

Ejercitación 1 - Responder el siguiente cuestionario

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

HTML es el lenguaje de marcado principal de la World Wide Web. Originalmente, HTML fue diseñado principalmente como un lenguaje para describir semánticamente documentos científicos. Su diseño general, sin embargo, ha permitido que se adapte, en los años posteriores, para describir varios otros tipos de documentos e incluso aplicaciones.

Versiones de HTML:

HTML ha evolucionado a lo largo de los años, adaptándose a las nuevas tecnologías y necesidades de la web. Algunas de las versiones más importantes son:

- **HTML 1.0:** La primera versión, publicada en 1991, introdujo las etiquetas básicas para dar formato al texto y crear enlaces.
- **HTML 2.0:** En 1995, se estableció como el primer estándar oficial de HTML, agregando elementos como tablas y formularios.
- **HTML 3.2:** Introdujo características más avanzadas como marcos y hojas de estilo en línea.
- **HTML 4.0:** Esta versión fue un gran paso adelante, proporcionando una estructura más sólida y soporte para características como scripts y hojas de estilo externas.
- **HTML 5:** La versión más reciente y ampliamente utilizada, publicada en 2014. HTML 5 introdujo muchas características nuevas, como el elemento de video, el almacenamiento local, la geolocalización y un mejor soporte para gráficos y multimedia.

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

Principios básicos recomendados por el W3C para la creación de documentos HTML:

Compatibilidad:

- **Semántica clara:** Utiliza las etiquetas HTML con su significado correcto para que los navegadores, motores de búsqueda y tecnologías asistivas puedan interpretar correctamente el contenido.

- Validación: Asegúrate de que tu código HTML sea válido según las especificaciones del W3C para evitar errores y garantizar una visualización consistente en diferentes navegadores.
- Evitar el uso de atributos y elementos obsoletos: Mantente actualizado con las últimas versiones de HTML y evita utilizar elementos que hayan sido deprecados.

Utilidad:

- Resolver problemas reales: Utiliza HTML para solucionar problemas del mundo real y crear experiencias de usuario significativas.
- Priorizar la accesibilidad: Diseña tus páginas para que sean accesibles a personas con discapacidades, siguiendo las pautas de WCAG (Web Content Accessibility Guidelines).
- Facilitar la internacionalización: Utiliza atributos como `lang` para indicar el idioma del contenido y asegúrate de que tu diseño sea adaptable a diferentes culturas y regiones.

Simplicidad:

- Minimizar la complejidad: Evita utilizar soluciones excesivamente complejas cuando una solución más simple sea suficiente.
- Promover la legibilidad: Escribe código HTML limpio y bien estructurado para facilitar su mantenimiento y comprensión.
- Fomentar la reutilización: Crea componentes HTML reutilizables para evitar la duplicación de código y mejorar la eficiencia.

Extensibilidad:

- Permitir la creación de nuevos elementos: HTML es un lenguaje extensible que permite crear nuevos elementos personalizados para satisfacer necesidades específicas.
- Facilitar la integración con otras tecnologías: Diseña tu HTML para que se integre fácilmente con otras tecnologías como CSS y JavaScript.

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

En las especificaciones de HTML, los términos "desaprobado" (deprecated) y "obsoleto" (obsolete) se utilizan para indicar el estado de ciertas características del lenguaje que ya no se recomiendan para su uso. Esto se hace con el objetivo de guiar a los desarrolladores hacia mejores prácticas y tecnologías más modernas.

Un elemento o atributo se considera **desaprobado** cuando:

- Existe una alternativa mejor: Hay una nueva característica o técnica que realiza la misma función de manera más eficiente, accesible o compatible.
- Presenta problemas de seguridad: El uso de la característica puede comprometer la seguridad de una aplicación web.
- Es redundante: La funcionalidad de la característica ya está cubierta por otras características estándar.

Un elemento o atributo se considera **obsoleto** cuando:

- Ha sido reemplazado por una característica estándar: La característica ha sido completamente sustituida por una nueva que ofrece una funcionalidad similar o mejorada.
- Ha sido eliminada de las especificaciones: La característica ya no forma parte del estándar y su uso no está garantizado en futuros navegadores.

