

Entornos Gráficos

Práctica N° 1

HTML

Docentes:

Prof. Ing. Daniela Díaz

Prof. Ing. Julián Butti

Año de cursado:

2023

Alumno:

Legajo
47019

Nombre y Apellido
Fadua Dora Sicardi

E-mail
faduadora77@gmail.com

ÍNDICE

ÍNDICE	1
EJERCICIOS	1
Ejercicio 1	1
1. ¿Qué es HTML, cuando fue creado, cuáles fueron las distintas versiones y cuál es la última?	1
2. ¿Cuáles son los principios básicos que el W3C recomienda seguir para la creación de documentos con HTML?	1
3. En las Especificaciones de HTML, ¿cuándo un elemento o atributo se considera desaprobado? ¿y obsoleto?	2
4. ¿Qué es el DTD y cuáles son los posibles DTDs contemplados en la especificación de HTML 4.01?	2
5. ¿Qué son los metadatos y cómo se especifican en HTML?	3
Ejercitación 2	3
Ejercitación 3	6
Ejercitación 4	14
Ejercitación 5	15

EJERCICIOS

Ejercicio 1

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

HTML es un lenguaje de marcado (marcado ya que representa con etiquetas a cada elemento). Es un estándar a cargo del W3C que se impuso en la visualización de las páginas web y el cual todos los navegadores han adoptado. Este estándar brinda una estructura básica y código HTML para “escribir” los elementos y contenidos de una página web (como por ejemplo textos, enlaces, videos, imágenes, etc).

HTML lo desarrolló Tim Berners-Lee, pero fue popularizado por el navegador Mosaic. En sus principios (1990 - 1995), tuvo varias revisiones y le agregaron extensiones. Es en el 95 cuando W3C experimenta con HTML 3.0, el cual terminó siendo un intento fallido de su extensión. En 1997 aparece la versión 3.2 con un enfoque más pragmático, y esta versión es completada el mismo año por una llamada 4.01. En el 1998 la evolución de HTML se frena, y W3C apuesta a XHTML 1.0, que era similar a la última versión de HTML pero con una nueva serialización que se completó en el año 2000. Se continúa la evolución de HTML en el año 2003 con la aparición de los formularios web XForms. En el 2006 surge la idea de crear HTML 5.0, y es en 2007 cuando los navegadores Apple, Mozilla y Opera permiten a W3C publicar estas especificaciones, trabajando en conjunto con WHATWG. Actualmente, HTML5 es la última versión de HTML y es el lenguaje principal de la web.

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

- Estructura semántica: Utilizar los elementos HTML apropiados para describir la estructura de la página web (usar los elementos HTML para definir el encabezado, el contenido principal, las secciones, los párrafos, las listas, los enlaces, etc).
- Separación de presentación y contenido: Separar el diseño visual y la presentación de la página del contenido mediante el uso de hojas de estilo CSS (hace que el contenido sea accesible y fácil de entender para una variedad de dispositivos y usuarios).
- Accesibilidad: Diseñar páginas web que sean accesibles para todos los usuarios incluyendo aquellos con discapacidades. Esto implica utilizar etiquetas y atributos HTML que permitan a los usuarios navegar y entender la página.
- Validación: Validar el código HTML usando herramientas de validación (por ej: el validador de HTML del W3C) para asegurar que el código esté bien estructurado y cumpla con los estándares.
- Optimización del rendimiento: Optimizar la carga y el rendimiento de la página web mediante la reducción del tamaño del archivo y la optimización de las imágenes para que la página cargue más rápido.
- Compatibilidad con múltiples navegadores: Diseñar páginas web que sean compatibles con múltiples navegadores web, incluyendo los que se usan en los dispositivos móviles. Esto asegura que la página sea accesible y utilizable por diferentes usuarios.

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

Un elemento o atributo se considera "desaprobado" cuando oficialmente se declaró que ya no se debe usar, pero aún es posible utilizarlo. Es decir que aunque todavía funcionen, ya no se recomiendan para su uso, y ya fueron reemplazados por alternativas más nuevas y estandarizadas.

Un elemento o atributo se considera "obsoleto" cuando oficialmente se declaró que no se deben usar, ya que su uso podría traer errores y problemas en la página web. Su uso no es recomendable.

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

El DTD es la declaración del tipo de documento (document type declaration), esta va a estar en la primera línea del archivo html. En HTML 4.01 se puede declararla como:

Declaración transitoria:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

Declaración estricta:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd"
```

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

Los metadatos son información de un documento. Con HTML se los puede especificar (a los metadatos) de distintas formas:

Cuando especificamos al autor del documento HTML, se usa el elemento "meta". Este elemento provee metadatos que no pueden ser expresados con otros elementos del encabezado (head), como title, base, link, style y script. Este elemento puede ser utilizado de tres maneras diferentes.

- Con el atributo charset: En este caso, el atributo debe estar solo y tener un valor que represente al conjunto de caracteres utilizado por el documento. Sólo puede haber una declaración meta con el atributo charset por documento.
- Con el atributo name: El elemento provee metadatos a nivel de documento organizados en pares nombre-valor. El "nombre" en este par es provisto por el atributo name, mientras que el "valor" es provisto por el atributo content en cada caso.
- Con el atributo http-equiv: El elemento representa una directiva pragma y su tipo es declarado en el atributo http-equiv. Los valores para su interpretación son definidos en el atributo content. El nombre de cada directiva es asociado a una descripción de cómo debería ser el valor del atributo content.

El uso de metadatos contribuye a la optimización de los motores de búsquedas, lo cual puede posicionar a la página web entre los primeros resultados de una búsqueda de Google por ejemplo. Esto se conoce como SEO (Search Engine Optimization).

Aquí un ejemplo:

```
<meta charset="UTF-8">  
<meta http-equiv="X-UA-Compatible" content="IE=edge">  
<meta name="viewport" content="width=device-width,  
initial-scale=1.0">  
<title>Primera pagina</title>
```

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 -->
```

Esto es un comentario, los comentarios pueden colocarse en cualquier parte del programa, pero es de gran ayuda ponerlos cuando se codifica algo importante, o se hace un cambio, o en cualquier situación que parece necesario dejar un comentario, para no olvidarnos que se quiso hacer o si el código lo usa o ve otra persona pueda entenderlo. Es muy útil aunque no es un elemento obligatorio. La estructura de los comentarios es: `<!--Aquí va lo que se comenta-->`

2.b)

```
<div id="bloque1">Contenido del bloque 1</div>
```

El elemento div es muy importante y sirve como “encapsulador” o “contenedor” de bloques del código. El div ocupa todo el ancho disponible en la pantalla, pero de alto ocupará solo lo ocupado. No podemos decir que es obligatorio, aunque es una muy buena práctica y se utiliza en todas las codificaciones. En este caso el atributo es id, y el valor del mismo es bloque1. Este elemento se usa en el body. La estructura del elemento div es:

```
<div "atributo1=valor1, atributo2=valor2, ..." >
```

Contenido del div `<!--dentro de un div pueden haber mas div, párrafos, imágenes, texto, etc -->`

```
</div>
```

2.c)

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

El elemento img sirve para mostrar una imagen. Cuenta siempre con el atributo source (src) que es donde va la ruta o ubicación de la imagen. Es muy buena práctica utilizar el atributo alt, este sirve para describir de manera simple la imagen puesta (brinda accesibilidad). La estructura de este elemento es:

```

```

En este caso el ejemplo dado no tiene ningún valor el elemento src, por lo que no va a mostrar ninguna imagen en la pantalla (lo cual no sería correcto).

2.d)

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

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

Estos elementos se ubican en el head. Los elementos meta proveen metadatos y no pueden ponerse en otro elemento del head que no sea el elemento meta.

En el primer ejemplo se usa el atributo name, seguido del atributo lang que se usa para especificar el idioma (es=español) y el content que contiene una descripción de la información.

El segundo ejemplo

2.e)

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

El elemento a se usa en el body y se usa para poner enlaces/links. Tiene siempre el atributo href que su contenido será el URI del enlace, y entre las Etiquetas del elemento, se pone una descripción del enlace para cliquear. Puede tener atributos como target, que sirve para especificar dónde/cómo abrir ese enlace. En este caso, se utilizó el atributo type que indica el tipo de contenido de lo enlazado en este caso es un archivo html, tiene el atributo hreflang que identifica al lenguaje del contenido enlazado, el atributo rel sirve para indicar el significado que tiene el documento enlazado, en este caso es help, es decir que el documento sirve de ayuda. Por último el atributo charset quedó obsoleto en html5.

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">1</th>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <th scope="row">2</th>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
</table>

```

Las tablas van en el body. Este elemento permite crear una tabla. La tabla cuenta con los atributos width y summary. El width sirve para especificar el ancho del elemento, en este caso 200px. El summary quedó obsoleto en html 5, pero era como un resumen del propósito de la tabla.

Luego dentro de la tabla está el elemento caption, que lo que hace es dar título a la tabla, el cual tiene el atributo align que indica cómo alinear el elemento caption, luego cuenta con los elementos básicos de una tabla como lo son tr, th y td. El tr indica todos los elementos de una fila (row). Dentro de los tr hay hijos: th y td. Los th sirven para indicar que esa celda es una celda de encabezado o titular. Y los td representan a celdas comunes o regulares. En este ejemplo vemos que la primera fila (primer par de tr) es de celdas de encabezado, y la primera columna tiene un espacio en blanco. La segunda fila es de una celda de encabezado y las demás son celdas comunes. La tercer fila es igual a la segunda.

Ejercitación 3

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

3.a)

```

<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>

```

En el 1er caso, tenemos un link para dirigirnos a la página de google, el cual se visualiza con el texto "Click aquí para ir a Google". Cuando cliqueamos, Google se abre en la misma pestaña.

En el 2do caso, se visualiza de la misma forma que el primero, pero en vez de abrirse en la misma pestaña, Google se abre en una nueva pestaña (gracias a que el valor del atributo target es _blank).

En el 3er caso, la etiqueta del elemento a no está cerrada, por lo que directamente no se visualiza nada en pantalla.

En el 4to caso, tenemos una visualización similar al 1er y 2do caso, pero al clicar, nos redirige a la misma página en la que estamos, ya que el href tiene el valor #.

En el 5to caso podemos visualizar el link que dice "Click para volver arriba", y al clicar sucede lo mismo que en el 4to caso, el link nos redirige a la misma página. Para que esto tuviera sentido debería haber algún elemento en la parte superior de la pagina que tenga un id="arriba", y al bajar un poco por la página, encontrarnos con este link, y ahí sí tendría sentido utilizar el href="#arriba".

En el 6to caso, no podemos visualizar nada y además no hay ningún documento enlazado ya que no está el atributo href.

3.b)

```
<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>
```

En el 1er, tenemos dentro de un párrafo, en primer lugar la visualización de una imagen, y debajo de la imagen, un link que se visualiza con el texto "Click aquí", que al clicar la página se redirige a abrir google, en esa misma pestaña.

En el 2do caso, sucede que se visualiza igual que en el 1ero, pero si cliqueamos en "Click aquí" no sucede nada, en cambio al clicar la imagen, nos redirige en la misma pestaña a google. Esto sucede porque el elemento a encierra al elemento img (en lugar de al texto Click aquí).

En el 3er caso, además de suceder al igual que en el 2do que al tocar la imagen nos dirigimos a la página de Google, se suma que también nos dirige a Google si cliqueamos el texto "Click aquí". Esto sucede porque tanto el elemento img como el texto de Click aquí quedaron dentro de las etiquetas del elemento a.

En el 4to caso sucede lo mismo que en el caso 3, pero en lugar de estar tanto el elemento img como el "Click aquí" dentro del mismo elemento a, se los puso en diferentes elementos

a, pero ambos redirigen a la página de Google y o hacen de la misma forma. Por lo que desde el lado del usuario esto no se percibe como “diferente”.

3.c)

<pre> xxx yyy zzz </pre>	<pre> xxx yyy zzz </pre>	<pre> xxx <li value="2">yyy <li value="3">zzz </pre>	<pre><blockquote> <p>1. xxx
 2. yyy
 3. zzz</p> </blockquote></pre>
--	--	--	---

1er columna: se visualiza una lista desordenada (items no numéricos), la primer fila es xxx, la segunda yyy, la tercera zzz.

2da columna: se visualiza una lista ordenada (los items ahora son números), donde las filas contienen lo mismo que lo mostrado en el 1er caso / 1er columna.

3era columna: se igual que el caso 2 (o columna 2), pero esto es gracias a que se usan los atributos value en cada li, si esto no se hiciese, se visualizan 3 items y ambos ítems son el número 1. Esto sucede porque para cada una de las filas que antes se espacian con li, ahora directamente se espacian con ol. Como lo es para listas ordenadas, comienza en 1 cada vez que se abre otra lista, por eso son todos 1 los ítems.

4ta columna, se ve “similar” las columnas 2 y 3, pero no tan espaciado como si estuviese hecho por listas en lugar de párrafo. Se percibe la diferencia del espaciado.

3.d)

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


Ambos casos se ven de igual manera, no es conveniente usar el 2do caso (el de la derecha), es un código más engorroso sin necesidad de serlo, ya que definiendo a Columna

1 y Columna 2 en las celdas de encabezado era suficiente. Si bien se ven/perciben de igual forma del lado del usuario, lo “correcto” sería hacerlo como el 1er caso (lado izquierdo). Sin embargo, si se le quisiera hacer un diseño más interesante al título, ahí sí tendría sentido

3.e)

<pre><table width="200"> <caption></pre>	<pre><table width="200"> <tr></pre>
--	---

5

 Facultad Regional Rosario Universidad Tecnológica Nacional	Entornos Gráficos Diaz / Butti
<pre>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></pre>	<pre><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></pre>

Sucede lo mismo que en el ejercicio 3.d), el caso de la derecha usa un código más engorroso y no es claro, si se quiere poner un título a la tabla, ya existe el elemento caption para hacerlo, no hay necesidad de hacer todo lo que se hace en el lado derecho, por más que se ve de igual manera que el 1er caso (lado izquierdo). Sin embargo, si se le quisiera hacer un diseño más interesante al título, ahí sí tendría sentido.

3.f)

<pre><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></pre>	<pre><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></pre>
--	--

Resultado:

Titulo		

Titulo		

colspan permite unir dos celdas de forma horizontal (une columnas) y rowspan permite unir celdas de forma vertical (une filas). El 1er caso se corresponde con la tabla de arriba, el 2do caso con la tabla de abajo. La tabla de arriba al tener un rowspan, une la celda de arriba y de abajo (por ser rowspan = 2). En cambio, en la tabla de abajo, al tener un colspan une la celda de la izquierda con la del medio (por tener un colspan=2).

3.g)

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

Este es el resultado, siendo la tabla de arriba correspondiente al 1er caso y la de abajo al 2do caso:

Titulo		

Titulo		

En el 1er caso, en la 1er fila se unen las celdas de arriba de forma horizontal (colspan=3), y luego al tener en la 2da fila un rowspan=2 y un colspan=2, se unen cuatro celdas= las dos de la 2da fila y las 2 de la 3era fila.

En el 2do caso, en la primer fila nos dice que hay un colspan=2, que es allí donde esta el titulo y donde se unen la celda 1 y celda 2 de forma horizontal (porque se están uniendo las columnas). En la segunda fila, hay un rowspan=2, por lo que se unirán de forma vertical esa celda y la de abajo. Además en este caso, el borde es todo "relleno" ya que está el cellpadding = 0, y ocupa un espacio menor porque el cellspacing = 0.

3.h)

<pre> <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" />
 Clave: <input type="password" id="clave1" name="clave1" value="xxx" /> </fieldset> <input type="submit" id="boton1" name="boton1" value="Enviar" /> </form> </pre>
<pre> <form id="form2" name="form2" action="" method="get" target="_blank"> LOGIN
 <label>Usuario: <input type="text" id="usu2" name="usu2" /></label>
 <label>Clave: <input type="text" id="clave2" name="clave2" /></label>
 <input type="submit" id="boton2" name="boton2" value="Enviar" /> </form> </pre>
<pre> <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" />
 Clave: <input type="password" id="clave3" name="clave3" /> </fieldset> <input type="reset" id="boton3" name="boton3" value="Enviar" /> </form> </pre>

El caso 1 y el caso 2 (arriba y medio) se ven practicamente iguales, la diferencia esta en el primero que tiene el fieldset que recuadra al formulario:

LOGIN

Usuario: xxx

Clave: ...

Enviar

LOGIN

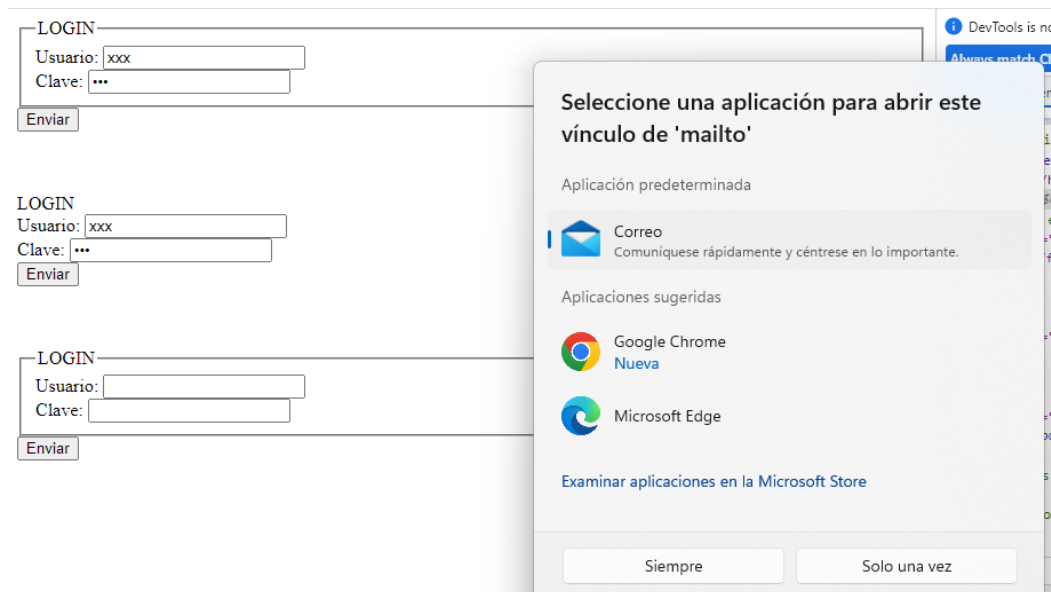
Usuario: xxx

Clave: ...

Enviar

Además otra diferencia es que el 2do caso no tiene action, por lo que al tocar el botón Enviar, simplemente redirige en otra ventana a la misma pagina, en cambio el 1ero abre otra ventana de error (ya que no existe procesar.php).

El 3er caso se ve similar al 1ero pero no muestra nada dentro de los input, y cuando cliqueamos en el boton Enviar, abre una nueva ventana para elegir por donde mandar el mail (que es lo que esta como accion)



3.i)

<pre> <label>Botón 1 <button type="button" name="boton1" id="boton1">
 CLICK AQUÍ</button></label> </pre>
<pre> <label>Botón 2 <input type="button" name="boton2" id="boton2" value="CLICK AQUÍ" /> </label> </pre>


En el primer caso podemos observar que posee una imagen dentro del botón mientras que en el segundo caso, el botón solo muestra texto.

En el caso 1, el elemento utilizado para representar al botón es la etiqueta `<button />`, mientras que en el caso 2, se utiliza un `<input />` con `type="button"` lo cual le permite al elemento tener el mismo comportamiento, es decir, que el elemento sea clickeable.

El elemento "button" para mostrar el texto y/o imagen requiere que estos elementos (``, ``) sean hijos.

En el caso del "input", el texto está dado por la etiqueta `value`. No es posible poner una imagen en el atributo "value" del input.

Botón 1 

Botón 2 

3.j)

3.j)

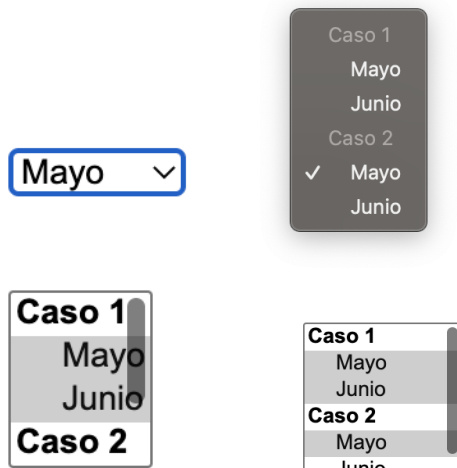
<pre><p><label><input type="radio" name="opcion" id="X" value="X" />X</label>
 <label><input type="radio" name="opcion" id="Y" value="Y" />Y</label></p></pre>
<pre><p><label><input type="radio" name="opcion1" id="X" value="X" />X</label>
 <label><input type="radio" name="opcion2" id="Y" value="Y" />Y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

Ambos casos se ven iguales, la única diferencia es que en el caso 1, solo puedo elegir X o Y en los radio button, esto es porque los nombres de ambos son iguales (opcion). En cambio en el caso 2 puedo seleccionar ambas opciones (ya que los nombres son distintos: opcion1 y opcion2) . En un conjunto de radio button, para que me permita elegir UNA opción, los nombre deben ser iguales (como en el caso 1), para elegir varias opciones, seria una buena practica usar checkbox antes que un radio button.

3.k)

<pre><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></pre>	<pre><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></pre>
--	--

Básicamente la diferencia entre el caso 1 y 2, es que el 2do permite seleccionar múltiples elementos. Visualmente, se puede notar una diferencia entre un selector colapsado y el otro expandido. El atributo "multiple" es el que le da la habilidad a la etiqueta <select /> de poder seleccionar varios elementos.



Ejercitación 4

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Practica 1</title>
</head>
<body>
  <header>
    <a href="https://www.frro.utn.edu.ar/"></a>
  </header>
  <br>
  <hr>
  <h1>AÑO 2008</h1>
  <table width="40%" cellpadding="0" border="1">
    <tr>
      <th colspan="2"><div align="center">Exámenes</div></th>
    </tr>
    <tr>
      <td rowspan="2">Fecha:</td>
      <td>Aprobados</td>
    </tr>
    <tr>
```

```

        <td>24/30</td>
    </tr>
</table>
<ol>
    <li><a href="#enunciado">Enunciado del Examen</a></li>
    <li><a href="resolucion.html"
target="_blank">Resolucion</a></li>
</ol>
<br>
<form id="formulario1" name="formulario1" action="mailto: x@x.com"
target="_blank">
    <label><strong>email:</strong> <input type="email" id="email1"
name="email1"></label> <br><br>
    <label><strong>consulta:</strong> <input type="text" id="cons1"
name="cons1"></label> <br><br>
    <input type="submit" id="boton1" name="boton1" value="ENVIAR">
    <input type="submit" id="boton2" name="boton2" value="BORRAR">
</form><br>
<h3 id="enunciado">Enunciado del Examen</h3>
<div width="30%">
    <p>
        Ejercicio 1:
        <br>
    </p>
    <p>
        Ejercicio 2:
        <br>
    </p>
</div>
<br>
<span>Para mas informacion dirigirse a </span><a
href="http://www.gmail.com.ar">contacto@xx.com</a>
<br>
</body>
</html>

```

Ejercitación 5

subida a github, se adjunta repositorio en el mail enviado.

