

CURSO DE HTML

Índice de contenido

1.- INTRODUCCIÓN.....	2
1.1 Definición.....	2
1.2 Versiones.....	2
1.3 Sintaxis.....	2
1.4 Test.....	3
2.- ESTRUCTURA DEL DOCUMENTO.....	4
2.1 Información.....	4
2.2 Cabecera.....	5
2.3 Cuerpo.....	6
2.4 Ejemplo práctico.....	6
2.5 Test.....	7
2.6 Ejercicio.....	8
3.- FORMATOS.....	9
3.1 Formatos de párrafo.....	9
3.2 Formatos de Frase.....	10
3.3 Formato de Letra.....	11
3.4 Estilo de texto.....	12
3.4 Estilo de texto.....	12
3.5 Inserción de espacios.....	14
3.6 Comentarios.....	15
3.7 Caracteres Especiales.....	15
3.8 Test.....	16
3.9 Ejercicio.....	17
4.- ENLACES.....	18
4.1 Tipos de Enlaces.....	19
4.2 Test.....	21
4.3 Ejercicio.....	22
5.- LISTAS.....	23
5.1 Test.....	28
5.2 Ejercicio.....	29
6.- IMÁGENES Y OBJETOS.....	30
6.1 Mapas de imagen.....	32
6.2 Test.....	33
6.3 Ejercicio.....	34
7.- TABLAS.....	35
7.1 Test.....	37
7.2 Ejercicio.....	38
8.- FORMULARIOS.....	39
8.1 Test.....	45
8.2 Ejercicio.....	46
9.- FRAMES.....	47
9.1 Test.....	50
9.2 Ejercicio.....	51
10.- MIGRACIÓN A XHTML.....	52
11.- EJERCICIO PRÁCTICO FINAL.....	54
12.- BIBLIOGRAFÍA.....	60

1.- INTRODUCCIÓN

1.1 Definición

A lo largo de este manual vamos a descubrir el lenguaje más utilizado para la creación de páginas web: el **Hyper Text Markup Language**, más conocido como **HTML**. Se puede afirmar que el HTML es el lenguaje usado por los navegadores para mostrar en la pantalla del usuario las páginas webs, es por tanto a día de hoy la interface más extendida en la red.

Con HTML podemos utilizar textos, sonidos, imágenes, colores, enlaces a otras páginas, tablas, formularios, etc. Lo que en un principio podía ser suficiente para hacer llegar la información al usuario, llegó un momento que el desarrollo de internet precisó que el lenguaje HTML se actualizara. Por ello se introdujeron tecnologías complementarias como es el caso de CSS y Javascript.

Otra cuestión a tener en cuenta, es el uso de estándares web. ¿Por qué? Porque las diversas versiones de los navegadores no lo implementan adecuadamente. Por eso el webmaster está obligado a comprobar que su web será interpretada correctamente.

1.2 Versiones

La primera versión más usada de HTML fue la 2.0 en el año 1995. En un principio cumplía más que suficiente la presentación del contenido, que primaba sobre el diseño. Pero debido a que Netscape empezó a incluir nuevas etiquetas no estándares, fue preciso crear un comité que estableciera las bases estándares web en la versión HTML 3.0. Dado que no fue muy aceptado, la principales compañías de informática crearon el nuevo comité W3C, que es la que actualmente elabora las versiones de HTML. En 1997 se presentó entonces la versión 3.2, que incluían nada más que lo ya hecho hasta el momento. En seguida, tal que el mismo año, se presenta una nueva versión que es la que continúa vigente en este momento la 4.0, y es la que veremos en este curso. En el año 2000 esta última versión junto con el lenguaje XML 1.0. fue reformulado en XHTML 1.0.

1.3 Sintaxis

La sintaxis del lenguaje HTML se basa en la denominada etiqueta. La etiqueta tiene dos partes: una apertura y un cierre. Encontrando en medio un elemento cuyo estilo vendrá definido con las reglas CSS.

<etiqueta> contenido</etiqueta>

La etiqueta que se encuentra al principio activa la orden y la última es la misma que la que abre precedida del signo / y lo que hace es desactivarla. No siempre es así, existen etiquetas con una etiqueta al principio y sin etiqueta de cierre. Y su sintaxis es:

<etiqueta />

Es importante resaltar que las mayúsculas o minúsculas son indiferentes a la hora de escribir las etiquetas. Pero debido al uso complementario de otras tecnologías es aconsejable escribirlas en minúsculas.

1.4 Test

1. La sintaxis de un código HTML común puede ser:

```
<etiqueta valor="atributo" >elemento</etiqueta>  
<etiqueta atributo="valor">elemento</etiqueta>  
<atributo etiqueta="valor">elemento</atributo>  
<valor etiqueta="atributo">elemento</valor>
```

2. Con respecto a los estándares de la web.

Son creados por la W2C.
El HTML debe escribirse en mayúscula.
No es necesario guiarse por ellos.
Ninguna es correcta.

3. HTML significa:

HyperText Match Language
Lenguaje de Marcado HiperTextual
HyperText Markup Language
Lenguaje de Marcado HiperTextónico

2.- ESTRUCTURA DEL DOCUMENTO

Un documento HTML se compone de las siguientes partes:

1. información sobre la versión del HTML
2. cabecera declarativa por la etiqueta <head>
3. cuerpo, que contiene el contenido real del documento indicada por <body>

Ejemplo:

```
<!doctype html public "-//w3c//dtd html 4.01//en"
"http://www.w3.org/tr/html4/strict.dtd">
<html>
  <head>
    <title>mi primer documento html</title>
  </head>
  <body>
    <p>ihola mundo!</p>
  </body>
</html>
```

2.1 Información

Un documento HTML válido declara qué versión de HTML se utiliza en el documento. Esta declaración está codificada y viene especificada por el DOCTYPE.

El objetivo es indicar a un Validador de Versiones, qué versión del HTML utilizar en la comprobación de sintaxis del documento. Según los estándares del HTML, cada documento requiere un tipo de declaración diferente. Se detallan a continuación las últimas definiciones del tipo de documento (DTD):

- <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

En esta declaración se incluyen todos los componentes excepto los frames.

- <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

En esta declaración se incluyen todos los componentes de HTML 4.0 Strict más aquellos desaconsejados de versiones anteriores, sin incluir los frames. Es la versión más utilizada aunque no obtenga todas las ventajas que persigue HTML 4.0, tiene un soporte mayor por parte de los browsers.

- <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">

En esta declaración se incluyen los componentes del HTML Transitional más los frames.

Si queremos seguir la normativa de la W3C: *"guiar la web hacia su máximo potencial a través del desarrollo de estándares"*, necesitamos crear nuestros documentos con el formato estándar XHTML, es decir, con la extensión del lenguaje de marcado hipertextual (eXtensible HyperText Markup Language). Algo que veremos más adelante.

Desde este punto de vista, estandarizar un sitio web con XHTML 1.0 strict, es la mejor forma de crear una pagina web correcta con el DTD.

La codificación, como segundo componente de una declaración, es el formato en el que se guardan los caracteres en el archivo. Se suele utilizar la Occidental charset=iso-8859-1 o bien Unicode charset=utf-8.

```
<meta http - equiv =" Content - Type " content =" text / html ; charset =utf -8" />
```

Después de la declaración nos encontramos con la etiqueta <html> en la que vamos a encontrar todo el contenido de la página web y a veces la indicación de qué lengua estamos usando.

```
<html xmlns =" http :// www.w3.org /1999/ xhtml \"xml: lang ="es" lang ="es">
```

en esta etiqueta vamos a encontrar 2 bloques:

<head>...</head> que señala la cabecera del documento y encontramos metadatos, scripts, enlaces a archivos externos de CSS, titulo, ...

<body>...</body> que contiene el cuerpo del documento, es decir, lo que visualizamos en el navegador con el formato correspondiente.

2.2 Cabecera

El elemento <head> contiene información sobre el documento actual, como el título, palabras clave que pueden ser de utilidad para motores de búsqueda, y otros datos que no se consideran estrictamente parte del contenido del documento.

- Title
Es muy conveniente utilizar el elemento <title> para identificar los contenidos de un documento.
- Metadatos

El elemento metadatos es un contenedor de metainformación sobre el documento relacionado con el protocolo HTTP. Se utiliza para identificar al documento y su contenido. Esta información es utilizada por los buscadores e índices. Por ejemplo: el autor, la fecha de caducidad, una lista de palabras clave, etc. Siempre van en la etiqueta head. Esta especificación no define un conjunto normativo de propiedades.

Los atributos más importantes de los metadatos son:

- **name:** identifica la información que se va a exportar.
- **content:** contenido de la información.
- **http-equiv:** es el nombre de un comando HTTP.

Cada atributo va a especificar una pareja propiedad/valor. Por ejemplo, la siguiente declaración establece un valor para la propiedad Author:

```
<META name="Author" content="Davor Rakleet">
```

El atributo name identifica la propiedad y el atributo content especifica el valor de la propiedad.

2.3 Cuerpo

El cuerpo de un documento es la parte que contiene el contenido del documento. El contenido puede ser presentado como un lienzo sobre el que aparece el contenido: texto, imágenes, colores, gráficos, etc. Para agentes de usuario por voz, el mismo contenido podría ser pronunciado.

El estilo que presentará el contenido de nuestra página web estará en un archivo css, pero el contenido vendrá definido por los siguientes elementos:

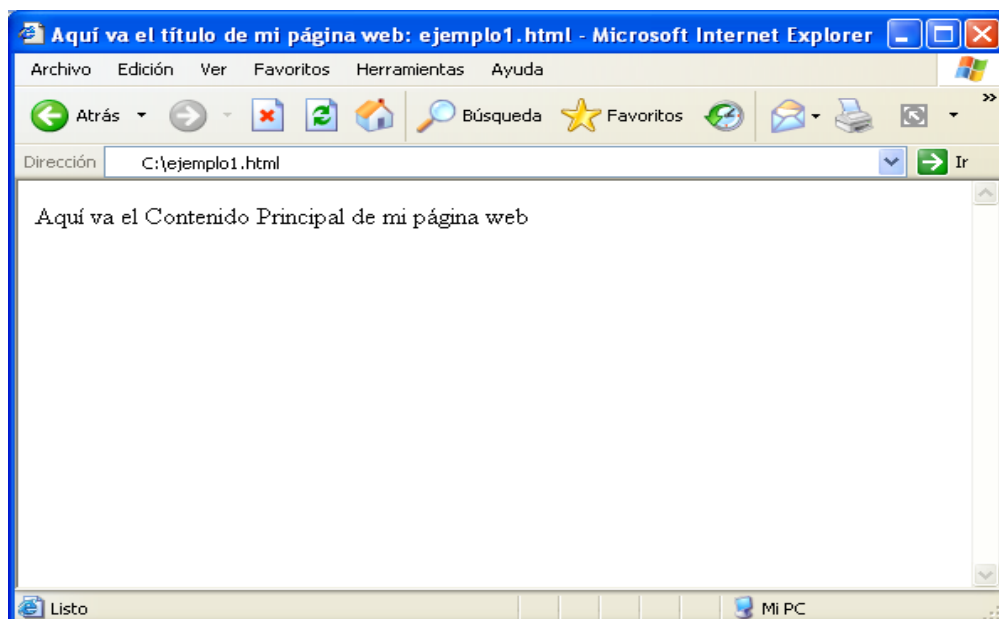
- identificadores de elementos como los atributos id y class
- elementos en bloque (div) y elementos en línea (span)
- encabezados (h1, h2, h3, h4, h5 y h6)

2.4 Ejemplo práctico

A continuación se propone un ejemplo resuelto con la una estructura básica de una página web:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Aquí va el título de mi página web</title>
</head>
<body>
Aquí va el Contenido Principal de mi página web
</body>
</html>
```

Resultado:



2.5 Test

1. Para hacer una declaración, que doctype de los siguientes es el correcto:

```
<!DOCTYPE html PUBLIC "-//W2C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 1.0 Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Cualquiera de los anteriores

2. En el elemento <head>:

Se incluye la etiqueta <hr>
Se incluye la etiqueta <h4>
Se incluye la etiqueta <bgcolor>
Ninguna de las anteriores

3. En un documento HTML podemos incorporar:

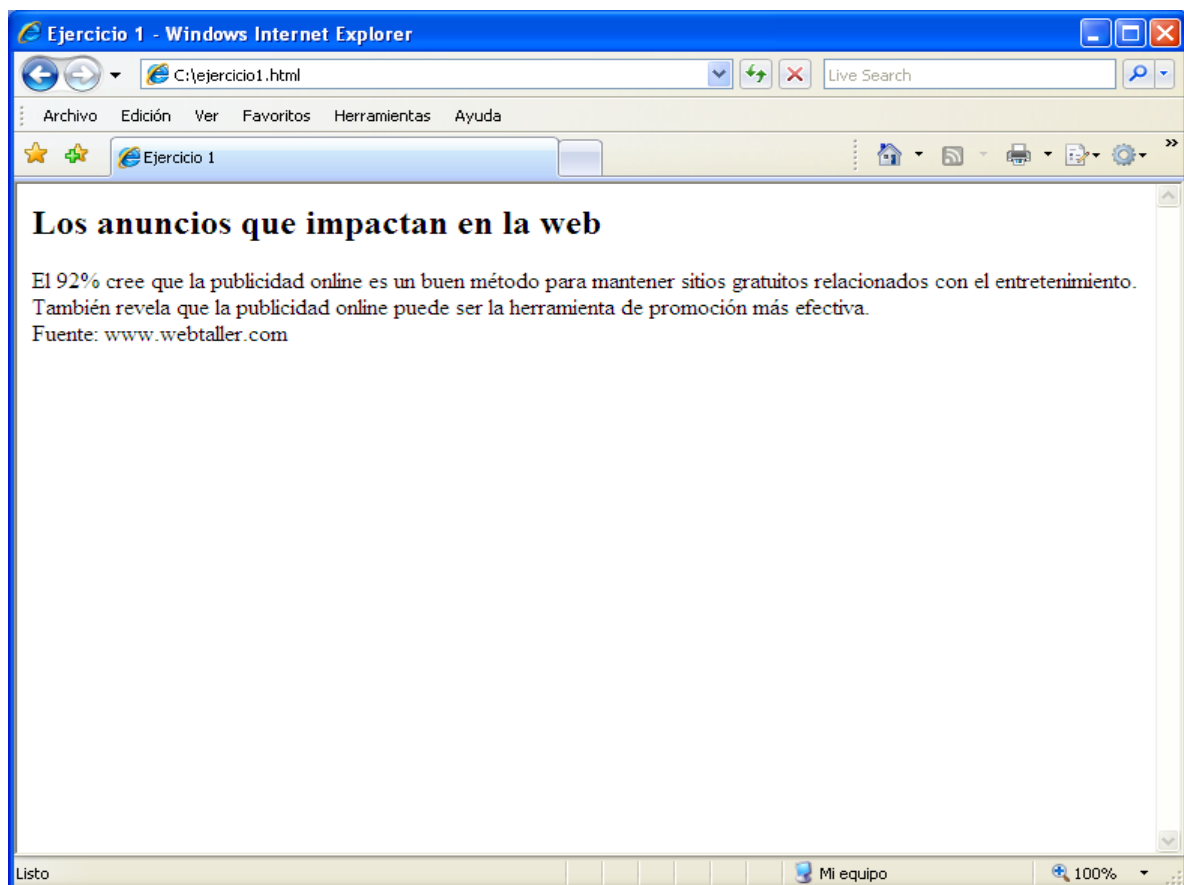
Hojas de estilo CSS
Formularios de contacto
Imágenes GIF
Todas las anteriores

2.6 Ejercicio

Le llamaremos ejercicio1.html, simplemente con un editor de texto (notepad, bloc de notas, ultra-edit, etc) escribimos el código y lo guardamos con un nombre y extensión html. Esto es interpretado por el navegador que tengamos instalado en nuestro ordenador. Las características de esta página web son:

- Utilizamos un DTD transicional.
- En los metadatos:
 - . http-equiv=content-type
 - . content=text/html
 - . charset=iso-8859-1
- El título del navegador es "Ejercicio 1"
- En el body pondremos un encabezado tipo <h2>: "Los anuncios que impactan en la web"
- En dos párrafos (<p>) incluir el siguiente texto:
 - . El 92% cree que la publicidad online es un buen método para mantener sitios gratuitos relacionados con el entretenimiento. También revela que la publicidad online puede ser la herramienta de promoción más efectiva.
 - . Fuente: www.webtaller.com

Resultado:



3.- FORMATOS

Es muy conveniente conocer las etiquetas con las que dar formato a los textos, ya que es la mayor parte del contenido de una página web. De esta forma podremos estructurar el texto en párrafos, titulares, citas, etc para darles un estilo.

3.1 Formatos de párrafo

Los párrafos sirven para dar estructura al contenido. Vamos a ver a continuación las etiquetas más importantes:

- ◆ `<p>` y `</p>`: Esta etiqueta enmarca un texto dentro de un párrafo. Crea un salto de línea y además deja una línea en blanco antes de continuar con el resto del documento.

Para una buena presentación visual de un párrafo hay que tener en cuenta aspectos técnicos y de estilo como:

- Tratamiento del espacio en blanco
 - Saltos de línea y cambio automático de línea
 - Justificación
 - División de palabras
 - Convenciones del idioma escrito y direccionalidad del texto
 - Formateo de los párrafos con respecto al contenido circundante
- ◆ `<center>` y `</center>`: Permite centrar el párrafo
 - ◆ `<pre>` y `</pre>`: Esta etiqueta indica que el texto contenido está "preformateado", es decir, ya viene con su fuente de letra, tamaño, etc. Por tanto el navegador respeta la forma en que está escrito. Resulta útil a la hora de representar código fuente.
 - ◆ `<blockquote>` y `</blockquote>`: Se utiliza para citar un texto ajeno. Al implementarse deja márgenes tanto a izquierda como a derecha.
 - ◆ `<div align>` y `</div>`: Hay ocasiones en que el texto está compuesto por varios párrafos, y que para poder darle el mismo estilo es conveniente incluirlos en una capa contenedora o más conocida como div. El atributo a utilizar es align, que permite justificar el texto del párrafo. Es un atributo que ayuda a posicionar el párrafo a la izquierda (`div align=left`), a la derecha (`div align=right`), al centro (`div align=center`) o a ambos márgenes (`div align=justify`).

Ejemplo:

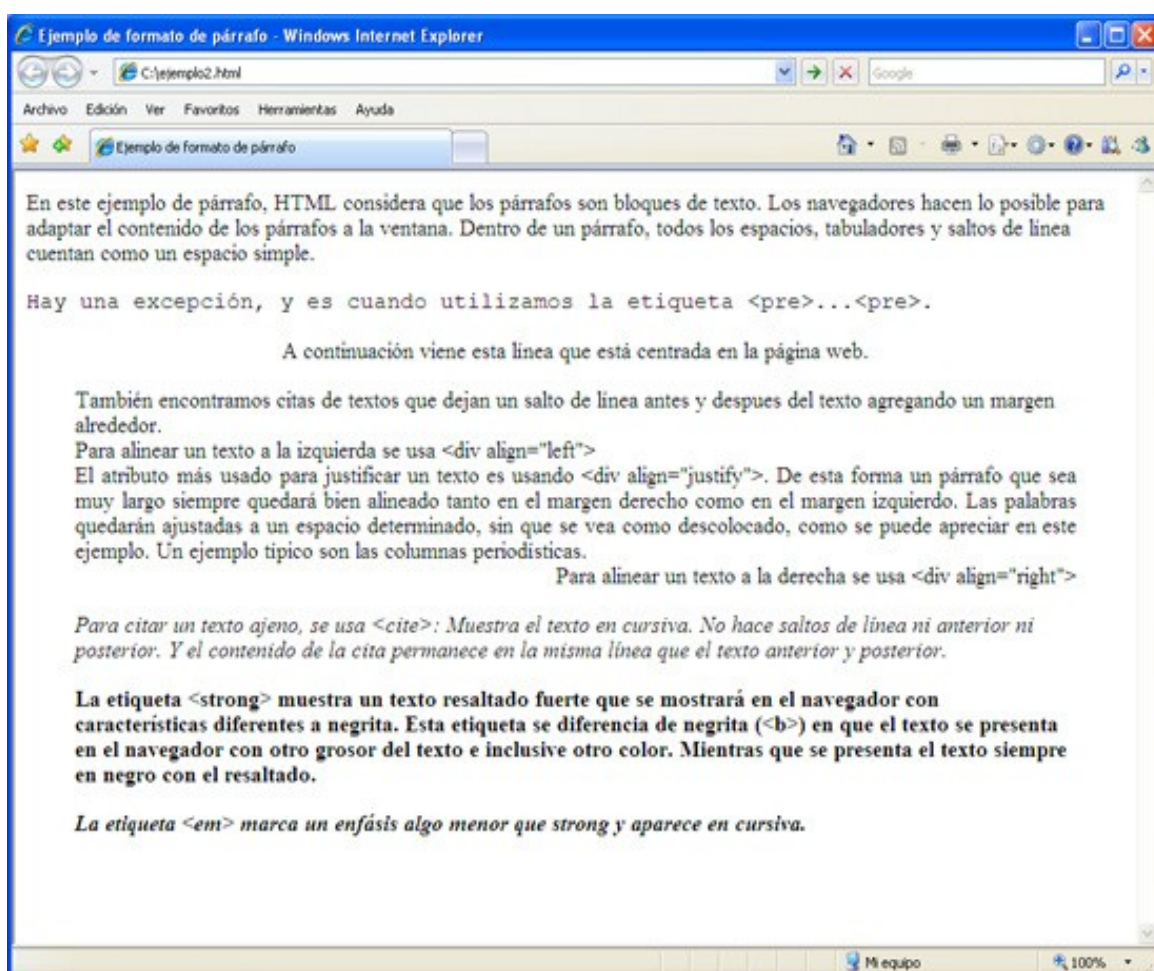
```
<div align="justify">
<p>Aquí va el primer párrafo</p>
<p>Aquí va el segundo párrafo</p>
</div>
```

- ◆ `<text-align>` y `</text-align>`: Indica la alineación del texto. Los valores que admite son left, right, center y justify.

3.2 Formatos de Frase

- ♦ `<cite>` y `</cite>`: Para citar un texto ajeno.
- ♦ `` y ``: Esta etiqueta muestra un texto resaltado fuerte que se mostrará en el navegador con características diferentes a negrita. Esta etiqueta se diferencia de negrita (``) en que el texto se presenta en el navegador con otro grosor del texto e inclusive otro color. Mientras que `` se presenta el texto siempre en negro con el resaltado.
- ♦ `` y ``: Marca un énfasis algo menor que strong y aparece en cursiva.

Ejemplo:



Una parte importante del texto suelen ser los títulos o encabezados. Un encabezado es una etiqueta que se refiere al título de una sección. Da formato al texto. Hay seis niveles de encabezados en HTML: H1, H2, H3, H4, H5 y H6, siendo H1 el más importante y H6 el menos importante. Los navegadores visuales pueden representar los encabezados más importantes con fuentes más grandes que los menos importantes.

Este encabezado es de nivel 1

Este encabezado es de nivel 2

Este encabezado es de nivel 3

Este encabezado es de nivel 4

Este encabezado es de nivel 5

Este encabezado es de nivel 6

3.3 Formato de Letra

- ♦ `` y ``: El texto aparece en negrita.

Ejemplo: **Este texto está muy negro**

- ♦ `<i>` y `</i>`: El texto aparece en cursiva.

Ejemplo: *Me gusta mirar de lado*

- ♦ `<u>` y `</u>`: El texto aparece subrayado.

Ejemplo: Es muy importante que aparezca subrayado

- ♦ `^{` y `}`: Muestra un texto como superíndice.

Ejemplo: $E=mc^2$

- ♦ `_{` y `}`: Muestra un texto como subíndice.

Ejemplo: $\sum_{i=1}$

- ♦ `<big>` y `</big>`: Incrementa el tamaño de letra.

Ejemplo: ME VEO GRANDE

- ◆ `<small>` y `</small>`: Decremento del tamaño de la letra.

Ejemplo: me veo pequeño

- ◆ `<blink>` y `</blink>`: El texto parpadea.

Ejemplo: Estoy parpadeando

3.4 Estilo de texto

- ◆ `<font-family>` y `</font-family>`: Determina el tipo de fuente que presenta el texto de la página web.

Ejemplo:

```
<font-family="Helvetica,Arial,Times">Tengo una fuente de letra Helvetica, Arial o Times.</font-family>
```

Resultado:

Tengo una fuente de letra Helvetica, Arial o Times.

- ◆ `<font-size>` y `</font-size>`: Determina el tamaño de fuente. Se puede indicar a través de los siguientes valores:

xx-small | x-small | small | medium | large | x-large | xx-large

o bien con cualesquiera de las unidades de medidas que se usa en el estándar de CSS.

Ejemplo:

```
<font-size="22px">Tamaño 22 píxeles</font>
```

Resultado:

Tamaño 22 píxeles

- ◆ `<font-color>` y `</font-color>`: Esta etiqueta se usa para indicar el color de la fuente. En el estándar CSS hay varias maneras de definir un color:

- Nombre: aunque existen miles de colores sólo hay unos 17 que se pueden indicar por su nombre, porque son los que están incluidos en el estándar CSS.

```
h1 {color: blue}
```

- Notación hexadecimal: Pueden definirse más de 16 millones de colores al especificarse los valores RGB de un color con la notación `#RRGGBB`, donde RR, GG y BB representan, cada uno, un número hexadecimal entre 00 y FF para rojo, verde y azul, respectivamente.

h1 {color: #0000FF} o en su forma abreviada h1 { color: #00F }

- Notación decimal: CSS también define una notación en función de sus proporciones relativas en porcentaje o en números enteros positivos (cada valor puede ser un número entero entre 0 y 255).

rgb(100%,100%,100%)

rgb(24,125,255)

Ejemplo:

```
<div style="color: #ff0000;">Text en color rojo.</div>
```

Resultado:

Texto en color rojo.

- ◆ <font-style> y </font-style>: Esta etiqueta nos da variados estilos al texto.
 - normal (estilo normal)
 - itálica (estilo cursiva que propone la fuente)
 - oblicua (estilo oblicuo por deformación de la fuente)
- ◆ <font-weight> y </font-weight>: Otorga diferentes grosores al tamaño de la fuente. Los valores que presenta esta propiedad son:
 - normal (valor por defecto, sin estilo)
 - bold (estilo negrita)
 - lighter (mas pesado)
 - bolder (mas liviano)
 - 100
 - 200
 - 300
 - 400 (grosor por defecto)
 - 500
 - 600
 - 700
 - 800
 - 900
- ◆ <text-decoration> y </text-decoration>: Con esta etiqueta se permite adornar o no el texto.
 - none: el texto no tiene decoración. Es muy útil para quitar los subrayados a los enlaces.
 - underline: el texto aparece subrayado
 - overline: el texto aparece sobrerayado
 - line-through: el texto aparece tachado

Por último, señalar que normalmente al asignar un formato a un texto indicamos la fuente, su tamaño, color, y/o estilo. Por ello se suelen abreviar los valores de la siguiente manera:

```
<style type="text/css">
```

```
body {font:bold italic 25px verdana;}
```

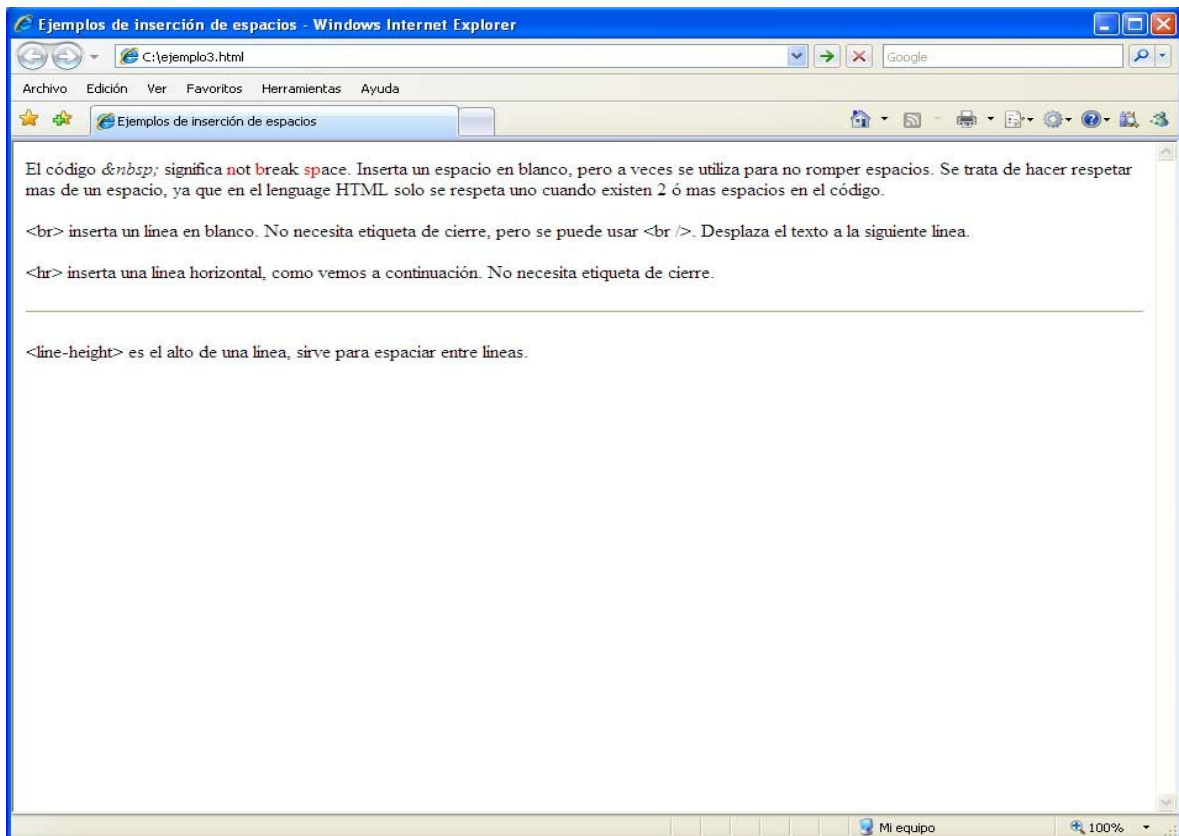
```
</style>
```

3.5 Inserción de espacios

HTML tiene un comportamiento especial con respecto a los espacios en blanco. Como sólo respeta los espacios existentes entre palabras, fue preciso crear etiquetas específicas para insertar espacios en blanco. Veamos cuales son:

- ◆ ` `: inserta un espacio en blanco
- ◆ `
`: inserta un línea en blanco. No necesita etiqueta de cierre, pero se puede usar `
`. Desplaza el texto a la siguiente línea.
- ◆ `<hr>`: inserta una línea horizontal. No necesita etiqueta de cierre.
- ◆ `<line-height>`: es el alto de una línea, sirve para espaciar entre líneas

Ejemplo:



3.6 Comentarios

Para insertar aclaraciones dentro del código fuente y que resulten invisibles en la pantalla del navegador, tenemos que escribirlo entre `<!--` y `-->`.

Ejemplo:

```
<!-- Inicio del menú -->
```

3.7 Caracteres Especiales

Es preciso tener en cuenta que los ordenadores manejan caracteres especiales para ciertas letras o símbolos. Para insertar el carácter adecuado se toma como referencia la tabla estándar ISO-Latin-1. Se detallan a continuación los más usados:

Código	Resultado
á Á é É í Í ...	á, Á, é, É, í, Í, ó, Ó, ú y Ú
ñ Ñ	ñ Ñ
¿	¿
¡	¡
 	Inserta espacio en blanco
<	<
>	>
&	&
"	"

3.8 Test

1. La etiqueta de fuente y sus atributos:

Actualmente no se promociona su uso con normalidad
No cumple con los estándares XHTML
Fue sustituida por propiedades CSS
Todas son correctas

2. Cual de las siguientes reglas es incorrecta:

<font-weight: bolder>
<text-decoration: overline>
<h7>Titulo</h7>
palabra enfatizada

3. Cual de las siguientes definiciones es correcta:

Los párrafos <p>son bloques de texto
La etiqueta <div> define los bloques de elementos. Esta estructura está creada para definir recuadros como menús o ventanas.
La etiqueta <title> está dentro de la cabecera. Aparece como título de la ventana en los navegadores.
Todas son correctas

3.9 Ejercicio

Para crear el siguiente documento web, llamado ejercicio2.html, insertaremos el siguiente título:

“2009 RIBA Award Winners Announced”

y el siguiente texto:

The Royal Institute of British Architects (RIBA) recently announced the winners of the 2009 RIBA Awards. RIBA Awards for architectural excellence have been presented across the country with 103 buildings in the UK and Europe winning awards (97 in the UK and six in the rest of the EU).

El estilo que daremos será el siguiente:

- Añadir el siguiente comentario en cualquier parte del código: “Noticias sobre arquitectura”.
- El título irá con el encabezado de nivel 1.
- Entre el encabezado y el texto insertar una línea.
- El texto, que va en formato párrafo, irá dentro de un div con una alineación justificada.
- El siguiente texto “Royal Institute of British Architects (RIBA)” estará enfatizado.

Resultado:



4.- ENLACES

En lo que se diferencia HTML de otros lenguajes es en sus características para hipertexto y documentos interactivos. Es decir, la posibilidad de conectar una parte con otra del documento web o con otra web dondequiera que se encuentre. Para ello se usan los vínculos como estructura básica y sencilla del hipertexto, lo cual ha posibilitado el éxito de la world wide web.

Un vínculo o enlace es una área de texto o imagen que al ser seleccionados nos trasladan a otro documento de hipertexto o a otra posición dentro del documento actual. El objetivo es que el usuario consiga navegar por los contenidos de la web de tal forma que consiga encontrar la información que busca. Veamos a continuación la estructura de un hipervínculo:

`contenido`

Un vínculo tiene dos extremos llamados anclas y una dirección. El vínculo comienza en el "ancla de origen", que corresponde al origen del vínculo y apunta al "ancla destino" que corresponde al destino del vínculo; éste puede ser cualquier recurso de la Web, desde una imagen a un videoclip, un archivo de sonido, un programa, un documento HTML, un elemento dentro de un documento HTML, etc. Veamos detalladamente cada parte del hipervínculo:

1º Para colocar un enlace, nos serviremos de las etiquetas `<a>` y ``. De forma que lo que se encuentre entre estas etiquetas se considera un enlace y se va a visualizar de forma diferente en el navegador y es fácilmente detectado por el usuario dentro de la página web. El texto aparecerá subrayado y de un color distinto al habitual, y las imágenes estarán rodeadas por un borde del mismo color que el del texto del enlace. Basta con deslizar el puntero del ratón sobre las imágenes o el texto y ver como estarán subrayados y coloreados en azul. En el caso de las imágenes que sirvan de enlace, veremos que están delimitadas por un marco azul por defecto. Al pulsar sobre el enlace, seremos enviados al documento que apuntaba el enlace.

Dentro de la etiqueta de apertura deberemos especificar asimismo el destino del enlace o dirección. Este destino será introducido bajo forma de atributo, el cual lleva por nombre href.

2º La dirección o destino del enlace estará en formato URL. Una URL nos indica tanto una dirección de Internet como el servicio que esperamos nos ofrezca el servidor al que corresponde la dirección. Tiene el siguiente formato:

`servicio://máquina:puerto/ruta/fichero@usuario`

donde el servicio podrá ser uno de los siguientes:

http: Es el servicio invocado para transmitir páginas web y el que usaremos normalmente en los enlaces.

https: Es una innovación sobre el anterior, que nos permite acceder a servidores (generalmente comerciales) que nos ofrecen el uso de técnicas de encriptación para proteger los datos que intercambiamos con él de terceras personas.

ftp: Permite transmitir ficheros desde servidores de *ftp* anónimo. Si no le pedimos un fichero sino un directorio, en general el navegador se encargará de mostrarnos el contenido del mismo para que podamos escogerlo cómodamente. Utilizando la @ podremos acceder a servidores privados.

mailto: Para poder mandar un mensaje. Por ejemplo, la URL *mailto:info@mipagina.com* enviaría un mensaje a la bandeja de entrada.

news: Para poder acceder a foros de discusión. Se indica el servidor y el grupo. Por ejemplo: *news://news.iber.net.es/es.comp.demos* nos conectaría con el foro *es.comp.demos* en el servidor de Telefónica.

telnet: No es implementado generalmente por los navegadores, que suelen invocar un programa externo. Nos permite conectarnos con otros ordenadores y entrar en ellos como si nuestro ordenador fuese una terminal del mismo.

La dirección de la máquina puede ser, o bien una serie de cuatro números entre 0 y 255 (123.3.5.65) o bien algo más fácil de recordar como es una serie de palabras separadas por puntos (*www.programacion.net*). El puerto generalmente no se indica, ya que el servicio predetermina uno.

La ruta es una serie de directorios separados por el símbolo /, que es el utilizado en UNIX (el sistema operativo más extendido en los servidores de Internet).

Existe otro formato de URL. Cuando queremos acceder a un fichero situado en la misma máquina que la página web que estamos creando podemos utilizar este formato:

ruta_relativa/fichero

En la ruta relativa podremos utilizar los dos puntos (..) para acceder al directorio padre o comenzar con la barra diagonal (/) para acceder a una ruta absoluta dentro de nuestro ordenador.

3º Para acceder al destino del enlace, primero debemos especificar el lugar del documento al que queremos acceder con el llamado ancla:

4.1 Tipos de Enlaces

En función del destino los enlaces pueden ser:

- ◆ **Enlaces internos:** los que se dirigen a otras partes dentro de la misma página.

Ejemplo:

En el lugar desde el que queremos saltar, escribimos:

```
<a href="#inicio"> Ir al Inicio</a>
```

Y en el punto de destino, escribimos la siguiente etiqueta:

```
<a name="inicio"> </a>
```

- ◆ **Enlaces locales:** los que se dirigen a otras páginas del mismo sitio web.

Ejemplo:

Para ir de la página inicial a la página final, escribimos:

```
<a href="pagina_final.htm">Ir a pagina final</a>
```

- ◆ **Enlaces remotos:** los dirigidos hacia páginas de otros sitios web.

Ejemplo:

Indicamos la dirección absoluta de una página web.

```
<a href="http://www.w3c.es/">Oficina española de la W3C</a>
```

- ♦ **Enlaces usando imágenes:** Con frecuencia se emplean imágenes como enlaces. A veces son enlaces a las mismas imágenes con más resolución y otras veces nos llevan a una localización cualquiera.

Ejemplo:

Para que un texto sea un enlace a una imagen, escribimos:

```
<a href="#inicio"></a>
```

Para que una imagen sea un enlace a otra imagen se debe escribir:

```
<a href="carpeta1/imagen1.jpg"></a>
```

- ♦ **Enlaces con direcciones de correo:** para crear un mensaje de correo dirigido a una dirección.

Ejemplo:

```
<a href="mailto: nombre@tienesunemail.es">
```

```
nombre_apellido@tienesunemail.es</a>
```

- ♦ **Enlaces con grupos de noticias:** para enviar y recibir mensajes, acceder a foros de discusión, etc.

Ejemplo:

```
<a href="news://news.tienesunemail.es/">Acceso al grupo de noticias  
de tienesunemail</a>
```

- ♦ **Enlaces con archivos:** los usuarios puedan hacer download de ficheros, es importante indicar la ruta donde se encuentran los ficheros.

Ejemplo:

```
<a href="../carpeta1/temario.zip">Descarga del archivo temario</a>
```

- ♦ **Enlaces que abren nuevas ventanas:** si queremos que la información a la que queremos acceder se muestre en una nueva ventana, solamente le añadimos target="nueva"

Ejemplo:

```
<a href="index.htm" target="nueva">Indice</a>
```

La forma en que se accede a los archivos pueden ser:

Las direcciones absolutas son aquellas que contienen la url completa. En estas direcciones podemos ver todos los elementos de la dirección.

Ejemplo:

```
<a href="http://webcompleta.com/html/imagenes/prueba.jpg" >
```

En las direcciones relativas vemos que no está la url completa. En este tipo de direcciones, las partes que faltan de la dirección el navegador las genera de la propia página en la que está el enlace.

Ejemplo:

```
<a href="/imagenes/prueba.jpg" >
```

4.2 Test

1. Señale la sintaxis correcta:

```
<a href="http://www.pagina.com">mi web</a>  
<href a="/pagina.html">mi web</href>  
<a href="www.pagina.com">mi web</a>  
<a title="http://www.pagina.com">mi web</a>
```

2. Nuestro documento HTML puede tener enlaces hacia:

Archivos .exe y .zip
Webs .html y .php
Documentos .pdf
Todas las anteriores

3. Indica cual es verdadera o falsa:

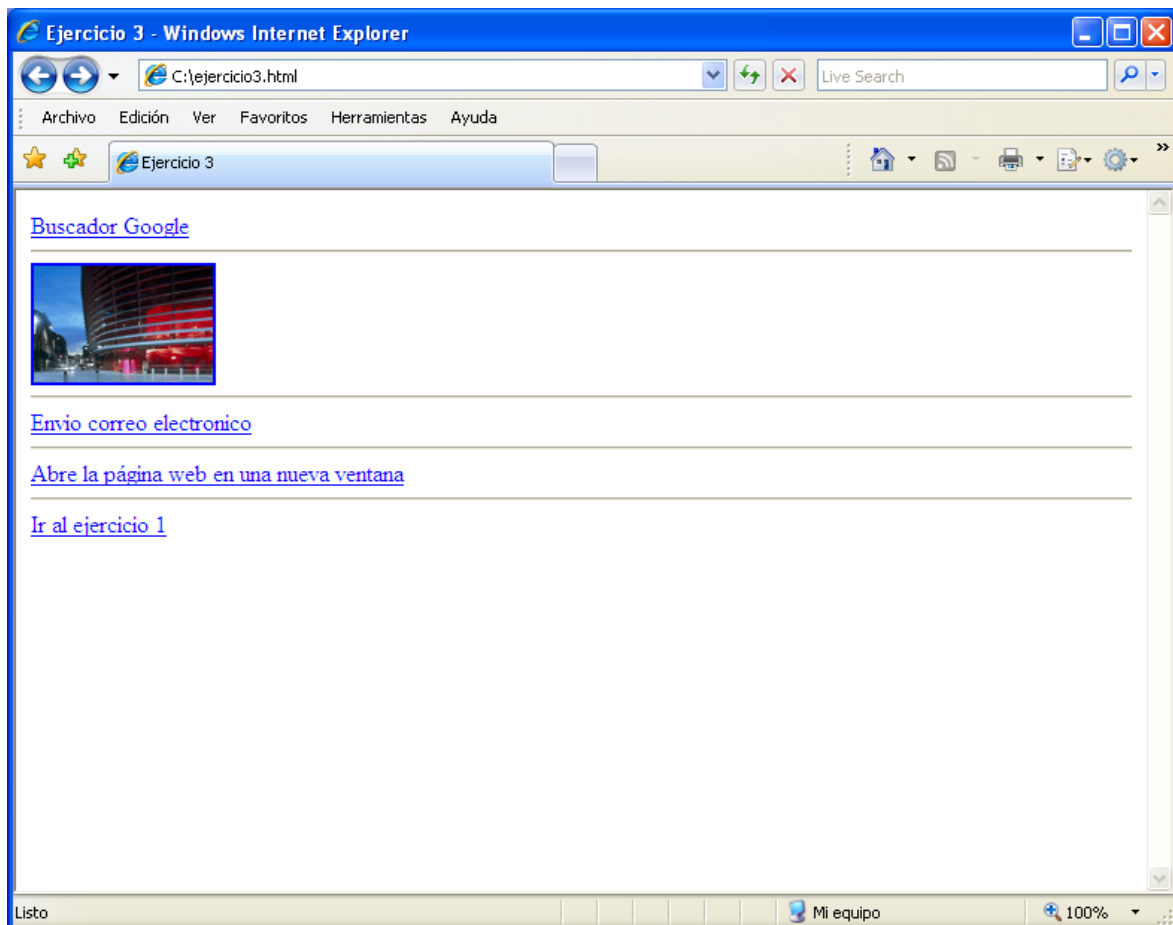
Las imágenes que sirven de enlace están delimitadas por un marco rojo
Un ejemplo de dirección absoluta: ``
El siguiente enlace lleva a la parte final de la página 2.htm
`Ir a sitio de la página 2`
Sólo se pueden crear 5 enlaces internos por cada página web

4.3 Ejercicio

Este ejercicio3.html va a mostrar diferentes formas de hipervínculos:

- Crear un enlace al buscador Google que se muestre en la misma ventana.
- Insertar una imagen, la cual al pinchar sobre ella sea un enlace al sitio externo www.noticiasarquitectura.info, y que se muestre en la misma ventana.
- Que el texto "Envío correo electronico" sea un enlace a una dirección de email.
- Mostrar la siguiente web <http://www.universitarios.org/> en una nueva ventana, al pinchar sobre el texto "Abre la página web en una nueva ventana".
- Que el texto "Ir al ejercicio 1" nos muestre en la misma ventana el ejercicio1.html.

Resultado:



5.- LISTAS

Las listas sirven para enumerar y definir elementos, los textos preformateados y las cabeceras o títulos. Al comienzo de cada línea se antepone un símbolo o número para indicar la secuencia de elementos, además su uso hace que el comienzo de la línea se desplace hacia la derecha.

Veamos los tipos de listas:

◆ Listas desordenadas

Se delimitan por las etiquetas `` y `` (unordered list). Cada uno de los elementos de la lista se cita con la etiqueta ``, aunque no es preciso cerrarlo.

Se puede definir el tipo de viñeta a emplear. Se especifica con el atributo `type`, y se incluye con la etiqueta de apertura ``, si queremos que el estilo sea válido para toda la lista, o dentro de la etiqueta `` si queremos hacerlo específico para un solo elemento. La sintaxis es del siguiente tipo:

```
<ul type="tipo de viñeta">  
<li type="tipo de viñeta">
```

donde tipo de viñeta puede ser `circle`, `disc`, `square`, etc. Indicar que en algunos navegadores estos símbolos no se presentan bien y muestran el típico redondo y negro.

Ejemplo:

Se muestra una lista con un cuadrado en lugar de un redondel, y en el último elemento colocaremos un círculo. Para ello vamos a colocar el atributo `type` en la etiqueta ``, con lo que afectará a todos los elementos de la lista.

```
<ul type="square">  
<li>Elemento 1  
<li>Elemento 2  
<li>Elemento 3  
<li type="circle">Elemento 4  
</ul>
```

Resultado

- Elemento 1
- Elemento 2
- Elemento 3
- Elemento 3
- Elemento 4

◆ Listas ordenadas

Las etiquetas que se utilizan son `` y `` (ordered list). Cada elemento será igualmente precedido de su etiqueta ``.

Ejemplo:

```
<p>Reglas de ordenación</p>
<ol>
  <li>Primer elemento
  <li>Segundo elemento
</ol>
```

Resultado:

Reglas de ordenación

1. Primer elemento
2. Segundo elemento

Las listas ordenadas ofrecen la posibilidad de modificar el estilo de las viñetas. Es posible especificar el tipo de numeración eligiendo entre números, letras y números romanos. Se utiliza, como ya hemos visto, el atributo `type`, el cual será situado dentro de la etiqueta ``, con uno de los siguientes valores:

1 Para ordenar por números : (1, 2, 3...),

a Por letras del alfabeto: (a, b, c...)

A Por letras mayúsculas del alfabeto: (A, B, C,...)

i Ordenación por números romanos en minúsculas: (i, ii, iii,...)

I Ordenación por números romanos en mayúsculas: (I, II, III,...)

Puede que en algún caso deseemos comenzar nuestra enumeración por un número o letra que no tiene por qué ser necesariamente el primero de todos. Para solventar esta situación, podemos utilizar un segundo atributo, `start`, que tendrá como valor un número. Este número, que por defecto es 1, corresponde al valor a partir del cual comenzamos a definir nuestra lista. Para el caso de las letras o los números romanos, el navegador se encarga de hacer la traducción del número a la letra correspondiente.

Ejemplo:

```
<p>Ordenamos por letras</p>
<ol type="a">
<li>Elemento a
<li>Elemento b
</ol>
```

```
<p>Ordenamos por números romanos empezando por el 10</p>
<ol type="i" start="10">
<li>Elemento x
<li> Elemento xi
</ol>
```

Resultado:

Ordenamos por letras

- a. Elemento a
- b. Elemento b

Ordenamos por números romanos empezando por el 10

- x. Elemento x
- xi. Elemento xi

◆ Listas de definición

Su objetivo es presentar una lista de definiciones, por un lado el objeto definido y por otro su definición. La etiqueta principal es `<dl>` y `</dl>` (definition list). La etiquetas del elemento y su definición son `<dt>` (definition term) y `<dd>` (definition definition) respectivamente.

Ejemplo:

```
<p>Definiciones</p>
<dl>
  <dt>Objeto uno
    <dd>Definición uno
  <dt>Objeto dos
    <dd>Definición dos
</dl>
```

Resultado:

Definiciones

Objeto uno

Definición uno

Objeto dos

Definición dos

Este tipo de etiquetas son usadas a menudo con el propósito de crear textos más o menos desplazados hacia la izquierda, este efecto lo produce la etiqueta <dd>.

◆ Anidar listas

Es factible conseguir listas mixtas, mediante su anidamiento y hasta cierto punto y en las que haya más de un tipo de lista.

Ejemplo:

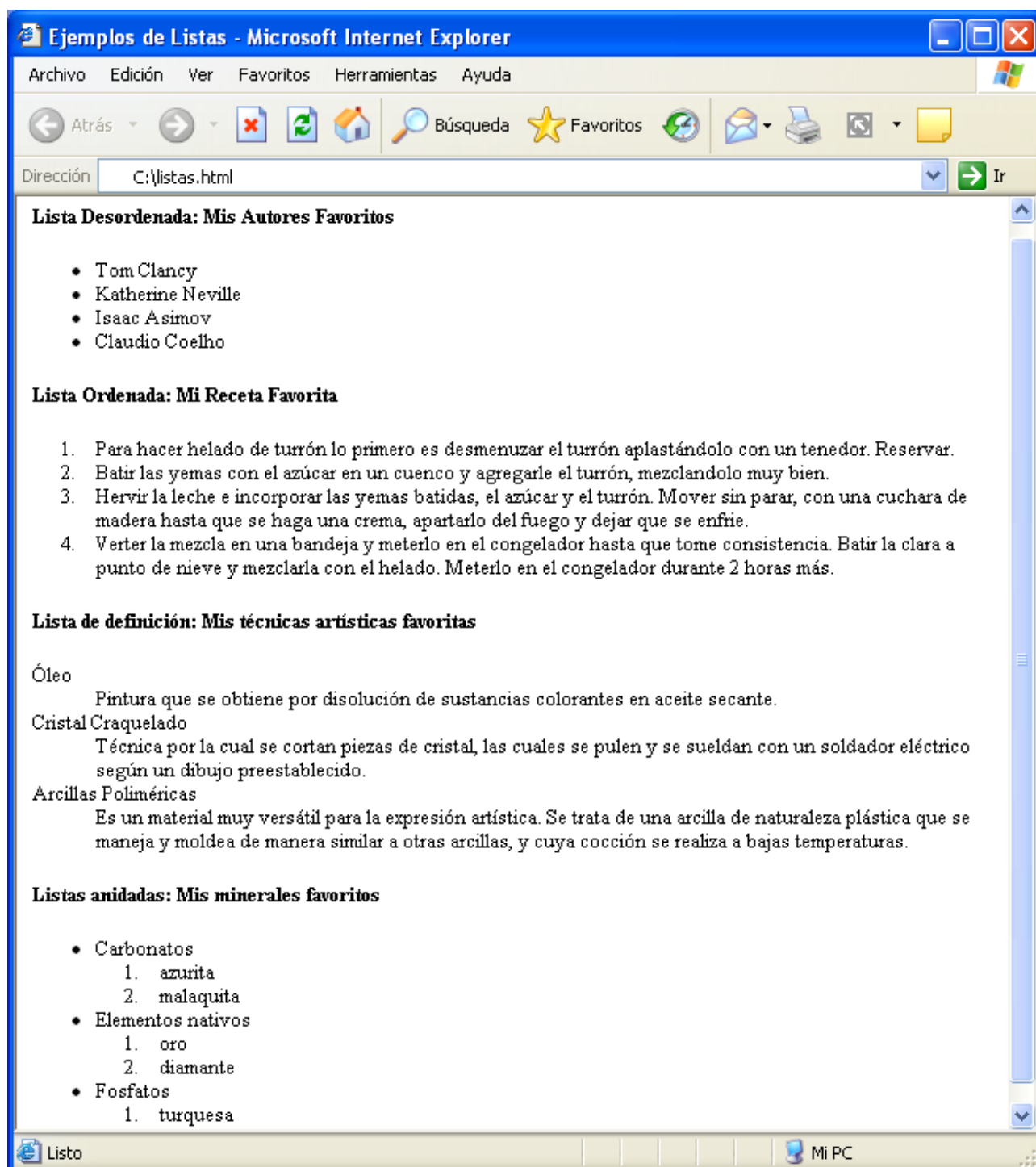
```
<p>Anidamiento de Listas</p>
<ul>
  <li>Lista desordenada uno
    <ol>
      <li>Elemento uno.uno
      <li>Elemento uno.dos
    </ol>
  <li>Lista desordenada dos
    <ol>
      <li>Elemento dos.uno
      <li>Elemento dos.dos
    </ol>
</ul>
```

Resultado:

Anidamiento de Listas

- Lista desordenada uno
 1. Elemento uno.uno
 2. Elemento uno.dos
- Lista desordenada dos
 1. Elemento dos.uno
 2. Elemento dos.dos

Ejemplos:



5.1 Test

1. En el siguiente código HTML queremos enumerar los elementos de una lista de forma ordenada, entonces X equivale a:

```
<X>  
  <li>Objeto</li>  
  <li>Cosa</li>  
  <li>Color</li>  
</X>
```

OL
DD
UL
DT

2. Es correcto el siguiente código:

```
<ul>  
  <li type="square">  
    <ol>Primero  
    <ol>Segundo  
  <li type="circle">Tercero  
</ul>
```

3. Indica cual es verdadera o falsa:

La etiquetas <dt> desplaza textos hacia la izquierda.

Se puede elegir el tipo de viñeta para

Tanto si es letra como número siempre empieza por a ó 1

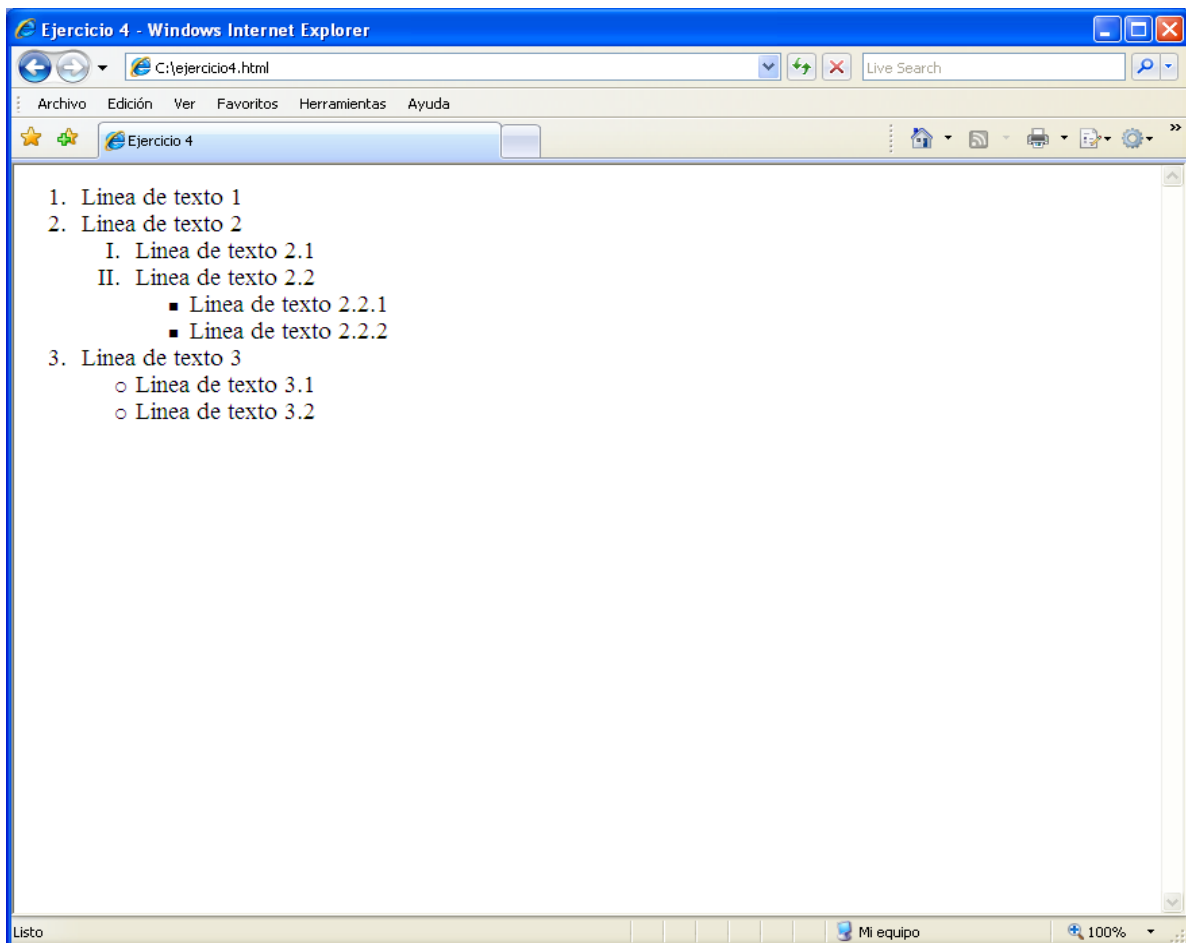
El grado de anidamiento es ilimitado

5.2 Ejercicio

En este ejercicio4.html vamos a crea una lista ordenada de la siguiente forma:

- Detallar una lista ordenada de 3 puntos numerados.
- En el punto 2 crear una lista de 2 puntos cuyo marca sea en números romanos en mayúsculas.
- En el punto 2.II crear una lista desordenada de 2 puntos cuyo tipo de viñeta sea cuadrado.
- Y en el punto 3º una lista desordenada de 2 puntos cuyo tipo de viñeta sea un circulo.

Resultado:



6.- IMÁGENES Y OBJETOS

No sólo encontramos texto en una páginas, también encontramos otros objetos que dinamizan estéticamente el aspecto de nuestra página web. Encontramos todo tipo de objetos, como como applets de Java y vídeos. Aunque el más común de los objetos son las imágenes.

Hay que precisar que algunas imágenes complementan la información textual, mientras que otras son meramente estéticas. Esta diferenciación nos marca cómo codificarlas. En el primer caso se incluyen directamente en el código HTML mediante la etiqueta `` seguido de `src`, para indicar la url donde se localiza el archivo y su nombre, mientras que las imágenes estéticas se deben incluir en el código de las hojas de estilos CSS para mostrarlas.

La sintaxis para inclusión de imágenes es:

`` como dirección relativa.

`` como dirección absoluta.

Veamos los atributos que tiene la etiqueta ``:

- ◆ `alt`: Es el texto que aparece cuando se sitúa el cursor sobre la imagen. También se muestra el texto si no la carga el navegador.
 - ◆ `align`: Indica la posición de la imagen respecto del texto. Toma los siguientes valores:
 - `top`: texto alineado con la parte superior de la imagen
 - `middle`: texto alineado con la parte central de la imagen
 - `bottom`: texto alineado con la parte inferior de la imagen
 - `left`: fuerza el texto arriba a la derecha dejando la imagen a la izquierda
 - `right`: fuerza el texto arriba a la izquierda dejando la imagen a la derecha
 - ◆ `width`: Indica la anchura de la imagen.
 - ◆ `height`: Indica la altura de la imagen.
- Si especificamos las dimensiones de las imágenes, las páginas se cargan más rápido, porque el navegador les reserva un espacio, mientras sigue cargando.
- ◆ `border`: Añade un borde a la imagen, como un marco alrededor.
 - ◆ `hspace`: Especifica el espacio horizontal que separa la imagen del texto.
 - ◆ `vspace`: Especifica el espacio vertical entre la imagen y el texto.

Existen muchos formatos para guardar imágenes que tienen diferentes extensiones. Pero básicamente las más extendidas son los gif y los jpg. Ambos formatos comprimen las imágenes para reducir su tamaño, asegurando una transferencia lo más rápida por la red.

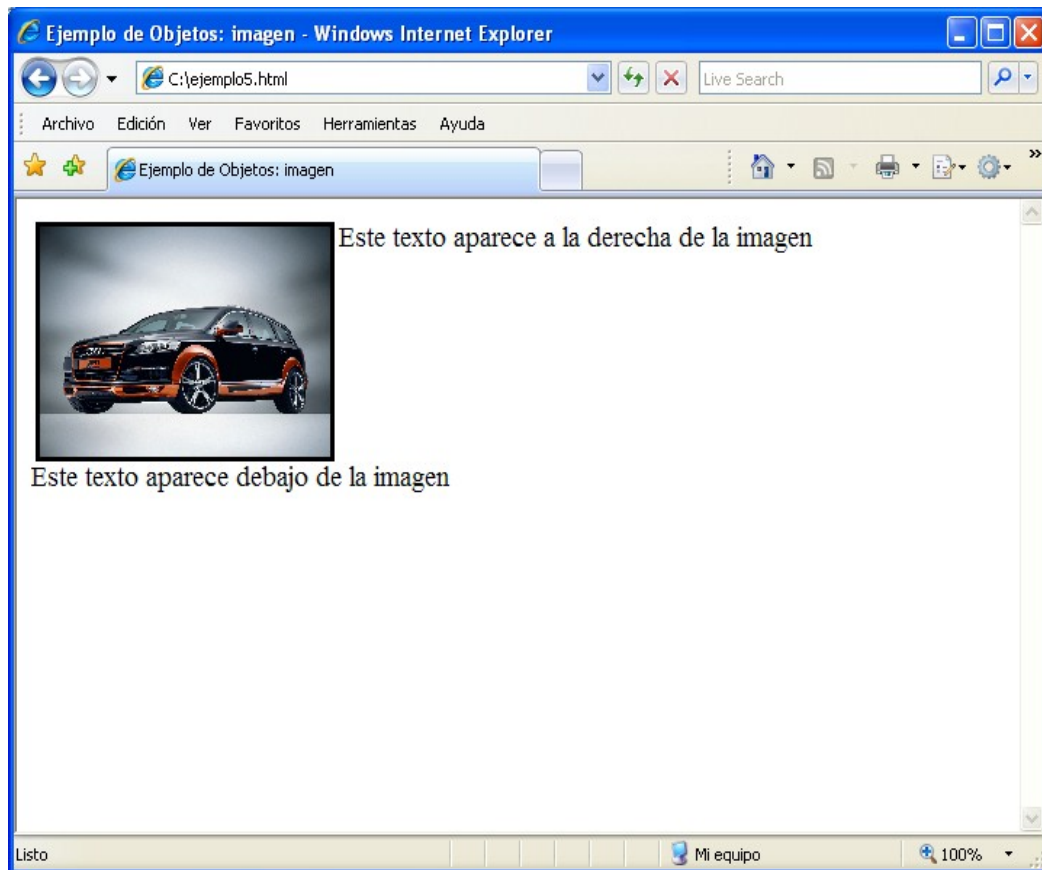
Y ¿porque usar este tipo de imágenes? Sencillamente porque si la imagen tarda en cargar, es posible que el usuario se canse de esperar y abandone la página. En los casos en que es necesario poner una imagen con una resolución elevada para ver los detalles, es mejor colocar una imagen pequeña y usarla como enlace que nos conduzca a la imagen final. De ese modo el visitante ya sabe que la imagen va a tardar en cargar y tiene que ser paciente.

Ejemplo:

En la siguiente imagen se le aplican los estilos para dar formato a la imagen, y de cómo se alinea el texto.

```
Este texto aparece a la derecha de la imagen  
<br clear=all>Este texto aparece debajo de la imagen
```

Resultado:



6.1 Mapas de imagen

Los mapas de imágenes es otra forma de navegación que permite definir diferentes zonas "pinchables" dentro de una imagen. Cada zona está definida por figuras geométricas y apunta a una url diferente, cuyo funcionamiento es exactamente el mismo que otros enlaces. Su uso se ha visto muy reducido debido a que ahora se usa tecnología Flash.

Son utilizados en webs que, por ejemplo, tienen portadas donde damos a conocer cada una de las secciones del sitio por medio de una imagen. También puede ser muy práctico en mapas geográficos donde cada ciudad, provincia o punto cualquiera representa un enlace a una página.

Para crear un mapa de imagen, en primer lugar se inserta la imagen original mediante la etiqueta ``. A continuación, se utiliza la etiqueta `<map>` para definir las zonas o regiones de la imagen. Cada zona se define mediante la etiqueta `<area>`.

Así pues, un mapa de imagen está compuesto de dos partes:

- La imagen propiamente dicha que estará situada como de costumbre dentro de la etiqueta `<body>` de nuestro documento HTML.
- Un código, situado en el interior de la etiqueta `<map>`, que delimitará por medio de líneas geométricas imaginarias cada una de las áreas de los enlaces presentados en la imagen. Estas líneas se definen por coordenadas x,y. Siendo el origen la esquina superior izquierda.

Dentro de ella queremos introducir un enlace a cada uno de los elementos que la componen. Para ello, definiremos nuestros enlaces como zonas circulares de pequeño tamaño que serán distribuidas a lo largo y ancho de la imagen.

Veamos a continuación el código que utilizaremos:

```

<map name="mapa_zonas">
<area shape="rect" coords="20,25,84,113" href="rectangulo.html" />
<area shape="polygon" coords="90,25,162,26,163,96,89,25,90,24"
href="triangulo.html"/>
<area shape="circle" coords="130,114,29" href="circulo.html" />
<area shape="rect" coords="19,156,170,211"
href="mailto:rectangulo@direccion.com" />
<area shape="default" nohref="nohref" />
</map>
```

Cada área se indica con una etiqueta `<area>`, que tiene los siguientes atributos:

- ♦ **alt:** Para indicar un texto explicativo que se mostrará cuando situemos el ratón en el área. Es recomendable indicarlo para hacer la navegación más intuitiva.
- ♦ **shape:** Indica el tipo de área, `rect`, `circle` y `poly`.
- ♦ **coords:** Las coordenadas que definen el área. Son un grupo de valores numéricos distintos dependiendo del tipo de área (`shape`) que definamos.
- ♦ **href:** Para indicar el destino del enlace correspondiente al área.

6.2 Test

1. "src" indica:

- La dirección de un formulario.
- La dirección de una imagen.
- La dirección de un enlace.
- La dirección de un texto.

2. Indica cual es verdadera o falsa:

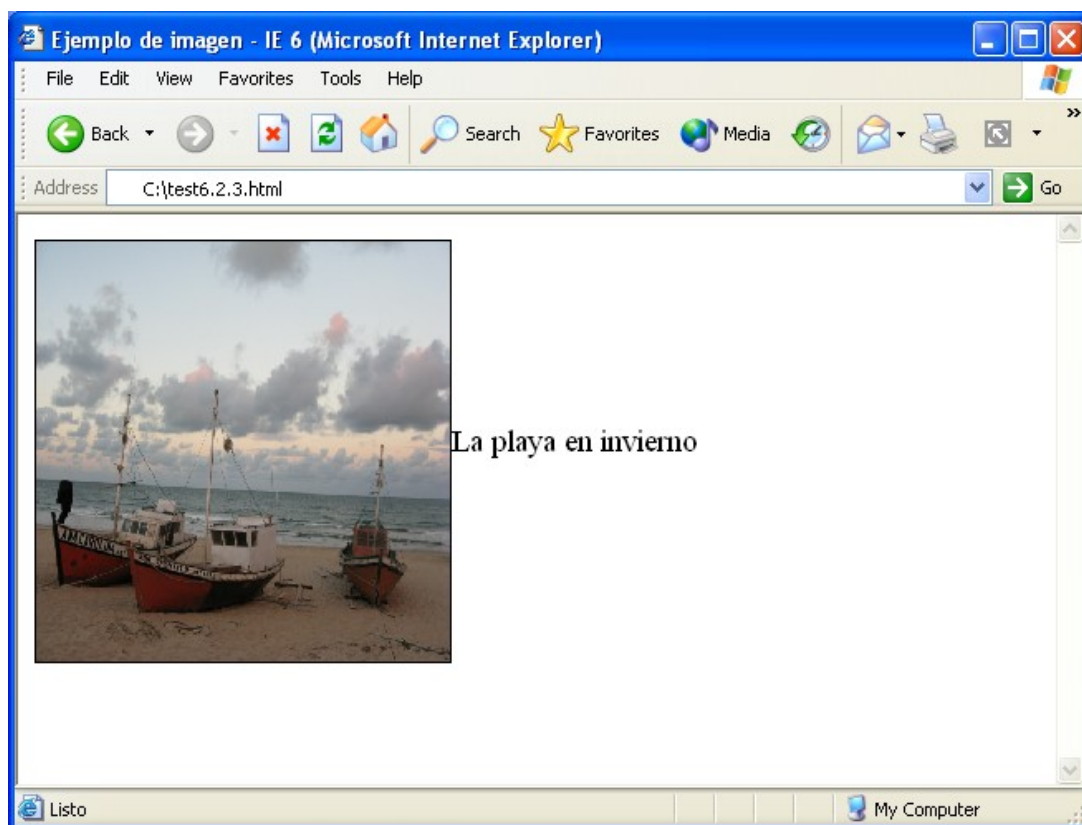
Las páginas se cargan más rápido si el navegador no les reserva un espacio, mientras sigue cargando.

`` es una dirección de tipo relativa

En un mapa de imagen las coordenadas determinan un área para formar un enlace de tipo línea

Un mapa de imagen podría servir para descargar un archivo

3. El código de la siguiente página web es:

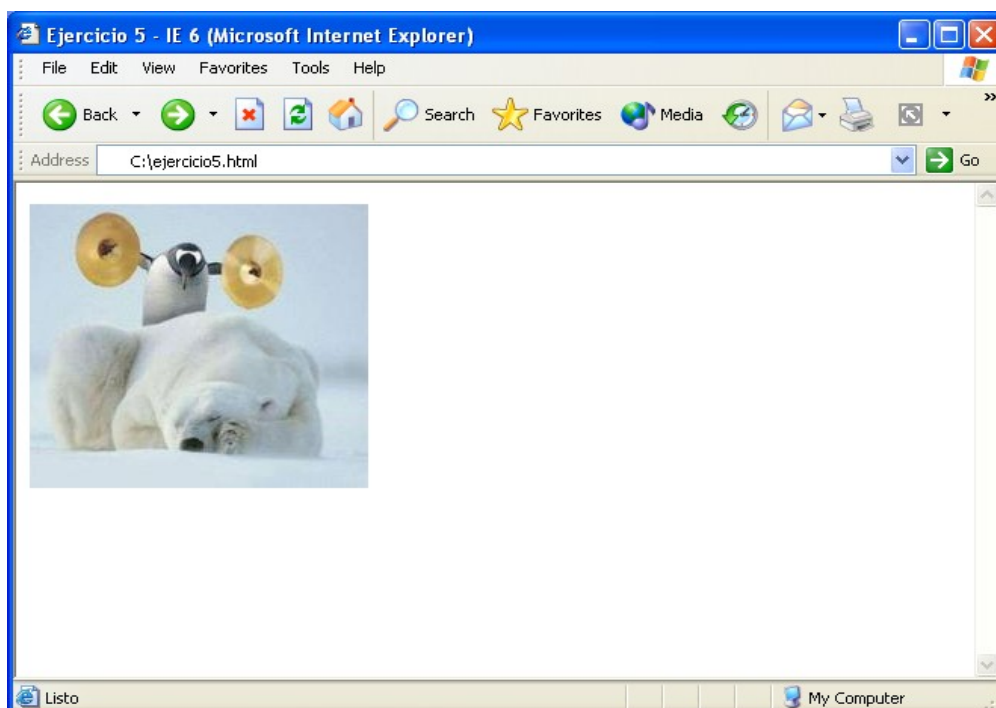
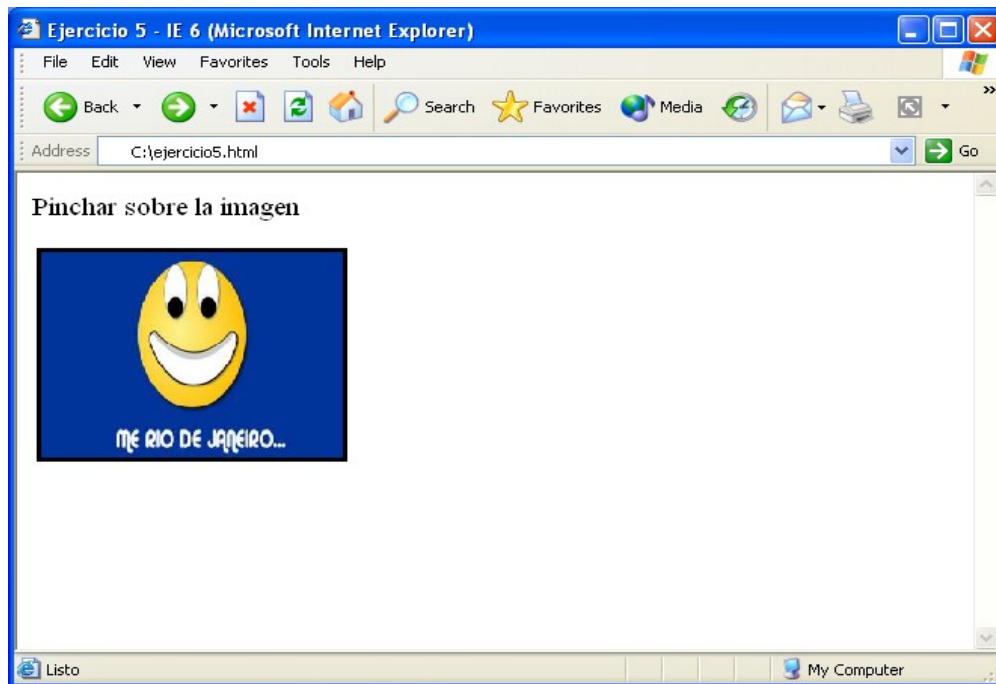


```
<a img src="images/barcos.jpg" align="center" width="250" height="250"
border="1">La playa en invierno</a>
La playa en invierno</img>
La playa en verano</img>
```

6.3 Ejercicio

En este ejercicio5.html vamos a insertar una imagen, con las siguientes dimensiones: ancho de 175 px y alto de 125 px, añadir un borde de 3 px y por último un mapa de imagen de forma rectangular, con sus correspondientes coordenadas. Al pinchar sobre esta imagen nos llevará a otra imagen con las dimensiones que traiga por defecto.

Resultado:



7.- TABLAS

Las tablas constituyen el mejor método para la organización de datos estructurados en celdas y organizados en filas (horizontales) y columnas (verticales). Es muy importante tener claro que organiza datos y no contenidos.

Aunque resulte muy cómodo para maquetar textos e imágenes, ya que presenta un aspecto en que todos los elementos esta muy bien colocados unos respecto a otros, un buen profesional lo hace usando CSS (capas). El código resultante es limpio y sobre todo estándar, para que todos los navegadores encuentren nuestra página web accesible y que supere la validación de la W3C.

Las etiquetas que necesitamos para definir una tabla son:

`<table>` y `</table>`: crea la tabla
`<tr>` y `</tr>`: indican el comienzo y final de una fila
`<th>` y `</th>`: indican si es una fila de cabecera
`<td>` y `</td>`: indican una celda

El orden en que se establece la tabla se va definiendo primero declarando una fila y a continuación las celdas que contiene esa fila, luego otra fila y sus celdas, etc. No es necesario que todas las filas contengan el mismo número de celdas.

◆ Atributos de la Tabla

`border`: Indica el tamaño del borde. Si no se indica nada no aparece.

`width`: Es el ancho de la tabla. Se puede indicar en píxeles o porcentaje.

`cellspacing`: Es el espacio entre las celdas, y por defecto es 2.

`cellpadding`: Es el espacio entre el contenido de las celdas y su borde. Por defecto es 1.

`align`: Para alinear la tabla a la izquierda (left), a la derecha (right) o en el centro (center).

◆ Atributos de fila y celda

`width`: Es el ancho de toda la fila o celda.

`align`: Para alinear el contenido a la izquierda (left), a la derecha (right) o en el centro (center).

`Valign`: Para alinear el contenido en la vertical, arriba (top), en el medio (middle) o abajo (bottom).

`colspan`: Indica el número de celdas que se fusionan en la columna.

`rowspan`: Indica el número de celdas que se fusionan en la fila.

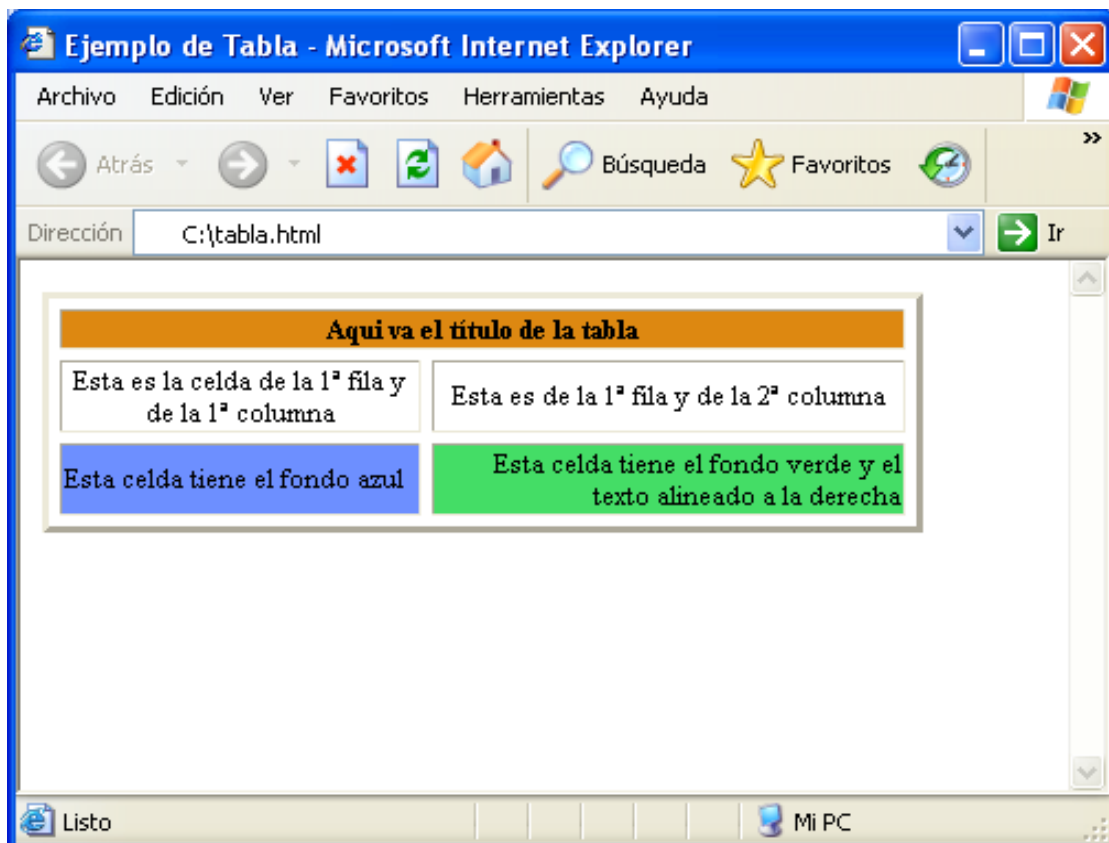
`bgcolor`: Asigna un color de fondo a la fila o celda.

En algunos caso resulta útil el uso de tablas anidadas. De forma que puede incluirse una tabla en una celda. El funcionamiento sigue siendo el mismo, aunque no hay que excederse en el grado de anidación, porque pueden aparecer problemas de visualización y también que la página web tarde más en cargar.

Ejemplo:

```
<table border="3" cellspacing="5" width="400">
  <tr>
    <th colspan="2" bgcolor="#dd8811">Aqui va el
      título de la tabla</th>
  </tr>
  <tr>
    <td align="center">Esta es la celda de la 1ª fila y de la 1ª columna</td>
    <td align="center">Esta es de la 1ª fila y de la 2ª columna</td>
  </tr>
  <tr>
    <td bgcolor="#6D8FFF">Esta celda tiene el fondo azul</td>
    <td align="right" valign="bottom" bgcolor="#44dd66">
      Esta celda tiene el fondo verde y el texto alineado a la derecha
    </td>
  </tr>
</table>
```

Resultado:



7.1 Test

1. En el siguiente código, X es igual a:

```
<table>
<tr> <X>Navegador</X> <X>Uso mundial</X> </tr>
<tr> <td>Firefox</td> <td>44%</td> </tr>
<tr> <td>I. Explorer</td> <td>46%</td> </tr>
<tr> <td>Otros</td> <td>10%</td> </tr>
</table>
```

td
th
tt

2. Las tablas se utilizan para:

Organizar contenidos y datos
Organizar contenidos
Organizar datos
Organizar canciones

3. Indica cual es verdadera o falsa:

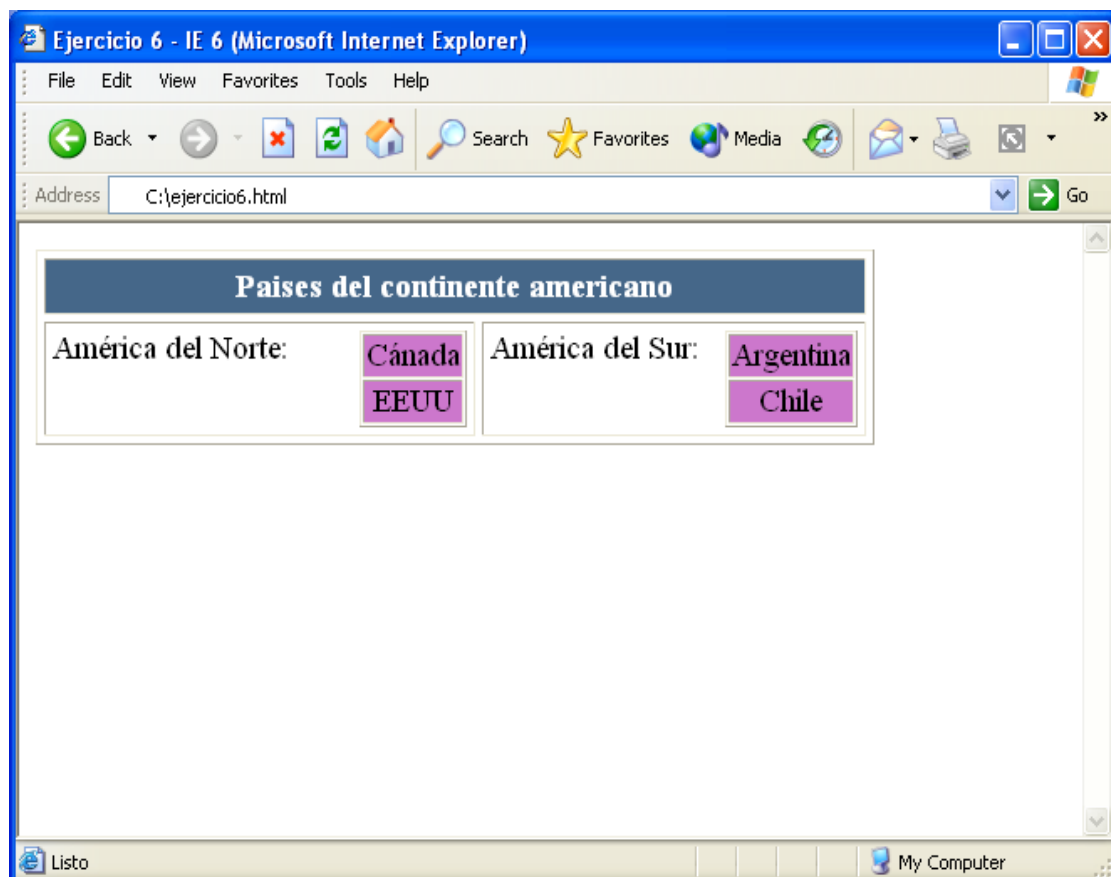
El espacio entre las celdas es cellpadding.
Para alinear el contenido arriba o abajo usamos align
Si el borde de la tabla es 0 px no aparece ningún borde de tabla
El grado de anidamiento es de 9 tablas

7.2 Ejercicio

En este ejercicio6.html vamos a crear una tabla anidada:

- La primera tabla tiene los espaciados cellpadding y cellspacing de 4px. Un borde de 1 px y un ancho total de 500 px. El tamaño es de 2 x 2.
- En la 1ª fila las dos celdas están unidas, color "456789" y texto "Países del continente americano" va centrado.
- En la 2ª fila tenemos 2 celdas y en cada una de ellas el texto "América del Norte" y "América del Sur" respectivamente se alinean a la derecha, con 1 px de cellpadding, cellspacing y de border.
- Insertamos en cada celda de la 2ª fila una segunda tabla con dos filas, color de fondo cc77cc y texto alineado en el centro. Con 2 países para cada celda según gráfico.

Resultado:



8.- FORMULARIOS

Uno de los aspectos que hacen atractiva la web es la interactividad, es decir, la posibilidad de intercambio de información. Cualquier usuario puede realizar numerosas acciones como comprar, enviar comentarios, apuntarse a foros, solicitar el envío de boletines, rellenar encuestas, etc. Para ello HTML nos proporciona una herramienta muy útil: los formularios.

Los formularios son un conjunto de cajas de texto y botones que se usan para realizar búsquedas o bien para introducir datos personales. Un usuario rellena este formulario con cierta información para realizar una determinada acción de su interés. Estos datos introducidos se envían a través del correo electrónico al administrador del formulario. Y a partir de aquí se empieza su procesamiento. El tratamiento de los datos requiere del uso de lenguajes como ASP o PHP que realizan buenos tratamientos de los formularios o bien usar programas prediseñados que están albergados en los servidores.

A continuación se va a explicar en qué consisten los elementos básicos para la realización de un formulario:

- ◆ Abrir y cerrar un formulario

Esto se hace con las etiquetas `<form>` y `</form>`. Entre medias irán todas las etiquetas que generan campos, casillas, botones, etc, con sus correspondientes estilos. La sintaxis básica es:

```
<form action="mailto:nombre@tienesunemail.es"
      METHOD="post" ENCTYPE="text/plain">
...
</form>
```

Veamos esta sintaxis por partes:

- action: indica la url que gestiona los datos del formulario, también vale una dirección de correo.
- method: indica cómo se mandará el formulario, usa valores GET o POST. Si tomamos como elección GET, todas las variables se enviarán por la dirección html, es decir, cuando recarguemos la página web al haber enviado el formulario, las variables aparecen en la barra de direcciones. Esto no es lo más recomendable, pues estamos mostrando datos confidenciales. Por eso usamos el método POST, porque consiste básicamente en enviar las variables de forma "oculta" para que nadie pueda ver los datos que ha enviado al pulsar el botón de envío de formulario.
- enctype: indica en qué forma viaja la información. Para el envío del formulario por correo electrónico usaremos "text/plain". No lo indicaremos en el caso de que queramos que lo procese un programa.

- ◆ Entrada de texto corto

Se crea una caja de texto con la etiqueta `<input>` e `</input>`, con la siguiente estructura:

```
<input type="text" size="30" name="Nombre">
```

Veamos esta sintaxis por partes:

- type: tipo de entrada de los datos
- size: es el tamaño de la caja de texto
- name: es el nombre del contenido

Luego tendremos una caja con un ancho de 30 y lo que escribamos nos llegará asociado a la palabra Nombre:

◆ Entrada de texto largo

Para obtener una caja de texto con varias líneas se utiliza la etiqueta `<textarea>` con su cierre `</textarea>` de la siguiente manera:

```
<textarea name="comentario" rows=3 cols=50>
    Escriba su mensaje
</textarea>
```

Veamos esta sintaxis por partes:

- name: es el nombre del contenido.
- rows: para definir el número de líneas
- cols: para definir el número de columnas

Escriba su mensaje

◆ Lista de opciones

La lista de opciones son unos menús desplegables que nos permiten escoger una o varias de las opciones que se proponen. Para su creación usamos la etiqueta `<select>` y `</select>` con su correspondiente cierre, dentro de la siguientes sintaxis:

```
<select name="flores" size="4">
  <option selected> rosa
  <option> lirio
  <option> buganvilla
  <option> calas
</select>
```

Veamos los atributos de select:

- option: cada opción que queramos que aparezca deberá ser incluida en una línea precedida de la etiqueta `<option>`
- name: asigna un nombre al contenido

- size: indica el número de valores mostrados o visibles de la lista. El resto se pueden ver utilizando la barra lateral de desplazamiento
- multiple: permitir seleccionar varios elementos de la lista a la vez
- selected: atributo de la etiqueta <option>, mediante el cual el elemento esté seleccionado por defecto.
- value: atributo de la etiqueta <option> al cual debemos asignar un valor numérico. Dicho valor será el enviado al programa o correo electrónico si el usuario selecciona esa opción.

◆ Botones de radio o de opción

Nos permite elegir una opción entre varias posibilidades. Al seleccionar una, se deselecta otra, es decir, las opciones son excluyentes. La etiqueta empleada en este caso es <input>, sin cierre, en la cual tendremos el atributo type que ha de tomar el valor radio. La sintaxis es:

```
<input type="radio" name="estacion" value="1">Primavera
<input type="radio" name="estacion" value="2">Verano
<input type="radio" name="estacion" value="3">Otoño
<input type="radio" name="estacion" value="4">Invierno
```

Veamos sus atributos:

- name: a todas las opciones se les asigna un nombre
- value: y para cada opción un valor distinto

El resultado es el siguiente:

```
○Primavera
○Verano
○Otoño
○Invierno
```

Si el usuario elige supuestamente Otoño, recibiremos en nuestro correo una línea como esta: estacion=3. Con lo que ya sabremos que 3 está asociado a otoño.

Cabe señalar que es posible preseleccionar por defecto una de las opciones. Esto puede ser conseguido por medio del atributo **checked**:

```
<input type="radio" name="estacion" value="2" checked>Verano
```

Veamos el efecto:

```
○Primavera
○Verano
○Otoño
○Invierno
```

◆ Casillas de verificación o Checkboxes

Las casillas de verificación nos permiten elegir entre varias posibilidades, bien una o más casillas. Recibiremos como dato el valor de la casilla señalada en name.

Ejemplo:

```
<input type="checkbox" name="color" value="azul">
<input type="checkbox" name="color" value="blanco">
<input type="checkbox" name="color" value="naranja">
```

Resultado:

☐ azul
☒ blanco
☐ naranja

◆ Botones envío y borrado

En los formularios existen dos botones imprescindibles. Son los botones de Enviar y Borrar. Su sintaxis es la siguiente:

```
<input type="submit" value="enviar">
<input type="reset" value="borrar">
```

Al presionar en el botón de enviar lo que ocurre es que una vez validadas las casillas y está todo correcto, el contenido de todo el formulario será enviado a la dirección de correo indicada. El botón de borrar lo que produce es un reseteo de todos los contenidos de los campos.

Ejemplo de un formulario de viajes:

```
<form action="mailto:nombre@tienesunemail.es" method="post"
enctype="text/plain">
<center><table bgcolor="#55cc11" border="0" cellpadding="6"
cellspacing="0" width="500">

<tr>
<td align="right" valign="top"><b>Nombre completo</b></td>
<td><input type="text" size="35" name="Nombre"> </td>
</tr>

<tr>
<td align="right" valign="top"><b>E-mail</b></td>
<td><input type="text" size="25" name="e-mail"> </td>
</tr>

<tr>
<td align="right" valign="top"><b>Provincia</b></td>
<td><input type="text" size="25" name="Provincia"> </td>
</tr>
```

```

<tr>
<td align="right" valign="top"><b>Pais</b></td>
<td><input type="text" size="25" name="Pais"> </td>
</tr>

<tr>
<td align="right" valign="top"><b>Frecuencia de los viajes</b></td>
<td><select name="utilizacion">
    <option value="1">Varias veces al dia
    <option value="2">Una vez al dia
    <option value="3">Varias veces a la semana
    <option value="4">varias veces al mes
</select>
</td>
</tr>

<tr>
<td align="right" valign="top"><b>¿En que estación del año prefiere viajar?</b></td>
<td>
<input type="radio" name="estacion" value="1">Primavera
<input type="radio" name="estacion" value="2">Verano<input
type="radio" name="estacion" value="3">Otoño<input type="radio"
name="estacion" value="4">Invierno
</td>
</tr>

<tr>
<td align="right" valign="top"><b>Comentarios</b></td>
<td><textarea cols="30" rows="7" name="comentarios">Introduzca aqui su
comentario</textarea><td>
</tr>

<tr>
<td align="right" valign="top"><b>Comentarios</b></td>
<td><input type="checkbox" name="recibir_info" checked>Deseo recibir
notificación de las novedades en las líneas de autobuses.</td>
</tr>

<tr>
<td></td>
<td align="center"><input type="submit" name="Enviar datos" value=" Enviar
"><input type="reset" value="Borrar" name="B1">
</td>
</tr>

</table>
</center>
</form>

```

Resultado:

The screenshot shows a Windows Internet Explorer window titled "Suscripción al boletín de viajes". The address bar displays "C:\ejemplo6.html". The menu bar includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". The toolbar contains icons for home, back, forward, stop, and search, along with a "Live Search" input field. The main content area has a green background and contains the following form elements:

- Nombre completo**: A text input field.
- E-mail**: A text input field.
- Provincia**: A text input field.
- País**: A text input field.
- Frecuencia de los viajes**: A dropdown menu with "Varias veces al día" selected.
- ¿En que estación del año prefiere viajar?**: Four radio buttons labeled "Primavera", "Verano", "Otoño", and "Invierno".
- Comentarios**: A text area with the placeholder text "Introduzca aqui su comentario".
- Comentarios**: A checkbox labeled "Deseo recibir notificación de las novedades en las líneas de autobuses." which is checked.
- Enviar** and **Borrar**: Two buttons at the bottom of the form.

The status bar at the bottom shows "Listo", "Mi equipo", and "100%".

8.1 Test

1. Cual de los siguientes atributos pertenece a select:

selected
value
size
Todas las anteriores

2. Un checkbox corresponde a:



3. Indica cual es verdadera o falsa:

Botones de radio nos permite elegir una opción entre varias posibilidades.
Botones de borrado envía en mensaje.
Casillas de verificación nos permiten seleccionar sólo una opción entre varias.
Los formularios son un conjunto de cajas de texto y botones que se usan para introducir datos con los que realizar cálculos.

4. Para crear una entrada de texto de 20 caracteres usaremos la etiqueta:

```
<input type="reset" value="Borrar" name="B1">  
<input type="text" size="20" name="Nombre">  
<input type="radio" name="estacion" value="20">  
<input type="checkbox" name="forma" value="redondo">
```

5. Al presionar un botón de tipo submit se envía la información de la forma "get" y "post", éstas se diferencian en que:

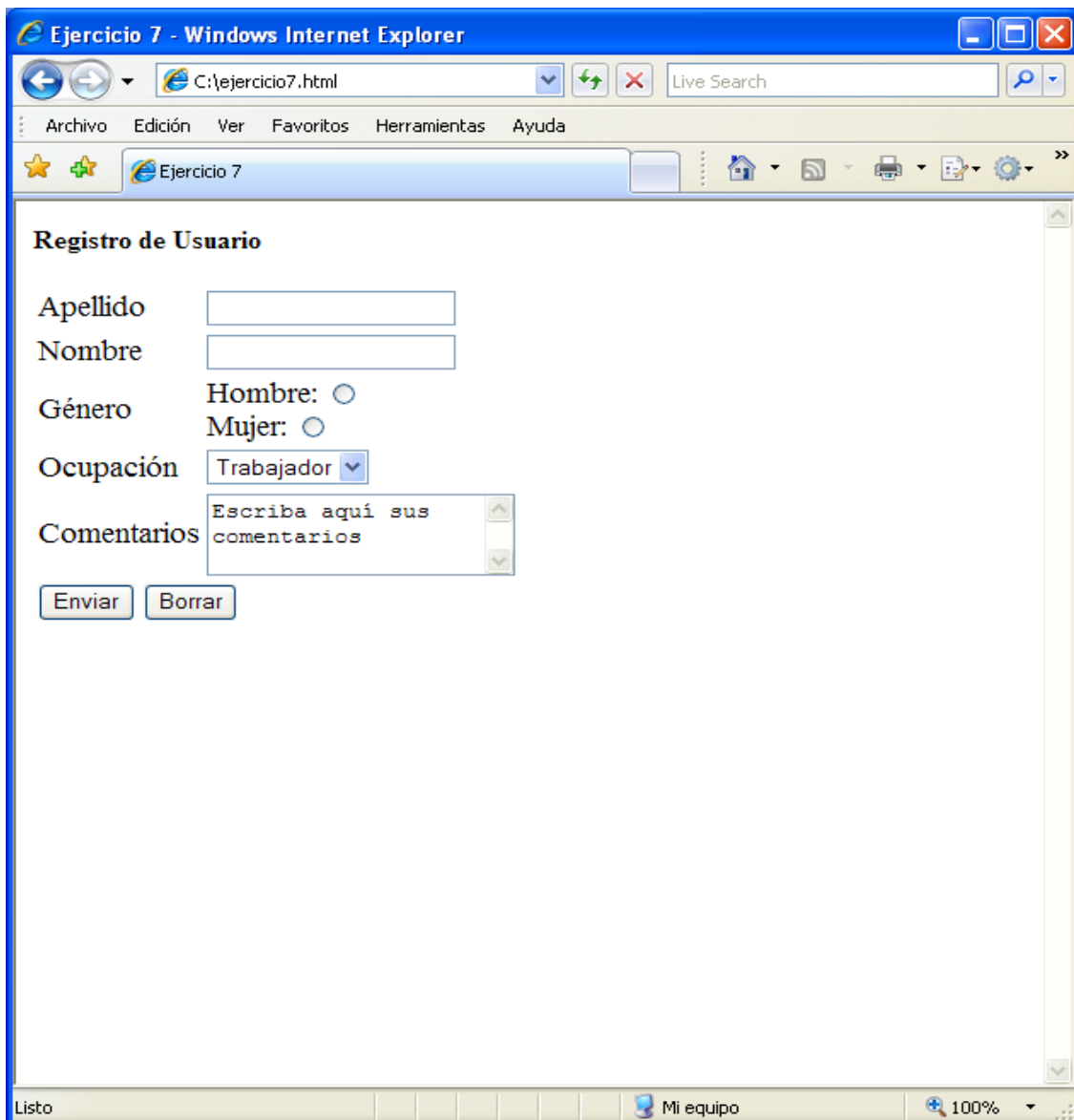
POST se encarga de ocultar y encriptar los valores del envío
GET muestra los valores del envío en la barra del navegador
Las dos anteriores son correctas
Las dos anteriores son incorrectas

8.2 Ejercicio

En el ejercicio7.html vamos a construir un formulario para usuarios:

- El título "Registro de Usuario" como una cabecera h5.
- Insertar el formulario y para que quede organizado crear una tabla de tamaño 6 x 2 y sin borde.
- En la 1ª fila Apellido y en la 2ª celda una caja de texto de tamaño 20.
- En la 2ª fila Nombre y en la 2ª celda una caja de texto de tamaño 20.
- En la 3ª fila Género y en la 2ª celda dos radiobuttons para Hombre y Mujer.
- En la 4ª fila Ocupación y en la 2ª celda un menú desplegable para Trabajador, Estudiante, Autónomo, Jubilado y Otro.
- En la 5ª fila los botones de Enviar y Borrar.

Resultado:



The screenshot shows a Windows Internet Explorer window titled "Ejercicio 7 - Windows Internet Explorer". The address bar shows "C:\ejercicio7.html". The page content is a form titled "Registro de Usuario". The form consists of a table with 2 columns and 5 rows. The first row contains "Apellido" and a text input field. The second row contains "Nombre" and a text input field. The third row contains "Género" and two radio buttons labeled "Hombre" and "Mujer". The fourth row contains "Ocupación" and a dropdown menu with "Trabajador" selected. The fifth row contains "Comentarios" and a text area with the placeholder text "Escriba aquí sus comentarios". Below the table are two buttons: "Enviar" and "Borrar".

Apellido	<input type="text"/>
Nombre	<input type="text"/>
Género	Hombre: <input type="radio"/> Mujer: <input type="radio"/>
Ocupación	Trabajador ▼
Comentarios	<div>Escriba aquí sus comentarios</div>

9.- FRAMES

Los frames o marcos nos sirven para dividir la ventana del navegador en diferentes áreas, cada una de las cuales es independiente. De forma que cada marco corresponde a un archivo html distinto con sus propias barras de desplazamiento, propiedades...

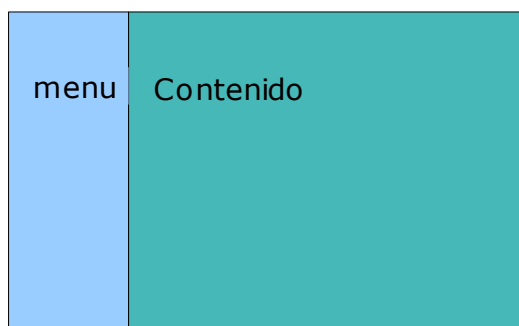
Se crean frames con la etiqueta `<frameset>`. Y el habitual `<body>` es sustituido por `frameset`. Dentro de esta etiqueta primero definimos cada uno de los marcos poniéndoles un nombre y especificando qué fichero html le corresponde mediante la etiqueta `<frame>`, y por último indicamos lo que debe de aparecer al usuario en el caso de que su navegador no soporte frames utilizando la etiqueta `<noframes>`.

Mediante la etiqueta `<frame>` y el atributo `src` definimos la procedencia de cada una de las filas o columnas. El valor del atributo `src` es la ruta y el nombre del archivo html que va a mostrar el frame. Habrá tantas etiquetas `<frame>` como particiones.

Ejemplo:

```
<html>
<head>
  <title>mi primera página con marcos</title>
</head>
<frameset cols="30%,70%">
  <frame name="menu" src="index.html">
  <frame name="contenido" src="content.html">
  <noframes>
    <p>Si ves esto quiere decir que tu navegador no tiene la capacidad de
      visualizar marcos.</p>
  </noframes>
</frameset>
</html>
```

Resultado:



Cada área en la que se divide la ventana viene definida por los parámetros `cols` (columnas) o `rows` (filas). En el atributo `cols` o `rows` colocamos entre comillas el número de particiones que deseamos realizar, indicando de paso el tamaño que va a asignarse a cada una. Los formatos de tamaño son los siguientes:

- Porcentajes: porcentaje referido al espacio total disponible.
- Absolutos: mediante un número especificando el tamaño en píxeles.
- Sobre el espacio sobrante: si colocamos un asterisco (*) indicamos que se va a utilizar todo el espacio sobrante. Si ponemos el asterisco en más marcos lo que hace es repartir por igual el espacio entre todos. Si queremos que algún marco sea mayor antepone un número al asterisco.

Ejemplo:

```
<frameset cols="10%,*,200,2*">
```

En este ejemplo vamos a tener 4 marcos para un ancho de 800 píxeles: el 1^{er} marco son 80 px, el 3^o 200 px y el 4^o tendrá el doble que el 2^o, entonces se asigna 140 px al 2^o y 280 px al 4^o.

Los parámetros de esta etiqueta son:

name: Asigna un nombre a un marco para que después podamos referirnos a él.

src: Indica la dirección del documento HTML que ocupará el marco.

scrolling: Decide si se colocan o no barras de desplazamiento al marco para que podamos movernos por su contenido. Su valor es por defecto auto, que deja al navegador la decisión. Las otras opciones que tenemos son yes y no.

noresize: Si lo especificamos el usuario no podrá cambiar de tamaño el marco.

frameborder: Al igual que su homónimo en la etiqueta <frameset>, si lo igualamos a cero se eliminará el borde con todos los marcos contiguos que tengan también este valor a cero.

marginwidth: Permite cambiar los márgenes horizontales dentro de un marco. Se representa en píxeles.

marginheight: Igual al anterior pero con márgenes verticales.

frameborder: Indica el grosor del borde. Si es 0 el navegador elimina el borde.

framespacing: Indica el hueco entre marcos.

¿Qué ocurre cuando pinchamos en un enlace dentro de un marco? Pues que el nuevo contenido se abre sólo en ese marco. Para que nos muestre el contenido donde nosotros queramos utilizaremos el parámetro target. Que va con las etiquetas <a> y <area> indicando el marco en el que abriremos ese enlace y con la etiqueta <base> con la que modificaremos el marco en el que se nos muestran los enlaces por defecto.

El valor que le asignaremos a target para que nos muestre la información que queremos es el nombre del frame:

Ejemplo:

```
<a href="contacto.html" target="contenido">
```

También existen cuatro nombres reservados que podremos utilizar:

_top: muestra la nueva página en la ventana completa del navegador.

_blank: muestra la nueva página en una ventana nueva y sin nombre del navegador.

_self: muestra la nueva página en el mismo frame.

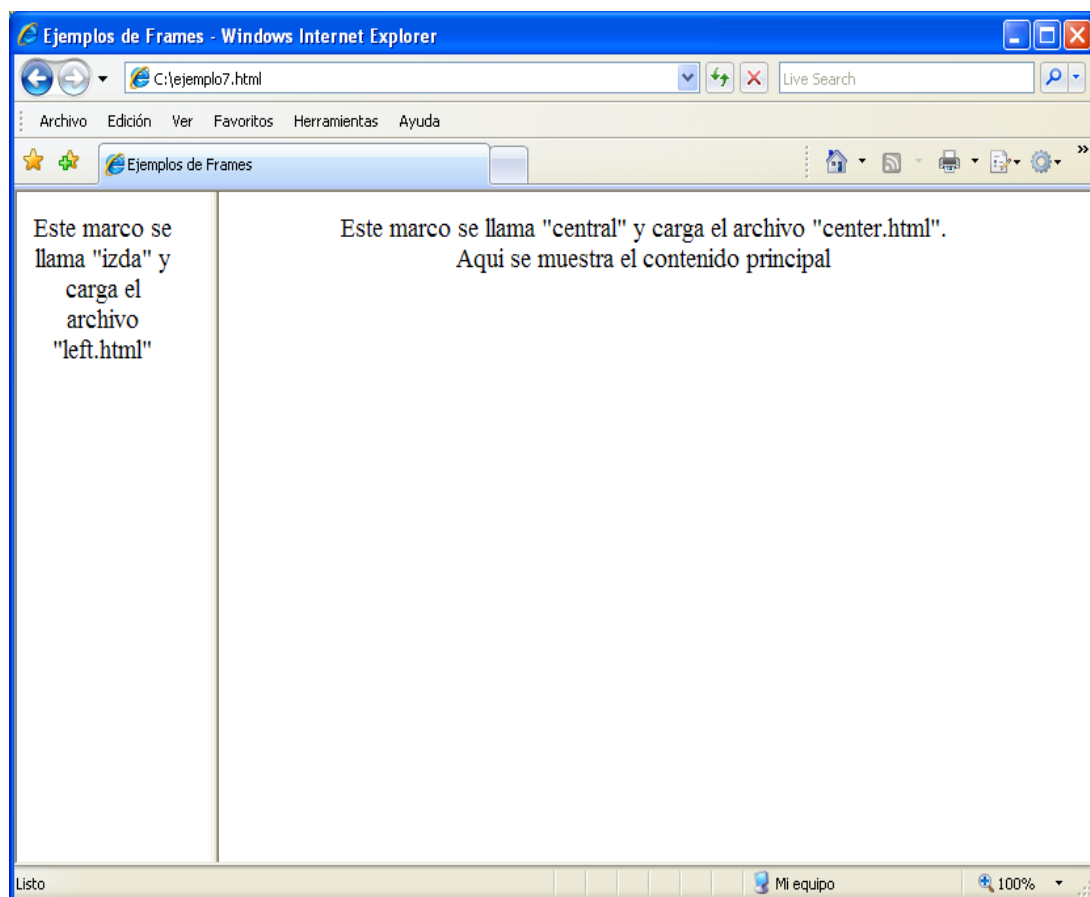
_parent: muestra la nueva página en el frame "padre" del que viene el actual.

nombre: muestra la nueva página en el frame llamado nombre.

Ejemplo:

```
<html>
<head>
<title>
    Ejemplos de Frames
</title>
</head>
    <frameset cols="150,*">
        <frame name="izda" src="left.html">
        <frame name="central" src="center.html">
    </frameset>
</frameset>
<html>
```

Resultado:



9.1 Test

1. La sintaxis para indicar que el contenido de un frame se muestre en una nueva ventana:

```
<a href="contacto.html" target="_blank">  
<a href="contacto.html" target="contenido">  
<a href="contacto.html" target="_parent">  
<a href="contacto.html" target="_top">
```

2. Identifica la notación correcta para un ancho de 800px:

```
<frameset cols="100,*,2*,20%">  
<frameset cols="500,10%,200,*">  
Ninguna de las anteriores  
Cualquiera de las anteriores
```

3. Indica cual es verdadera o falsa:

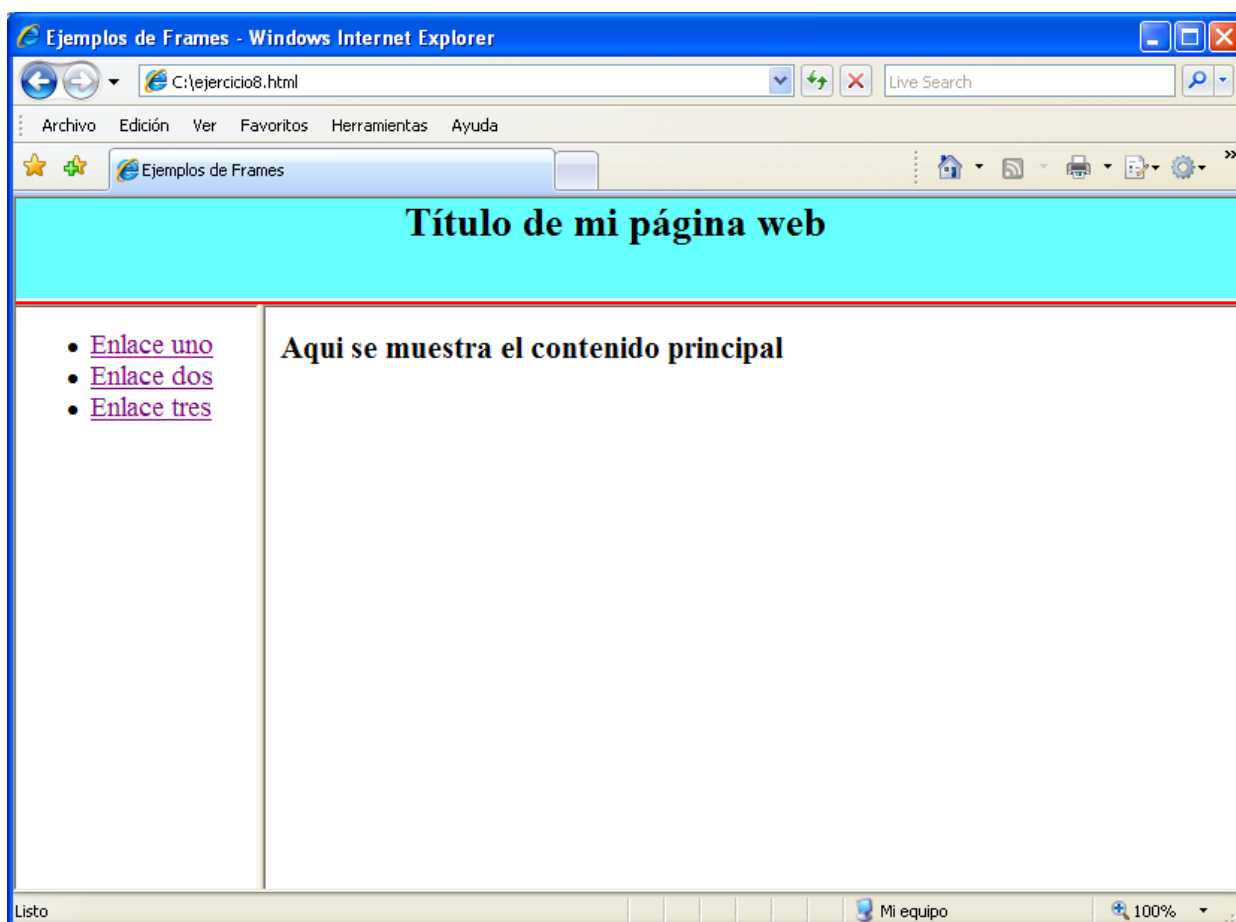
Los frames se sitúan en el body y poseen la capacidad de anidarse
Para mostrar el contenido en la ventana completa del navegador usamos _top
Para modificar el tamaño del marco usamos la etiqueta noresize
Cada frame o marco es independiente del resto de frames

9.2 Ejercicio

En este ejercicio8.html vamos a crear un frame típico, cuyo espacio se divide en tres marcos. Veamos sus características:

- En el ejercicio8.html se divide el espacio en 2 filas, la 1ª con el 15% del espacio y que carga head.html y la 2ª fila con el resto. En esta 2ª fila creamos 2 columnas, la 1ª con el 20% del espacio y que carga menu.html y la 2ª con el resto que carga cuerpo.html.
- En head.html ponemos un fondo de color #66ffff, colocar el "título de mi página web" con un encabezado h2 y centrado.
- En menu.html creamos una lista desordenada con los epígrafes enlace uno, enlace dos y enlace tres que actuarán como hipervínculo y mostrarán su contenido en el frame cuerpo.
- En cuerpo.html se va a mostrar el texto "Aquí se muestra el contenido principal". En este espacio se puede apreciar la funcionalidad del menú, donde se mostrará el contenido de cada epígrafe.
- Crear enlace1.html, con un contenido a elegir desde un texto, imágenes, etc. Para comprobar que al pinchar sobre el enlace uno, éste funciona adecuadamente, y muestre su contenido en el frame cuerpo.html.

Resultado:



10.- MIGRACIÓN A XHTML

Hasta aquí ya sabemos HTML 4.0 para crear nuestras páginas web, y entonces ¿para que migrar a XHTML 1.0? debido a la rapidez con que el mundo web está cambiando, se demanda otra manera de acceder a nuestros contenidos en la red.

El objetivo principal de migrar a XHTML es la creación de una familia de tipos de documentos que pretende sistematizar el lenguaje para dar soporte a cualquier dispositivo, y que a partir de su modularización, y dependiendo de las características y necesidades de cada plataforma (ordenadores, televisores, consolas de juego, lectores de voz, dispositivos de mano, teléfonos, etc.), se dé soporte para más o menos módulos.

Por estos motivos los nuevos sitios web han de crearse ajustándose a los estándares CSS, han de ser optimizados para buscadores y otras herramientas y han de cargar rápidamente en el navegador para que el acceso resulte rápido y eficaz.

Veamos algunas generalidades para la migración:

- ➔ Los documentos deben ser "gramaticalmente correctos"
A partir de ahora todos los elementos deben tener etiquetas de cierre. Existen casos como `` y `
` que no tienen. Esto se resuelve añadiendo la barra / y un espacio antes del cierre, es decir, `` y `
`.
- ➔ Los elementos deben estar bien anidados.
Se refiere a que elementos block (párrafos, headings, listas, etc) no se pueden meter dentro de elemento inline (etiquetas de formato, enlaces, etc)
- ➔ Minúsculas y comillas.
Los nombres de elementos y atributos deben escribirse en minúscula, ahora no significa lo mismo `` que ``. Todos los atributos tienen que estar entre comillas dobles.
- ➔ Etiquetas y atributos que desaparecen.
Debido a que XHTML deja de lado la presentación del documento, porque ya es controlado por CSS, existen etiquetas que ya no tiene sentido utilizar. Es el caso de ``, atributos referentes a color, imágenes de fondo, alineación, negrita, cursiva o lo que tenga que ver con estilo serán sustituidos por reglas CSS.
- ➔ Accesibilidad de un documento
Hay que facilitar la comprensibilidad de un sitio web. Para ello podemos usar el atributo alt de la etiqueta ``, que muestra un texto cuando la imagen no se puede cargar o al pasar el cursor por encima. Existe otro atributo muy útil en los hipervínculos, que usado en la etiqueta `<a>`, nos informa de a dónde nos dirige ese enlace y se llama title.
- ➔ Frames y Tablas
En cuanto a los frames no se recomienda en absoluto su uso ya que supone ir en contra de la accesibilidad de un documento. Y por consiguiente, el atributo target de la etiqueta `<a>` tampoco.
En cuanto a las tablas, es muy importante resaltar que no se usan para maquetar. Por supuesto que una tabla pasaría el análisis del validador del W3C, pero se ha diseñado para representar información tabulada, no para diseñar maquetas o layouts.

➔ Caja contenedora

Y entonces, ¿cómo se maqueta una página web?. Pues con CSS. Se dispone una caja contenedora en la que iremos poniendo todos los demás elementos. Es una buena manera de que empecemos a concebir el diseño de forma más parecida al diseño gráfico, de forma que en esta especie de contenedor, al que le damos unas dimensiones determinadas, se colocan todos los elementos sin que sobresalga nada y para que el conjunto funcione correctamente. Usaremos una estructura como la siguiente:

```
<div id="contenedor"> ... </div>
```

➔ Caracteres especiales

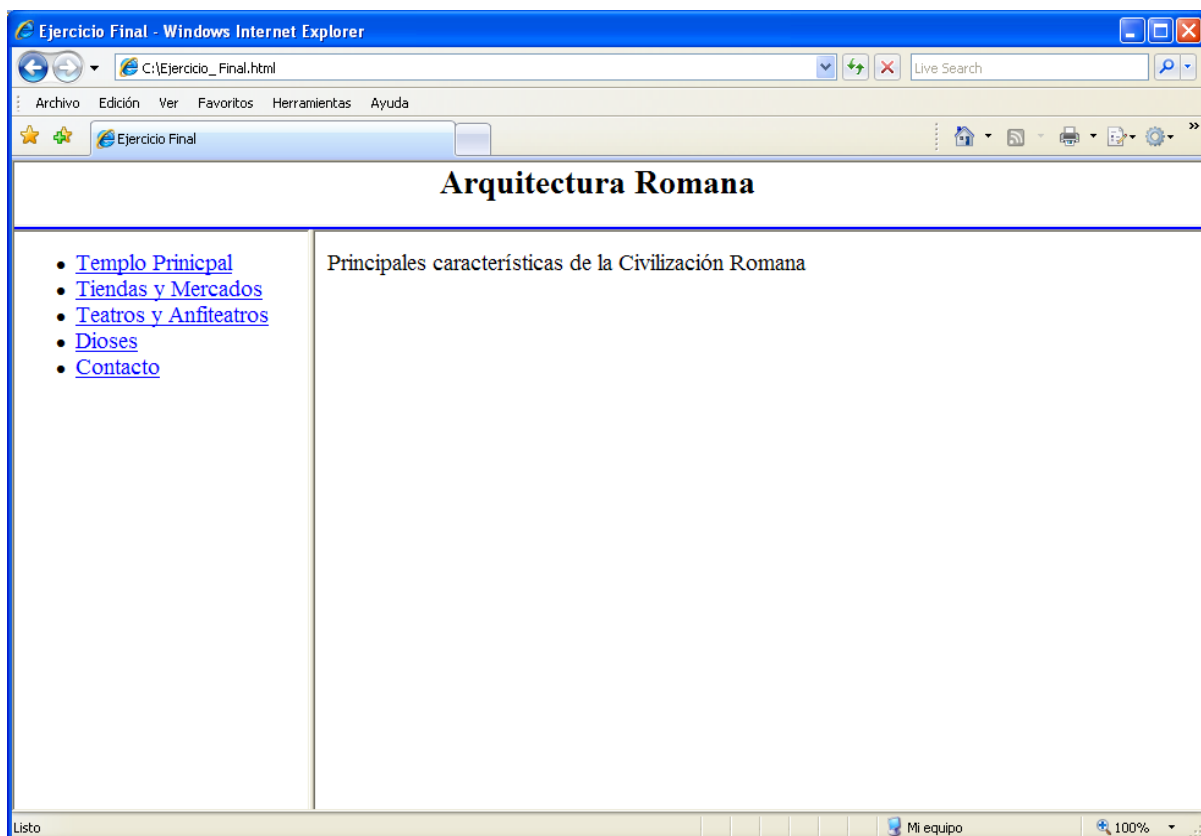
El juego de caracteres o "*charset*" que definimos al principio del documento le dirá al navegador cómo debe interpretar los caracteres que se encuentren en el texto. Para nuestro idioma es el "*Occidental ISO-8859-1*", que contiene todos los caracteres de los idiomas occidentales como la *eñe*, las vocales con tilde o los circunflejos del francés, y el "*UTF-8*", que contiene miles de caracteres de los alfabetos de muchos de los idiomas de todo el mundo.

El famoso *&* da muchos problemas al validador del W3C, así que es necesario insertar ciertos caracteres de forma segura. En el caso de *&* sustituirlo por *&*, los caracteres con acentos, como *á*, ahora es *á*.

11.- EJERCICIO PRÁCTICO FINAL

En este Ejercicio_Final.html vamos a crear un sitio web básico:

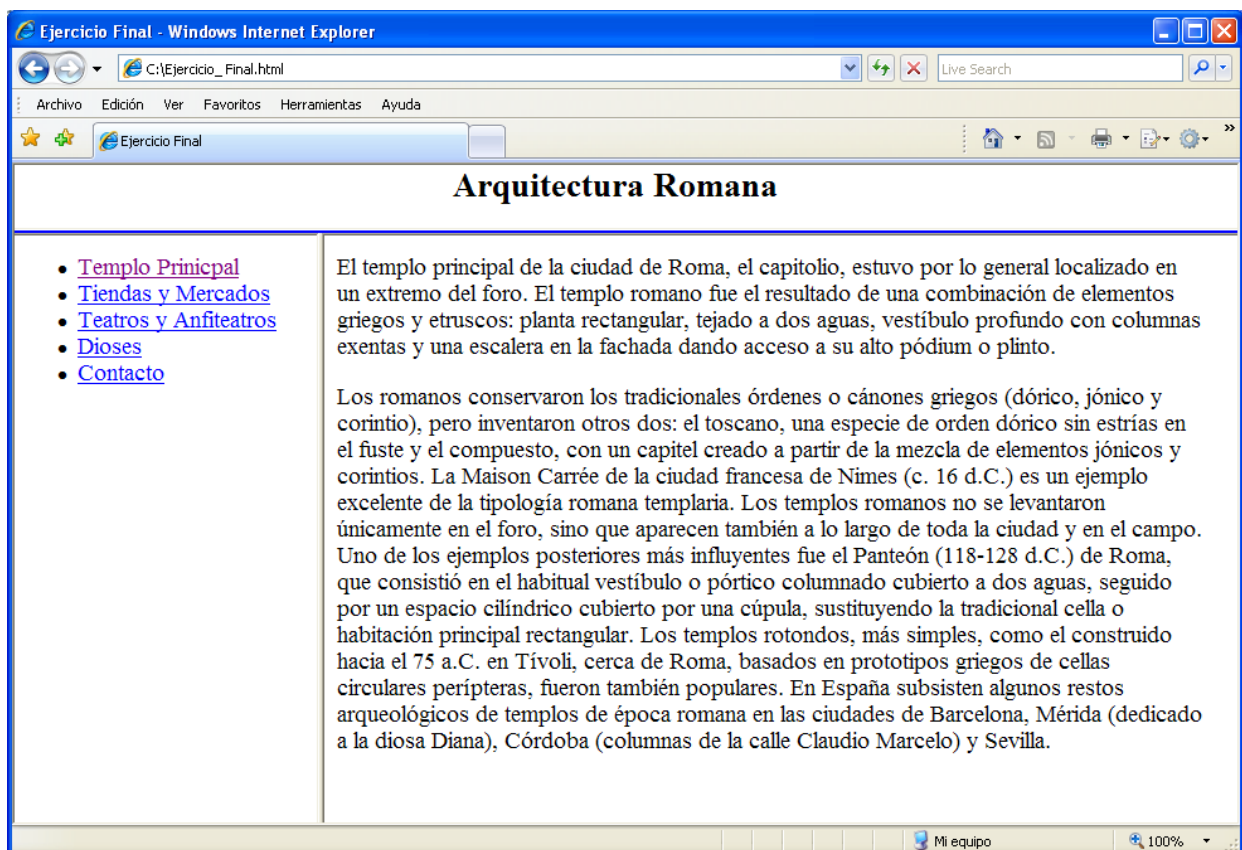
- En Ejercicio_Final.html vamos a crear dos frames. La fila superior tiene un tamaño del 10% del total del espacio y se llama cabecera.html. Y la fila inferior abre otro frame con una columna a la izquierda llamado barra.html y dimensión del 25% del ancho total y la columna derecha llamada contenido.html.
- En cabecera.html insertamos el texto alineado “Arquitectura Romana” en h2.
- En barra.html disponemos en un listado los epígrafes siguientes:
 - Templo Principal
 - Tiendas y Mercados
 - Teatros y Anfiteatros
 - Dioses
 - Contacto
- En contenido.html simplemente poner una introducción con el texto “Principales Características de la Civilización Romana”



- Bajo el epígrafe Templo Principal, hay un hipervínculo a templo.html en el que se muestra el siguiente contenido, dividido en dos párrafos:

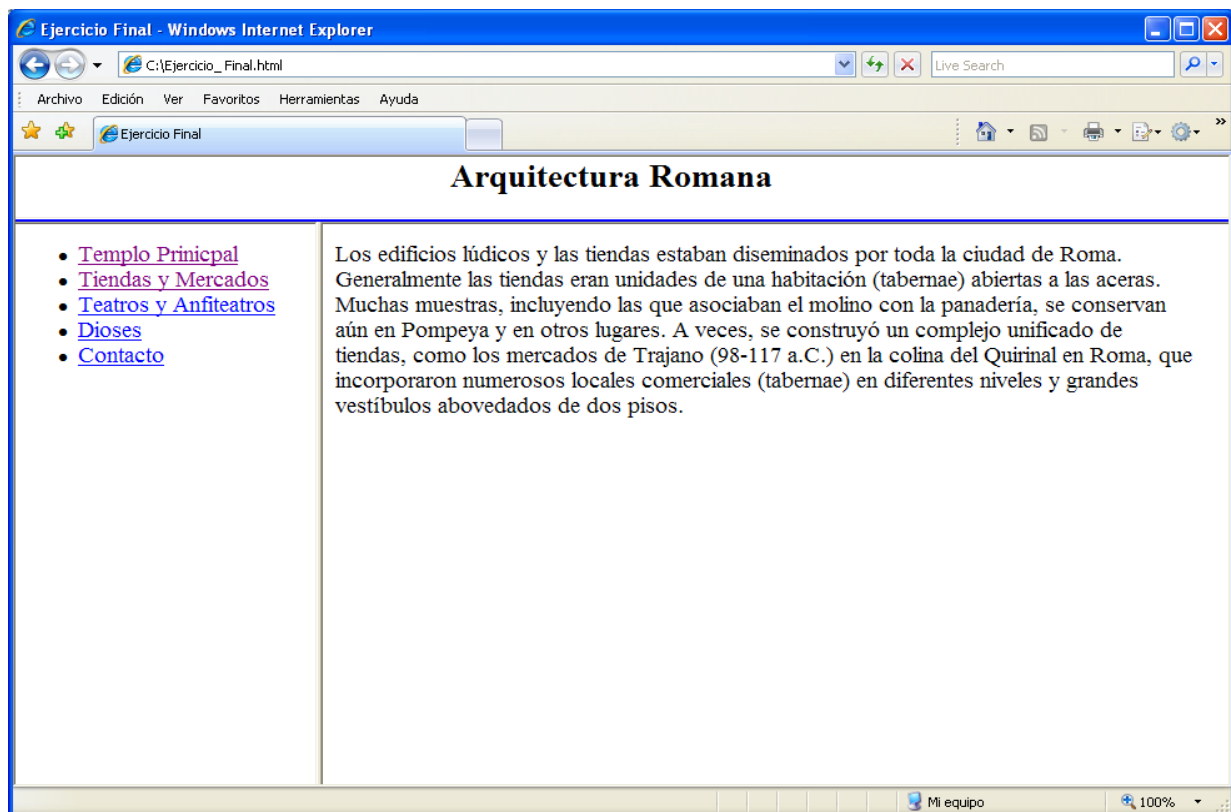
“ El templo principal de la ciudad de Roma, el capitolio, estuvo por lo general localizado en un extremo del foro. El templo romano fue el resultado de una combinación de elementos griegos y etruscos: planta rectangular, tejado a dos aguas, vestíbulo profundo con columnas exentas y una escalera en la fachada dando acceso a su alto pódium o plinto.

Los romanos conservaron los tradicionales órdenes o cánones griegos (dórico, jónico y corintio), pero inventaron otros dos: el toscano, una especie de orden dórico sin estrías en el fuste y el compuesto, con un capitel creado a partir de la mezcla de elementos jónicos y corintios. La Maison Carrée de la ciudad francesa de Nimes (c. 16 d.C.) es un ejemplo excelente de la tipología romana templaria. Los templos romanos no se levantaron únicamente en el foro, sino que aparecen también a lo largo de toda la ciudad y en el campo. Uno de los ejemplos posteriores más influyentes fue el Panteón (118-128 d.C.) de Roma, que consistió en el habitual vestíbulo o pórtico columnado cubierto a dos aguas, seguido por un espacio cilíndrico cubierto por una cúpula, sustituyendo la tradicional cella o habitación principal rectangular. Los templos rotundos, más simples, como el construido hacia el 75 a.C. en Tívoli, cerca de Roma, basados en prototipos griegos de cellas circulares perípteras, fueron también populares. En España subsisten algunos restos arqueológicos de templos de época romana en las ciudades de Barcelona, Mérida (dedicado a la diosa Diana), Córdoba (columnas de la calle Claudio Marcelo) y Sevilla.”



- Bajo el epígrafe Tiendas y Mercados, hay un hipervínculo a tienda.html en el que se muestra el siguiente contenido, en un único párrafo:

“Los edificios lúdicos y las tiendas estaban diseminados por toda la ciudad de Roma. Generalmente las tiendas eran unidades de una habitación (tabernae) abiertas a las aceras. Muchas muestras, incluyendo las que asociaban el molino con la panadería, se conservan aún en Pompeya y en otros lugares. A veces, se construyó un complejo unificado de tiendas, como los mercados de Trajano (98-117 a.C.) en la colina del Quirinal en Roma, que incorporaron numerosos locales comerciales (tabernae) en diferentes niveles y grandes vestíbulos abovedados de dos pisos.”

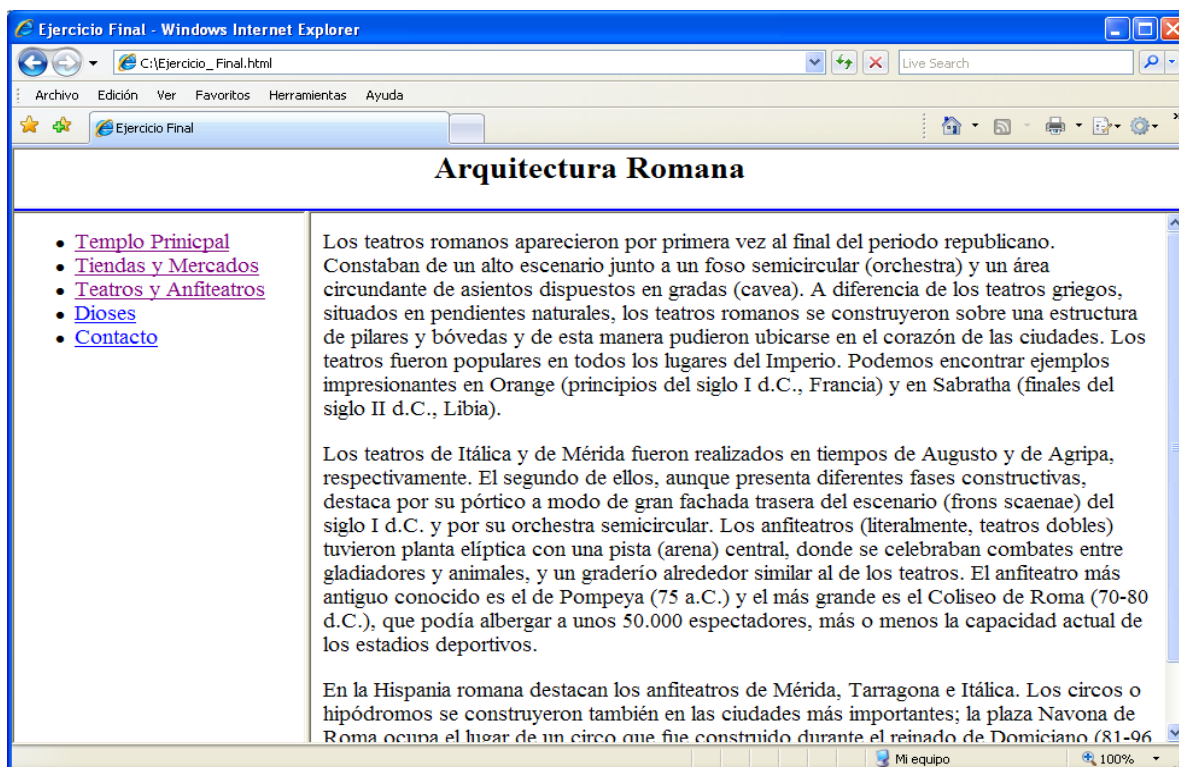


- Bajo el epígrafe Teatros y Anfiteatros, hay un hipervínculo a teatro.html en el que se muestra el siguiente contenido, dividido en tres párrafos:

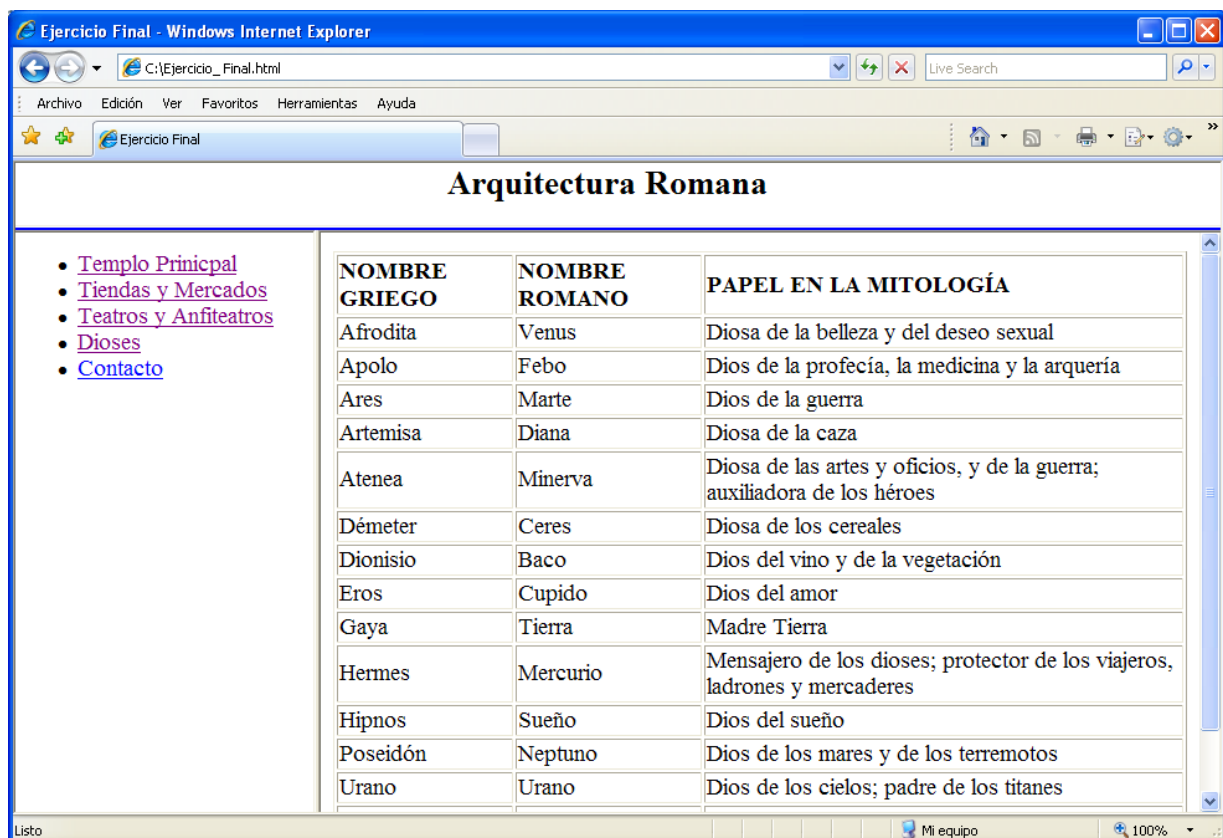
“Los teatros romanos aparecieron por primera vez al final del periodo republicano. Constaban de un alto escenario junto a un foso semicircular (orchestra) y un área circundante de asientos dispuestos en gradas (cavea). A diferencia de los teatros griegos, situados en pendientes naturales, los teatros romanos se construyeron sobre una estructura de pilares y bóvedas y de esta manera pudieron ubicarse en el corazón de las ciudades. Los teatros fueron populares en todos los lugares del Imperio. Podemos encontrar ejemplos impresionantes en Orange (principios del siglo I d.C., Francia) y en Sabratha (finales del siglo II d.C., Libia).

Los teatros de Itálica y de Mérida fueron realizados en tiempos de Augusto y de Agripa, respectivamente. El segundo de ellos, aunque presenta diferentes fases constructivas, destaca por su pórtico a modo de gran fachada trasera del escenario (frons scaenae) del siglo I d.C. y por su orchestra semicircular. Los anfiteatros (literalmente, teatros dobles) tuvieron planta elíptica con una pista (arena) central, donde se celebraban combates entre gladiadores y animales, y un graderío alrededor similar al de los teatros. El anfiteatro más antiguo conocido es el de Pompeya (75 a.C.) y el más grande es el Coliseo de Roma (70-80 d.C.), que podía albergar a unos 50.000 espectadores, más o menos la capacidad actual de los estadios deportivos.

En la Hispania romana destacan los anfiteatros de Mérida, Tarragona e Itálica. Los circos o hipódromos se construyeron también en las ciudades más importantes; la plaza Navona de Roma ocupa el lugar de un circo que fue construido durante el reinado de Domiciano (81-96 d.C.). En las ciudades de Tarragona, Sagunto y Toledo pueden hoy día contemplarse algunos restos de antiguos circos romanos.”



- Bajo el epígrafe Dioses, hay un hipervínculo a tabla.html en el que se muestra el siguiente contenido, estructurado en una tabla de 3 x 15:
La tabla tiene un ancho del 100% del espacio.
Los títulos son NOMBRE GRIEGO, NOMBRE ROMANO y PAPEL EN LA MITOLOGÍA
El contenido para las celdas son
Afrodita / Venus / Diosa de la belleza y del deseo sexual
Apolo / Febo / Dios de la profecía, la medicina y la arquería
Ares / Marte / Dios de la guerra
Artemisa / Diana / Diosa de la caza
Atenea / Minerva / Diosa de las artes y oficios, y de la guerra; auxiliadora de los héroes
Démeter / Ceres / Diosa de los cereales
Dionisio / Baco / Dios del vino y de la vegetación
Eros / Cupido / Dios del amor
Gaya / Tierra / Madre Tierra
Hermes / Mercurio / Mensajero de los dioses; protector de los viajeros, ladrones y mercaderes
Hipnos / Sueño / Dios del sueño
Poseidón / Neptuno / Dios de los mares y de los terremotos
Urano / Urano / Dios de los cielos; padre de los titanes
Zeus / Júpiter / Soberano de los dioses olímpicos



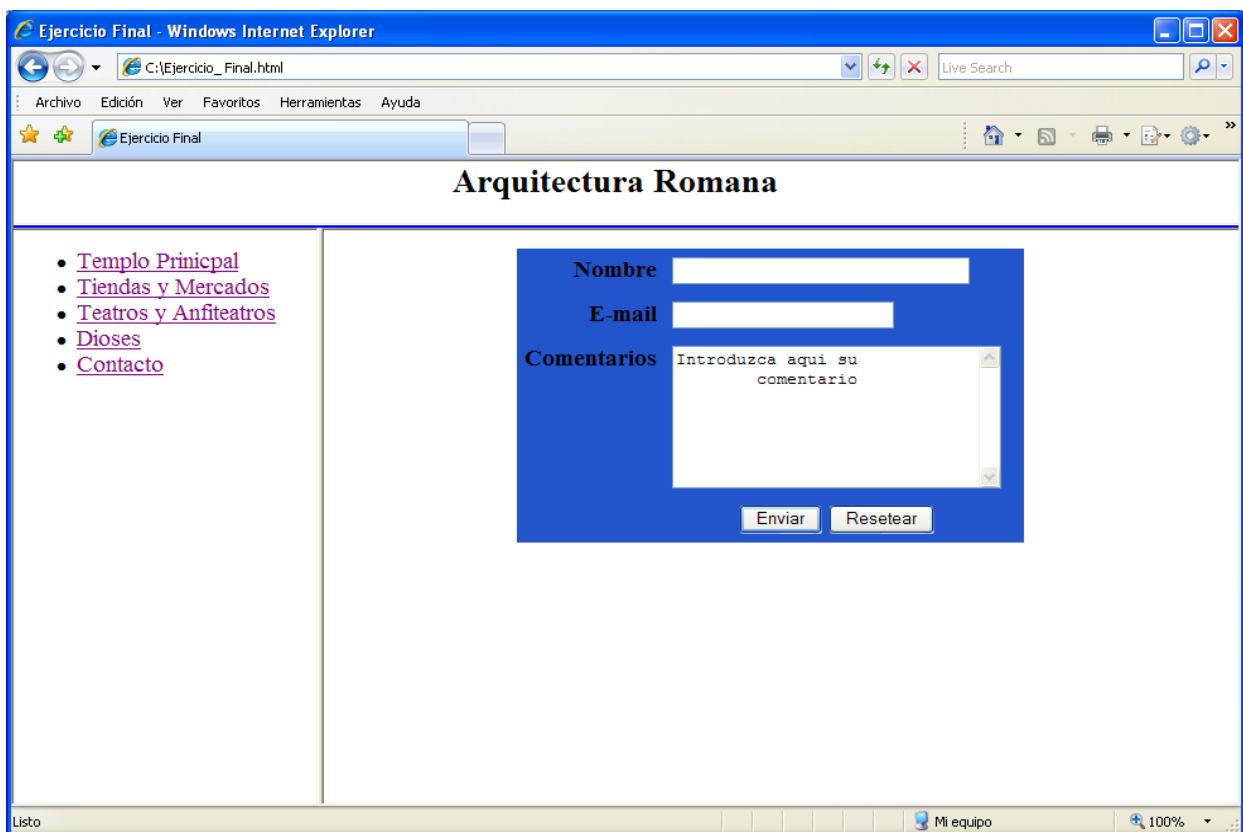
- Bajo el epígrafe Contacto, hay un hipervínculo a contacto.html en el que se muestra un formulario de contacto:

El formulario es de method post y action con tu email de contacto.

En la tabla donde va el formulario poner un color de fondo #2255cc, sin borde con cellpadding 6, el ancho de la tabla es de 200 px.

Los campos son Nombre con su caja de texto, E-mail con su caja de texto y Comentarios con una caja de texto de 30 columnas y 7 filas.

Añadir al final dos botones Enviar que envía datos a la dirección de correo electrónico indicado en el form y Borrar que resetea los datos introducidos.



12.- BIBLIOGRAFÍA

- <http://www.w3c.es/>
- <http://es.html.net/>
- <http://www.virtualnauta.com/>