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

Un **DTD** (Document Type Definition, o Definición del Tipo de Documento en español) es un conjunto de reglas que definen la estructura legal de un documento XML o SGML (Standard Generalized Markup Language). En el contexto de HTML, el DTD especifica los elementos, atributos y su relación entre sí que son válidos en un documento HTML.

DTDs en HTML 4.01

HTML 4.01, una versión ya obsoleta de HTML, contemplaba **tres DTDs principales**:

1. **Strict**: Este DTD era el más estricto y exigía un código HTML muy limpio y estructurado. Prohibía el uso de elementos y atributos presentacionales que no estuvieran definidos en el estándar.
2. **Transitional**: Este DTD era un poco más flexible que el Strict, permitiendo el uso de algunos elementos presentacionales obsoletos pero ampliamente utilizados en ese momento, como ****, **<center>**, etc.
3. **Frameset**: Este DTD estaba diseñado específicamente para crear páginas web que utilizaban marcos (frames). Los marcos dividían la ventana del navegador en múltiples secciones, cada una con su propio contenido. Sin embargo, el uso de marcos ha caído en desuso debido a sus limitaciones y problemas de accesibilidad.

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

Los metadatos, en el contexto de HTML, son datos que proporcionan información adicional sobre una página web. Esta información no se muestra directamente en la página, sino que se utiliza para describirla y clasificarla. Los metadatos son como un resumen o una etiqueta que le da a los motores de búsqueda y a otros sistemas una idea del contenido de la página.

¿Para qué sirven los metadatos?

- **Motores de búsqueda:** Los metadatos ayudan a los motores de búsqueda a entender de qué trata una página y a clasificarla en los resultados de búsqueda.
- **Redes sociales:** Cuando compartes una página en redes sociales, los metadatos se utilizan para generar una vista previa atractiva.
- **Navegadores:** Los metadatos pueden proporcionar información al navegador sobre la codificación de caracteres, la descripción de la página o la compatibilidad con dispositivos móviles.

¿Cómo se especifican los metadatos en HTML?

Los metadatos se especifican dentro de la sección **<head>** de un documento HTML utilizando la etiqueta **<meta>**. Esta etiqueta tiene varios atributos que permiten definir diferentes tipos de información.

Los atributos más comunes de la etiqueta **<meta>** son:

- **name:** Indica el tipo de metadato. Algunos valores comunes son "description", "keywords", "author", "viewport", etc.
- **content:** Especifica el valor del metadato.
- **http-equiv:** Se utiliza para definir encabezados HTTP. Por ejemplo, para especificar el tipo de contenido de la página.

Ejercitación 2

Analizar los siguientes segmentos de código indicando en qué sección del documento HTML se colocan, cuál es el efecto que producen y señalar cada uno de los elementos, etiquetas, y atributos (nombre y valor), aclarando si es obligatorio.

2.a)

<!-- Código controlado el día 12/08/2009 →

Este es un comentario HTML. Los comentarios son texto o código que se incluye en el documento HTML pero que no se muestra en el navegador ni afecta la página. Se utilizan para dejar notas, explicaciones, o para desactivar código temporalmente durante el desarrollo.

Se puede colocar en cualquier parte del documento HTML.

Los comentarios en HTML comienzan con `<!--` y terminan con `-->`. Todo lo que esté entre estas marcas se considera parte del comentario y no se procesa como HTML.

No son obligatorios, pero son muy recomendados para mejorar la legibilidad y el mantenimiento del código.

2.b)

```
<div id="bloque1">Contenido del bloque1</div>
```

Este segmento se coloca dentro de la sección **<body>** del documento HTML, ya que está destinado a ser parte del contenido visual que se muestra en la página.

<div> es un elemento de contenedor genérico en HTML que se utiliza para agrupar otros elementos de contenido.

El contenido dentro de este **<div>** es **"Contenido del bloque1"**, que se mostrará tal cual en la página web.

El atributo **id** asigna un identificador único al elemento **<div>**.

En este caso, el valor del **id** es **"bloque1"**.

Obligatoriedad:

- **<div>**: El uso del **<div>** no es obligatorio en sí mismo, pero es fundamental en la estructuración de páginas HTML. Se usa ampliamente para organizar y controlar la disposición del contenido en la página.
- **id**: El **id** no es obligatorio para un **<div>**, pero si se utiliza, debe ser único dentro del documento.

