

TEMA 01 – INTRODUCCIÓN A LENGUAJE DE MARCAS

Qué es

- Modo de codificar un documento de texto donde junto al contenido se incluyen etiquetas, marcas o anotaciones
- Fin: añadir información relativa a la estructura del texto o su formato de presentación
- Algunas de las marcas más empleadas se delimitan con los símbolos "<" y ">"
- No es un lenguaje de programación
 - o Puede contener partes de códigos de otros lenguajes de programación

Características generales

- Texto plano: pueden ser editados con cualquier procesador
- Lenguaje compacto: las etiquetas se mezclan con el contenido
- Facilidad de procesa-: fácil desarrollar, comprender y aprender
 - o Especializados el función del tipo de contenido
- Flexibilidad: posibilidad de combinación con otros lenguajes (gráficos vectoriales, servicios web, interfaces...)

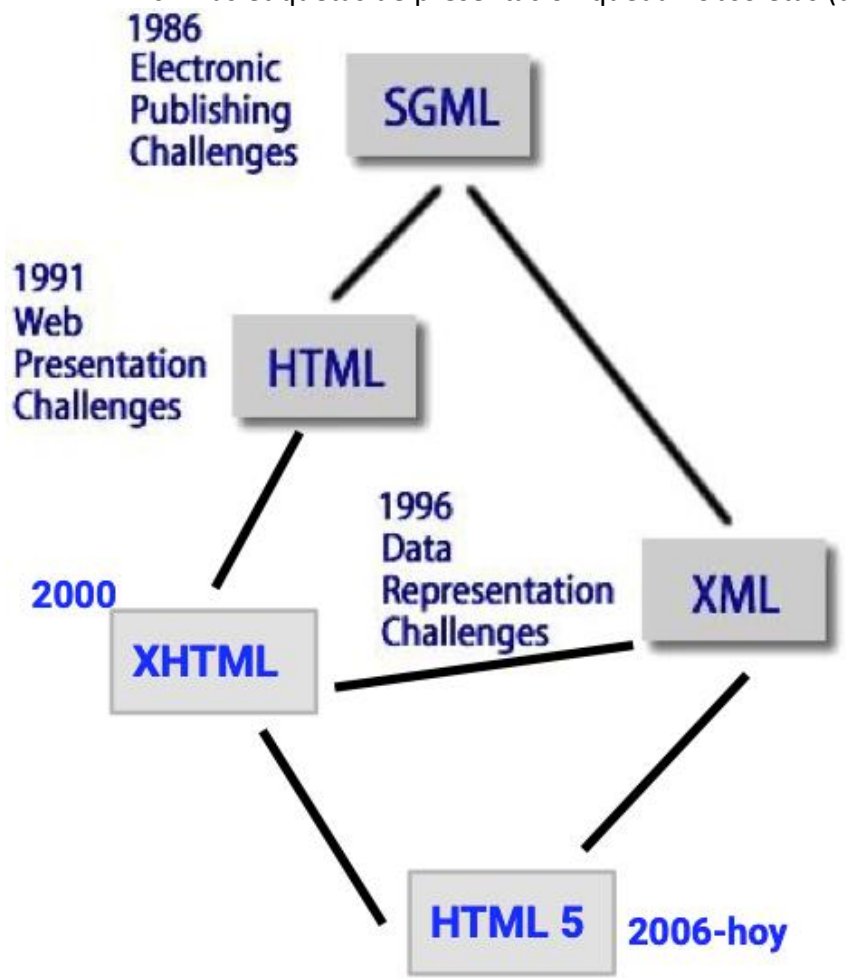
Definición del tipo

- Todo lenguaje de marcas está definido en un documento DTD (Document Type Definition)
- Se establecen:
 - o Marcas y elementos
 - o Etiquetas y atributos
 - o Sintaxis
 - o Normas de uso
- Tipos
 - o Lenguajes de presentación (ej.: Rich Text Format de Microsoft)
 - Define el formato (apariencia) del texto
 - Suelen ocultar las etiquetas y mostrar el texto con su formato
 - o Lenguajes de procedi-s (LaTeX, HTML)
 - También orientado a la presentación
 - El programa que representa el documento debe interpretar las etiquetas para realizar acciones en función de ellas
 - o Lenguajes descriptivos o semánticos (XML)
 - Describen las diferentes partes en las que se estructura
 - Es decir, definen su contenido, pero sin especificar cómo deben representarse

