

# Lenguaje de Marcas y Sistemas de Gestión

## UF 2 - HTML

### UA 2.6 - Introducción HTML: Formularios



Centro Profesional  
**Universidad**  
**Europea**

Prof. Raúl Rodríguez  
CFGs ASIR / DAM / DAW

# UA 2.6: Introducción HTML: Formularios



## Objetivos

- Definir qué es una página web
- Utilizar lenguajes de marcas para la transmisión de información a través de la Web analizando la estructura de los documentos e identificando sus elementos.



# UA 2.6: Introducción HTML: Formularios



## Contenidos

- Estructura básica y características principales de HTML5
- Cómo usar y crear formularios en HTML5.



# UA 2.6: Introducción HTML: Formularios



## Definición de Formulario

- Con los **Formularios** vamos a dotar a nuestras páginas estáticas escritas en HTML de la posibilidad de hacerlas interactivas.
- Son los elementos, que además de los enlaces, nos van a permitir interactuar con la página web.
  - ✓ Recogen información del usuario
  - ✓ Envían esos datos para que se guarden, se procesen o cualquier otra operación que sea requerida.
- Son elementos complejos, quizás el elemento más complejo de la especificación HTML, ya que pueden contener multitud de elementos y cada uno con muchas opciones diferentes.



# UA 2.6: Introducción HTML: Formularios



## Definición de Formulario

- Los **objetivos** que tenemos que tener a la hora de **crear un formulario HTML5** son los siguientes:
  - ✓ Hacer que sea lo más **sencillo** posible el proceso de inserción de datos por parte del usuario.
  - ✓ Que la experiencia de usuario sea lo más **agradable** posible.
  - ✓ Que los datos introducidos por el usuario estén en un **formato predecible** y esperado.
  - ✓ Reducir los errores al introducir datos en la medida de lo posible y comunicarlo claramente si ocurren.



# UA 2.6: Introducción HTML: Formularios



## Creación de Formulario

- Para crear un formulario, utilizaremos la etiqueta `<form>...</form>` que delimita el principio y el final del formulario, y contendrá todos los controles que lo formen.



- La etiqueta `<form>` dispone de varios **atributos** para utilizar, como son los siguientes:

Atributo	Valor	Descripción
action	<u>URL</u>	Dirección URL del back-end donde se enviará la información del formulario.
method	get   post	Método HTTP de envío. <b>GET</b> a través de URL, <b>POST</b> para envío extenso.
name	<u>nombre</u>	Nombre del formulario. Útil para procesar posteriormente.
target	<u>destino</u>	Nombre del lugar donde se abrirá el formulario. <b>_blank</b> para nueva pestaña.
enctype	<u>tipo</u>	Codificación para el envío del formulario. Importante para envío de archivos.
accept-charset	<u>codificación</u>	Fuerza a utilizar una codificación en los parámetros de texto del formulario.
autocomplete	on   off	Activa o desactiva el autocompletado para todos los campos del formulario.

# UA 2.6: Introducción HTML: Formularios



## Creación de Formularios

- El código de un formulario base que tendríamos aún vacío podría ser el siguiente:

```
<form name="formulario" method="post" action="http://pagina.com/send.php">
</form>
```

- Fijaros, que utilizamos tres atributos base en `<form>` que son fundamentales como son los siguientes:
  - ✓ **action:** indica la dirección donde se enviarán los datos cuando el usuario pulse el botón de enviar. En caso de omitirse esta información, se enviarían los datos a la propia página actual, volviéndola a cargar.
  - ✓ **method:** se define el tipo de método HTTP a utilizar. Utilizaremos el método **GET** cuando nos interese que los datos se envíen en la propia URL. En caso contrario, **POST**.
  - ✓ **name:** nombre del formulario para cuando necesitemos hacer referencia a él. Suele hacerse desde JavaScript o desde el Backend.

# UA 2.6: Introducción HTML: Formularios



## Tipos de datos a obtener en los Formularios

- Cuando creamos un formulario, tenemos que saber a priori que información tenemos que solicitar al usuario para de esta manera saber qué tipo de campo de entrada de datos es más conveniente colocar. Los más comunes y los que veremos con mayor detalle son los siguientes:

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

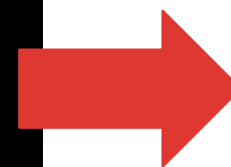
# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

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

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



Nombre de usuario:

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

# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Atributos etiqueta <input>

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

- **type** → nos permitirá indicar que tipo de campo de dato mostrará en el navegador (text, number, date...).
- **name** → proporciona un nombre al campo de texto para que cuando enviemos la información del formulario, poderlos manejar desde javascript o desde el back-end.

# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Etiqueta <input>

- La etiqueta HTML <input> a parte de usar el atributo **type** para permitir al usuario introducir **información de texto**, puede utilizar la etiqueta <textarea> para cantidades de texto más grandes como varios párrafos. Según nuestras necesidades, tendremos varias opciones:

Tipo de información a obtener	Etiqueta a utilizar	Ejemplo
Texto alfanumérico libre (texto corto)	<code>&lt;input type="text"&gt;</code>	<input type="text" value="Gordon Freeman"/>
Texto para búsquedas	<code>&lt;input type="search"&gt;</code>	<input type="text" value="HTML5"/>
Número de teléfono o móvil	<code>&lt;input type="tel"&gt;</code>	<input type="text" value="555-424258"/>
Dirección URL	<code>&lt;input type="url"&gt;</code>	<input type="text" value="http://www.google.es"/>
Dirección de correo email	<code>&lt;input type="email"&gt;</code>	<input type="text" value="correo@dominio.com"/>
Clave o contraseña	<code>&lt;input type="password"&gt;</code>	<input type="password" value="...."/>
Campo oculto (no mostrar al usuario)	<code>&lt;input type="hidden"&gt;</code>	
Texto alfanumérico libre (texto extenso)	<code>&lt;textarea&gt;</code>	<div>Esto es un campo de texto más extenso, útil para textos de uno o más párrafos.</div>

# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### <input type = "text">

- Utilizaremos esta etiqueta para fragmentos de texto corto: nombres, apellidos, direcciones físicas, etc... En resumen, pequeños fragmentos de texto menores a un par de palabras. Un sencillo ejemplo:

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



Introduce tu nombre de usi

- En el ejemplo, hemos incluido dos atributos adicionales:
  - ✓ **Autocompletado desactivado:** No aparecerá información escrita anteriormente en ese campo.
  - ✓ **Placeholder:** Texto que aparece modo de sugerencia/ayuda (en gris, ligeramente menos visible que el texto del campo).

# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Textos cortos específicos

- En este apartado, se recogen las etiquetas `search`, `tel`, `url` y `email`.

