

## Ejercicio 1

### 1. Qué es HTML, cuando fue creado, cuáles fueron las distintas versiones y cuál es la última?

El origen de HTML se remonta a 1980, cuando el físico Tim Berners-Lee, trabajador del CERN (Organización Europea para la Investigación Nuclear) propuso un nuevo sistema de "hipertexto" para compartir documentos.

Los sistemas de "hipertexto" habían sido desarrollados años antes. En el ámbito de la informática, el "hipertexto" permitía que los usuarios accedieran a la información relacionada con los documentos electrónicos que estaban visualizando. De cierta manera, los primitivos sistemas de "hipertexto" podrían asimilarse a los enlaces de las páginas web actuales.

Tras finalizar el desarrollo de su sistema de "hipertexto", Tim Berners-Lee lo presentó a una convocatoria organizada para desarrollar un sistema de "hipertexto" para Internet. Después de unir sus fuerzas con el ingeniero de sistemas Robert Cailliau, presentaron la propuesta ganadora llamada WorldWideWeb (W3).

Para publicar información y distribuirla globalmente, se necesita un lenguaje entendido universalmente, una especie de lengua franca de publicación que todas las computadoras puedan comprender potencialmente. El lenguaje de publicación usado por la World Wide Web es el HTML (acrónimo de HyperText Markup Language, Lenguaje para el Formato de Documentos de Hipertexto). El HTML da a los autores las herramientas para:

- Publicar documentos en línea con encabezados, textos, tablas, listas, fotos, etc.
- Obtener información en línea a través de vínculos de hipertexto, haciendo clic con el botón de un ratón.
- Diseñar formularios para realizar transacciones con servicios remotos, para buscar información, hacer reservas, pedir productos, etc.
- Incluir hojas de cálculo, videoclips, sonidos, y otras aplicaciones directamente en sus documentos.

Los estándares oficiales HTML son el HTML 2.0, el HTML 3.2, el HTML 4.0, el HTML 4.01 y el HTML 5. El HTML 5 es la última especificación oficial y se espera que continúe evolucionando a lo largo de los próximos años. El XHTML, una forma más avanzada del HTML que se suponía iba a sustituir a éste, podemos considerar que ha quedado integrado dentro del HTML 5.

Versiones de HTML:

HTML 2.0	En 1995 se publica el estándar HTML 2.0. A pesar de su nombre, HTML 2.0 es el primer estándar oficial de HTML, es decir, el HTML 1.0 no existió como estándar. HTML 2.0 no soportaba tablas. Se simplificaba al máximo la estructura del documento para agilizar su edición, donde la declaración explícita de los elementos body, html y head es opcional.
----------	--

HTML 3.2	La versión HTML 3.2 se publicó en 1997 y es la primera recomendación de HTML publicada por el W3C (Consortio internacional). Esta revisión incorporó los últimos avances de las páginas web desarrolladas hasta 1996, como applets de Java y texto que fluye alrededor de las imágenes.
HTML 4.01	La última especificación oficial de HTML se publicó en diciembre de 1999 y se denomina HTML 4.01. Desde la publicación de HTML 4.01, el W3C se centró en el desarrollo del estándar XHTML. Por este motivo, en el año 2004, las empresas Apple, Mozilla y Opera mostraron su preocupación por la falta de interés del W3C en HTML y decidieron organizarse en una nueva asociación llamada WHATWG (Web Hypertext Application Technology Working Group) que comenzó el desarrollo del HTML 5, cuyo primer borrador oficial se publicó en enero de 2008. Debido a la fuerza de las empresas que forman el grupo WHATWG y a la publicación de los borradores de HTML 5.0, en marzo de 2007 el W3C decidió retomar la actividad estandarizadora de HTML, dentro del cual decidió integrar el XHTML.
HTML 5 HTML 5.1 HTML 5.2	El consorcio internacional W3C, después de una evolución de varios años, liberó el HTML 5 como estándar oficial a finales de octubre de 2014. HTML 5 incorpora nuevos elementos no contemplados en HTML 4.01. Hay diversos cambios respecto a HTML 4.01. Hay nuevas etiquetas, se introduce la posibilidad de introducir audio y video de forma directa en la web sin necesidad de plugins o complementos en los navegadores, y otras novedades. El W3C irá lanzando progresivamente nuevas evoluciones del HTML 5.