Evolución

- Generalized Markup Language (GML): nace ante la necesidad de gestionar documentación entre distintos departamentos y aplicaciones
- Standard GML (SGML): evolución de 1986 a la estandarización del GML
- HyperText Markup Language (HTML) (1991)
 - o Con el nacimiento del World Wide Web
 - o Combinación de ASCII y SGML
 - o Desventajas
 - No soporta tareas de impresión y diseño
 - No es flexible (etiquetas limitadas)
 - No permite mostrar contenido dinámico
 - La estructura y el diseño están mezclados en el documento

- eXtensible Markup Language (XML) (1996)
 - Estándar del W3C
 - Lenguaje estructural puro
 - Ventajas
 - Permite definir etiquetas propias
 - Permite asignar atributos a las etiquetas
 - Utiliza un esquema para definir de forma exacta las etiquetas y los atributos
 - Estructura y diseño independientes
- eXtensible HyperText Markup Language (2000)
 - para expresar el HTML como un lenguaje XML válido
 - convertido de un lenguaje de marcas de procedi-s a un lenguaje de marcas descriptivo
 - sin mucho éxito
- HTML5
 - HTML semántico
 - Las etiquetas de presentación quedan obsoletas (deprecated)



Sintaxis

- Etiquetas entre símbolos "<" y ">"
- Todas las etiquetas tienen su correspondiente etiqueta de cierre: <h1></h1>
- Etiquetas y atributos en minúsculas
- Valor de los atributos entre comillas simples o dobles

TEMA 02 – INTRODUCCIÓN A LENGUAJE DE MARCAS XML

XML: eXtensible Markup Language

- 1996
- Lenguaje de marcas porque está definido por marcas o etiquetas
- Metalenguaje porque define las reglas a cumplir por un lenguaje de marcas
- Define la estructura de un documento
- Extensión: .xml
- Documento en texto plano
- Objetivo: estandarización del HTML -> XHTML (sin éxito)
- Uso actual:
 - o Intercambio de info entre aplicaciones
 - o Computación distribuida
 - o Documentación empresarial, etc.

Características básicas

- Concebido para trabajar en la web
 - o Compatible con protocolos HTTP
- Todo documento que verifique las reglas de XML está conforme con SGML
- Fáciles de crear
 - o No se requieren conocimientos de programación para tareas sencillas
- Diseño formal y conciso
- Extensible y adaptable a variedad de situaciones
- Orientado a objetos
- Se compone exclusivamente de datos de marcado y datos de carácter entremezclados

Algunos conceptos

- Caracteres y codificación: ISO-8859-1, UTF-8, etc.
- Procesador XML (XML parser)
 - o La aplicación recurre a un procesador XML para leer un documento XML
 - o Este procesador es el que lee, analiza y le pasa la información en un formato estructurado a la aplicación
 - o Etiquetas (tags) o marcas (mark-up)
 - Las etiquetas son marcas que sirven para identificar un contenido concreto del resto del contenido del documento
- Pueden estar formados por 2 partes:
 - o Prólogo (opcional)
 - Declaración XML
 - Declaración del tipo de documento
 - o Ejemplar: contenido del documento

Prólogo

- Debe preceder al ejemplar del documento
- Facilita el procesamiento de la información del ejemplar
- Declaración XML
 - o Opcional, pero obligatorio si se escribe el atributo "versión"
 - o Encoding y standalone son opcionales
 - o Encoding: siempre después de versión y standalone el último
 - o Standalone indica si el documento XML depende de un DTD externo (sirve a nivel de validación)

- Atrib. **version** `<?xml version="1.0" ?>`
- Atrib. **encoding** `<?xml version="1.0" encoding="iso-8859-1" ?>`
- Atrib. **standalone** `<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>`

