dirección válida de IPV4. Para eso debemos proceder de la siguiente manera:

```
<input title="0.0.0.0" type="text" name="ipv4"/>
```

Cuatro bloques de uno a tres dígitos es la notación punto-decimal de una dirección válida de IPV4. El uso de `required` y `pattern` asegura que la entrada debe ser rellenada siguiendo la expresión regular.

```
<input type="text" id="referencia"
name="referencia" required pattern="[A-Z]{3}
[0-9]{4}" title="El formato debe coincidir con 3
letras mayúsculas y 4 números."/>
<form action="enviar.php">
<label for="twitter">Usuario de Twitter: </
label>
<input type="text" pattern="^[A-Za-z0-9_
{1,15}$" name="twitter_account" id="twitter" />
<input type="submit" value="Enviar" />
</form>
```



Si cumple con la expresión regular indicada `^[A-Za-z0-9_]{1,15}$` (admite caracteres alfanuméricos y guion bajo hasta un máximo de 15), el formulario será enviado normalmente, sino, el navegador mostrará una advertencia y evitará que el formulario se envíe (1).

El atributo novalidate

El atributo **novalidate** especifica que el formulario o el campo de entrada de formulario no deben ser validados cuando se envíen. Funciona con: **<form>** y los siguientes tipos **<input>**: **text**, **search**, **url**, **telephone**, **email**, **password**, **date pickers**, **range**, y **color**.

Por ejemplo:

```
<form action="formu.php" method="post"
novalidate="true">
E-mail: <input type="email" name="user_
email" />
<input type="submit" />
</form>
```

Atributo formnovalidate

El atributo **formnovalidate** puede ser aplicado en **input** y **button**. Si está presente, la validación queda sin efecto. En el siguiente ejemplo tenemos dos botones, uno en donde se realiza la validación y otro en el que no.

```
<input type="text" id="referencia"
name="referencia" required pattern="[A-Z]{3}
[0-9]{4}"
title="El número de referencia consta de 3
letras mayúsculas y 4 dígitos."/>
```

```
<input type="submit" formnovalidate
value="Grabar">
```

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

→ Desempeños

→ Desempeño 1

Ahora te proponemos codificar los siguientes formularios. Aplicá los estilos aprendidos cuando sean necesarios ya que en la próxima clase veremos CSS aplicado a los controles de formulario.

Por favor, rellena estos datos

Nombre:	<input type="text"/>
Apellidos:	<input type="text"/>
Dirección:	<input type="text"/>
Código Postal:	<input type="text"/>
Marca tu equipo favorito:	<input checked="" type="radio"/> Real Madrid <input type="radio"/> Atlético de Madrid
Marca tu música favorita:	<input checked="" type="checkbox"/> Rock <input checked="" type="checkbox"/> Pop <input type="checkbox"/> Heavy <input type="checkbox"/> Tecno
Elige un escritor:	<input type="text" value="Cervantes"/>
¿Algún comentario?	<div><div></div></div>

→ Desempeño 2

→ Desempeño 3

Para este ejercicio sólo programá la estructura en HTML5, ya que en la clase siguiente se le aplicará CSS3, a modo de ejemplo.

Formulario de alta

Nombre y apellidos *

Nombre

Primer apellido

Segundo apellido

Dirección *

Calle, número, piso, puerta

Código postal Municipio

Provincia País

Email

Teléfono *

Fijo Móvil

Nombre:

Apellidos:

E-Mail: @

Población:

Provincia:

Edad:

☐ 0-20

☒ 20-40

☐ 40-60

☐ 60-100

¿Cómo nos conociste?

☐ A través de un amigo.

☐ A través de un buscador.

☐ Navegando por la red.

☐ Otros

Opinión sobre nuestra página web

Escriba aquí su opinión...

Tiene alguna sugerencia...

Escriba aquí sus sugerencias...

→ Desempeño 4

→ Desempeño 5

* Seleccione el producto

* Como supo de nuestro producto?

- ☐ Amigos
- ☐ Avisos Online
- ☐ Folletería
- ☐ Radio
- ☐ Vendedores
- ☐ Supermercado
- ☐ TV
- ☐ Otros

* Despues de la primera adquisicion del producto, que tanto se demoro en probarlo nuevamente?

- ☐ Inmediatamente
- ☐ En pocos dias
- ☐ En pocas semanas
- ☐ En pocos meses
- ☐ Pasado un año

* Que factores influyen a la hora de seleccionar el producto?

- ☐ Publicidad
- ☐ Marca
- ☐ Recomendacion de un amigo
- ☐ Recomendacion del vendedor
- ☐ Precio
- ☐ Garantia de fabrica
- ☐ Garantia del Supermercado

* Donde compro el producto?

- ☐ Supermercado
- ☐ Almacen
- ☐ Regalo
- ☐ Online
- ☐ Otros

* Que tal se siente con los atributos del producto?