## 2. ¿Cuáles son los principios básicos que el W3C recomienda seguir para la creación de documentos con HTML?

Se recomienda seguir los siguientes principios para la creación de documentos HTML:

**1. Separar estructura y presentación:** El HTML tiene sus raíces en SGML, que siempre ha sido un lenguaje para la especificación de código estructural. A medida que el HTML madura, un número cada vez mayor de sus elementos y atributos presentacionales ha sido reemplazado por otros mecanismos, en particular las hojas de estilo. La experiencia ha demostrado que separando la estructura de la presentación se reduce el coste de servir a un amplio espectro de plataformas, medios, etc. y se facilitan las revisiones del documento.

**2. Considerar la accesibilidad universal a la Web:** Para hacer la Web más accesible a todos, en especial a aquéllos con discapacidades, los autores deberían considerar cómo pueden representarse sus documentos en diferentes plataformas: navegadores basados en voz, lectores braille, etc. No estamos recomendando a los autores que limiten su creatividad, sólo que consideren representaciones alternativas de sus diseños. El HTML ofrece un número de mecanismos con este fin (p.ej., el atributo alt, el atributo accesskey, etc.).

Además de esto, los autores deberían recordar que sus documentos pueden llegar a una audiencia muy lejana con diferentes computadoras y configuraciones. Para que los documentos sean correctamente interpretados, los autores deberían incluir en sus

documentos información sobre el idioma natural y la dirección del texto, cómo está codificado el documento, y otras cuestiones relacionadas con la internacionalización.

**3. Ayudar con la representación incremental:** Mediante un diseño cuidadoso de las tablas y haciendo uso de las nuevas características de representación incremental, los autores pueden ayudar a los agentes de usuario a representar los documentos más rápidamente.

### **3. En las Especificaciones de HTML, ¿cuándo un elemento o atributo se considera desaprobado? ¿y obsoleto?**

Los tags y atributos desaprobados son partes del estándar HTML cuyo uso ya no es recomendado. Esto sucede dado que el estándar es actualizado de vez en cuando (por ejemplo, de la versión 3.0 a la 4.0) y alguno tags o características del lenguaje son agregada, removidas o desaprobadas. Debido a ello, los autores de documentos HTML son provistos de un mayor número de herramientas cuando un nuevo elemento es agregado, pero también son alentados a desechar aquellos que han sido reemplazados o que se vuelven obsoletos. La decisión de usar tags y/o atributos desaprobados o no, es librada a la consideración del autor.

Para preservar la compatibilidad, en favor de la satisfacción de sus clientes, muchos fabricantes de navegadores proveen soporte para los elementos desaprobados, aunque pasado un determinado tiempo lo retiran. La recomendación general es intentar otras formas de lograr los efectos de los elementos desaprobados, siempre que sea posible.

Con el paso y transición de versiones anteriores (HTML4 o XHTML) a HTML5, hay muchas etiquetas HTML que han sido marcadas como obsoletas y se recomienda dejar de utilizarlas cuanto antes. Es posible que algunas de ellas aún funcionen en navegadores actuales, pero llegará un momento que dejarán de ser soportadas por el navegador.

Por ejemplo, la etiqueta obsoleta “<bgsound>” que especifica un sonido de fondo, se recomienda cambiarla por la etiqueta “<audio>”. Existen muchas etiquetas que son marcadas como obsoletas.