Texto para búsquedas	<code>&lt;input type="search"&gt;</code>	<input type="text" value="HTML5"/>
Número de teléfono o móvil	<code>&lt;input type="tel"&gt;</code>	<input type="text" value="555-424258"/>
Dirección URL	<code>&lt;input type="url"&gt;</code>	<input type="text" value="http://www.ejemplo.com/"/>
Dirección de correo email	<code>&lt;input type="email"&gt;</code>	<input type="text" value="correo@dominio.com"/>

**Ejemplo:**

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



Teléfono:  Email:  Página web:

# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Campos para contraseñas

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

```
<form name="formulario" method="post" action="http://pagina.com/send.php">  
  <!-- Campo de entrada de usuario -->  
  Usuario: <input type="text" name="nombre" placeholder="Por ejemplo, Juan">  
  <!-- Campo de entrada de password -->  
  Password: <input type="password" name="pass" placeholder="Contraseña">  
</form>
```



Usuario:

Password:

# UA 2.6: Introducción HTML: Formularios




## Campos de entrada de información

### Campos extensos de texto y ortografía

- Si queremos indicar un campo para escribir textos más extensos, lo más apropiado es utilizar la etiqueta <textarea> con los siguiente atributos:

Atributo	Valor	Descripción
cols	<u>número de columnas</u>	Número de caracteres que caben en horizontal ( <u>columnas</u> ).
rows	<u>número de filas</u>	Número de caracteres que caben en vertical ( <u>filas</u> ).

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



Este es el valor por defecto

# UA 2.6: Introducción HTML: Formularios

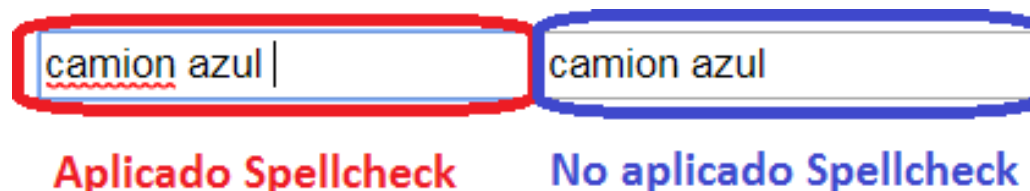


## Campos de entrada de información

### Campos extensos de texto y ortografía

- Tanto en las etiquetas `<input>` como `<textarea>`, podemos utilizar un atributo denominado **spellcheck** mediante el cuál indicamos al navegador si debe revisar la ortografía del texto escrito en dicho elemento (aparece una línea roja inferior cuando la palabra no se encuentra en el diccionario).

```
<form name="formulario" method="post" action="http://pagina.com/send.php">  
  <input type="text" name="objeto1" value="camion azul " spellcheck="true">  
  <input type="text" name="objeto2" value="camion azul " spellcheck="false">  
</form>
```



- Si el atributo **spellcheck** está a **true**, el navegador comprobará la ortografía, si se encuentra a **false**, no la comprobará

# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Campos numéricos

- Si deseamos obtener información o cantidades numéricas en un formulario, tenemos a nuestra disposición dos tipos de etiquetas `<input>`:
  - ✓ La primera de ellas es **number**, un tipo con el que podremos utilizar un campo de texto donde el usuario sólo puede escribir números (o caracteres como +, - o el número e, entre otros). La diferencia respecto a otros campos es que al mover el ratón sobre él, nos aparecen dos flechas que nos permiten aumentar o disminuir la cantidad numérica del campo.
  - ✓ Por otro lado, tenemos la etiqueta `<input type="range">`, que en lugar de mostrarnos un campo de texto que podemos modificar introduciendo directamente el número, podemos hacerlo mediante un «slider» o **barra de desplazamiento** que representa un número entre un **rango numérico** específico. Tanto en una como en otra podemos indicar los atributos **min**, **max** y **step**, para indicar límites y restricciones.

# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Campos numéricos

Tipo de información a obtener	Etiqueta a utilizar	Ejemplo
<u>Número o cantidad numérica</u>	<code>&lt;input type="number"&gt;</code>	<input type="text"/>
<u>Rango numérico</u>	<code>&lt;input type="range"&gt;</code>	<input type="range"/>

Atributo	Valor	Significado del atributo
min	<u>número mínimo</u>	Número mínimo permitido para pasar validación.
max	<u>número máximo</u>	Número máximo permitido para pasar validación.
step	<u>incremento o decremento</u>	Variación permitida entre números para pasar validación.

# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Campos numéricos

```
<form name="formulario" method="post" action="http://pagina.com/send.php">
  <!-- Número entre 10 y 50, de 5 en 5. Valor por defecto: 25 -->
  <input type="number" name="numero" value="25" min="10"
                                         max="50" step="5">

  <!-- Su misma versión, utilizando el slider en un rango numérico -->
  <input type="range" name="numrango" value="25" min="10"
                                         max="50" step="5">

</form>
```

Las diferencias son las siguientes:

- ✓ En el campo `<input type="number">` es posible introducir manualmente números no permitidos.
- ✓ En el campo `<input type="range">` no es posible salirse del rango indicado.
- ✓ Tanto en uno como en otro, no es posible enviar el formulario si no se cumplen las restricciones.

De cualquier forma, si indicamos `min=10`, `max=50` y `step=5` en una etiqueta numérica `<input>`, los valores permitidos por el usuario serían 10, 15, 20, 25, 30, 35, 40, 45 y 50.

# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Campos de Fecha y Hora

- Los atributos **type** de fecha/hora existentes en los campos **<input>** serían los dos primeros, mientras que los tres últimos se añadirán en HTML 5.1:

Tipo de información a obtener	Etiqueta a utilizar	Ejemplo
Fecha	<code>&lt;input type="date"&gt;</code>	<input type="text" value="dd/mm/aaaa"/>
Hora	<code>&lt;input type="time"&gt;</code>	<input type="text" value="--:--"/>
Fecha y hora local	<code>&lt;input type="datetime-local"&gt;</code>	<input type="text" value="dd/mm/aaaa --:--"/>
Mes	<code>&lt;input type="month"&gt;</code>	<input type="text" value="----- de ----"/>
Semana	<code>&lt;input type="week"&gt;</code>	<input type="text" value="Semana --, ----"/>



# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Campos de Fecha y Hora

Ejemplo básico con las dos primeras, `<input type="date">` e `<input type="time">`:

```
<form name="formulario" method="post" action="http://pagina.com/send.php">
  <!-- Campo de entrada de fecha -->
  Selecciona la fecha deseada:
  <input type="date" name="fecha" min="2018-03-25"
                                max="2018-05-25" step="2">

  <!-- Campo de entrada de hora -->
  Selecciona la hora deseada:
  <input type="time" name="hora" min="18:00"
                                max="21:00" step="3600">

</form>
```

Selecciona la fecha deseada:

Selecciona la hora deseada:

- Los atributos **min** y **max** actúan estableciendo un límite de fecha mínima o máxima permitida, y el campo **step** se utilizan para restricciones al variar unidades. En este caso, sólo aceptan días de dos en dos desde el 25 de marzo de 2018 hasta el 25 de mayo de 2018 y en el campo de la hora sólo se permitirá variar las horas, ya que se ha indicado 3600 segundos (salto cada 60x60seg, o sea cada hora).

# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Campos de Fecha y Hora en HTML 5.1

- Los tres últimos pertenecen a HTML 5.1, aunque ya varios navegadores lo soportan. Con ellos podremos seleccionar la fecha y hora local, así como meses específicos o semanas concretas del año:

```
<form name="formulario" method="post" action="http://pagina.com/send.php">
  <!-- Campo de entrada de mes -->
  Selecciona el mes deseado:
  <input type="month" name="mes" value="2018-05"
        min="2018-02" max="2018-06" step="2">
  <!-- Campo de entrada de semana -->
  Selecciona la semana deseada:
  <input type="week" name="semana" value="2018-W18"
        min="2018-W10" max="2018-W20" step="2">
</form>
```

- En el primer campo <input> seleccionamos el mes con el formato yyyy-mm, donde yyyy es la cifra de 4 dígitos del año y mm el mes con 2 dígitos. En el segundo campo <input> seleccionamos la semana con el formato yyyy-Wss, donde nuevamente yyyy es el año en formato de 4 dígitos, W es siempre esa letra y ss es el número de la semana en formato de 2 dígitos. El atributo step="2" lo que hace que sólo se puedan seleccionar semanas o meses impares.

# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Campos de Fecha y Hora en HTML 5.1

Selecciona el mes deseado: mayo de 2018 x ▴ ▾ Selecciona la semana deseada: Semana 18, 2018

mayo de 2018 ▾ ◀ ● ▶

lun	mar	mié	jue	vie	sáb	dom
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

Selecciona el mes deseado: mayo de 2018 Selecciona la semana deseada: Semana 18, 2018 x ▴ ▾

abril de 2018 ▾ ◀ ● ▶

Semana	lun	mar	mié	jue	vie	sáb	dom
13	26	27	28	29	30	31	1
14	2	3	4	5	6	7	8
15	9	10	11	12	13	14	15
16	16	17	18	19	20	21	22
17	23	24	25	26	27	28	29
18	30	1	2	3	4	5	6

# UA 2.6: Introducción HTML: Formularios



## *Campos de entrada de información*

### **Controles: Casillas o Botones**

- En un formulario, podemos utilizar **casillas de verificación** o **botones de opción** (botones de radio) para la selección de opciones. La diferencia principal entre estos dos tipos de botones es la siguiente:
  - ✓ Las **casillas de verificación** se marcan para establecer una opción como **activada** (verdadero). Se puede añadir el atributo **checked** para forzar que esa casilla esté activada inicialmente.
  - ✓ Los **botones de radio** se marcan para elegir **sólo una opción de varias posibles**. Si las opciones son demasiadas, se suele optar por una lista de selección. Para tener varios botones radio en un mismo grupo, y que sólo se active uno de ellos, hay que poner el mismo nombre en el atributo **name** a cada opción. Luego, en el atributo **value** hay que colocar el valor que tendrá que seleccionar el usuario

# UA 2.6: Introducción HTML: Formularios

## Campos de entrada de información

### Controles: Casillas o Botones

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

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



Sexo: ☒ Hombre ☐ Mujer

¿Eres autónomo o empresa? ☒ ☐

# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Listas de Selección

- Podemos utilizar **listas de selección** si las casillas de verificación se nos quedan cortas o necesitamos mostrar una lista más extensa de datos. Estas listas nos permiten mostrar al usuario varias opciones disponibles para elegir una de ellas. Podemos encontrar los siguientes tipos de listas:

Tipo de información a obtener	Etiqueta a utilizar	Ejemplo
Lista (cerrada) de opciones	<code>&lt;select&gt;</code> y <code>&lt;option&gt;</code>	<input type="text" value="Opción 1"/>
Lista (abierta) de opciones	<code>&lt;datalist&gt;</code>	<input type="text"/>

```
<select name="transporte">
<option value="1">Coche</option>
<option value="2">Avión</option>
<option value="3">Tren</option>
</select>
```

```
<input list="fruits">
<datalist id="fruits">
  <option value="Apple"/>
  <option value="Cherry"/>
  <option value="Lemon"/>
  <option value="Orange"/>
  <option value="Peach"/>
  <option value="Strawberry"/>
</datalist>
```

# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Listas de Seleccionables

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

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



Selecciona la opción deseada:

Opción 2 ▼
Opción 1
Opción 2
Opción 3

# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Listas de selecciones múltiples

- A la etiqueta `<select>` le podemos aplicar el atributo **multiple**, de modo que desaparecerá la posibilidad de desplegar la lista, mostrando directamente las opciones en otro formato que hará posible pulsar la tecla **CTRL** y marcar varias de las opciones de la lista. También podemos utilizar la etiqueta `<optgroup>`, que es una etiqueta contenedora para agrupar varias etiquetas `<option>`. De esta forma, podemos crear pequeñas agrupaciones para tener más organizada la lista de opciones. La propia etiqueta `<optgroup>` no es seleccionable por el usuario, aparecerá en negrita y sólo estará disponible para organizar las opciones:

```
<form name="formulario" method="post" action="http://pagina.com/send.php">
  <!-- Lista de selección múltiple -->
  <select name="combo" multiple>
    <!-- Formato alternativo con atributo label -->
    <optgroup label="Opciones básicas">
      <option value="1" label="Opcion 1">
    </optgroup>
    <optgroup label="Opciones avanzadas">
      <option value="2" label="Opcion 2">
      <option value="3" label="Opcion 3">
    </optgroup>
  </select>
</form>
```