- La declaración define el tipo de documento
 - Se define el tipo de documento y dónde encontrar la información sobre su DTD
 - Mediante un identificador público (PUBLIC) que hace referencia a dicho DTD
 - O mediante Identificador Universal de Recursos (URI) precedido por la palabra SYSTEM
 - Este archivo valida y define los datos que contiene nuestro documento XML
 - DTD en archivo externo:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE Persona SYSTEM "persona.dtd">
```

DTD embebido en documento

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE persona [
  <!ELEMENT persona (nombre,apellido,ciudad)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT apellido (#PCDATA)>
  <!ELEMENT ciudad (#PCDATA)>
]>

<persona>
  <nombre>Carlos</nombre>
  <apellido>Carnero</apellido>
  <ciudad>Valencia</ciudad>
</persona>
```

Contenido del
fichero DTD si
fuera externo.

Ejemplar y elementos

- El ejemplar contiene los datos reales del documento XML (contenido)
- Formado por un elemento raíz que a su vez contendrá otros elementos anidados
- Elemento: componente lógico de un documento XML
 - Comienza por una etiqueta de apertura y termina con la etiqueta de cierre correspondiente (o bien consiste en una única etiqueta vacía)
 - El contenido de un elemento es todo lo que se encuentra entre las etiquetas
 - También existen elementos vacíos

- Elemento con texto como contenido:

```
<autor>EsCif</autor>
```

- Elementos anidados:

```
<falla>
```

```
<autor>EsCif</autor>
```

```
</falla>
```

- Elemento vacío o sin contenido:

```
<ingreso />
```

o también puede escribirse así:

```
<ingreso></ingreso>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
```

```
<!DOCTYPE persona>
```

```
<persona>
```

```
<nombre>Carlos</nombre>
```

```
<apellido>Carnero</apellido>
```

```
<nacimiento>
```

```
<anyo>1956</anyo>
```

```
<mes>Abril</mes>
```

```
<dia>13</dia>
```

```
</nacimiento>
```

```
<ciudad>Alicante</ciudad>
```

```
</persona>
```

Busquemos prólogo,
ejemplar, el elemento
raíz y resto de
elementos

Atributos

- Componente de las etiquetas que consiste en una pareja nombre/valor
- Se puede encontrar en las etiquetas de apertura o vacías
- No puede +1 atributo con mismo nombre por etiqueta
- Sintaxis: nombreAtributo="valorAtributo"

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
```

```
<!DOCTYPE persona>
```

```
<persona nombre="Alfonso" apellidos="García Fernández">
```

```
<nacimiento dia='13' mes='mar' anyo='2019'></nacimiento>
```

```
<ciudad cp="46001">Valencia</ciudad>
```

```
</persona>
```

Definir atributos en el DTD

- Usaremos la etiqueta !ATTLIST indicando a continuación el nombre del elemento al que pertenecen los elementos que se van a definir

```
<!ELEMENT persona (nombre,apellido,ciudad)>
```

```
<!ELEMENT nombre (#PCDATA)>
```

```
<!ELEMENT apellido (#PCDATA)>
```

```
<!ELEMENT ciudad (#PCDATA)>
```

```
<!ATTLIST ciudad cp CDATA #REQUIRED
```

```
provincia CDATA "Valencia">
```

TEMA 03 – INTRODUCCIÓN A LENGUAJE DE MARCAS HTML

DTD de un documento HTML

- DTD público
- Su metaetiqueta DOCTYPE se ha simplificado al extremo
- Antes: `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`
- Ahora: `<!DOCTYPE html>`

Sintaxis básica

- `<>`
- Etiqueta cierre (menos img, br, hr e input)
- Etiquetas y atributos en minúsculas
- Comentarios entre `<!-->`



`<p class="foo">Esto es un párrafo</p>`

Estructura de un documento HTML5

- Prólogo
 - o DTD
- Ejemplar
 - o Cabecera (head)
 - Title
 - Meta
 - Estilos (mejor CSS)
 - Enlaces (como para el CSS)
 - o Cuerpo
 - Contenido
 - Scripts (enlaces)