También existen atributos que son marcados como obsoletos. Por ejemplo, el caso del atributo “language” que pertenece a “<script>” e indica el lenguaje utilizado, se recomienda cambiarlo por “type”.

#### 4. Qué es el DTD y cuáles son los posibles DTDs contemplados en la especificación de HTML 4.01?

Una DTD es un documento que define la estructura de un documento XML: los elementos, atributos, entidades, notaciones, etc, que pueden aparecer, el orden y el número de veces que pueden aparecer, cuáles pueden ser hijos de cuáles, etc. El procesador XML utiliza la DTD para verificar si un documento es válido, es decir, si el documento cumple las reglas del DTD.

Un documento HTML válido declara qué versión de HTML se utiliza en él. La declaración del tipo de documento especifica la definición del DTD que se usa en él (ISO8879). HTML4.01 especifica tres DTDs, que varían en cuanto a los elementos que soportan:

- El DTD HTML 4.01 Estricto (Strict DTD) incluye todos los elementos y atributos que no han sido desaprobados o que no aparecen en documentos con marcos.
- El DTD HTML 4.01 Transicional(Transitional DTD) incluye todo lo que incluye el DTD estricto más los elementos y atributos desaprobados (la mayoría de los cuales están relacionados con la presentación visual).
- El DTD HTML 4.01 para Documentos con Marcos(Frameset DTD incluye todo lo que incluye el DTD Transicional más los marcos.

#### 5. Qué son los metadatos y cómo se especifican en HTML?

El HTML permite a los autores especificar metadatos -- *información sobre un documento más que contenido del propio documento* -- de diferentes maneras. Por ejemplo, para especificar el autor: `<META name="Author" content="Dave Raggett">`

El elemento META puede utilizarse para identificar propiedades de un documento (p.ej., el autor, la fecha de caducidad, una lista de palabras clave, etc.) y para asignar valores a esas propiedades. Puede utilizarse el atributo lang de META para especificar el idioma del valor del atributo content. Esto permite a los sintetizadores de voz aplicar reglas de pronunciación dependientes del idioma. Ejemplo: declara que el nombre del autor está en francés: `<META name="Author" lang="fr" content="A. Le Hors">`

El siguiente ejemplo especifica la codificación de caracteres `<META http-equiv="Content-Type" content="text/html; charset=ISO-8859-5">`

Un uso común de META es especificar palabras clave que pueden usar los motores de búsqueda para mejorar la calidad de los resultados de una búsqueda. Cuando se proporcionan varios elementos META con información para varios idiomas, los motores de búsqueda pueden utilizar el atributo lang como filtro para mostrar los resultados de la búsqueda usando las preferencias de idioma del usuario:

Para hablantes de inglés americano: `<META name="keywords" lang="en-us" content="vacation, Greece, sunshine">`

Para hablantes de español: `<META name="keywords" lang="es" content="vacaciones, Grecia, sol">`

## Ejercicio 2

2a) `<!-- Código controlado el día 12/08/2009 -->`

Dentro del Body.

Indica un comentario o nota, dentro del código. Corresponde a una buena práctica, el agregado de notas o comentarios dentro del código que se está programando.

2b) `<div id="bloque1">Contenido del bloque1</div>`

Dentro del Body.

Corresponde a una sección llamada bloque1. El contenido es un texto que dice "Contenido del bloque1".

2c) `<img src="" alt="lugar imagen" id="im1" name="im1" width="32" height="32" longdesc="detalles.htm" />`

Dentro del Body.

Corresponde a una imagen, que en caso de no cargar o visualizarse erróneamente, aparece el texto "lugar imagen". Tiene una id y un name para localizarlos en un form, también tiene establecido un ancho y alto. Longdesc es la descripción de la URL utilizada, complementa al alt.