	Super Conforme	Conforme	Indiferente	Disconforme	Nada Conforme
Indispensable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
De facil Uso	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Esencial	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Divertido	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Moderno	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Popular	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Costoso	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Información sobre el producto

Datos básicos

Nombre

Descripción

Foto Ningún archivo seleccionado

☐ Añadir contador de visitas

Datos económicos

Precio € Impuestos

Promoción

- ☒ Ninguno
- ☐ Transporte gratuito
- ☐ Descuento 5%

→ Desempeño 6

→ Desempeño 7

Nombre

Email

Población

Sexo ☒ Hombre ☐ Mujer Frecuencia de los viajes Comentarios sobre su satisfacción personal

☒ Deseo recibir notificación de las novedades en las líneas de autobuses.

iDESWEB

Formulario de registro

Los campos marcados con un asterisco son obligatorios

¿Quién eres?

Nombre * Apellidos *

Correo electrónico *

Fecha de nacimiento

¿De dónde eres?

Ciudad

Código postal

País

¿Cómo quieres iniciar sesión?

Nombre de usuario * (letras del alfabeto inglés y números, mínimo 4 caracteres)

Contraseña * (letras del alfabeto inglés, números, guión "-", subrayado "_", punto ".", mínimo 8 caracteres)

Vuelve a escribir la contraseña *

Condiciones de registro

Deseo recibir ofertas de iDESWEB ☐ Una vez al día ☐ Una vez a la semana ☐ Una vez al mes

☐ Acepto el acuerdo de servicios, la declaración de privacidad y la declaración de uso de cookies

Formulario de Subscripcion

* Nombre * Apellido

* Calle Nro/Dpto/Plao

Barrio

* Ciudad * Provincia

* CP * Telefono

Subscripcion

Usted debera aceptar la subscripcion de 1 año como mínimo.

* % por categoria

☐ Movilidad y Pasajes (\$120) ☐ Informatica y Tecnología (\$250)

☐ Indumentaria (\$100) ☐ Alimentos (\$100)

☐ Compras via Web (\$130) ☐ Otros (\$90)

Informacion de la Tarjeta de Credito



Sus pagos sera tramitado por PayPal.
Una vez que su formulario haya sido enviado,
cuando ingrese la informacion del pago.

* Campos requeridos en forma obligatoria.

Continuar....

Agregar Congreso

DATOS PRINCIPALES

Clave

Nombre Completo Sede Tipo de Evento

Fecha de Inicio Fecha de Termino Fecha Larga

Descripción del Evento (No mayor a 500 caracteres)

CONFIGURACIÓN GENERAL

Tipo de Moneda Costo (incluyendo impuestos) Tipo de cambio a MXN

STATUS

Status del Evento Activar idioma Inglés


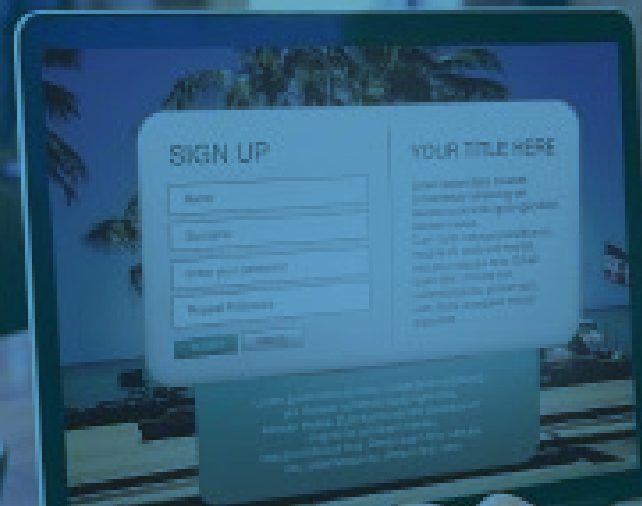
☐ Activo ☐ Activo ☒ Registro Normal

☐ Registro con claves

Link de Retomo (ej. http://www.dominio.com)

guardar

→ Desempeño 10



*** Nombre** *** Apellido** *** Fecha de Ingreso** *** Ingreso**

Direccion 1 *** Direccion 2** *** Fecha Salida** *** Salida**

*** Ciudad** *** Provincia** *** CP** *** Mayores** *** Niños/Menores**

*** Telefono** *** Email** *** Habitacion** ☐ Comun ☐ Deluxe ☐ Suite

Desea algun servicio mas?