Opciones básicas  
Opcion 1

Opciones avanzadas  
Opcion 2

# UA 2.6: Introducción HTML: Formularios

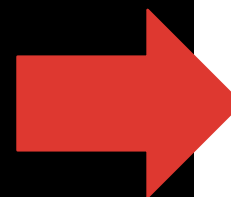


## Campos de entrada de información

### Listas de selecciones abiertas

- Podemos crear una lista abierta, donde el usuario puede seleccionar opciones sugeridas mediante un `<datalist>` o indicar la suya propia escribiéndola manualmente. Para ello, utilizaremos una etiqueta `<datalist>` que funcionará exactamente igual que un `<select>`, conteniendo las opciones posibles en etiquetas `<option>`.
- Al hacer esto, y al contrario que las etiquetas `<select>`, el elemento `<datalist>` no mostrará visualmente nada. Sin embargo, la diferencia radica en que le daremos un atributo `id` al `<datalist>` y lo combinaremos con un campo de texto `<input>` a través del atributo `list`:

```
<form name="formulario" method="post" action="http://pagina.com/send.php">
  <!-- Campo de texto combinado con lista de opciones -->
  <input type="text" list="items">
  <!-- Lista de opciones -->
  <datalist id="items">
    <option value="1">Opción 1</option>
    <option value="2">Opción 2</option>
    <option value="3">Opción 3</option>
  </datalist>
</form>
```



▼	
1	Opción 1
2	Opción 2
3	Opción 3

# UA 2.6: Introducción HTML: Formularios

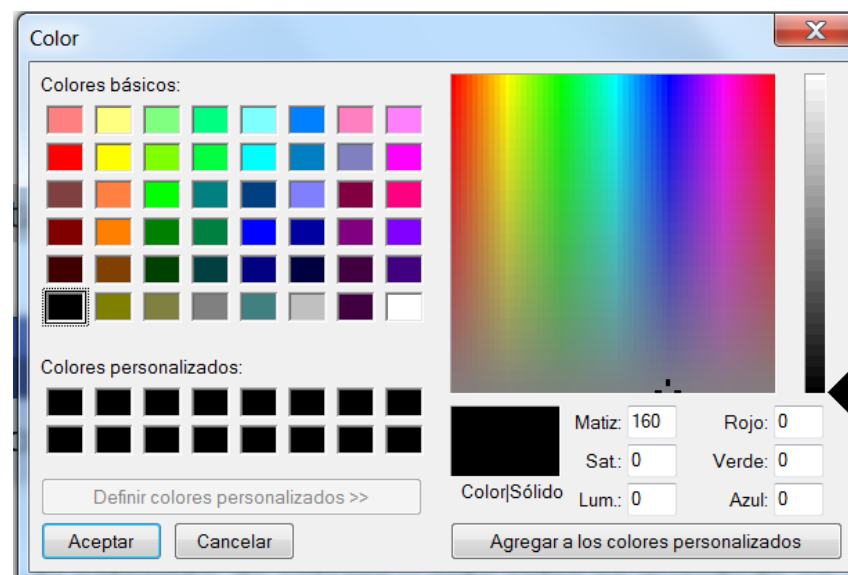
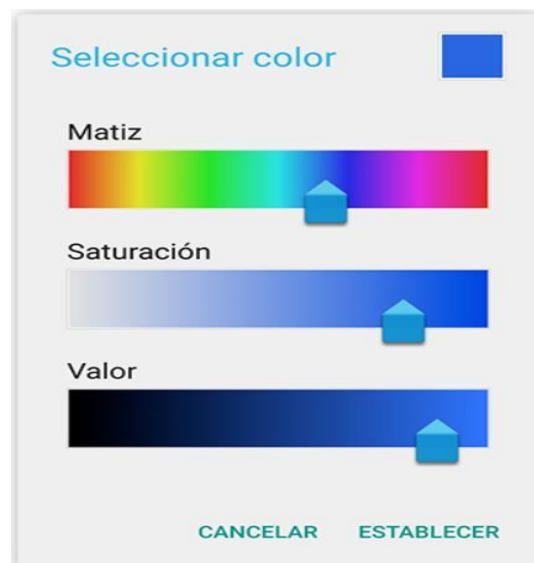


## Campos de entrada de información

### Selección de Color

- La etiqueta que utilizaremos para mostrar la selección de color será `<input>` con el atributo `type="color"`:

Tipo de información a obtener	Etiqueta a utilizar	Ejemplo
Campo de entrada de datos de selección de color	<code>&lt;input type="color"&gt;</code>	<input type="color"/>

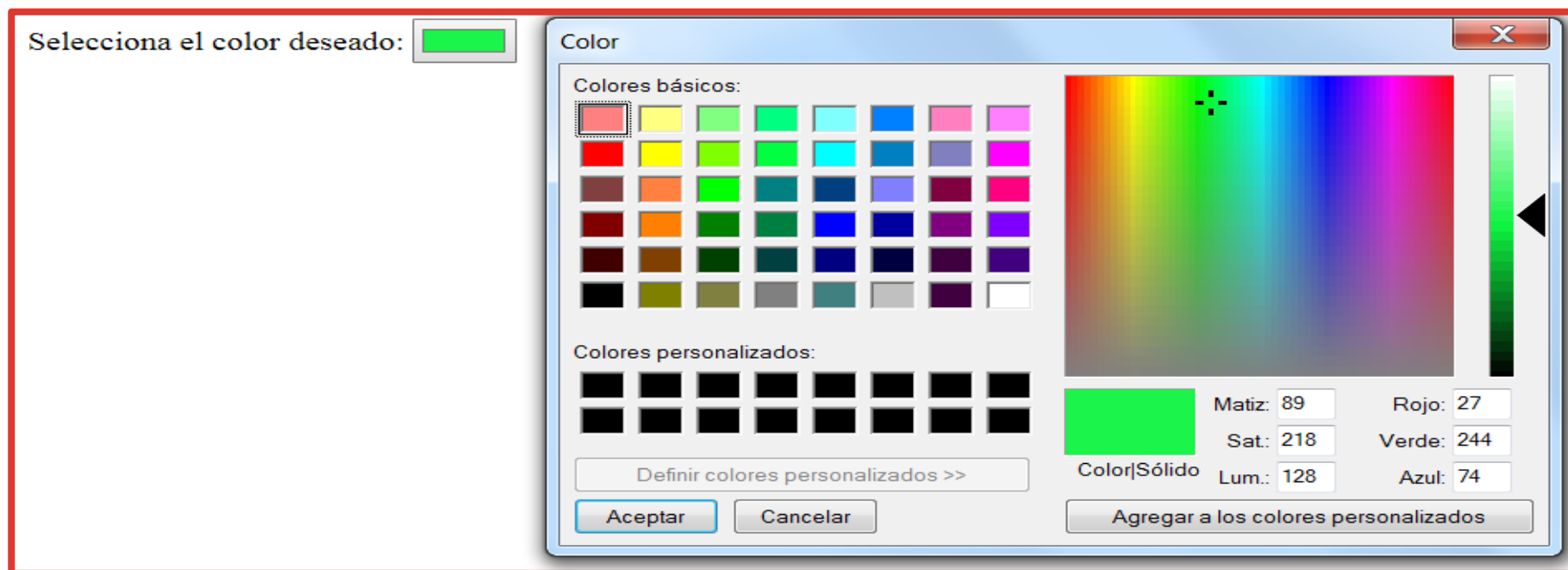


# UA 2.6: Introducción HTML: Formularios

## Campos de entrada de información

### Selección de Color

```
<form name="formulario" method="post" action="http://pagina.com/send.php">  
  <!-- Selección de color -->  
  Selecciona el color deseado:  
  <input type="color" name="color" value="#1BF44A">  
  <!-- Color verde por defecto HEX(1B, F4, 4A) -->  
</form>
```



# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Selección de Archivos

Tipo de información a obtener	Etiqueta a utilizar	Ejemplo
Adjuntar archivo	<code>&lt;input type="file"&gt;</code>	<input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado

- En primer lugar, debemos saber que si queremos adjuntar archivos a nuestro formulario, es necesario indicar en la etiqueta `<form>` el atributo **enctype="multipart/form-data"**. Con esto nos aseguramos que las cabeceras del formulario indican que estamos enviando archivos adjuntos.
- También se nos permite añadir el atributo **accept** a la etiqueta `<input type="file">` a modo de sugerencia, para indicar los formatos de archivos permitidos al usuario en el campo de selección de archivos.

# UA 2.6: Introducción HTML: Formularios

## Campos de entrada de información

### Selección de Archivos

#### Ejemplo básico

```
<form name="formulario" method="post" action="http://pagina.com/send.php"
      enctype="multipart/form-data"> <!-- ¡No olvides el enctype! -->
  <!-- Campo de selección de archivo -->
  <input type="file" name="adjunto" accept=".pdf,.jpg,.png" multiple>
</form>
```



- Si indicamos el atributo `multiple` en la etiqueta `<input>`, se le da la opción al usuario de pulsar la tecla CTRL para adjuntar varios archivos a la vez, y no uno solo.

# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Agrupación de Campos

- En los formularios podemos utilizar una serie de etiquetas que nos ayuden a organizar los elementos reagrupándolos por categorías o temáticas, entre otras cosas.
- Las etiquetas que veremos son las siguientes:

Tipo de información a obtener	Etiqueta a utilizar	Ejemplo
Agrupación visual o temática de campos de entrada	<code>&lt;fieldset&gt;</code>	<input type="text"/>
Leyenda para la etiqueta <code>&lt;fieldset&gt;</code>	<code>&lt;legend&gt;</code>	<div>Leyenda</div>
Relación de campo y texto	<code>&lt;label&gt;</code>	

# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Etiquetas <fieldset> y <legend>

- La primera etiqueta es <fieldset>, un elemento utilizado como etiqueta contenedora para agrupar mediante un trazo simple todos los campos de un formulario que estén relacionados (misma temática o apartado, por ejemplo). La etiqueta <legend>, suele incluirse en el interior de <fieldset> para mostrar un título en el trazo de esa agrupación.

```
<form name="formulario" method="post" action="http://pagina.com/send.php">
  <!-- Agrupación de campos -->
  <fieldset>
    <legend>Datos personales</legend> <!-- Leyenda -->
    <!-- Campo de texto -->
    Nombre de usuario:
    <input type="text" name="nombre" placeholder="Por ej: John Smith">
    <!-- Campo numérico -->
    Edad:
    <input type="number" name="edad" min="18" max="100" placeholder="18">
    <!-- Botón radio de opción -->
    Sexo:
    <input type="radio" name="sexo" value="H"> Hombre
    <input type="radio" name="sexo" value="M"> Mujer
  </fieldset>
</form>
```

Datos personales

Nombre de usuario:  Edad:  Sexo: ☐ Hombre ☐ Mujer

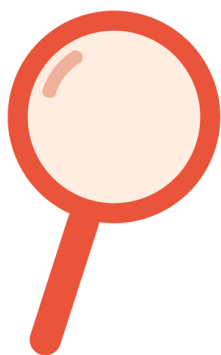
# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Etiqueta <label>

- Podemos utilizar la etiqueta **label** para mejorar la usabilidad de los usuarios.
- Si utilizamos la etiqueta contenedora **<label>**, podemos establecer una relación semántica de un texto con un campo de entrada de datos.
- Al incluir texto y campo de texto dentro de la etiqueta **<label>**, conseguiremos tener el mismo efecto que al pulsar en el campo de texto del propio texto (esto es muy útil también en casillas de verificación).



**Nota:** A las etiquetas **<label>** se le puede indicar opcionalmente un atributo **for** con un nombre específico, y así poder establecer la relación con una etiqueta HTML con **id** con el mismo nombre, sin necesidad de que estén contiguos o adyacentes

# UA 2.6: Introducción HTML: Formularios

## Campos de entrada de información

### Etiqueta <label>

- Veamos el ejemplo anterior, mejorado con las etiquetas <label>:

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

Datos personales

Nombre de usuario:

Edad:

Sexo: ☐ Hombre ☐ Mujer

# UA 2.6: Introducción HTML: Formularios

## Campos de entrada de información

### Botones en los Formularios

- En un formulario, podemos usar varios tipos de botones que actúan sobre él:

Tipo de botón de control	Etiqueta a utilizar	Ejemplo
Botón de envío de formulario	<code>&lt;input type="submit"&gt;</code>	
Botón de envío con imagen	<code>&lt;input type="image"&gt;</code>	
Botón de borrar formulario	<code>&lt;input type="reset"&gt;</code>	
Botón sin funcionalidad	<code>&lt;input type="button"&gt;</code> o <code>&lt;button&gt;</code>	

# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Botón con Envío a Formulario

- El botón `<input type="submit">` sirve para enviar el formulario una vez el usuario ha rellenado todos sus campos y pulsa en este botón. El texto por defecto del botón será **Enviar** (o equivalente según el idioma). Podemos modificar el texto del botón mediante el atributo `value`.

```
<form name="formulario" method="post" action="mailto:raul.rodriguez@universidadeuropea.es">
  <!-- Datos del formulario -->
  Usuario: <input type="text" name="usuario">
  <!-- Botón de envío de formulario -->
  <input type="submit" value="Enviar formulario">
</form>
```

0

```
<form name="formulario" method="post" action="http://pagina.com/send.php">
  <!-- Datos del formulario -->
  Usuario: <input type="text" name="usuario">
  <!-- Botón de envío de formulario -->
  <input type="submit" value="Enviar formulario">
</form>
```

Usuario:

Enviar formulario

## UA 2.6: Introducción HTML: Formularios

### Campos de entrada de información

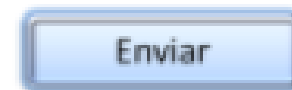
#### Botón con imagen

- El botón `<input type="image">` es un botón que tiene la misma funcionalidad que el botón anterior, pero es un híbrido entre el botón `<input type="submit">` y una etiqueta de imagen `<img>`, puesto que también posee los atributos `src`, `alt`, `width` y `height` para indicar una imagen que actuará de botón:

```
<form name="formulario" method="post" action="http://pagina.com/send.php">  
  <!-- Datos del formulario -->  
  Usuario: <input type="text" name="usuario">  
  <!-- Botón de envío de formulario con imagen -->  
  <input type="image" src="enviar.png" alt="Enviar"  
    width="80" height="28">  
</form>
```



Usuario:



# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Botón de Reset en el Formulario

- El botón `<input type="reset">` nos ofrece la posibilidad de disponer de un botón que al pulsarse borre completamente el formulario y reestablece todos sus campos de entrada de datos a sus **valores por defecto** iniciales.
- Es muy útil si queremos que el usuario disponga de la posibilidad de borrar todos los campos del formulario fácilmente y no tener que hacerlo uno por uno.

```
<form name="formulario" method="post" action="http://pagina.com/send.php">  
  <!-- Datos del formulario -->  
  Usuario: <input type="text" name="usuario">  
  <!-- Botón de envío de formulario -->  
  <input type="submit" value="Enviar formulario">  
  <!-- Botón de reseteo de formulario -->  
  <input type="reset" value="Borrar formulario">  
</form>
```



Usuario:

## UA 2.6: Introducción HTML: Formularios



### Campos de entrada de información

#### Botón sin funcionalidad predefinida

- En los formularios también tenemos el botón `<input type="button">` o la etiqueta `<button>`, que en ambos casos añade un botón sin ninguna funcionalidad por defecto.
- Podemos redefinir la funcionalidad de este botón a través de Javascript con el fin de dotarle de una funcionalidad concreta, por lo que no podríamos hacer mucho sin este lenguaje. A continuación un pequeño ejemplo muy básico:

```
<button onClick="alert('¡Hola, señor!')">Saludar</button>
```

Saludar



# UA 2.6: Introducción HTML: Formularios



## Campos de entrada de información

### Otro tipo de Botones: Funcionalidad Alternativa

- Los botones de envío de formulario habituales funcionan en base a los atributos **action**, **enctype**, **method**, **target** y **novalidate** de la etiqueta **<form>** que los contiene.
- Sin embargo, es posible crear botones de envío alternativos que tengan un comportamiento de envío diferente a los botones por defecto. Para ello, tenemos que crear un botón de envío **<input type="submit">** o **<input type="image">** e incluir los atributos **formaction**, **formenctype**, **formmethod**, **formtarget** y **formnovalidate** respectivamente, según nos interese, ya que sobre escribirán el comportamiento por defecto del formulario:

```
<form name="formulario" method="post" action="http://pagina.com/send.php">
  <!-- Datos del formulario -->
  Usuario: <input type="text" name="usuario">
  <!-- Botón por defecto, hace caso a atributos de <form> -->
  <input type="submit" value="Enviar">
  <!-- Botones con datos de envío alterados -->
  <input type="submit" value="Enviar al dpto. marketing"
    formaction="http://pagina.com/send-mkt.php">
  <input type="submit" value="Enviar al dpto. informática"
    formaction="http://pagina.com/send-sat.php">
</form>
```

Usuario:

## UA 2.6: Introducción HTML: Formularios



### Validación de Formularios en HTML

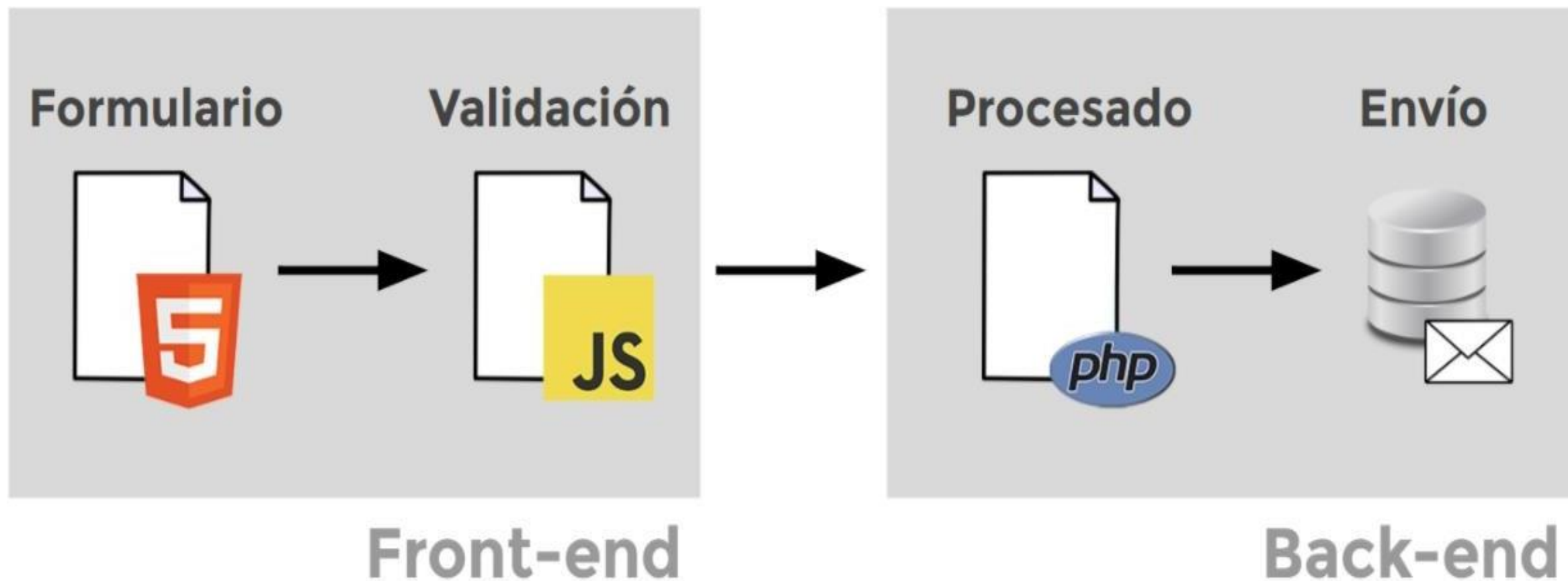
- Cuando creamos un formulario, tenemos que tener en cuenta una cosa: **los usuarios se equivocan siempre al rellenarlo.**
- El caso es que debemos estar preparados y anticiparnos a estos errores, para intentar que los datos lleguen correctamente a su destino y evitar cualquier tipo de moderación o revisión posterior.
- Para ello, utilizamos un proceso automático que se llama **validación**: establece las pautas para que si el usuario introduce alguna información incorrecta, deba modificarla o en caso contrario no podrá continuar ni enviar el formulario correctamente.



# UA 2.6: Introducción HTML: Formularios

## Validación de Formularios en HTML

### Esquema Tradicional de Validación



# UA 2.6: Introducción HTML: Formularios



## Validación de Formularios en HTML

### Tipos de Validaciones

- 1 **El formulario no tiene validación:** El usuario puede escribir la información y el sistema no comprobará los datos, ni realizará ningún tipo de validación. Es el peor escenario posible.
- 2 **El formulario tiene validación sólo en el front-end** (cliente: Los datos son verificados en el navegador del usuario antes de enviarse, pero carecen de validación en el **back-end**)
- 3 **El formulario tiene validación sólo en el back-end:** Con esta forma, garantizamos que un usuario malintencionado no podrá eliminar el proceso de validación, y los datos siempre se comprobarán. Sin embargo, la desventaja de este método es que el usuario puede rellenar un formulario y es necesario que lo envíe, se procese en el back-end y al devolver un error, el usuario tenga que retroceder al formulario y en algunos casos, incluso tener que volver a rellenar todos los campos de nuevo.
- 4- **El formulario tiene validación en el front-end y en el back-end (doble validación):** Es el caso ideal y en este caso, el formulario es sometido a un proceso de validación en la parte del front-end, y si lo supera, vuelve a pasar otro proceso de validación en el back-end. Es el sistema más estricto y seguro.

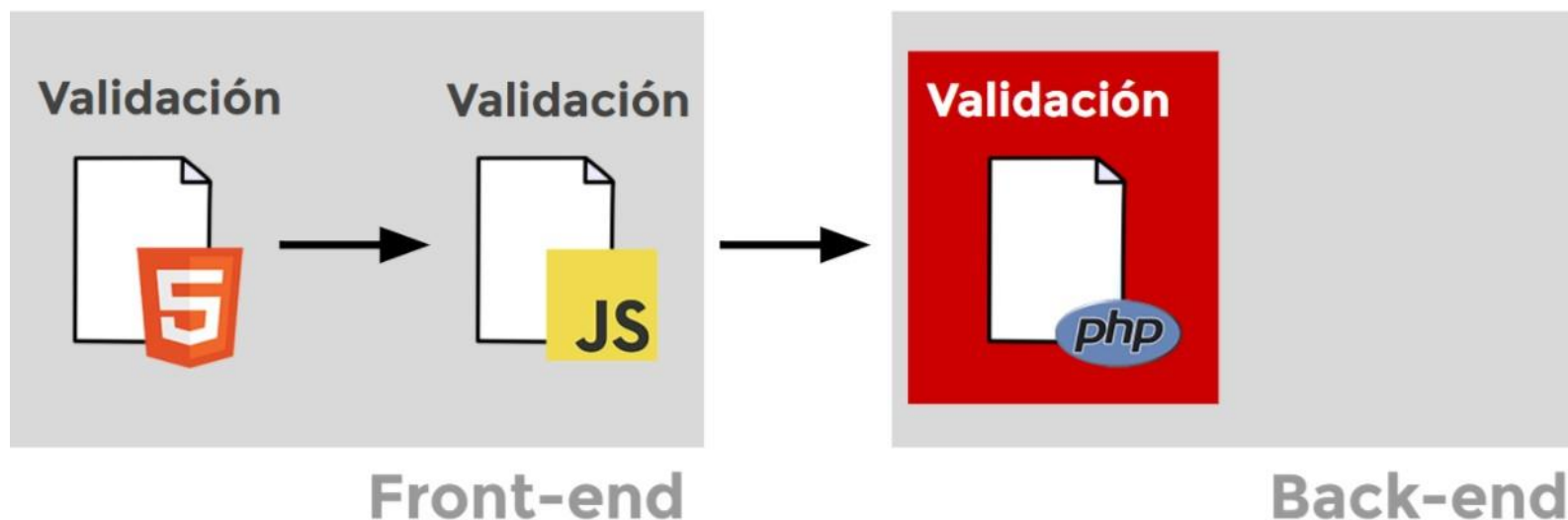
# UA 2.6: Introducción HTML: Formularios



## Validación de Formularios en HTML

### Tipos de Validaciones

- Tradicionalmente, la validación de un formulario se ha hecho siempre en Javascript, sin embargo, HTML5 introduce unos nuevos atributos para formularios que permiten realizar la validación del formulario directamente en HTML5, sin necesidad de recurrir a Javascript (aunque es posible hacerlo si se desea):



- Recordar que siempre se tiene que realizar la validación en el servidor o back-end (en rojo), ya que esta acción puede evitar problemas graves de seguridad.

# UA 2.6: Introducción HTML: Formularios



## Atributos Básicos de Validación en Formularios

Atributo	Valor	Puede actuar sobre...	Descripción
minlength	<u>número</u>	Campos de texto	Establece la longitud mínima del texto requerida.
maxlength	<u>número</u>	Campos de texto	No permite escribir textos superiores a <u>número</u> caracteres.
min	<u>número</u>	Campos numéricos	Establece el número mínimo permitido.
	<u>fecha</u>	Campos de fecha	Establece la fecha mínima permitida.
	<u>hora</u>	Campos de hora	Establece la hora mínima permitida.
max	<u>número</u>	Campos numéricos	Establece el número máximo permitido.
	<u>fecha</u>	Campos de fecha	Establece la fecha máxima permitida.
	<u>hora</u>	Campos de hora	Establece la hora máxima permitida.
step	<u>número</u>	Campos numéricos	Establece el salto de números permitido. Por defecto, 1.
	<u>fecha</u>	Campos de fecha	Establece el salto de días permitido. Por defecto, 1.
	<u>hora</u>	Campos de hora	Establece el salto de segundos permitido. Por defecto, 1.
required		Campos en general	Campo obligatorio. Se debe rellenar para enviar formulario.
disabled		Campos en general	Campo desactivado. No se puede modificar. No se envía.
readonly		Campos en general	Campo de sólo lectura. No se puede modificar. Se envía.

## UA 2.6: Introducción HTML: Formularios

### Atributos básicos de Validación en Formularios

- Con estos atributos, podemos crear validaciones básicas en nuestros campos de entrada de datos, obligando al usuario a rellenar un campo obligatorio, forzando a indicar valores entre un rango numérico o permitiendo sólo textos con un tamaño específico, entre otros.

```
<!Doctype html>
<html>
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" href="new.css" type="text/css" media="all" />
</head>
<body>

<form name="formulario" method="post" action="http://pagina.com/send.php">
  <!-- Nombre de usuario.
       Campo obligatorio, entre 3-20 caracteres -->
  Usuario: <input type="text" name="nombre" placeholder="Por ejemplo, Raul"
              minlength="3" maxlength="20" required>
  <!-- Contraseña.
       Campo obligatorio, mínimo 4 caracteres -->
  Password: <input type="password" name="pass" placeholder="Contraseña"
                  minlength="4" required>

</form>

</body>
</html>
```

```
input:valid, textarea:valid {
  background:green;
}
input:invalid, textarea:invalid {
  background:red;
}
```



# UA 2.6: Introducción HTML: Formularios



## Atributos básicos de Validación en Formularios

- En este ejemplo hemos utilizado pseudoclasas de CSS de validación como son `:valid` e `:invalid` podemos aplicar estilos a los campos `<input>` y `<textarea>` teniendo en cuenta su validación. Aplicando el siguiente código CSS se mostrará un fondo verde o rojo, dependiendo de si cumple o no la validación, respectivamente:

```
<!Doctype html>
<html>
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" href="new.css" type="text/css" media="all" />
</head>
<body>

<form name="formulario" method="post" action="http://pagina.com/send.php">
  <!-- Nombre de usuario.
       Campo obligatorio, entre 3-20 caracteres -->
  Usuario: <input type="text" name="nombre" placeholder="Por ejemplo, Raul"
             minlength="3" maxlength="20" required>
  <!-- Contraseña.
       Campo obligatorio, mínimo 4 caracteres -->
  Password: <input type="password" name="pass" placeholder="Contraseña"
                minlength="4" required>
</form>

</body>
</html>
```

```
input:valid, textarea:valid {
  background:green;
}
input:invalid, textarea:invalid {
  background:red;
}
```



Usuario:  Password:

Usuario:  Password:

# UA 2.6: Introducción HTML: Formularios



## Patrones de Validación en HTML5

- Los patrones de validación están formados por **expresiones regulares**, que consisten en una cadena de texto que representa un posible patrón de coincidencias, que se aplican mediante el atributo **pattern** en los campos que queramos validar.

Expresión regular	Carácter especial	Denominación	Descripción
.	Punto	Comodín	Cualquier carácter (o texto de tamaño 1)
A B	<i>Pipe</i>	Opciones lógicas	Opciones alternativas (o A o B)
C(A B)	Paréntesis	Agrupaciones	Agrupaciones alternativas (o CA o CB)
[0-9]	Corchetes	Rangos de caracteres	Un dígito (del 0 al 9)
[A-Z]			Una letra mayúscula de la A a la Z
[^A-Z]	^ en corchetes	Rango de exclusión	Una letra que no sea mayúscula de la A a la Z
[0-9]*	Asterisco	Cierre o clausura	Un dígito repetido 0 ó más veces (vacío incluido)
[0-9]+	Signo más	Cierre positivo	Un dígito repetido 1 ó más veces
[0-9]{3}	Llaves	Coincidencia exacta	Cifra de 3 dígitos (dígito repetido 3 veces)
[0-9]{2,4}		Coincidencia (rango)	Cifra de 2 a 4 dígitos (rep. de 2 a 4 veces)
b?	Interrogación	Carácter opcional	El carácter b puede aparecer o puede que no
\.	Barra invertida	Escape	El carácter . literalmente (no como comodín)

# UA 2.6: Introducción HTML: Formularios

## Patrones de Validación en HTML5

### Ejemplos de Patrones

- ✓ Tipo de campo: Nombre de usuario
- ✓ Campo obligatorio: **required**.
- ✓ Entre 5-40 caracteres: **minlength="5" maxlength="40"**
- ✓ Sólo se permiten letras (mayúsculas y minúsculas) y números: **pattern="[A-Za-z0-9]+"**

```
<form name="formulario" method="post" action="http://pagina.com/send.php">  
  <!-- Nombre de usuario.  
        Campo obligatorio, entre 5-40 caracteres  
        Sólo se permiten letras y números -->  
  <input type="text" name="nombre" placeholder="Su nombre de usuario"  
        minlength="5" maxlength="40" required pattern="[A-Za-z0-9]+">  
</form>
```

# UA 2.6: Introducción HTML: Formularios



## Patrones de Validación en HTML5

### Ejemplos de Patrones

- Notar que de no incluir los atributos `minlength` y `maxlength`, el usuario no tendría limitación en cuanto al tamaño. Esto también puede incorporarse en la propia expresión regular, y prescindir de dichos atributos

```
<form name="formulario" method="post" action="http://pagina.com/send.php">  
  <!-- Nombre de usuario.  
    Campo obligatorio, entre 5-40 caracteres  
    Sólo se permiten letras y números -->  
  <input type="text" name="nombre" placeholder="Su nombre de usuario"  
    required pattern="[A-Za-z0-9]{5,40}"  
    title="Letras y números. Tamaño mínimo: 5. Tamaño máximo: 40">  
</form>
```

## UA 2.6: Introducción HTML: Formularios

### Patrones de Validación en HTML5

#### Ejemplos de Patrones

- Sin embargo, en este caso, no se limitará al usuario a la hora de escribir, como hace **maxlength**, sino que permitirá al usuario escribir la información que desee y en caso de no pasar la validación, mostrará un mensaje de advertencia y no lo dejará continuar hasta que termine. Podemos ampliar el mensaje de advertencia incluyendo el texto en el atributo **title**.

Su nombre de usuario Enviar

! Completa este campo

required (Chrome, Windows 10)

Marco\$ Enviar

! Utiliza un formato que coincida con el solicitado  
Letras y números. Tamaño mínimo: 5.  
Tamaño máximo: 40

pattern (Chrome, Windows 10)

Marco\$ Enviar consulta

Ajústese al formato solicitado: Letras y números.  
Tamaño mínimo: 5. Tamaño máximo: 40.

pattern (Firefox, Windows 10)

# UA 2.6: Introducción HTML: Formularios



## Patrones de Validación en HTML5

### Ejemplos de Patrones

- En el siguiente caso, se pide al usuario que indique el modelo de coche que posee, en un posible formulario de servicio técnico. Los modelos posibles son A1, A3, A4 y A15. En lugar de mostrar una lista de selección, podemos mostrar un campo de texto y colocar una validación como la siguiente:

```
<form name="formulario" method="post" action="http://pagina.com/send.php">
  <!-- Modelo de coche
        Campo obligatorio, opciones posibles: A1, A3, A4 y A15
        Sólo se permiten letras y números -->
  <input type="text" name="coche" placeholder="Su modelo de coche"
        required pattern="A|a(1|3|4|15)"
        title="Modelos posibles: A1, A3, A4 y A15">
</form>
```