2d) `<meta name="keywords" lang="es" content="casa, compra, venta, alquiler" /> <meta http-equiv="expires" content="16-Sep-2019 7:49 PM" />`

Etiqueta meta, que se encuentra dentro del head. Indica que los elementos se encuentran en idioma español. Dentro del content, indica los datos que se quieren asociar al name. El name es el nombre al cual se le asocia este meta.

La segunda etiqueta meta, indica que el documento se considera expirado después de la fecha dentro del content.

2e) `<a href="http://www.e-style.com.ar/resumen.html" type="text/html" hreflang="es" charset="utf-8" rel="help">Resumen HTML </a>`

Dentro del Body.

Corresponde a una etiqueta de enlace a la página del href. El type indica el tipo de contenido de la URL enlazada. Hreflang indica el idioma, español. Charset es la codificación de caracteres de la URL. Rel=help, indica que la URL enlazada sirve de ayuda a la página actual.

2f) `<table width="200" summary="Datos correspondientes al ejercicio vencido"> <caption align="top"> Título </caption> <tr> <th scope="col">&nbsp;</th> <th scope="col">A</th> <th scope="col">B</th> <th scope="col">C</th> </tr> <tr> <th scope="row">1º</th> <td>&nbsp;</td> <td>&nbsp;</td> <td>&nbsp;</td> </tr> <tr> <th scope="row">2º</th> <td>&nbsp;</td> <td>&nbsp;</td> <td>&nbsp;</td> </tr> </table>`

Dentro del Body.

Tabla de ancho 200. En su resumen, indica que son datos correspondientes al ejercicio vencido. Título alineado al centro y arriba. A, B y C, son columnas cabeceras. Por defecto está en negrita y centrado. Se diferencian con el td, en que el td puede contener datos. 1º y

2º son filas cabeceras, al igual que A,B y C. Los datos de las celdas (&nbsp;) restantes indican un espacio en blanco.

### Ejercicio 3

```
3a) <a href="http://www.google.com.ar">Click aquí para ir a Google</a>
<a href="http://www.google.com.ar" target="_blank">Click aquí para ir a Google</a>
<a href="http://www. google.com.ar" type="text/html" hreflang="es" charset="utf-8"
rel="help">
<a href="#">Click aquí para ir a Google</a>
<a href="#arriba">Click aquí para volver arriba</a> <a name="arriba" id="arriba"></a>
```

Corresponde a una serie de etiquetas de enlaces al sitio <http://www.google.com.ar> . En la segunda, en target se aclara que se abra en otra pestaña con el valor `_blank`. En la primera se abre en la misma pestaña, porque de manera default se asigna el valor `_self` a target. En el tercer enlace, carece de texto donde hacer click para dirigirnos a la página. Si tuviese, indica que es de tipo texto, en español, con la codificación utf 8 y corresponde a una página de ayuda a la primera. En el cuarto, `href="#"`, al no seguirle nada después del numeral, lleva a la parte de arriba de la página. En el ultimo (Click para volver arriba), nos lleva al id "arriba".

```
3b)
<p><a href="http://www.google.com.ar">Click
aquí</a></p>
<p><a href="http://www.google.com.ar"></a> Click
aquí</p>
<p><a href="http://www.google.com.ar">Click
aquí</a></p>
<p><a href="http://www.google.com.ar"></a> <a
href="http://www.google.com.ar">Click aquí</a></p>
```

Corresponde a 4 etiquetas de párrafo p.

La primera posee dos componentes, una imagen y un enlace con texto.

La segunda posee un enlace, que se accede haciendo click en la imagen. Luego se cierra el enlace y sigue el elemento de texto *Click aquí*, pero no funciona porque ya se cerró la parte de enlace.

El tercero posee un enlace, que se puede acceder o bien haciendo click en la imagen o bien haciendo click en el texto, ya que, a diferencia del segundo, la parte del texto si se encuentra dentro de la etiqueta de enlace a.

La cuarta funciona igual que la tercera, pero son dos etiquetas de enlaces diferentes (aunque lleven al mismo sitio), una la imagen y otra el texto.