```
2 <html lang="es">
3   <head>
4     <title>Mi primera pagina web</title>
5   </head>
6   <body>
7   </body>
8 </html>
```

- o Elemento raíz: `<html></html>`

Marcas de formato

- Encabezados: <h1></h1>... <h6></h6>
- Párrafos: <p>Lorem</p>
- Salto de línea:

- Separador horizontal: <hr />
- Comentarios: <!-- ... -->
- Marcas de lista: ...
 - o Ordenadas:
 - o No ordenadas:
- Listas de definición
 - o <dl>
 - <dt>
 - <dd>
- Enlace: <a> ...
 - o Enlace externo:
Ir a Google
 - o Enlace a otro documento de la web:
Contacto
 - o Ancla a sección del mismo documento:
Aficiones
 - Para indicar el párrafo al que desas enlazar:
aficiones
 - o Enlace a correo electrónico:
Mandar email a micorreo@dominio.es
 - o Enlace nulo:
Sin salida
- Inserción imagen:

TEMA 04 – ETIQUETAS HTML

Marcas de aspecto

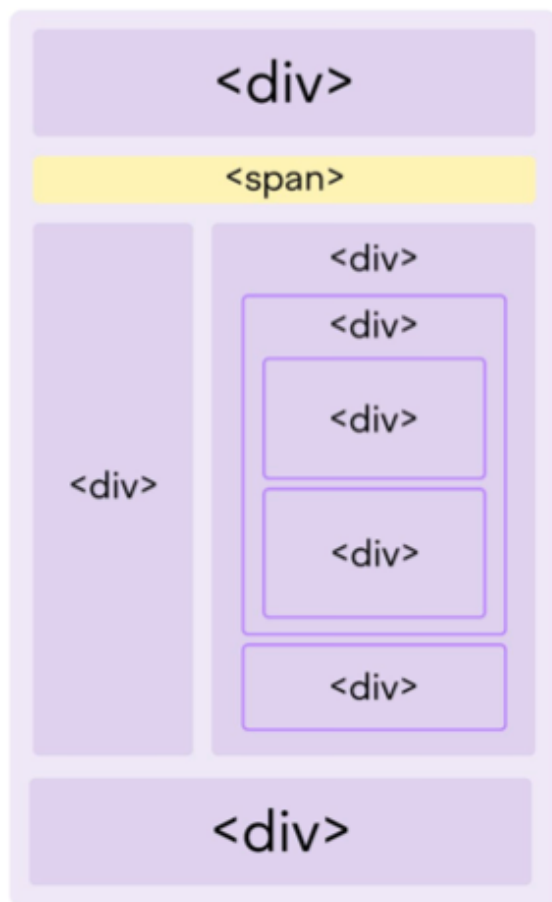
- Énfasis: ` ... `
- Negrita: ` ... `
- Marcado: `<mark> ... </mark>`
- Entrecorillado: `<q> ... </q>`
- Preformateado: `<pre> ... </pre>`
- Otras:
 - o Código: `<code> ... </code>`
 - o Subrayado: `<ins> ... </ins>`
 - o Tachado: ` ... `
 - o Subíndice: `_{...}`
 - o Superíndice: `^{...}`
 - o Texto pequeño: `<small> ... </small>`
 - o Texto grande: `<big> ... </big>`
- Se pueden anidar

Tablas

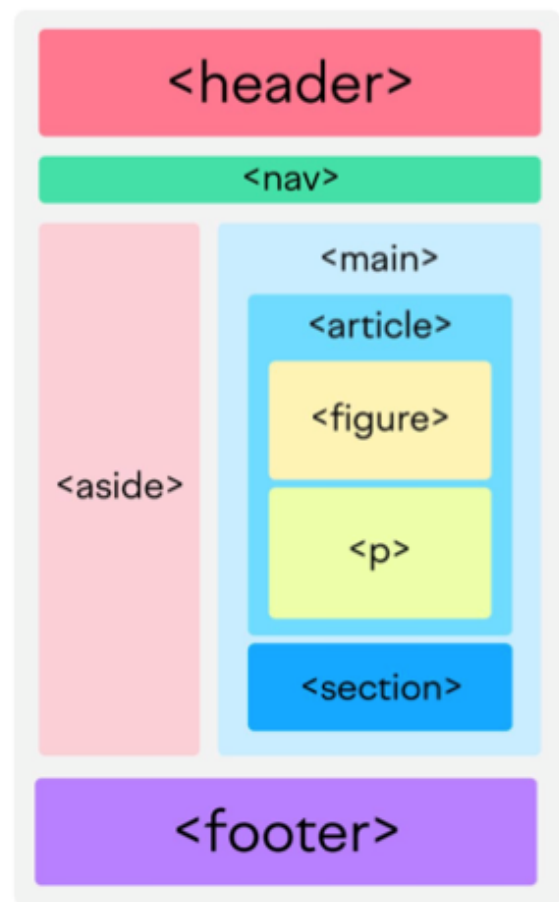
```
<table>
  <thead>
    <tr>
      <th>Name</th>
      <th>Instrument</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <th>Name</th>
      <th>Instrument</th>
    </tr>
  </tfoot>
```