2.c)

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

Este segmento se coloca dentro de la sección **<body>** del documento HTML, ya que está destinado a mostrar una imagen como parte del contenido visual de la página.

**** es un elemento que se utiliza para insertar imágenes en una página web.

Atributos:

- **src (obligatorio):** Especifica la URL de la imagen. Dejarlo vacío no es común y puede generar problemas visuales.
- **alt (recomendado):** Proporciona un texto alternativo, es altamente recomendado para accesibilidad.
- **id:** Proporciona un identificador único para el elemento.
- **name:** Asigna un nombre al elemento, aunque es menos común en imágenes.
- **width:** Define el ancho de la imagen en píxeles.
- **height:** Define la altura de la imagen en píxeles.
- **longdesc:** Ofrece una descripción más detallada de la imagen a través de un enlace, útil para mejorar la accesibilidad.

2.d)

```
<meta name="keywords" lang="es" content="casa, compra, venta, alquiler " />
```

```
<meta http-equiv="expires" content="16-Sep-2019 7:49 PM" />
```

Estos segmentos se colocan dentro de la sección **<head>** del documento HTML. Los elementos **<meta>** proporcionan metadatos sobre el documento, que no se muestran directamente en la página, pero son útiles para navegadores, motores de búsqueda, y otros servicios.

<meta name="keywords" lang="es" content="casa, compra, venta, alquiler" />:

- **name="keywords":** Este atributo define el tipo de metadato como una lista de palabras clave relevantes para el contenido de la página.
- **lang="es":** Indica que el idioma de las palabras clave es español.
- **content="casa, compra, venta, alquiler":** Contiene las palabras clave separadas por comas. Estas palabras clave pueden ser usadas por motores

de búsqueda para indexar la página, aunque en la actualidad, muchos motores de búsqueda, como Google, ya no utilizan este atributo.

<meta http-equiv="expires" content="16-Sep-2019 7:49 PM" />:

- **http-equiv="expires":** Este atributo indica que el contenido de la página debe considerarse expirado después de la fecha y hora especificadas. Esto puede afectar cómo los navegadores y cachés manejan la página.
- **content="16-Sep-2019 7:49 PM":** Especifica la fecha y hora en que el contenido expira. Después de esta fecha, el navegador puede solicitar una versión actualizada de la página en lugar de utilizar una copia en caché.

Obligatoriedad:

- **Etiqueta <meta>:** No es obligatorio incluir estas etiquetas, pero pueden ser útiles para la optimización en motores de búsqueda o para controlar el almacenamiento en caché.
- **Atributos name, lang, y content del primer <meta>:** No son obligatorios, pero son necesarios para que la etiqueta <meta> cumpla con su propósito específico.
- **Atributos http-equiv y content del segundo <meta>:** No son obligatorios, pero se usan para controlar la caducidad del contenido de la página.

2.e)

`Resumen HTML `

Este segmento se coloca dentro de la sección **<body>** del documento HTML. La etiqueta **<a>** es un enlace que se muestra como parte del contenido visual en la página.

Atributos:

- **<href> (obligatorio):** Especifica la URL del recurso al que se enlaza.
- **<type>:** Indica el tipo MIME del recurso. Es opcional y no siempre se incluye.
- **<hreflang>:** Indica el idioma del recurso vinculado. Es opcional, pero útil para accesibilidad y SEO.

- **<charset>**: Especifica la codificación de caracteres del recurso. Es opcional y menos común en enlaces.
- **<rel>**: Describe la relación entre el documento actual y el documento vinculado. Es opcional y útil para mejorar la semántica del enlace.