3c)

```
<ul> <li>xxx</li> <li>yyy</li> <li>zzz</li> </ul>
<ol> <li>xxx</li> <li>yyy</li> <li>zzz</li> </ol>
<ol> <li>xxx</li> </ol> <ol> <li value="2">yyy</li> </ol> <ol> <li value="3">zzz</li> </ol>
<blockquote> <p>1. xxx<br /> 2. yyy<br /> 3. zzz</p> </blockquote>
```

La primera etiqueta *ul*, genera una lista no ordenada, con elementos (la etiqueta *li* declara elementos de una lista).

La segunda etiqueta *ol*, genera una lista ordenada (por eso se le agregan índices a los elementos). Los componentes se encuentran bajo la etiqueta *li*.

El tercer bloque corresponde a 3 etiquetas *ol*, generando 3 "listas ordenadas". En el primero, como tiene un solo ítem, se le asigna el índice 1 a ese ítem. En el segundo y en el tercer *ol*, se le asigna valor a los índices y por eso es que se visualizan 2 y 3, siendo que ambos son primeros de su lista ordenada. Se fuerza a que tengan esos valores.

Por último *blockquote*, indica un bloque que se utiliza para citar todo lo que se encuentra dentro (en este caso son 3 párrafos con bajada de línea de por medio). *Blockquote* tiene un hermano menor: *q*. *blockquote* sirve para citas en bloque y *q* para citas en línea.

3d)

```
<table border="1" width="300"> <tr> <th>Columna 1</th> <th>Columna 2</th> </tr> <tr>
<td>Celda 1</td> <td>Celda 2</td> </tr> <tr> <td>Celda 3</td> <td>Celda 4</td> </tr>
</table>
```

```
<table border="1" width="300"> <tr> <td><div align="center"><strong>Columna 1</strong></div></td>
<td><div align="center"><strong>Columna 2</strong></div></td> </tr> <tr> <td>Celda 1</td>
<td>Celda 2</td> </tr> <tr> <td>Celda 3</td> <td>Celda 4</td> </tr> </table>
```

Corresponde a 2 tablas, ambas con el mismo borde y el mismo ancho.

En la primera, tenemos dos columnas, con datos de cabecera Columna 1 y Columna 2. Luego le siguen 2 filas de dos columnas cada una, con datos dentro de las celdas (acá se diferencian las etiquetas *th* con las *td*, las *th* son datos de cabecera o headers, y las *td* son de datos dentro de las celdas). Por defecto, pone en negrita los datos de cabecera.

En la segunda, tenemos 2 columnas, con datos *td* Columna 1 y Columna 2, centrados y en negrita (al ser una etiqueta *td* y no *th*, no la pone en negrita de manera default). Luego, igual que en la primera tabla, 2 filas más con datos dentro de cada celda, siguiendo la estructura de la tabla con 2 columnas.

3e)

```
<table width="200">
  <caption>
    Título
  </caption>
  <tr> <td bgcolor="#dddddd">&nbsp;</td> <td bgcolor="#dddddd">&nbsp;</td> <td
bgcolor="#dddddd">&nbsp;</td> </tr>
  <tr> <td bgcolor="#dddddd">&nbsp;</td> <td bgcolor="#dddddd"> &nbsp;</td> <td
bgcolor="#dddddd">&nbsp;</td> </tr>
</table>
<table width="200">
  <tr> <td colspan="3"><div align="center">Título</div></td> </tr>
  <tr> <td bgcolor="#dddddd">&nbsp;</td> <td bgcolor="#dddddd">&nbsp;</td> <td
bgcolor="#dddddd">&nbsp;</td> </tr>
  <tr> <td bgcolor="#dddddd">&nbsp;</td> <td bgcolor="#dddddd">&nbsp;</td> <td
bgcolor="#dddddd">&nbsp;</td> </tr>
</table>
```

La primera corresponde a 1 tabla con un caption, el cual es el encargado de darle un título descriptivo a las tablas. Además posee 2 filas con 3 columnas cada una, rellenas con color. La segunda corresponde a una tabla, con un ancho específico. Tiene una primer fila de ancho 3 columnas (colspan="3"), que contiene un título alineado al centro. Luego, posee 2 filas más de 3 columnas cada una, rellenas con color.