```
<tbody>
  <tr>
    <td>John Lennon</td>
    <td>Rhythm Guitar</td>
  </tr>
  <tr>
    <td>Paul McCartney</td>
    <td>Bass</td>
  </tr>
  <tr>
    <td>George Harrison</td>
    <td>Lead Guitar</td>
  </tr>
  <tr>
    <td>Ringo Starr</td>
    <td>Drums</td>
  </tr>
</tbody>
</table>
```


Non-Semantic HTML



Semantic HTML



- Iguales a una etiqueta div genérica
- Nombres simple- para dar sentido a su contenido

Header: nombre del sitio y logotipo

Footer: detalles legales, copyright, etc.

Nav: menu principal (puede ir dentro del header)

Main

- Contenido importante o principal
- No descendiente de un elemento que no sea el body

Section

- Contenido relacionado entre sí
- Normal- contiene una o varias etiquetas article

Article

- Información que tiene estructura común dentro de su section
- Todas las etiquetas article dentro de una etiqueta section tienen la misma estructura

Aside

- Contenido secundario (anuncios, enlaces externos, cita del texto...)
- Normalmente en los lados
- Recoge contenido secundario con respecto a su etiqueta madre (body, section o article)

TEMA 06 – FORMULARIOS

Formularios

- Para recoger datos de los usuarios (pedido, encuesta, preguntas...)
- Una vez enviados, se arranca un programa que recibirá los datos y hará el tratamiento correspondiente
- Se necesitan algunas nociones de programación (PHP o JS)
- Formato por etiquetas, campos de texto, menús y botones
- HTML5: validaciones nativas y nuevas etiquetas

Etiquetas

- Form
 - `<form> ... </form>`
 - Para crear el formulario
 - Encierra todos los elementos del formulario
 - Atributos:
 - `id="x"`: representa el id o nombre del formulario para el CSS o scripts
 - `action="url"`: indica la URL donde se enviarán los datos y que se encargará de procesarlos
 - Esta aplicación también se encarga de generar la respuesta que muestra el navegador
 - `method="x"`: especifica el método HTTP que se empleará para enviar los datos
 - `get`: por defecto
 - No adjuntos
 - 500 bytes máximo
 - Se ve en la barra de direcciones del navegador
 - Para formularios que no modifican la información (ej: formulario de búsqueda)
 - `post`:
 - `enctype="application/x-www-form-urlencoded"` o `"multipart/form-data"`
 - Tipo de codificación empleada al enviar el formulario al servidor
 - Solo se indica explícita- en los formularios que permiten adjuntar archivos
 - `accept="tipo_de_contenido_ MIME"`: lista separada por comas de todos los tipos de archivos aceptados por el servidor desde el formulario
 - La mayoría utilizar solo id, action y method
- Input
 - `<input />`
 - La mayoría de controles del formulario
 - Definición y lista de atributos muy extensa
 - Atributos
 - `'src'`: permite crear botones con imágenes (ruta de la imagen)
 - `'step'` (HTML5): permite establecer la cantidad de valores posibles dentro de un rango
 - `'value'` indica el valor inicial del control
 - `'autofocus'` (HTML5): fuerza el foco una vez carga la página

- En JS igual con la función focus()
- 'required' (HTML5): obligatorio
- 'placeholder' (HTML5): texto por defecto hasta tener el foco
- 'autocomplete' (HTML5): activa o desactiva (on/off) el auto completado en base a los textos introducidos anterior- por el usuario
 - Puede usarse como atributo para un campo o para el form
- 'alt': cuando el tipo es una imagen y no se puede cargar en la interfaz de usuario, se muestra este texto alternativo