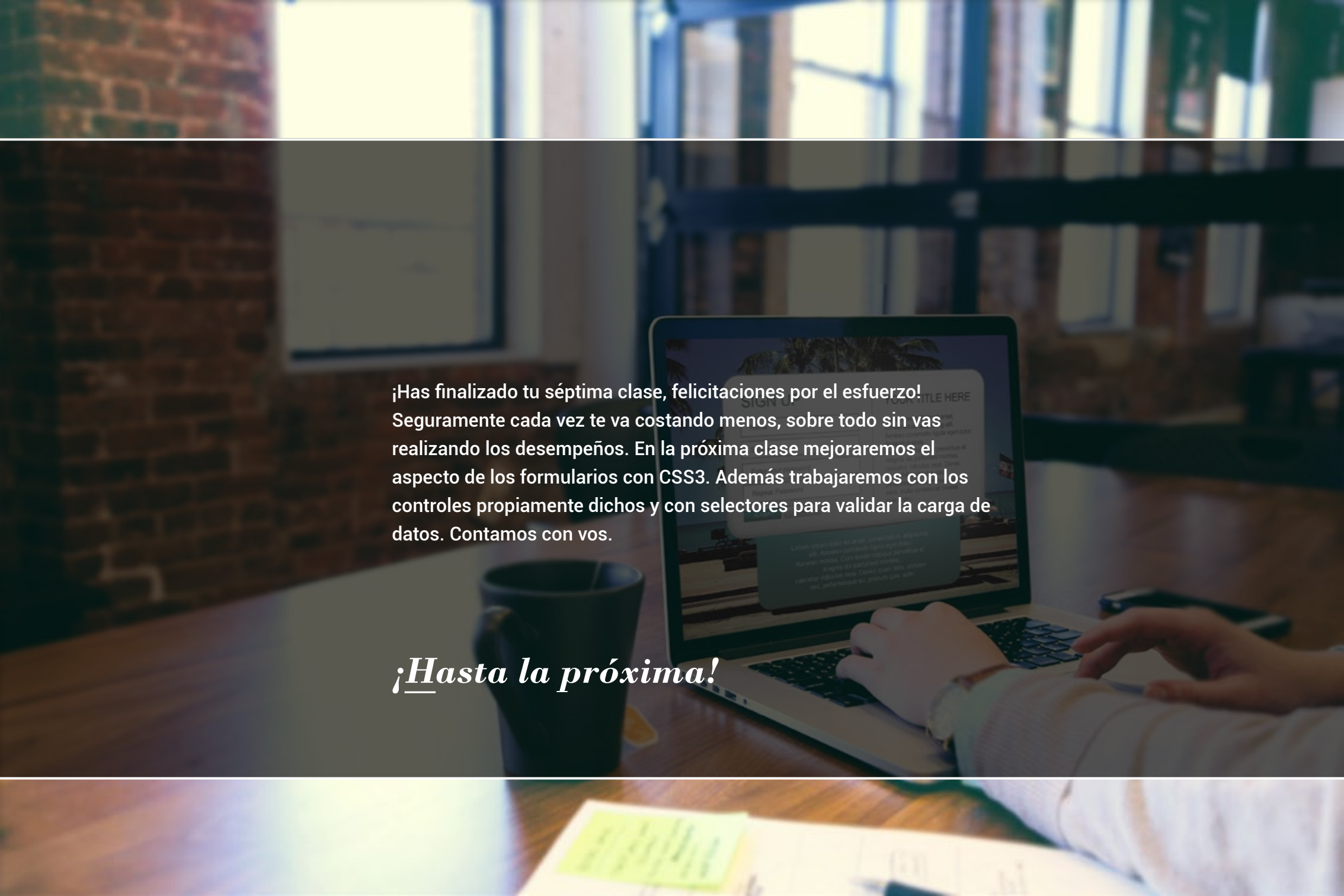
*** Campos requeridos**

Reservar

Autoevaluación

Ahora, a modo de autoevaluación, contestá las siguientes preguntas:

- 01 |** ¿Qué diferencia existe entre los Radio buttons y las casillas de verificación?
- 02 |** ¿Para qué sirve agrupar controles de un formulario?
- 03 |** ¿Cuál es la función del atributo Autocomplete?
- 04 |** ¿Cuál es el propósito del tag<LABEL>?
- 05 |** ¿Cuál es la función de la etiqueta <OPTION>?
- 06 |** ¿Qué pasa si se insertan varios botones de opción con el mismo nombre?
- 07 |** ¿Que indica el atributo disabled?
- 08 |** ¿Cuál es la función del atributo hidden?
- 09 |** ¿Cuál es la función del atributo pattern?
- 10 |** ¿Cuál es la diferencia entre POST y GET?



¡Has finalizado tu séptima clase, felicitaciones por el esfuerzo! Seguramente cada vez te va costando menos, sobre todo sin vas realizando los desempeños. En la próxima clase mejoraremos el aspecto de los formularios con CSS3. Además trabajaremos con los controles propiamente dichos y con selectores para validar la carga de datos. Contamos con vos.

¡Hasta la próxima!