3f)

```
<table width="200">
  <tr>
    <td colspan="3"><div align="center">Título</div></td>
  </tr>
  <tr>
    <td rowspan="2" bgcolor="#dddddd">&nbsp;</td>
    <td bgcolor="#dddddd">&nbsp;</td>
    <td bgcolor="#dddddd">&nbsp;</td>
  </tr>

  <tr>
    <td bgcolor="#dddddd">&nbsp;</td>
    <td bgcolor="#dddddd">&nbsp;</td>
  </tr>

</table>

<table width="200">
  <tr>
    <td colspan="3"><div align="center">Título</div></td>
  </tr>
```



```

<tr>
  <td colspan="2" bgcolor="#dddddd">&nbsp;</td>
  <td bgcolor="#dddddd">&nbsp;</td>
</tr>
<tr> <td bgcolor="#dddddd">&nbsp;</td>
  <td bgcolor="#dddddd">&nbsp;</td>
  <td bgcolor="#dddddd">&nbsp;</td>
</tr>
</table>

```

La primer tabla, tiene un ancho establecido y 3 filas, en la primera el título, que abarca un ancho de 3 columnas (no está con las etiquetas de header th, sino con datos td). Luego, le sigue la segunda fila, que posee 3 elementos. El primero, tiene una altura de 2 filas (rowspan="2"), tiene como dato un color y de texto un espacio en blanco (&nbsp; corresponde a un espacio en blanco). Luego le siguen 2 columnas mas, todas con un color y de texto un espacio vacío.

En la tercer fila, ya la primera columna se encuentra establecida, porque el elemento de la primer columna de la segunda fila ocupa dos filas. Luego, le siguen 2 columnas más con un color y un texto de espacio vacío

La segunda tabla tiene un ancho establecido y 3 filas, en la primera el título, que abarca un ancho de 3 columnas (no está con las etiquetas de header th, sino con datos td). Luego, le sigue la segunda fila, que posee 2 filas, una con el ancho de 2 columnas y la otra con el ancho de 1. Ambas poseen color y texto de espacio vacío. La tercer fila posee 3 columnas, todas con un mismo color y espacio de texto vacío.

3g)

```

<table width="200" border="1">
  <tr>
    <td colspan="3"><div align="center">Título</div></td>
  </tr>
  <tr>
    <td colspan="2" rowspan="2">&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td width="50%">&nbsp;</td>
  </tr>
</table>
<table width="200" border="1" cellpadding="0" cellspacing="0">
  <tr>
    <td colspan="2"><div align="center">Título</div></td>
  </tr>
  <tr>
    <td rowspan="2">&nbsp;</td> <td>&nbsp;</td>
  </tr>

```

```
<tr>
  <td width="50%">&nbsp;</td>
</tr>
</table>
```

Corresponde a 2 tablas. Como en los casos anteriores, en ambas existe un título que abarca las 2 columnas que tienen las tablas (aunque en el primero abarque 3, si se agrega otra columna en la fila de abajo, la primera fila abarca completamente también a esta nueva columna agregada). Luego, poseen 2 filas más. La primera columna de la segunda fila ocupa el espacio de 2 filas y 2 columnas. Luego le sigue una fila más con contenido vacío. La tercera fila, empieza con la columna establecida desde antes, y la segunda columna tiene un ancho del 50% del total del ancho de la tabla, por eso es que aparece centrada.