2.f)

```
<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>

Este segmento de código se coloca dentro de la sección **<body>** del documento HTML. La etiqueta **<table>** se usa para crear una tabla que organiza los datos en filas y columnas.

Etiqueta <table>:

- **<width="200">**: Especifica el ancho de la tabla en píxeles.
- **<summary="Datos correspondientes al ejercicio vencido">**: Proporciona un resumen de la tabla para ayudar a los usuarios de tecnologías de asistencia a entender el propósito de la tabla.

Etiqueta <caption>:

- **<align="top">**: Especifica la alineación de la leyenda en la parte superior de la tabla.

Etiqueta <tr>:

- Define una fila dentro de la tabla.

Etiqueta <th>:

- **<scope="col">**: Indica que la celda de encabezado está relacionada con una columna.
- **<scope="row">**: Indica que la celda de encabezado está relacionada con una fila.

Etiqueta <td>:

- Define una celda de datos dentro de la fila.

Entidad :

- Representa un espacio en blanco no rompible en HTML.

Obligatoriedad:

- **Etiqueta <table>**: Es obligatoria para crear una tabla.
- **Atributo <width>**: Es opcional y se usa para definir el ancho de la tabla.
- **Atributo <summary>**: Es opcional, pero recomendado para accesibilidad.
- **Etiqueta <caption>**: Es opcional, pero útil para dar contexto a la tabla.
- **Etiqueta <tr>**: Es obligatoria dentro de una tabla para crear filas.
- **Etiqueta <th> y <td>**: Son obligatorias dentro de <tr> para crear celdas de encabezado y datos, respectivamente.

- **Atributo <scope>:** Es opcional, pero recomendado para mejorar la accesibilidad y la semántica de las tablas.

Ejercitación 3

En cada caso, explicar las diferencias entre los segmentos de código y sus visualizaciones:

3.a)

OP1) `Click aquí para ir a Google`

OP2) `Click aquí para ir a Google`

OP3) ``

OP4) `Click aquí para ir a Google`

OP5) `Click aquí para volver arriba`

``

Opción 1: este enlace estándar apunta a `http://www.google.com.ar`. Al hacer clic en el enlace, el usuario será llevado a la página de Google en Argentina.

Opción 2: Similar al primer enlace, pero con el atributo `target="_blank"`, que hace que el enlace se abra en una nueva pestaña o ventana del navegador.

Opción 3: Funciona igual que el primer enlace, pero los atributos adicionales pueden ser utilizados por navegadores, motores de búsqueda, y tecnologías de asistencia para proporcionar una experiencia más contextual y accesible.

Opción 4: es un enlace que recarga la página actual o la desplaza hacia arriba, pero no lleva al usuario a un nuevo recurso.

Opción 5: crea un enlace que permite navegar dentro de la misma página, moviendo la vista a una sección específica.

3.b)

OP1) `<p>Click aquí</p>`

OP2) <p> Click aquí</p>

OP3)<p>Click aquí</p>

OP4)<p> Click aquí</p>

Opción 1: La imagen se muestra al lado del enlace de texto "Click aquí". El enlace es independiente de la imagen, por lo que solo se puede hacer clic en el texto para ir a <http://www.google.com.ar>.

Opción 2: Al hacer clic en la imagen, el usuario es redirigido a la página de Google. El texto "Click aquí" no tiene efecto de enlace y es solo texto normal.

Opción 3: En este caso, tanto la imagen como el texto "Click aquí" están envueltos dentro del mismo enlace.

Opción 4: Al hacer clic en la imagen o en el texto "Click aquí", el usuario será redirigido a la página de Google. Sin embargo, son dos enlaces distintos que conducen al mismo destino.

3.c)

Código 1:

```
<ul>
<li>xxx</li>
<li>yyy</li>
<li>zzz</li>
</ul>
```

Este código utiliza una lista no ordenada (), donde cada elemento de la lista se define mediante .

Los elementos de la lista (**xxx**, **yyy**, **zzz**) se muestran con viñeta.

Código 2:

```
<ol>
<li>xxx</li>
```

```
<li>yyy</li>
```

```
<li>zzz</li>
```

```
</ol>
```

Este segmento utiliza una lista ordenada (****), con tres elementos de lista (**xxx**, **yyy**, **zzz**).

Los elementos se muestran con números secuenciales (1., 2., 3.).

Código 3:

```
<ol>
```

```
<li>xxx</li>
```

```
</ol>
```

```
<ol>
```

```
<li value="2">yyy</li>
```

```
</ol>
```

```
<ol>
```

```
<li
```

```
value="3">zzz</li>
```

```
</ol>
```

Este código tiene tres listas ordenadas (****) separadas, donde cada una contiene un solo elemento. Los valores de numeración se ajustan manualmente utilizando el atributo **value** en ****.

Código 4:

```
<blockquote>
```

```
<p>1. xxx<br />
```

```
2. yyy<br />
```

```
3. zzz</p>
```

</blockquote>

Aquí se utiliza un bloque de citas (**<blockquote>**), dentro del cual se enumera manualmente cada elemento con números dentro de un párrafo (**<p>**).

3.d)

Código 1:

```
<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>
```

- Se utiliza una tabla (**<table>**) con un borde de grosor 1 (**border="1"**) y un ancho de 300 píxeles (**width="300"**).
- La primera fila (**<tr>**) contiene encabezados de columna definidos con **<th>** para "Columna 1" y "Columna 2".
- Las siguientes filas contienen celdas de datos (**<td>**) con texto "Celda 1", "Celda 2", "Celda 3" y "Celda 4".
- La tabla tiene encabezados centrados y en negrita de forma predeterminada, debido a la naturaleza de **<th>**. Las celdas de datos están alineadas de manera normal, sin estilo adicional.

Código 2:

```
<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>
```

Este código también utiliza una tabla con borde y ancho similar, pero en lugar de usar **<th>** para los encabezados, emplea **<td>** con contenido envuelto en un **<div>**.

Dentro de cada **<div>**, el texto de los encabezados ("Columna 1" y "Columna 2") se centra con **align="center"** y se hace negrita con ****.

3.e)

Código 1:

```
<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>

```

En este código se utiliza una tabla (**<table>**) con un ancho de 200 píxeles (**width="200"**).

El título de la tabla se coloca dentro de la etiqueta **<caption>**, que se utiliza para describir el contenido de la tabla.

Las celdas de la tabla están coloreadas con un fondo gris claro mediante el atributo **bgcolor="#dddddd"**.

El símbolo ** ** dentro de las celdas representa un espacio en blanco.

Código 2:

```

<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>

```

En lugar de usar **<caption>**, este código coloca el título dentro de una celda (**<td>**) que abarca las tres columnas (**colspan="3"**).

El título se centra dentro de la celda utilizando un **<div align="center">**.

Las celdas tienen el mismo color de fondo gris claro (**bgcolor="#dddddd"**) y están vacías con un espacio en blanco representado por ** **.

3.f)

Código 1:

```

<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>
```

En este código, la tabla tiene un título centrado que ocupa las tres columnas de la primera fila (**colspan="3"**).

La segunda fila tiene una celda que se extiende verticalmente a lo largo de dos filas (**rowspan="2"**) y abarca la primera columna. Las otras dos celdas de la fila tienen un fondo gris claro.

La tercera fila completa la tabla con dos celdas adicionales, también con fondo gris claro.

Código 2:

```
<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>

Al igual que en el primer código, la tabla tiene un título centrado que ocupa las tres columnas de la primera fila.

La segunda fila contiene una celda que se extiende horizontalmente a lo largo de dos columnas (**colspan="2"**), mientras que la tercera columna tiene una celda independiente. Todas las celdas tienen un fondo gris claro.

La tercera fila está compuesta de tres celdas con fondo gris claro.

3.g)

Código 1:

```
<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>
```

La tabla tiene un borde de grosor 1 (**border="1"**) y un ancho de 200 píxeles (**width="200"**).

El título está centrado y ocupa las tres columnas en la primera fila (**colspan="3"**).

En la segunda fila, la primera celda se extiende tanto horizontal (**colspan="2"**) como verticalmente (**rowspan="2"**), ocupando las dos primeras columnas y las dos primeras filas. La última celda de la segunda fila es normal.

La tercera fila contiene solo una celda en la segunda columna (**width="50%"**), y la primera columna es la celda que se extiende desde la fila anterior.

Código 2:

```
<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>
```

Similar al primer código, esta tabla también tiene un borde de grosor 1, pero añade **cellpadding="0"** y **cellspacing="0"** para eliminar el espaciado interno y entre celdas.

El título está centrado y ocupa dos columnas en la primera fila (**colspan="2"**).