- 'checked': para indicar si el elemento está seleccionado por defecto en los controles checkbox y radiobutton
- 'disabled': booleano que indica que está desactivado (no admite entrada de datos) y no se envía al servidor
- 'readonly': booleano que indica que el valor es solo lectura
- 'formenctype' (HTML5): indica el modo de envío
 - 'application/x-www-form-urlencoded'
 - 'multipart/form-data'
 - 'text/plain'
- 'formaction' (HTML5): dirección URL que se utiliza al activar el formulario
- 'height' (HTML5): altura del elemento tipo image
- 'width' (HTML5): anchura del elemento tipo image
- 'size': tamaño inicial del control
 - En text/password: cantidad de caracteres, en resto píxeles
- 'max' (HTML5): valor máximo que se puede entrar
- 'min' (HTML5): valor mínimo que se puede entrar
- 'maxlength': longitud máxima para text y password
- 'formnovalidate' (HTML5): valor booleano que indica que el formulario no se valida antes de enviarse
- 'multiple' (HTML5): valor booleano que indica que se permite la selección de valores múltiples
- 'name': nombre que identifica al campo de datos (importante para identificación por el servidor)
- 'pattern' (HTML5): expresión regular que se utiliza para validar entradas de datos
 - Permite hacer validaciones muy complejas y utilizando correcta- puede ahorrar muchas líneas de código
- Elementos (<input type="valor" />)

Value		Descripción
button		Define un botón que se pulse (en su mayoría se utiliza con un JavaScript para activar una secuencia de comandos)
checkbox		Define una casilla de verificación
color	New	Define un selector de color
date	New	Define un control de fecha (año, mes y día (no hay tiempo))
datetime	New	Define un control de fecha y hora (año, mes, día, hora, minuto, segundo y fracción de segundo, basado en la zona horaria UTC)
datetime-local	New	Define un control de fecha y hora (año, mes, día, hora, minuto, segundo y fracciones de segundo (sin zona horaria))
email	New	Define un campo para una dirección de e-mail
file		Define un campo de archivo de selección y un "Examinar ..." botón (para la carga de archivos)
hidden		Define un campo de entrada oculto
image		Define una imagen como botón submit
month	New	Define un mes y control del año (sin zona horaria)
number	New	Define un campo para introducir un número
search	New	Define un campo de texto para introducir una cadena de búsqueda
submit		Define un botón de envío
tel	New	Define un campo para introducir un número de teléfono
text		Default. Define un campo de texto de una sola línea (ancho predeterminado es de 20 caracteres)
time	New	Define un control para entrar en un tiempo (sin zona horaria)
url	New	Define un campo para introducir una URL
week	New	Define a la semana y el control al año (no zona horaria)
password		Define un campo de contraseña (caracteres se enmascaran)
radio		Define un botón de opción
range	New	Define un control para introducir un número cuyo valor exacto no es importante (como un control deslizante)
reset		Define un botón de reinicio (restablece todos los valores del formulario a valores por defecto)

- Todos:
 - <https://mardeasa.es/descargas/formularios/Formularios-camposHTML5/Formularios-camposHTML5.html>
- Text
 - Pedir datos de entrada relativamente cortos
 - 'name': identificador donde se guardará el valor del usuario y como referencia en scripts (más recomendado id para esto)
 - 'value': valor predeterminado en el campo de entrada
 - 'size': longitud visible
 - 'maxlength': longitud máxima de caracteres

```
<input type="text" name="nombre" value="Introduce tu nombre" size="30" maxlength="40"/>
```

Nombre

- Password:
 - Igual que text, pero aparece oculto

- No codificado, solo oculto en pantalla
- Checkbox
 - Opciones individuales
 - Cada uno es independiente del resto
 - Se utilizan cuando se pueden activar varias opciones relacionadas, pero no excluyentes
 - Aunque resulta redundante que ID y valor sea iguales, es obligatorio porque los atributos en XHTML no pueden tener valores vacíos

Deportes practicados:

```
<input type="checkbox" value="futbol" id="futbol" name="deportes" checked>
<label for="futbol">Fútbol </label>
<input type="checkbox" value="baloncesto" id="baloncesto" name="deportes">
<label for="baloncesto">Baloncesto </label>
<input type="checkbox" value="ciclismo" id="ciclismo" name="deportes">
<label for="ciclismo">Ciclismo </label>
```

- Radio
 - Mutuamente excluyentes
 - 'name' para relacionarlos

Sexo


```
<input type="radio" name="sexo" value="hombre" checked="checked" /> Hombre <br/>
<input type="radio" name="sexo" value="mujer" /> Mujer
```

- Submit
 - Enviar el formulario al servidor
 - Si no hay atributo 'value', el navegador muestra el texto predefinido 'Enviar consulta'

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

- Image
 - Aspecto del botón cambiado por una imagen
 - Permite personalizar la estética de los botones
 - Ralentiza la carga

```
<input type="image" name="enviar" src="disco.jpg">
```

- Reset
 - Antes era muy popular
 - Borra todos los datos
 - 'value' por defecto: Restablecer

```
<input type="reset" name="limpiar" value="Borrar datos del formulario" />
```

- Button
 - Uso genérico
 - Se le pueden asociar distintos eventos por script
 - Uso frecuente: comprobar coherencia y validez de los datos para versiones anteriores a HTML5

```
<input type="button" name="botonAceptar" value="Aceptar" onclick="validaDatos()"/>
```

- Hidden
 - No se muestra en pantalla
 - A veces es preciso enviar información sin que el usuario tenga que introducirla (número de sesión sin que sea visible por el usuario)

```
<input type="hidden" name="url_previa" value="/articulo/primer.html" />
```

- File
 - Obligatorio 'enctype' en el <form>: multipart/form-data:

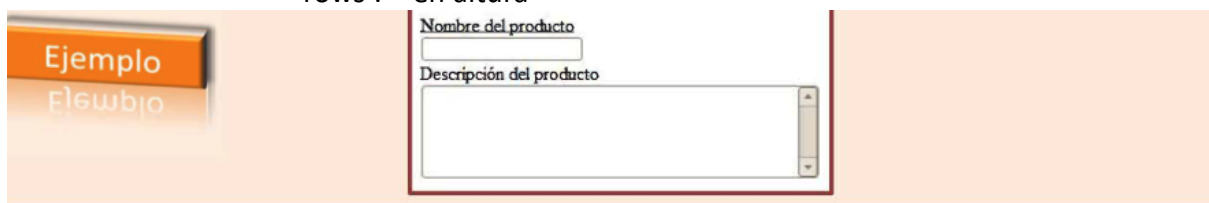
```
<form action="..." method="post" enctype="multipart/form-data">
```

- Accept para filtrar los tipos de archivos:

separados por comas: un tipo MIME de medio de Internet; una extensión de archivo (como ".jpg" o ".pdf"); o una de las cadenas "audio/*", "video/*" e "image/*" representando archivos de sonido, video e imagen, respectivamente.

```
<input type="file" name="imagen_subida" accept="image/png, .jpeg, .jpg, image/gif">
```

- Textarea
 - <textarea></textarea>
 - Caja de filas y columnas en las cuales se introduce texto
 - Campo de texto multilínea
 - Textos largos (descripciones, etc.)
 - No límite caracteres
 - Atributos:
 - 'cols': anchura en columnas visibles en número de caracteres
 - 'rows': = en altura



```
<label for="nombre">Nombre del producto</label> <br/>
<input type="text" id="nombre" name="nombre" value="" />
```

```
<label for="descripcion">Descripción del producto</label> <br/>
<textarea id="descripcion" name="descripcion" cols="40" rows="5"></textarea>
```

- Button
- Select
 - <select> ... </select>
 - Con los <option> ... </option> en su interior
 - Opcionalmente <optgroup> ... </optgroup>
 - Todo 'select' al menos un 'option'
 - 'optgroup' agrupa opciones
 - Atributos del 'option':
 - 'selected': preseleccionada inicial-
 - 'value': valor inicial de la opción a visualizarse
 - Si no se especifica, es el texto entre las etiquetas
 - Obligatorio, ya que es el dato que se envía al servidor
 - 'label': permite especificar un rótulo más corto para la opción


```

<label for="so">Sistema operativo</label> <br>
  <select id="so" name="so">
    <option value="" selected>- selecciona -</option>
    <option value="windows">Windows</option>
    <option value="mac">Mac</option>
    <option value="linux">Linux</option>
    <option value="otro">Otro</option>
  </select>

<label for="so2">Sistema operativo</label> <br>
  <select id="so2" name="so2" size="5">
    <option value="windows" selected>Windows</option>
    <option value="mac">Mac</option>
    <option value="linux">Linux</option>
    <option value="otro">Otro</option>
  </select>

<label for="so3">Sistema operativo</label> <br>
  <select id="so3" name="so3" size="5" multiple="multiple">
    <option value="windows" selected>Windows</option>
    <option value="mac">Mac</option>
    <option value="linux">Linux</option>
    <option value="otro">Otro</option>
  </select>

```

- Atributos del 'outgroup': label



```

<label for="programa">Programa seleccionado</label> <br>
<select id="programa" name="programa">
  <optgroup label="Sistemas Operativos">
    <option value="Windows" selected>Windows</option>
    <option value="Mac">Mac</option>
    <option value="Linux">Linux</option>
    <option value="Other">Otro</option>
  </optgroup>
  <optgroup label="Navegadores">
    <option value="Internet Explorer" selected>Internet Explorer</option>
    <option value="Firefox">Firefox</option>
    <option value="Safari">Safari</option>
    <option value="Opera">Opera</option>
    <option value="Other">Otro</option>
  </optgroup>
</select>

```

- Fieldset

- <fieldset> ... </fieldset>: agrupa campos del formulario
- <legend> --- </legend>: nombre a cada grupo

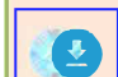
seleccionar un valor
El segundo tipo de select
sólo permite seleccionar
pero muestra varios
Por último, el tercer
desplegable es
muestra varios valores
realizar selecciones

Sistema operativo
- selecciona -

Sistema operativo
Windows
Mac
Linux
Otro

Sistema operativo
Windows
Mac
Linux
Otro

Programa seleccionado
Linux
Sistemas Operativos
Windows
Mac
Linux
Otro
Navegadores
Internet Explorer
Firefox
Safari
Opera
Otro



Formulario estructurado

Datos personales	
Nombre	<input type="text"/>
Apellidos	<input type="text"/>
DNI	<input type="text"/>
Datos de conexión	
Nombre de usuario	<input type="text"/>
Contraseña	<input type="password"/>
Repite la contraseña	<input type="password"/>

- Label
 - `<label></label>`
 - Para establecer el título de cada campo del formulario
 - El HTML queda mejor estructurado y mejora su accesibilidad
 - Al pulsar el label, el puntero se posiciona para escribir su input

```
<label for="nombre">Nombre</label> <br/>
<input type="text" id="nombre" name="nombre" value="" />
```
- Datalist
- Keygen
- Output
-