La segunda tabla añade los valores de diseño `cellpadding="0"` `cellspacing="0"`. El atributo `cellpadding` permite especificar la distancia entre el margen interior de la celda y su contenido y `cellspacing` permite especificar la distancia entre las celdas y el margen exterior de la tabla.

3h)

```
<form id="form1" name="form1" action="procesar.php" method="post" target="_blank">
    <fieldset>
        <legend>LOGIN</legend>
        Usuario: <input type="text" id="usu1" name="usu1" value="xxx" /><br />
        Clave: <input type="password" id="clave1" name="clave1" value="xxx" />
    </fieldset>
    <input type="submit" id="boton1" name="boton1" value="Enviar" />
</form>

<form id="form2" name="form2" action="" method="get" target="_blank"> LOGIN<br />
    <label>Usuario: <input type="text" id="usu2" name="usu2" /></label><br />
    <label>Clave: <input type="text" id="clave2" name="clave2" /></label><br />
    <input type="submit" id="boton2" name="boton2" value="Enviar" />
</form>

<form id="form3" name="form3" action="mailto:xx@xx.com" enctype="text/plain"
method="post" target="_blank">
    <fieldset> <legend>LOGIN</legend>
        Usuario: <input type="text" id="usu3" name="usu3" /><br />
        Clave: <input type="password" id="clave3" name="clave3" /> </fieldset>
        <input type="reset" id="boton3" name="boton3" value="Enviar" />
</form>
```

Corresponde a 3 forms, que al momento de hacer click en el input, de tipo submit, envía los valores de los campos usuario y contraseña, según sea su id.

En el primer form, los envia al metodo post del codigo procesar.php. El campo de usuario es de tipo texto y ya tiene un valor precargado. El campo de clave es de tipo contraseña y tambien ya tiene un valor precargado.

En el segundo form, se envía al método get, pero no especifica el action. El título LOGIN, aparece distinto porque no se encuentra en la leyenda, como sí en los otros dos. Luego, tiene 2 labels, usuario y contraseña, ambos de tipo texto (dejan ver lo que se está ingresando).

En el tercer form, envía un email con los datos de los campos usu3 y clave 3 a la dirección [xx@xx.com](mailto:xx@xx.com).

3i)

```
<label>Botón 1 <button type="button" name="boton1" id="boton1"> <br /> <b>CLICK
AQUÍ</b></button></label>
```

```
<label>Botón 2 <input type="button" name="boton2" id="boton2" value="CLICK AQUÍ" />
</label>
```

Son dos botones, con una indicación de botón 1 y botón 2, antes de cada uno. El botón 1 tiene una imagen (especificando ancho y alto) y un texto. Tiene además una descripción por si la imagen no carga bien o no se encuentra. El botón 2 posee solo texto.

3j)

```
<p><label><input type="radio" name="opcion" id="X" value="X" />X</label><br />
<label><input type="radio" name="opcion" id="Y" value="Y" />Y</label>
</p>
<p><label><input type="radio" name="opcion1" id="X" value="X" />X</label><br />
<label><input type="radio" name="opcion2" id="Y" value="Y" />Y</label>
</p>
```

Corresponde a 4 selecciones, 2 de las cuales tienen el mismo nombre (las dos primeras), por lo cual cuando se selecciona una, y se quiere seleccionar la otra, la selección de la primera se borra. Esto es así ya que el tipo de input *radio*, permite solo la selección de un valor dentro del conjunto de posibilidades.

3k)

```
<select name="lista">
  <optgroup label="Caso 1"> <option>Mayo</option>
  <option>Junio</option>
</optgroup>
  <optgroup label="Caso 2">
  <option>Mayo</option> <option>Junio</option>
  </optgroup>
</select>
<select name="lista[]" multiple="multiple">
  <optgroup label=" Caso 1"> <option>Mayo</option>
  <option>Junio</option>
  </optgroup>
  <optgroup label=" Caso 2"> <option>Mayo</option>
```

```
<option>Junio</option>
</optgroup>
</select>
```

Corresponde a 2 listas de opciones *select*, agrupadas por grupos (cuyo titulos son Caso 1 y Caso 2). La diferencia entre el primero y el segundo es el valor múltiple dentro de la etiqueta *select*, lo que hace que aparezcan todas las opciones (segun el tamaño que tengan) visibles, con diferencia al primero, que hay que hacer click y ahi se despliegan todas las opciones posibles.

## Ventajas de HTML5

- Nueva estructura de etiquetas mejorada: esta nueva estructura permite definir por separado el encabezado, la barra de navegación, las secciones de la página web, los textos del sitio, los diálogos y el pié de página de los sitios web. Ofrece la posibilidad de obtener un código más limpio y fácil de depurar, que los códigos de los estándares anteriores.
- Inclusión de las etiquetas video y audio: dicha etiqueta soporta de manera eficiente y estable cualquier opción de ejecución de video y audio, sin generar errores o incluir código flash en nuestro sitio web.
- ofrece la posibilidad de guardar información útil en el dispositivo del cliente mientras se sincroniza nuevamente al servidor. Esta característica no sólo hace que la aplicación funcione aún si el cliente está desconectado, sino que mejora el rendimiento.
- Eliminación total de las etiquetas obsoletas, que tienen los estándares anteriores de HTML.
- Gracias a que HTML5 ofrece una gran variedad de herramientas, se pueden diseñar y presentar sitios y aplicaciones Web que superen las expectativas de los usuarios. Consecuentemente, aumentando las visitas y proporcionando una buena experiencia.
- Los navegadores modernos y populares como (Chrome, Firefox, Safari y Opera) soportan HTML5, es decir, sin importar qué navegador empleen los usuarios el contenido se puede visualizar correctamente.
- Es compatible con los navegadores móviles, de modo que cada página realizada en HTML5 que se ve en ordenadores, también se puede adaptar a los dispositivos móviles. Esta característica del lenguaje HTML5 es probablemente una de las más útiles, se puede acceder a cualquier página o aplicación web desde un dispositivo móvil y permite que la experiencia sea igual de buena que al visitar una página web en ordenadores.

## **Formatos de audio soportados por HTML5**

- WAV, o WAVE. Es un formato de audio digital sin compresión de datos. Desarrollado por Microsoft e IBM.
- MP3, formato de compresión de audio digital. Desarrollado por el Moving Picture Experts Group (MPEG)
- AAC, o Advanced Audio Coding. Es un codec digital con pérdida que se corresponde al estándar internacional "ISO/IEC 13818-7" como una extensión de MPEG-2. Es muy utilizado en dispositivos Apple.
- Vorbis, codec digital con pérdida, conseguida mediante la compresión de datos, y de formato abierto. El contenedor utilizado es Ogg. Desarrollador por Xiph.org
- Opus, codec digital con pérdida, de formato abierto. El contenedor utilizado es Ogg. Está estandarizado por el Internet Engineering Task Force (IETF).

## **Formatos de video soportados por HTML5**

- MP4. Presenta un buen soporte.
- WebM. Es una alternativa libre a MP4 de Google. Presenta un soporte medio.
- Ogv. Presenta un soporte medio.
- MKV. Tiene una buena compresión pero requiere un alto consumo del CPU.
- AVI. Presenta menor compresión que MP4. No es recomendado su uso.
- HEVC. Codificación de vídeo de alta eficiencia. Es el último y más avanzado estándar de compresión de vídeo creado para ofrecer una mayor eficiencia de codificación y una mejor calidad de vídeo. Se utiliza mayoritariamente en Streamings.
- Otros formatos como MOV, FLV, 3GP, MPG, RMVB o ASF (WMV) no se recomiendan para su utilización en web ya que son anticuados, propietarios o poco eficientes.