La segunda fila tiene una celda que se extiende verticalmente a lo largo de dos filas (**rowspan="2"**) en la primera columna. La segunda celda en esta fila es normal.

La tercera fila contiene solo una celda en la segunda columna (**width="50%"**), similar al Código 1.

3.h)

Código 1:

```
<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>
```

Este formulario envía los datos a un archivo PHP llamado **procesar.php** mediante el método **post**.

Los datos se envían en una nueva ventana (**target="_blank"**).

Usa un **<fieldset>** con **<legend>** para organizar el contenido dentro de un recuadro titulado "LOGIN".

Los campos de entrada (**<input>**) tienen valores predefinidos (**value="xxx"**).

El botón de envío es un **<input type="submit">** que envía los datos al archivo PHP.

Código 2:

```
<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>
```

Este formulario utiliza el método **get** para enviar los datos, lo que significa que los datos se añadirán a la URL en formato de consulta.

action="" indica que los datos se enviarán a la misma URL donde se encuentra el formulario.

Los campos de entrada están dentro de etiquetas **<label>** para mejorar la accesibilidad.

El formulario carece de valores predeterminados.

El botón de envío es un **<input type="submit">** y también abrirá una nueva ventana/tab para procesar los datos.

Código 3:

```
<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>
```

Este formulario envía los datos a una dirección de correo electrónico especificada (**action="mailto:xx@xx.com"**), usando el método **post** y el tipo de codificación **text/plain**.

El formulario también usa un **<fieldset>** con **<legend>** para agrupar y titular los campos como "LOGIN".

El botón es un **<input type="reset">** etiquetado como "Enviar"

3.i)

Código 1:

```
<label>Botón 1
```

```
<button type="button" name="boton1" id="boton1">
```

```
<br />
```

```
<b>CLICK AQUÍ</b></button></label>
```

Este código crea un botón utilizando la etiqueta **<button>**.

Dentro del botón, se inserta una imagen (****) y debajo de ella, el texto **CLICK AQUÍ** en negrita.

El botón se encuentra dentro de una etiqueta **<label>**, que puede asociarse con campos de formulario o, en este caso, simplemente etiquetar el botón.

Código 2:

```
<label>Botón 2
```

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

```
</label>
```

Este código utiliza un botón de entrada (**<input type="button">**) que se etiqueta con el valor "CLICK AQUÍ".

Similar al Código 1, está envuelto en una etiqueta **<label>**, que aquí funciona como un texto de presentación para el botón.

3.j)

Código 1:

```
<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>
```

En este código, se crean dos botones de opción (**<input type="radio">**) dentro de etiquetas **<label>**.

Ambos botones de opción tienen el mismo nombre (**name="opcion"**), lo que los agrupa en el mismo conjunto. Esto significa que el usuario puede seleccionar solo una opción de las dos disponibles (X o Y).

Las etiquetas **<label>** están asociadas a sus respectivos botones de opción y facilitan la interacción con ellos.

Código 2:

```
<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>
```

Este código también crea dos botones de opción dentro de etiquetas **<label>**.

Sin embargo, aquí los botones tienen nombres diferentes: **name="opcion1"** y **name="opcion2"**.

Al tener diferentes nombres, los botones no están en el mismo grupo, lo que permite al usuario seleccionar ambas opciones simultáneamente (X y Y).

3.k)

Código 1:

```
<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>
```

Crea una lista desplegable donde solo se puede seleccionar una opción de las disponibles en los dos grupos ("Caso 1" y "Caso 2")

Este código no tiene el atributo **multiple**. Esto significa que solo se puede seleccionar una opción a la vez.

<select>:

- **Función:** Define una lista desplegable desde la cual el usuario puede seleccionar una o más opciones.
- **Atributos clave:**

name: Asigna un nombre al elemento para identificarlo en el envío de formularios.

multiple: Permite al usuario seleccionar múltiples opciones a la vez.

<optgroup>:

- **Función:** Agrupa opciones relacionadas dentro de un elemento <select>.

Código 2:

```
<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>
```

El segundo código crea una lista de selección donde se pueden seleccionar múltiples opciones ya que incorpora **multiple** del elemento **<select>**

Escribir el código HTML completo necesario para visualizar la siguiente salida en el navegador: