

# Arquitecturas Distribuidas

TEMA 3. Lenguajes estructurados

# Contenido del Tema 3

- I. HTML
- II. CSS
- III. SGML
- IV. XML
- V. DOM

# I. HTML

1. Introducción
  - 1.1. Motivación
  - 1.2. SGML
  - 1.3. Lenguaje HTML: introducción
  - 1.4. Evolución de HTML
2. Lenguaje de Marcado de Hipertexto (HTML)
  - 2.1. Etiquetas y elementos
  - 2.2. Estructura general
  - 2.3. Disposición del texto
  - 2.4. Listas
  - 2.5. Hiperenlaces
  - 2.6. Imágenes
  - 2.7. Tablas
  - 2.8. Limitaciones del HTML 3.2
  - 2.9. HTML 4.0

# Motivación

- Los S.D. son heterogéneos: diversidad de dispositivos, aplicaciones, sistemas operativos, lenguajes de programación, etc.
- ¿Cómo expresamos los datos de manera que puedan ser interpretados sin ambigüedad y pérdida de datos por sistemas heterogéneos?
  - Distintas soluciones posibles
  - Una de las más utilizadas consiste en utilizar lenguajes de marcas.
  - Las marcas añaden información estructural, semántica y/o de presentación a los datos

# Lenguajes de marcas

- Lenguaje de marcas (markup language)
- Vocabulario que combina datos (contenido) con notación (marcas)
- Proporciona metadatos: “datos sobre los datos”, es decir, las marcas proporcionan un contexto a los datos
- Permiten separar claramente la estructura y el contenido del documento de la presentación del mismo
- Son inteligibles para las personas (human-readable)

# SGML

- **Standard Generalized Markup Language**
- Características que lo hacen apropiado para S.D.:
  - Marcas descriptivas: anota los datos en términos de su estructura o semántica y NO en función de lo que se debe hacer con ellos => Separa los datos de la estructura y por implicación de su representación
  - Proporciona un medio para asociar un documento marcado con un *tipo de documento* (DTD)
- **Document Type Definition (DTD):** define el tipo de documento, las marcas usadas, la relación entre ellas y su significado.
- El DTD es la plantilla sobre la cual el contenido marcado puede ser entendido, interpretado y validado: “Manual del lenguaje de marcas”.
- **SGML proporciona una sintaxis para declarar DTDs**, es decir, lenguajes de marcas
- A partir de SGML se definen diversos lenguajes de marcas.
  - Entre ellos, el más usado: HTML
- SGML es demasiado complejo y ambiguo en ciertos aspectos: XML se desarrolla para solucionar estos problemas

# Lenguaje HTML: introducción

- Es el lenguaje de publicación de la web.
- Desarrollado originalmente por Tim Berners-Lee
- Su primer objetivo era proporcionar un medio sencillo de exhibir información de manera inteligible para un ser humano.
- El software que lo representa se denomina “navegador”
- HTML está definido mediante la sintaxis estándar de SGML: separa la estructura de los datos mediante marcas, llamadas *elementos*.
- La marca o etiqueta incluye el nombre del elemento entre <>
- Existen etiquetas de comienzo y de cierre <X> </X>
- **Un elemento está compuesto por una etiqueta de comienzo y de cierre y lo que contienen.**
- **Un elemento puede contener datos textuales y atributos.**
- Los elementos se anidan. Un elemento puede contener otros elementos. Estructura en forma de árbol.
- HTML no aportó (ya desde un primer momento) toda la generalidad necesaria, y ha sido (y es) objeto de revisiones.

# Lenguaje HTML

- Ejemplo página web:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> Ejemplo </TITLE>
```

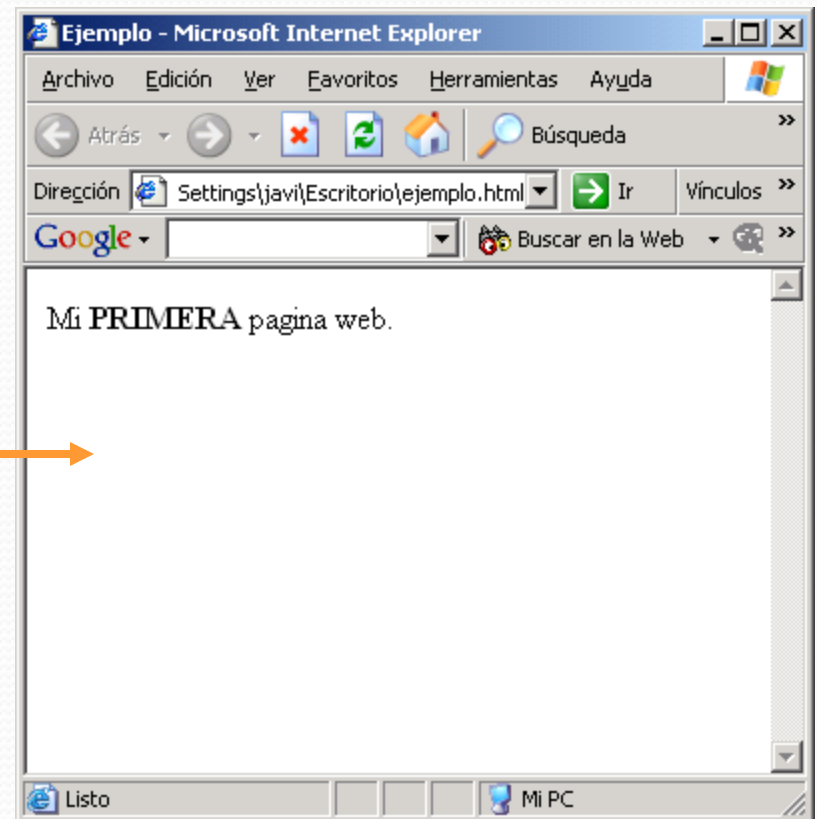
```
</HEAD>
```

```
<BODY>
```

```
Mi <B> PRIMERA </B> pagina web.
```

```
</BODY>
```

```
</HTML>
```





# Etiquetas y elementos

- HTML especifica un conjunto de **etiquetas** que indican como estructurar un documento:
  - Etiquetas van entre “< >”:
    - `` Indica la presencia de una imagen. El atributo `src` proporciona la URL de la imagen
  - La *mayoría* de las etiquetas están apareadas: existe una etiqueta de inicio y otra de fin:
    - `<title>` y `</title>` encierran el título de una página.

# Etiquetas y elementos

Formalmente, un **elemento** HTML es un objeto encerrado entre un par de etiquetas:

`<title>Mi página web</title>` es un elemento TITLE

`<b>Texto en negrita</b>` es un elemento BOLD

Los elementos pueden contener otros elementos:

`<p> Texto con parte en <b>negrita</b></p>`

Es un elemento PARAGRAPH que contiene un elemento con BOLD.

# Etiquetas y elementos

- Parámetros de los elementos:
  - Los elementos pueden tener diferentes parámetros o “atributos” asociados.
  - Su sintaxis es:

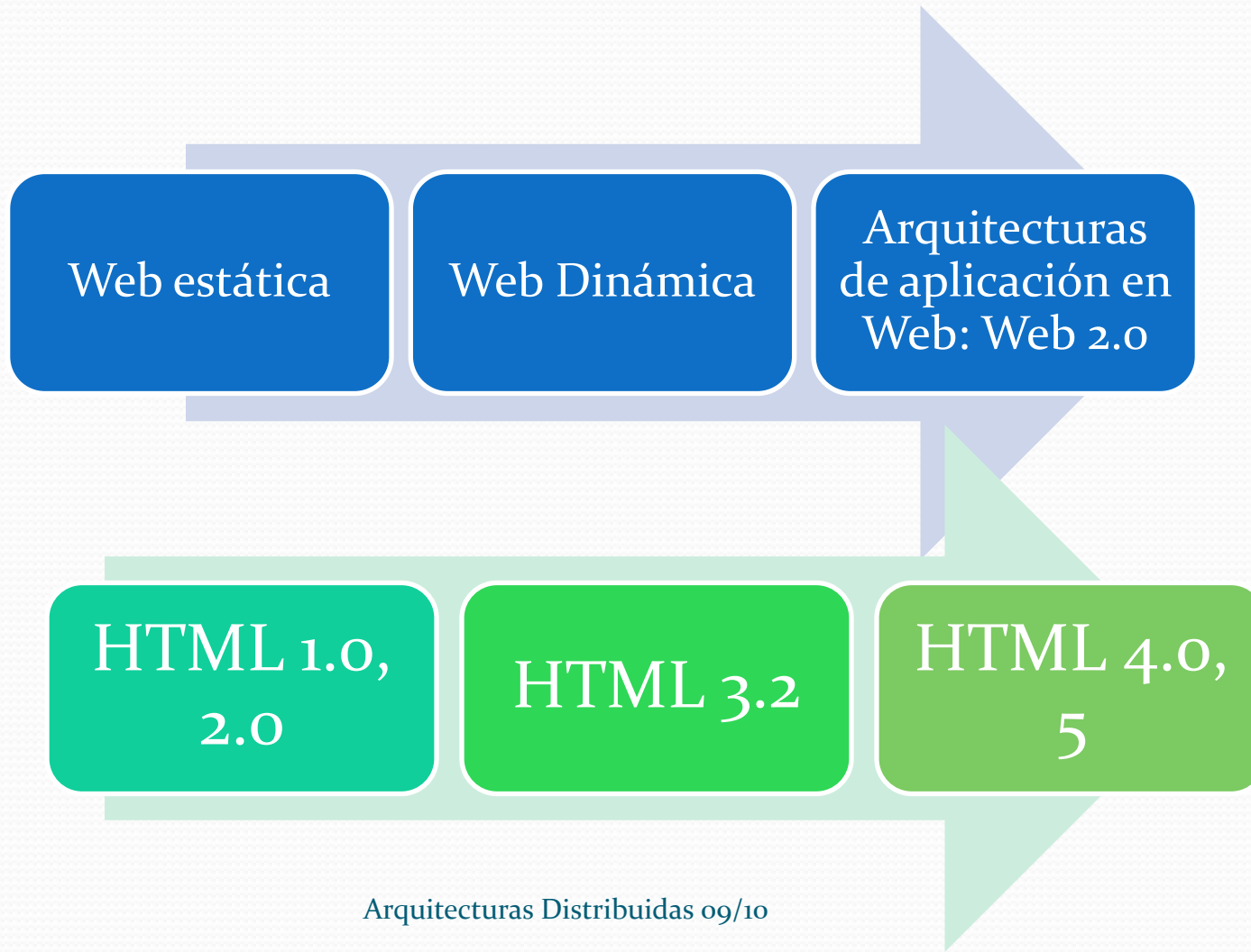
```
<TAG PAR1="VALOR1" PAR2="VALOR2" ... >
```

# Evolución de HTML

HTML es un estándar en evolución (se desarrolla a medida que nuevas tecnologías y herramientas están disponibles).

- **HTML 1** (Berners-Lee, 1989): muy básico, integración limitada de multimedia en 1993, Moscaic añadió muchas características nuevas (por ejemplo, imágenes integradas).
- **HTML 2.0** (IETF, 1994): intentó estandarizar éstas y otras propiedades, pero más tarde, entre 1994-96, Netscape y IE añadieron muchas nuevas (y divergentes) funcionalidades.
- **HTML 3.2** (W3C, 1996): trato de unificar HTML en un único estándar, pero de nuevo topó con características imprevistas, como *applets* Java y flujos de video.
- **HTML 4.0** (W3C, 1997): estándar actual, se diseñó para adelantarse a futuros desarrollos del web, no solo para crear otro estándar post-industrial.
- **XHTML 1.0** (W3C, 2000): versión de HTML 4.01 modificada para ser compatible con XML → Cambio de "forma", pero no de "fondo".
- **HTML 5** (W3C, 2004): estándar propuesto, sustituye a los anteriores e introduce nuevas marcas

# Lenguaje HTML: evolución



# I. HTML

## 1. Introducción

### 1.1. Motivación

### 1.2 SGML

### 1.3 Lenguaje HTML: introducción

### 1.4. Hipertexto y HTML

### 1.5. Evolución de HTML

## 2. **Lenguaje de Marcado de Hipertexto (HTML)**

### 2.1. Etiquetas y elementos

### 2.2. Estructura genera

### 2.3. Disposición del texto

### 2.4. Listas

### 2.5. Hiperenlaces

### 2.6. Imágenes

### 2.7. Tablas

### 2.8. Limitaciones del HTML 3.2

### 2.9. HTML 4.0

# Estructura general

- Un documento HTML está formado por:
  - DEFINICIÓN DE TIPO DE DOCUMENTO (DTD) con la indicación de qué formato de HTML está siendo utilizado en el resto del documento.
  - CABECERA (HEAD) que contiene información para el funcionamiento del navegador y de la página web.
    - Ejemplo, el título para la ventana del navegador, definiciones de estilo, código de *script*, etc.
  - CUERPO (BODY) con los contenidos reales que se mostrarán en la página web.
    - Ejemplo: texto, enlaces a otras páginas, imágenes, etc.

# Estructura general

- Ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.d
td">
<html>
<!-- Ejemplo 1 -->
<!-- Esto son comentarios -->

<head>
  <title>Titulo de la pagina</title>
</head>

<body>
  Contenido de la pagina.
</body>

</html>
```

La primera línea del documento debe ser la definición del tipo. Cualquier documento “conforme” debe tener esa línea.

Documentos HTML comienzan y finalizan con las etiquetas `<html>` y `</html>`, respectivamente

Es posible usar comentarios con `<!--` y `-->`

La CABECERA (HEAD) se inserta entre `<head>` y `</head>`

La sección con el CUERPO (BODY) va entre `<body>` y `</body>`



# Estructura general

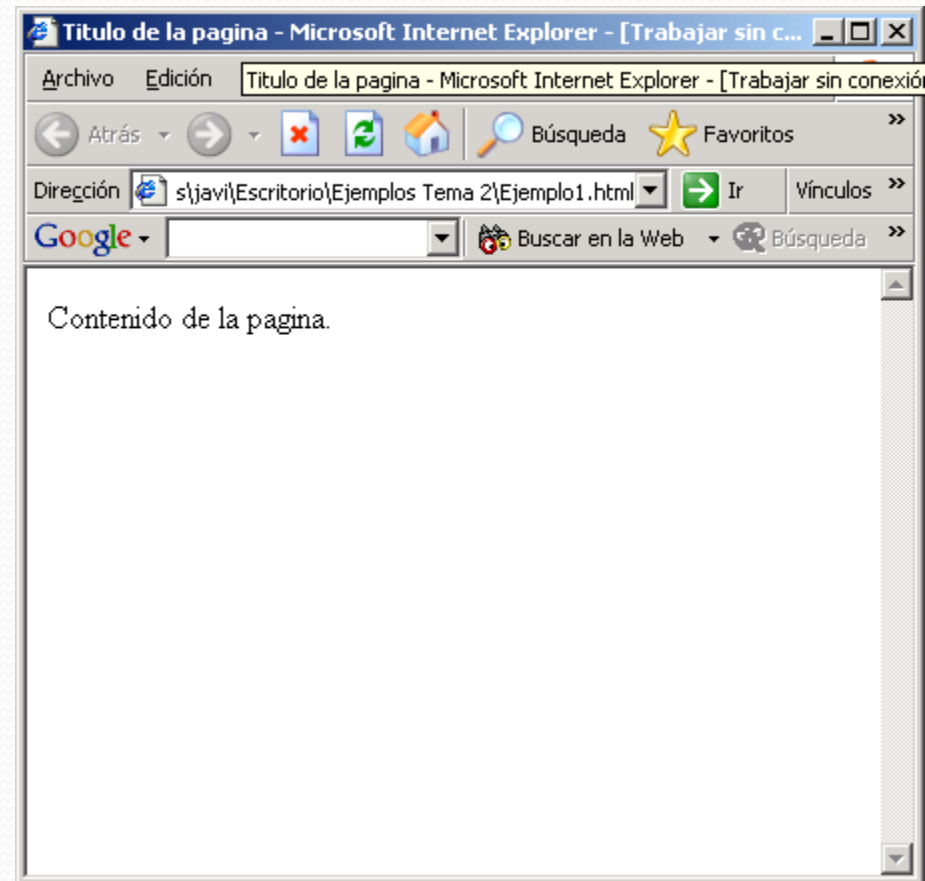
- Ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.d
td">
<html>
<!-- Ejemplo 1 -->
<!-- Esto son comentarios -->

<head>
  <title>Titulo de la pagina</title>
</head>

<body>
  Contenido de la pagina.
</body>

</html>
```



## Estructura general:

### Jerarquía de los elementos del cuerpo

- Elementos BLOQUE → Aquellos que provocan el comienzo de un nuevo párrafo: <H1>...<H6>, <P>, <LI>, etc
- Elementos TEXTO → No provocan un nuevo párrafo: <IMG>, <A>, <FONT>, etc.
- BLOQUE están formados de otros BLOQUE y TEXTO.
- TEXTO sólo pueden contener otros elementos TEXTO.
- Modelo exacto depende del elemento concreto.
  - ¿Cómo se define tal modelo? DTD, más adelante

# Disposición del texto

```
<!doctype HTML PUBLIC "-//W3C/DTD
HTML 3.2 Final//EN">

<html>
<!-- Ejemplo 2 -->

<head>
  <title>Titulo</title>
</head>

<body>
  Aquí aparecerá el
  texto que deseemos mostrar en la
  pagina web
  no
  importa que formato tenga
  aquí, ya que el
  navegador lo ignora
  .
  .
  .
</body>

</html>
```

- El CUERPO contiene múltiples líneas de texto.
- Su disposición en el documento, así como el espaciado son ignorados por el navegador
- Cada secuencia de espacios en blanco es interpretada como un solo espacio.
- El navegador automáticamente corta el texto para que encaje en el tamaño de la ventana.
- El texto dentro del documento puede estructurarse del modo más conveniente, ya que no afecta a como se representa en el navegador.

# Disposición del texto

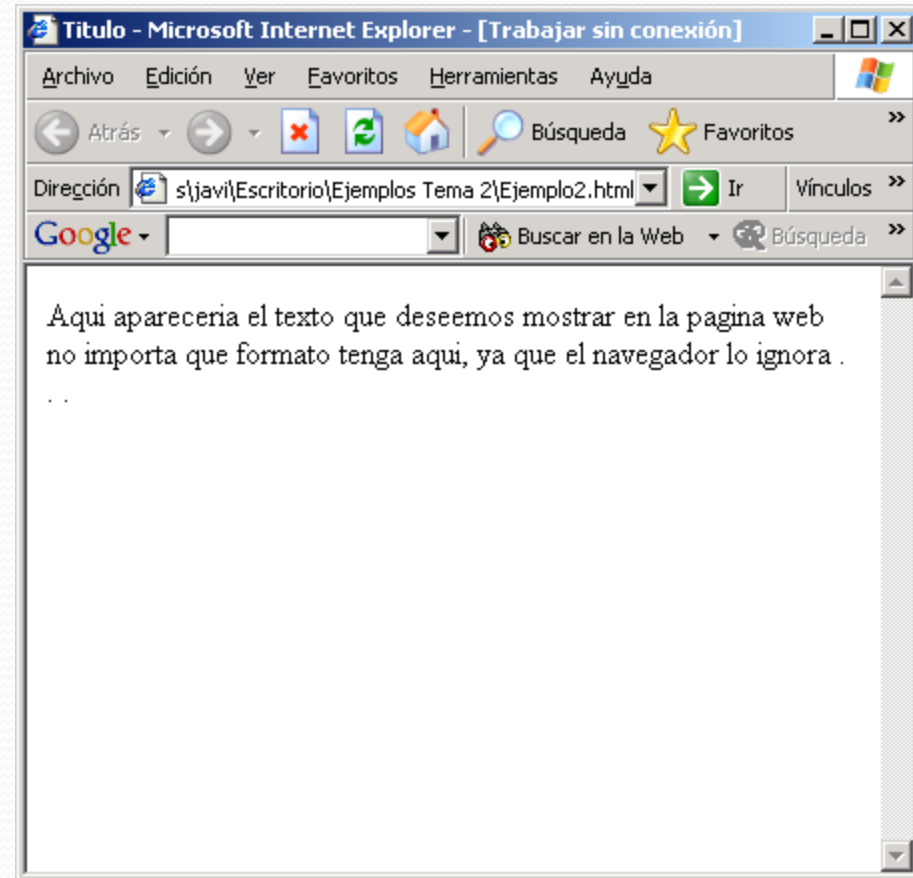
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.d
td">

<html>
<!-- Ejemplo 2 -->

<head>
  <title>Titulo</title>
</head>

<body>
  Aquí aparecería el
  texto que deseemos mostrar en la
  página web
  no
  importa que formato tenga
  aquí, ya que el
  navegador lo ignora
  .
  .
  .
</body>

</html>
```



# Cambios en la disposición del texto

```
<html>
<!-- Ejemplo 3 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <p>
  Parrafo con <br>
  dos lineas
  </p>

  <p>
  Parrafo que inclye una
  &nbsp; SEPARACION &nbsp; entre
  algunas
  de las palabras.
  </p>

  <p>
  &nbsp;&nbsp; Este parrafo esta
  <br/>
  tabulado en la primera linea
  <br/>
  pero no en las siguientes.
  </p>
</body>

</html>
```

## Cambios en la disposición:

- Puede provocarse un salto de línea usando la etiqueta simple `<br>`
- Puede crearse un nuevo parrafo (que comience en una nueva línea) con `<p>...</p>`
- Pueden forzarse espacios en blanco con el símbolo de espacio “sin ruptura” (non-breaking space): `&nbsp;`

# Cambios en la disposición del texto

```
<html>
<!-- Ejemplo 3 -->

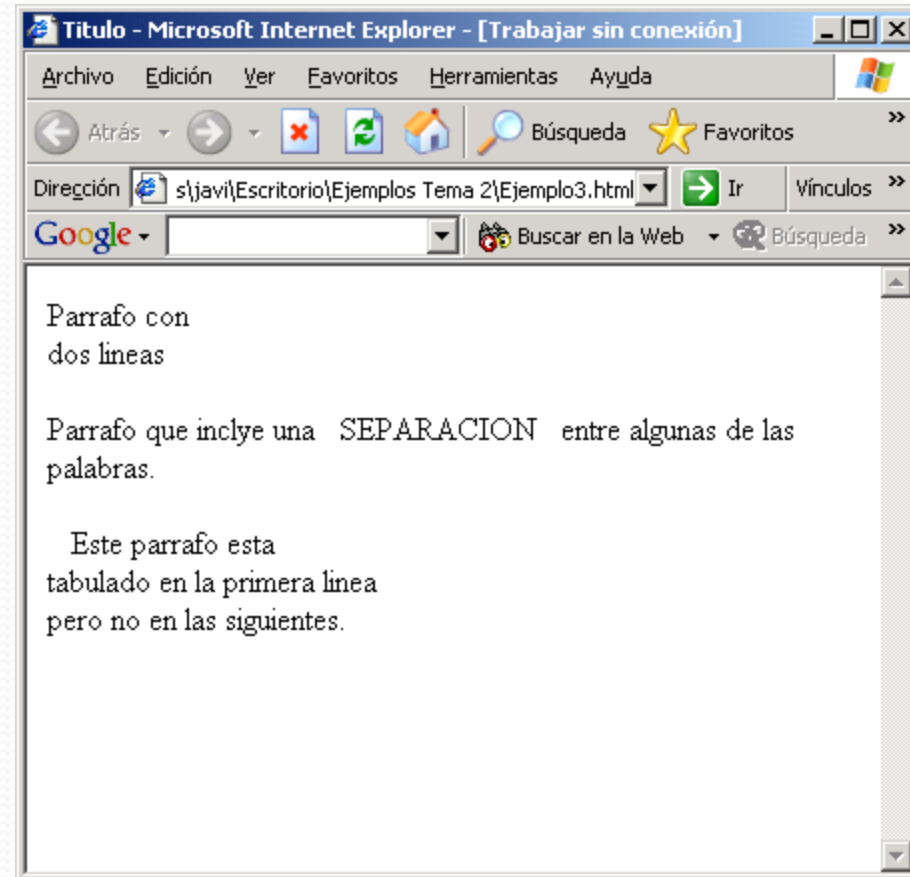
<head>
  <title>Titulo</title>
</head>

<body>
  <p>
  Parrafo con <br>
  dos lineas
  </p>

  <p>
  Parrafo que inclye una
  &nbsp; SEPARACION &nbsp; entre
  algunas
  de las palabras.
  </p>

  <p>
  &nbsp;&nbsp; Este parrafo esta
  <br/>
  tabulado en la primera linea
  <br/>
  pero no en las siguientes.
  </p>
</body>

</html>
```



# Separación de los bloques de texto

```
<html>
<!-- Ejemplo 4 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <h1> Primer encabezado </h1>
  <p>
  Texto cualquiera.
  </p>

  <h2> Subcabecera </h2>
  <p>
  Mas texto.
  </p>

  <h1> Segundo encabezado </h1>
  <p> Mas texto. </p>
</body>

</html>
```

- Pueden especificarse encabezados para párrafos y bloques de texto.
  - Etiquetas `<h1>...</h1>` producen una cabecera con un formato que denota mucha importancia (el formato final depende del navegador)
  - `<h2>...</h2>` producen encabezados de “menor importancia”.
  - etc
  - `<h6>...</h6>` producen los encabezados de menor nivel.

# Separación de los bloques de texto

```
<html>
<!-- Ejemplo 4 -->

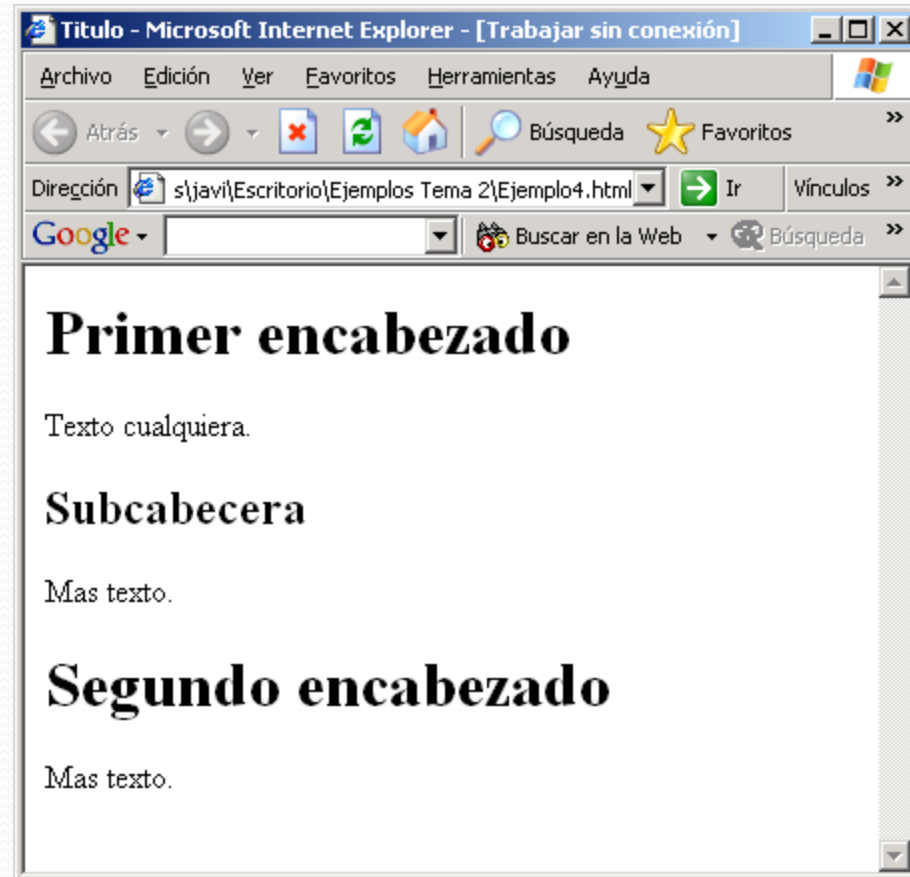
<head>
  <title>Titulo</title>
</head>

<body>
  <h1> Primer encabezado </h1>
  <p>
    Texto cualquiera.
  </p>

  <h2> Subcabecera </h2>
  <p>
    Mas texto.
  </p>

  <h1> Segundo encabezado </h1>
  <p> Mas texto. </p>
</body>

</html>
```





# Separación de los bloques de texto

```
<html>
<!-- Ejemplo 5 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <p> Texto cualquiera. </p>

  <hr>

  <p> Mas texto. </p>

  <hr width="50%">

  <p> Mas texto. </p>

  <hr size="10">

</body>
</html>
```

- Pueden insertarse separaciones entre secciones:
- `<hr>` dibuja una línea en la pantalla
- `<hr width="50%">` fija el ancho de la pantalla ocupado
- `<hr size=10 />` fija el grosor de la línea de separación

# Separación de los bloques de texto

```
<html>
<!-- Ejemplo 5 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <p> Texto cualquiera. </p>

  <hr>

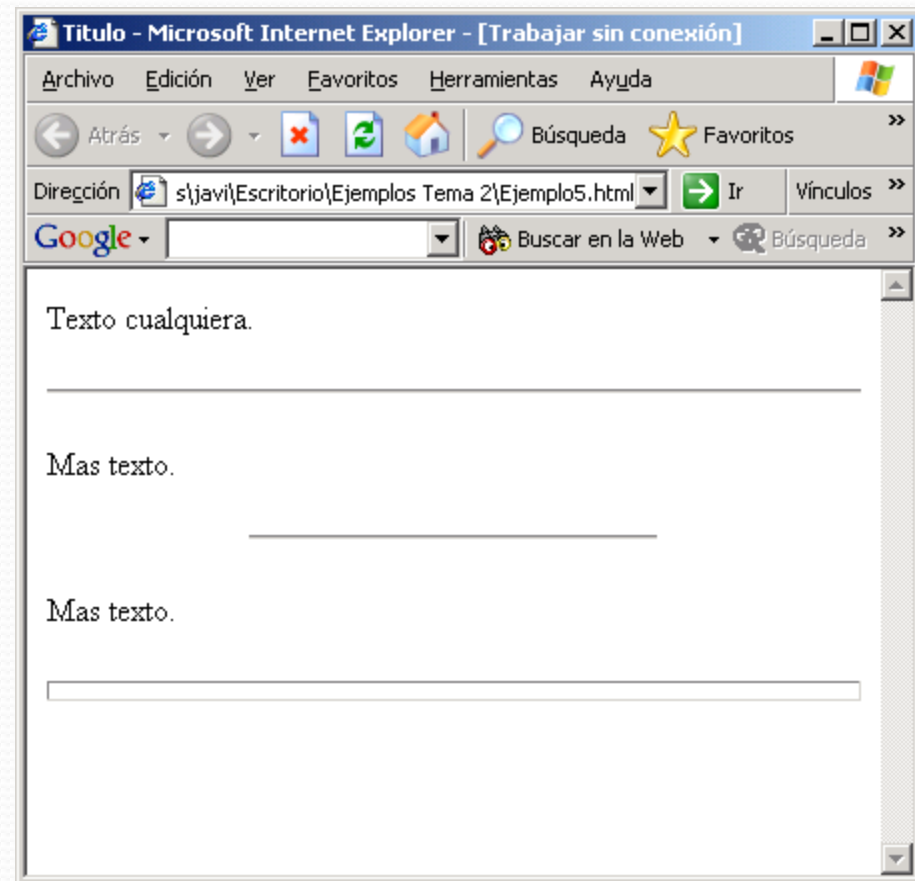
  <p> Mas texto. </p>

  <hr width="50%">

  <p> Mas texto. </p>

  <hr size="10">

</body>
</html>
```



# Alineación del texto

```
<html>
<!-- Ejemplo 6 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <h1 align="center"> Encabezado
centrado</h1>

  <p> Texto alineado a la izquierda (por
defecto) </p>

  <p align="center"> Texto centrado </p>

  <p align="right"> Texto a derechas </p>

  <div align="right">
    <p> Todo el texto del </p>
    <p> conjunto se alinea a </p>
    <p> derechas </p>
  </div>

</body>
</html>
```

- Es posible elegir la alineación del texto (por defecto alineado a la izquierda)
  - Atributo "align" de algunos elementos.
- Un conjunto de elementos puede alinearse si se agrupa en un elemento DIV:
  - <div> .. </div>

# Alineación del texto

```
<html>
<!-- Ejemplo 6 -->

<head>
  <title>Titulo</title>
</head>

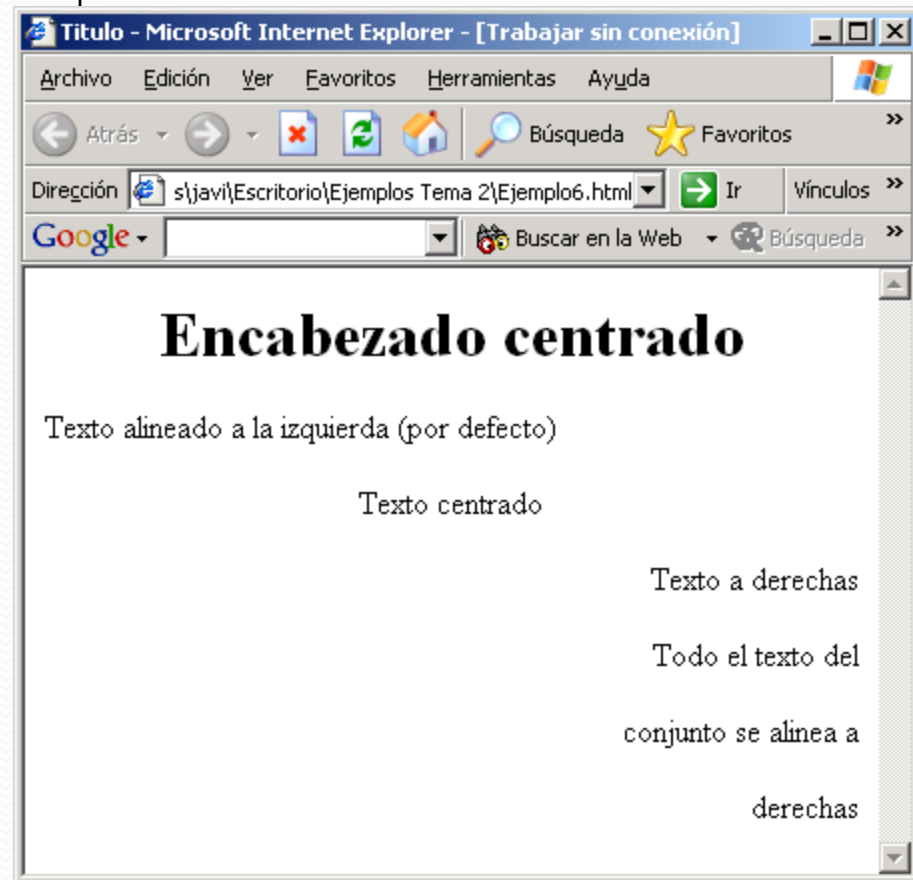
<body>
  <h1 align="center"> Encabezado
centrado</h1>

  <p> Texto alineado a la izquierda (por
defecto) </p>

  <p align="center"> Texto centrado </p>

  <p align="right"> Texto a derechas </p>

  <div align="right">
    <p> Todo el texto del </p>
    <p> conjunto se alinea a </p>
    <p> derechas </p>
  </div>
</body>
</html>
```



# Estilos de texto

```
<html>
<!-- Ejemplo 7 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <p>
    El texto puede ser señalado usando
    <b>negrita</b>, <i>cursiva</i>, incluso
    <big>haciendolo mayor</big>. <br>
    <u>Subrayar</u> el texto no esta
    aconsejado ya que parece un hiperenlace
    <br>
    La tetra de tipo fijo permite
    distinguir elementos como el codigo:
    <small><tt>sum = sum + i;</tt></small>
  </p>
</body>

</html>
```

- ESTILOS DE TEXTO:
  - <b>... </b> para **negrita**
  - <i>... </i> para *cursiva*
  - <u>... </u> para subrayados
  - <tt>... </tt> letra tipo fijo
  - <big>... </big> incrementa el tamaño de la letra
  - <small>... </small> decrementa el tamaño de la letra
  - **No deben utilizarse: utilizar hojas de estilo para el formato**

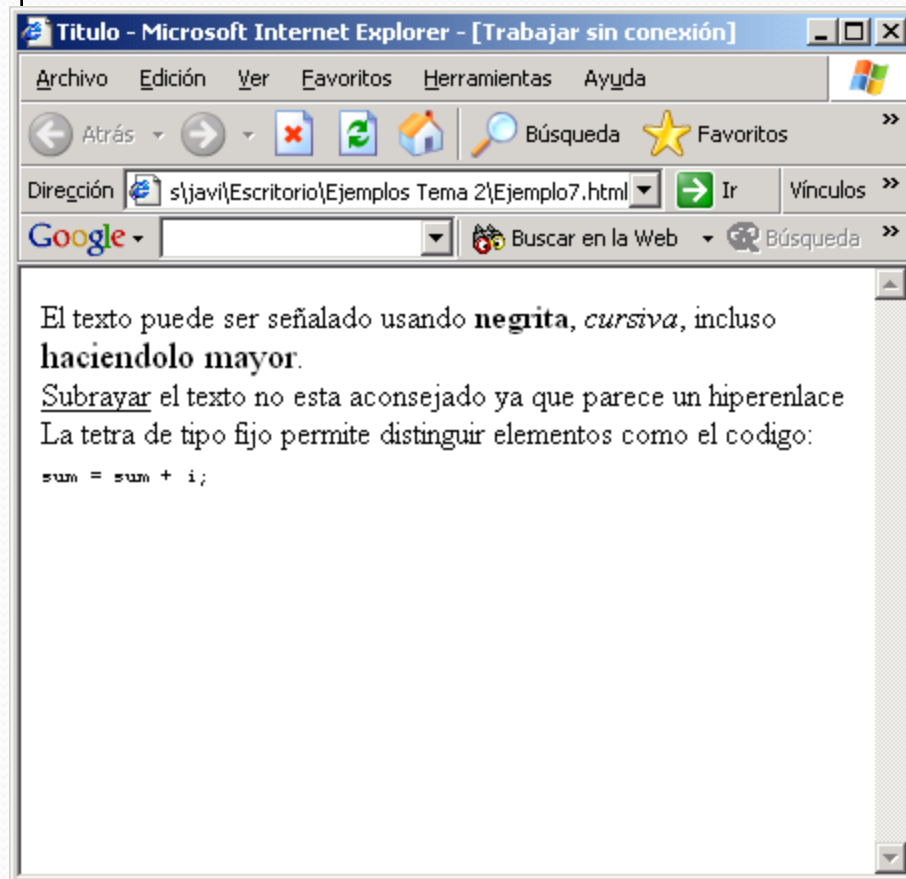
# Estilos de texto

```
<html>
<!-- Ejemplo 7 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <p>
    El texto puede ser señalado usando
    <b>negrita</b>, <i>cursiva</i>, incluso
    <big>haciendolo mayor</big>. <br>
    <u>Subrayar</u> el texto no esta
    aconsejado ya que parece un hiperenlace
  <br>
    La tetra de tipo fijo permite
    distinguir elementos como el codigo:
    <small><tt>sum = sum + i;</tt></small>
  </p>
</body>

</html>
```



# Estilos de texto

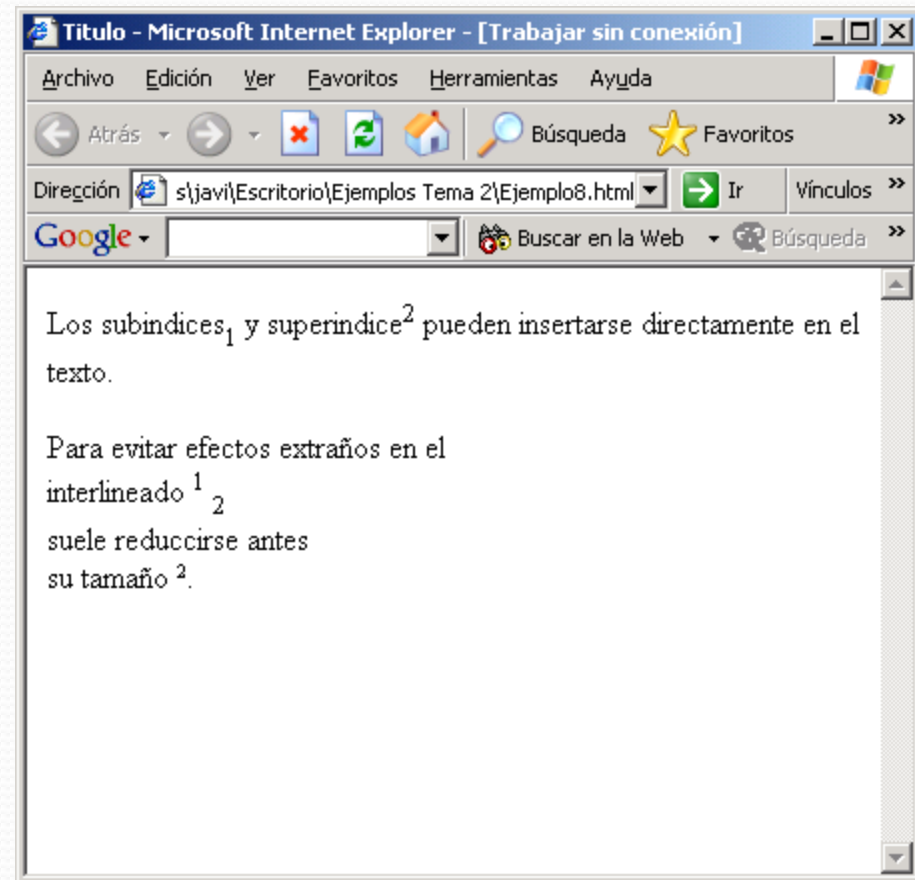
```
<html>
<!-- Ejemplo 8 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <p>
    Los subindices<sub>1</sub>
    y superindice<sup>2</sup>
    pueden insertarse directamente
    en el texto.
  </p>

  <p> Para evitar efectos extraños en el
  <br>
  interlineado <sup>1</sup> <sub>2</sub>
  <br>
  suele reducirse antes <br>
  su tamaño <small><sup>2</sup></small>.
  </p>

</body>
</html>
```



# Agrupaciones de texto

```
<html>
<!-- Ejemplo 9 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <p>
    <tt><pre>
      for (i = 0; i < 10; i++) {
        sum = sum + i;
      }
    </pre></tt>
  </p>

  <p>
    Comentario del ultimo teorema de Fermat (siglo
    XVII):
    <blockquote>
      Es imposible dividir un cubo en suma de otros dos o
      un bicuadrado en otros dos bicuadrados, en general
      una potencia cualquiera superior a dos en dos
      potencias del mismo grado; he descubierto una
      demostración maravillosa pero en este margen es
      demasiado estrecho para contenerla.
    </blockquote>
  </p>
</body>

</html>
```

Inserción de texto respetando saltos de línea:

```
<pre>...</pre>
```

Útil para insertar código, por ejemplo.

Texto con indentación en ambos márgenes:

```
<blockquote>...</blockquote>
```

Útil para insertar “citas”, por ejemplo.



# Agrupaciones de texto

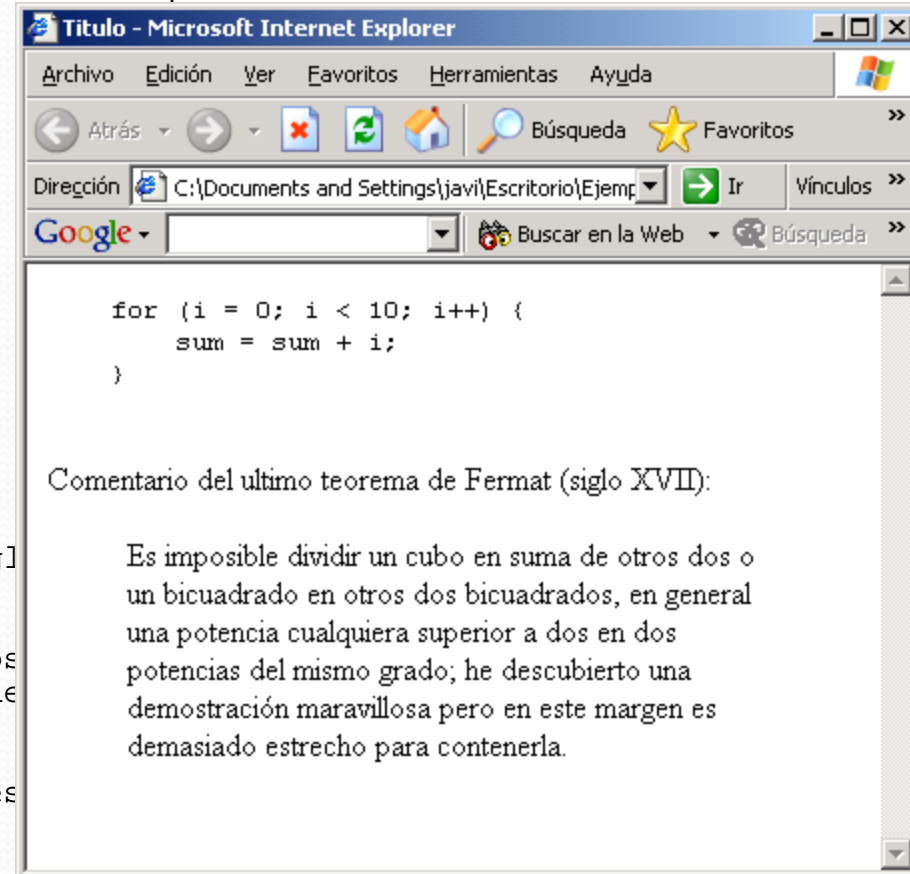
```
<html>
<!-- Ejemplo 9 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <p>
    <tt><pre>
      for (i = 0; i < 10; i++) {
        sum = sum + i;
      }
    </pre></tt>
  </p>

  <p>
    Comentario del ultimo teorema de Fermat (siglo
    XVII):
    <blockquote>
      Es imposible dividir un cubo en suma de otros dos o
      un bicuadrado en otros dos bicuadrados, en general
      una potencia cualquiera superior a dos en dos
      potencias del mismo grado; he descubierto una
      demostración maravillosa pero en este margen es
      demasiado estrecho para contenerla.
    </blockquote>
  </p>
</body>

</html>
```



# Listas

```
<html>
<!-- Ejemplo 10 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <p>
    <ol>
      <li> Primer elemento
      <li> Segundo elemento
      <li> Este debe ser el tercero
    </ol>
  </p>

  <ul>
    <li> Texto 1
    <li> Texto 2
  </ul>

  <p>
    <dl>
      <dt> HTML <dd> HyperText Markup Language
      <dt> HTTP <dd> HyperText Transfer Protocol
    </dl>
  </p>
</body>
</html>
```

## Existen tres tipos de listas:

- Listas ordenadas:  
<ol>...</ol> (cada elemento se precede por una letra o un número)
  - <li> identifica cada elemento de la lista
  - Se puede establecer el estilo y el índice inicial.
- Listas no ordenadas:  
<ul>...</ul> (cada elemento se precede con una señal)
  - <li> identifica cada elemento de la lista.
- Listas de definición:  
<dl>...</dl>
  - <dt> identifica cada termino
  - <dd> identifica su definición

# Listas

```
<html>
<!-- Ejemplo 10 -->

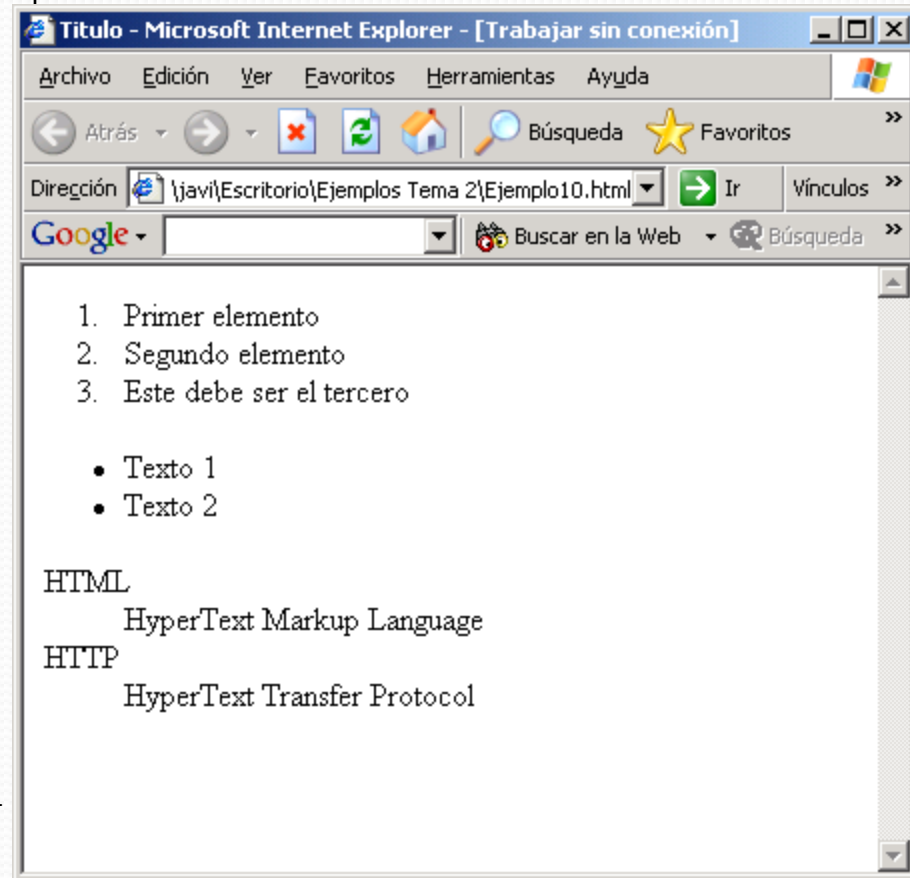
<head>
  <title>Titulo</title>
</head>

<body>
  <p>
    <ol>
      <li> Primer elemento
      <li> Segundo elemento
      <li> Este debe ser el tercero
    </ol>
  </p>

  <ul>
    <li> Texto 1
    <li> Texto 2
  </ul>

  <p>
    <dl>
      <dt> HTML <dd> HyperText Markup Language
      <dt> HTTP <dd> HyperText Transfer Protocol
    </dl>
  </p>
</body>

</html>
```



# Hiperenlaces

```
<html>
<!-- Ejemplo 11 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <p>
    <a href="http://www.upct.es">
    Universidad Politecnica de
    Cartagena</a>
    <br>
    <a href="http://www.teleco.upct.es">
    ETT Telemática</a>
    <br>
  </p>
</body>

</html>
```

- El elemento más importante de HTML son los hiperenlaces: Elemento “ANCHOR”.

- `<a href="URL">...</a>`

- URL indica el recurso con el que se enlaza este “ancla”

- **!!!URL RELATIVAS O ABSOLUTAS!!!**

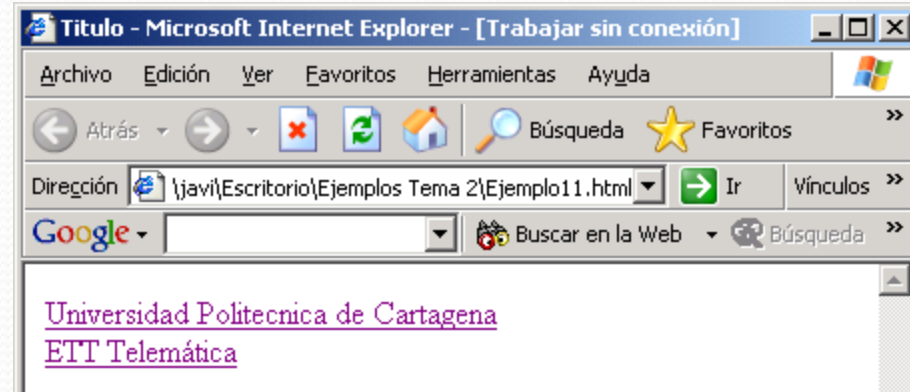
# Hiperenlaces

```
<html>
<!-- Ejemplo 11 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <p>
    <a href="http://www.upct.es">
      Universidad Politecnica de
      Cartagena</a>
    <br>
    <a href="http://www.teleco.upct.es">
      ETT Telemática</a>
    <br>
  </p>
</body>

</html>
```



# Hiperenlaces

```
<html>
<!-- Ejemplo 12 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <p align="center">
    [ <a href="#HTML">HTML</a> |
      <a href="#HTTP">HTTP</a> |
      <a href="#IP">IP</a> |
      <a href="#TCP">TCP</a> ]
  </p>
  <p>
  Acronimos:
  <dl>
    <a name="HTML"></a><dt>HTML
    <dd>HyperText Markup Language
    <a name="HTTP"></a><dt>HTTP
    <dd>HyperText Transfer Protocol
    <a name="IP"></a><dt>IP
    <dd>Internet Protocol
    <a name="TCP"></a><dt>TCP
    <dd>Transfer Control Protocol
  </p>
</body>
</html>
```

- Es también posible referenciar partes del mismo documento con un hiperenlace:

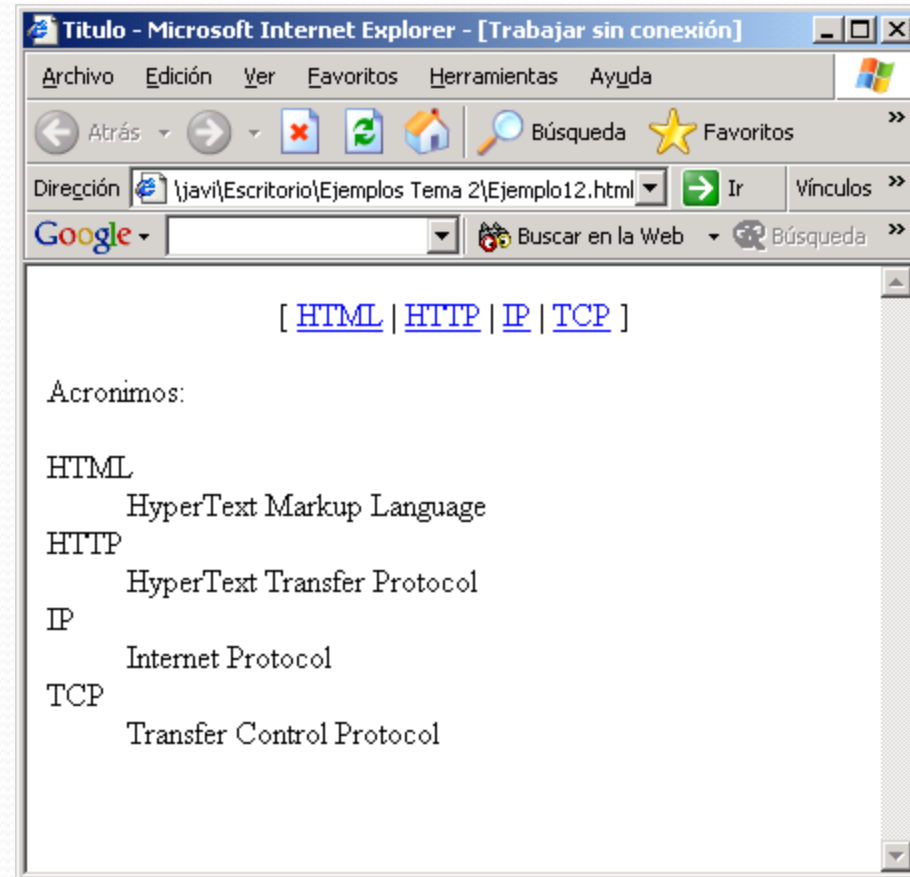
- `<a name="parte1">...</a>`
  - donde "parte1" es el identificador de la zona.
- `<a href="#parte1">...</a>`
  - Salta a la zona "parte1" de este documento.
- `<a href="URL#parte">...</a>`
  - Salta hasta la zona de la página identificada por la URL dada.

# Hiperenlaces

```
<html>
<!-- Ejemplo 12 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <p align="center">
    [ <a href="#HTML">HTML</a> |
      <a href="#HTTP">HTTP</a> |
      <a href="#IP">IP</a> |
      <a href="#TCP">TCP</a> ]
  </p>
  <p>
    Acronimos:
    <dl>
      <a name="HTML"></a><dt>HTML
      <dd>HyperText Markup Language
      <a name="HTTP"></a><dt>HTTP
      <dd>HyperText Transfer Protocol
      <a name="IP"></a><dt>IP
      <dd>Internet Protocol
      <a name="TCP"></a><dt>TCP
      <dd>Transfer Control Protocol
    </p>
  </body>
</html>
```



# Imágenes

```
<html>
<!-- Ejemplo 13 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <div align="center">
    
    <p>Prueba de imagen</p>
  </div>
</body>
</html>
```

Para incluir una imagen se usa el elemento IMG:

- Por defecto, todos los navegadores soportan .jpg, .gif y png.
- Para visualizar otros tipos puede ser necesario un *plug-in*.

```

```

- La imagen se identifica con su URL.
- La etiqueta de imagen no contiene ningún elemento

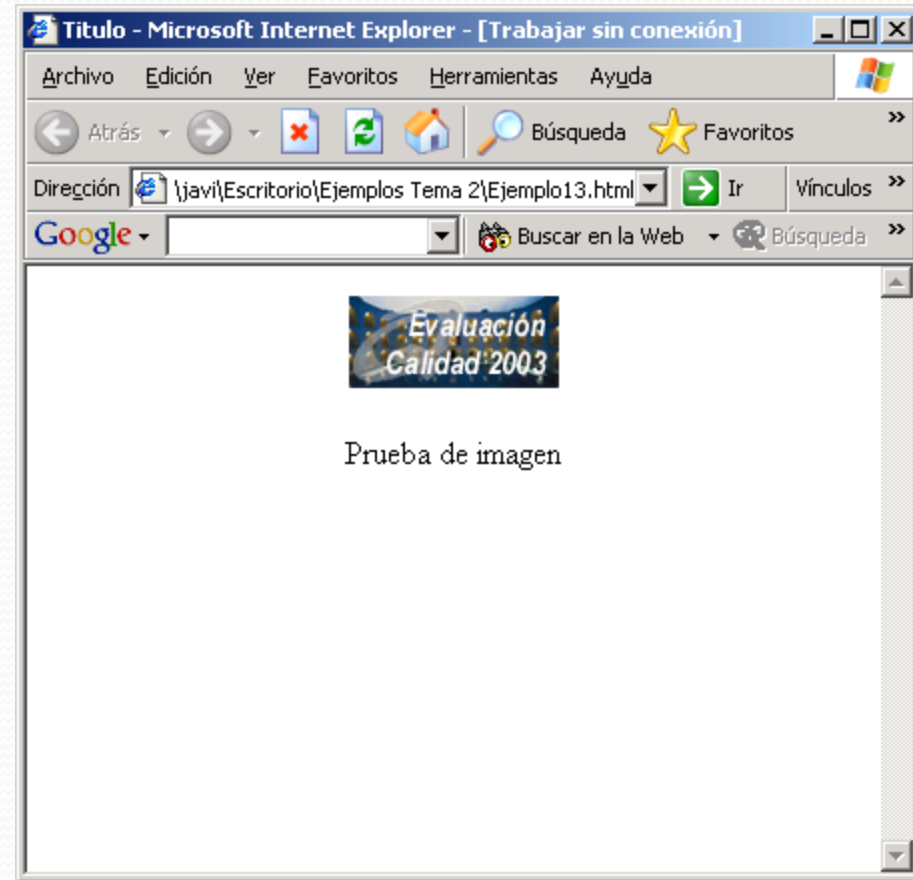


# Imágenes

```
<html>
<!-- Ejemplo 13 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <div align="center">
    
    <p>Prueba de imagen</p>
  </div>
</body>
</html>
```



# Tablas

```
<html>
<!-- Ejemplo 14 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <table>
    <tr>
      <td>foo</td> <td>bar</td>
    </tr>
    <tr>
      <td>biz</td> <td>baz</td>
    </tr>
  </table>
</body>
</html>
```

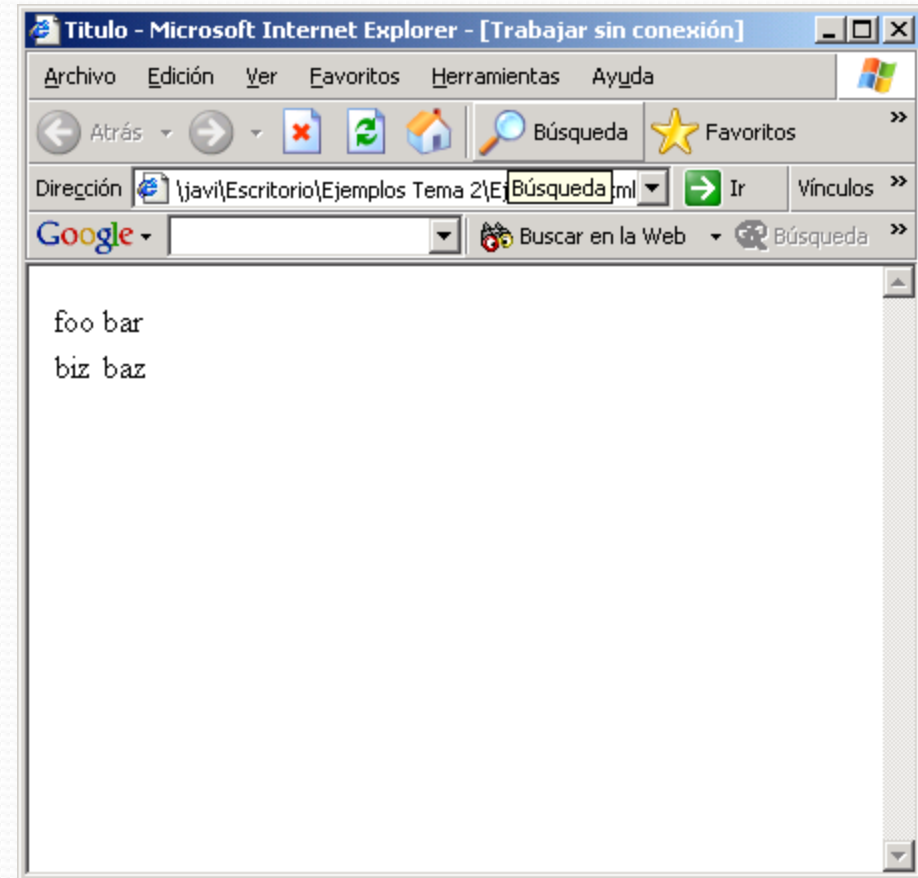
- Se utilizan para presentar de forma ordenada colecciones complejas de objetos
  - Una tabla divide los objetos en “filas” y “columnas”.
- `<table>...</table>` indica un elemento tabla
- `<tr>...</tr>` indica una fila (*row*)
- `<td>...</td>` indica un dato de la tabla, es decir, cada una de las columnas de la tabla.

# Tablas

```
<html>
<!-- Ejemplo 14 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <table>
    <tr>
      <td>foo</td> <td>bar</td>
    </tr>
    <tr>
      <td>biz</td> <td>baz</td>
    </tr>
  </table>
</body>
</html>
```



# Disposición de la tabla

```
<html>
<!-- Ejemplo 15 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <table border=1>
    <tr align="center">
      <td>foo<br>foo</td>
      <td valign="top">bar</td>
    </tr>
    <tr>
      <td>bizbiz</td>
      <td>booboo</td>
    </tr>
  </table>
</body>
</html>
```

Usando el atributo BORDER puede indicarse que aparezca un borde:

```
<table border=1>
```

Si se incrementa el número el borde se hará más grueso

El formato horizontal y vertical puede cambiarse para cada "celda":

```
<td align="center">
<td align="right">
<td valign="top">
<td valign="bottom">
```

O bien, para cada fila:

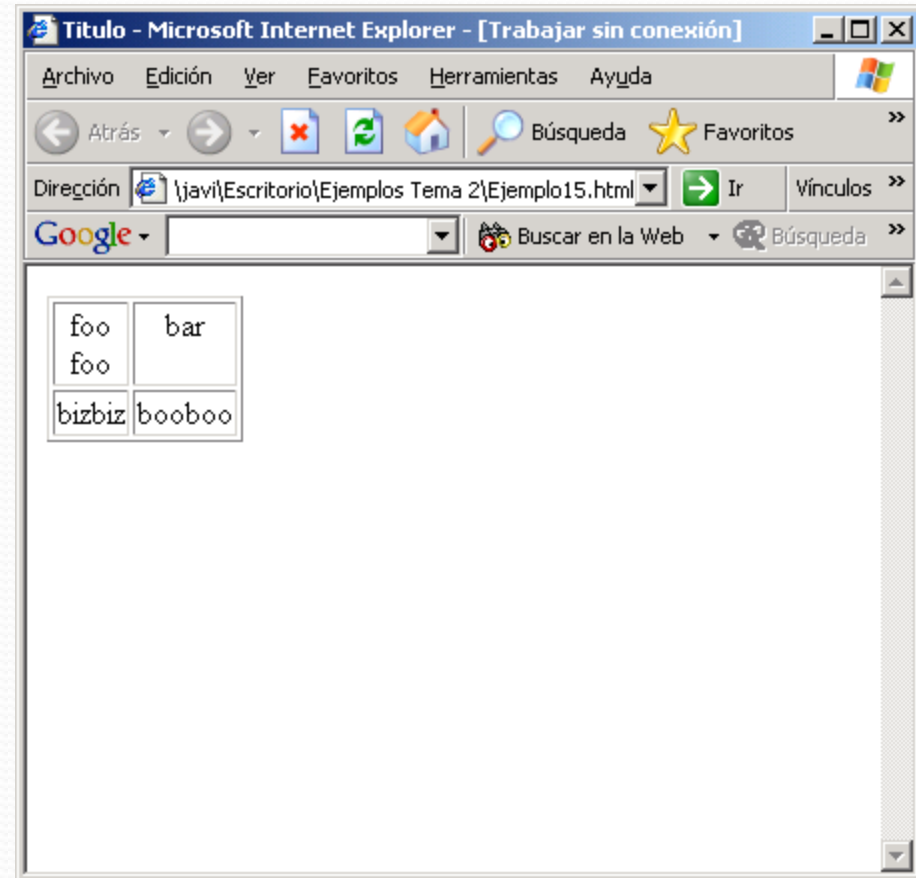
```
<tr align="center">
<tr valign="top">
```

# Disposición de la tabla

```
<html>
<!-- Ejemplo 15 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <table border=1>
    <tr align="center">
      <td>foo<br>foo</td>
      <td valign="top">bar</td>
    </tr>
    <tr>
      <td>bizbiz</td>
      <td>booboo</td>
    </tr>
  </table>
</body>
</html>
```



# Disposición de la tabla

```
<html>
<!-- Ejemplo 16 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <table width="100%">
    <tr>
      <td>A la izquierda
      <td align="right">a la
          derecha</td>
    </tr>
  </table>
</body>
</html>
```

Por defecto, el navegador escoge el tamaño de la tabla para que encajen todos los elementos.

Sin embargo, puede escogerse el tamaño de la tabla en proporción al ancho de la página:

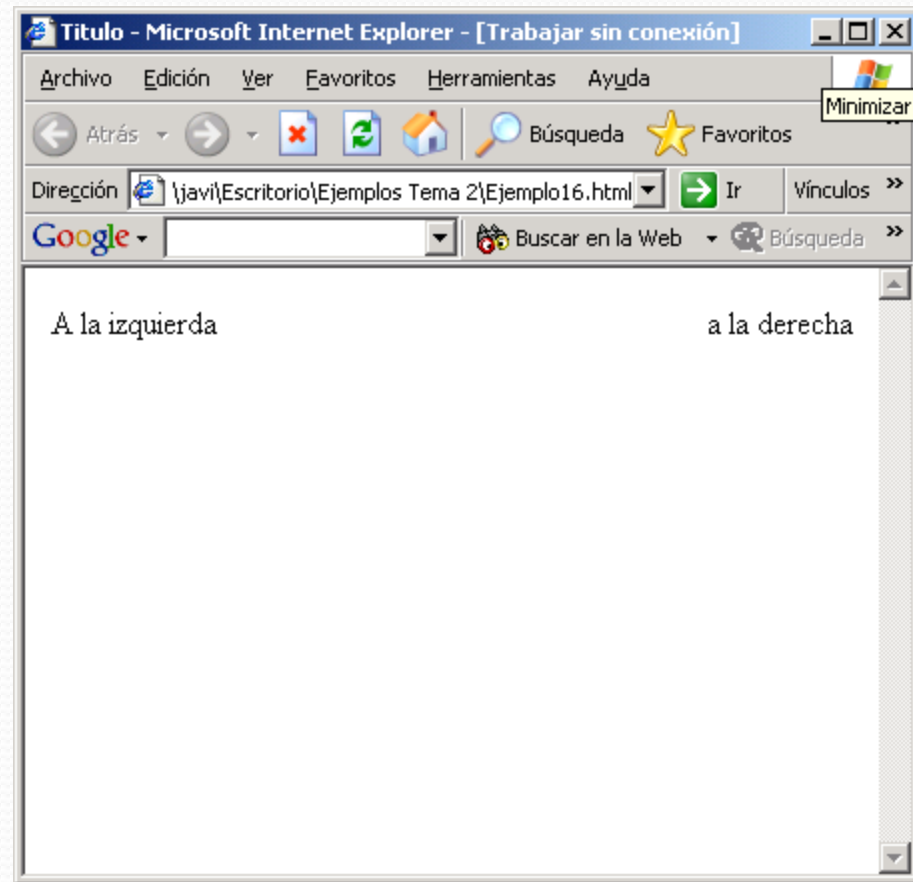
```
<table width="60%">
```

# Disposición de la tabla

```
<html>
<!-- Ejemplo 16 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <table width="100%">
    <tr>
      <td>A la izquierda
      <td align="right">a la
          derecha</td>
    </tr>
  </table>
</body>
</html>
```



# Otras opciones para tablas

```
<html>
<!-- Ejemplo 17 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <table border=1
    cellspacing=4 cellpadding=8>
    <tr> <th>CABECERA1</th>
      <th>CABECERA2</th>
      <th>CABECERA3</th>
    </tr>
    <tr>
      <td>uno</td> <td>dos</td> <td>tres</td>
    </tr>
    <tr>
      <td rowspan=2 align="center"> cuatro
</td>
      <td colspan=2 align="center"> cinco
</td>
    </tr>
    <tr>
      <td> seis </td> <td> siete </td>
    </tr>
  </table>
</body>
</html>
```

Es posible controlar el espacio entre “celdas” y los márgenes dentro de ellas:

```
<table cellspacing=5>
<table cellpadding=5>
```

Añadir cabeceras:

<th> es parecido a <td> pero muestra el contenido centrado y en negrita

Datos que ocupan más de una columna:

```
<td colspan=2>
```

O más de una fila:

```
<td rowspan=2>
```

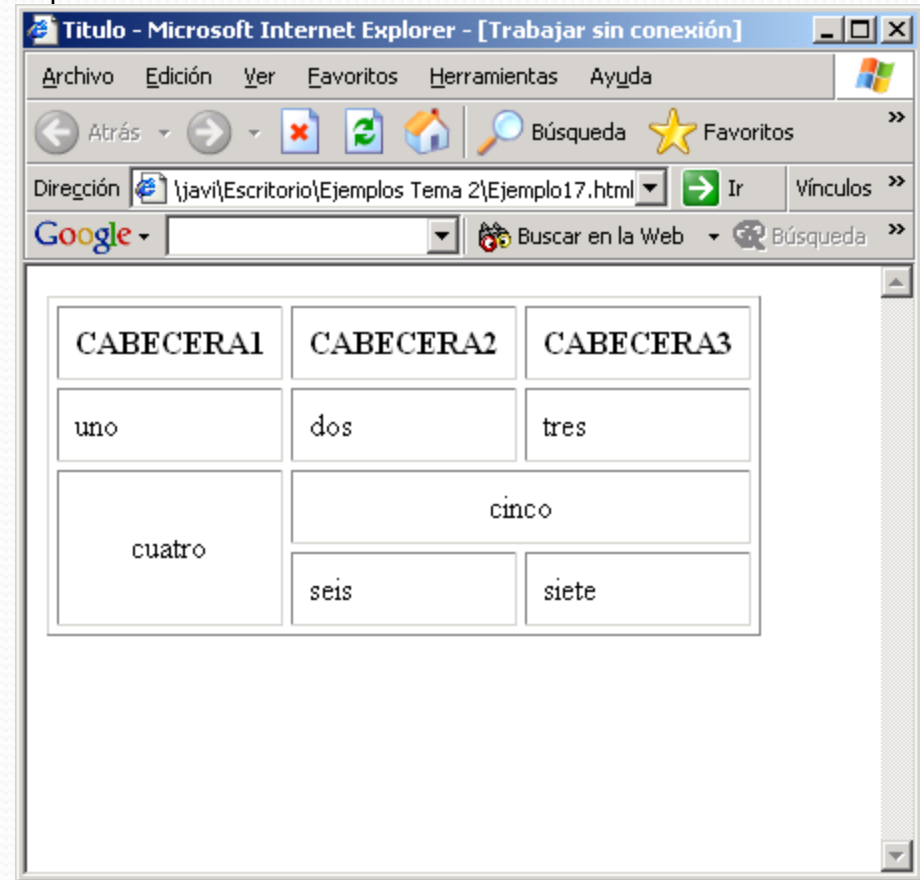


# Otras opciones para tablas

```
<html>
<!-- Ejemplo 17 -->

<head>
  <title>Titulo</title>
</head>

<body>
  <table border=1
    cellspacing=4 cellpadding=8>
    <tr>
      <th>CABECERA1</th>
      <th>CABECERA2</th>
      <th>CABECERA3</th>
    </tr>
    <tr>
      <td>uno</td> <td>dos</td> <td>tres</td>
    </tr>
    <tr>
      <td rowspan=2 align="center"> cuatro
    </td>
      <td colspan=2 align="center"> cinco
    </td>
    </tr>
    <tr>
      <td> seis </td> <td> siete </td>
    </tr>
  </table>
</body>
</html>
```



# Limitaciones de HTML 3.2

- Contenido y formato están entrelazados
  - Dificulta la actualización e inserción de contenidos.
  - Dificulta la selección de un formato determinado, y sobre todo, su cambio.
  - En ocasiones (ej. portal corporativo) todas las páginas deben adoptar un mismo formato: con HTML 3.2 esto resulta tedioso y obliga a la modificación de todas las páginas.
  - Por si mismo, HTML no posee la suficiente flexibilidad en cuanto al formato
- Solución → EL FORMATO DEBE DEFINIRSE INDEPENDIENTEMENTE → HOJAS DE ESTILO (en el siguiente tema)

# Limitaciones de HTML

- Contenido estático: Independiente de la versión de HTML
  - HTML *per se* sólo permite definir contenidos estáticos. Esto no es conveniente si:
    - La página sufre modificaciones frecuentes (ej: www con valores bursátiles).
    - Se desea realizar páginas personalizadas (ej: publicidad asociada al cliente que visita la página)
    - Etc.
- Solución → GENERACIÓN POR PROGRAMA DE HTML → VÍA APLICACIONES EXTERNAS, O BIEN USANDO LENGUAJES DE *SCRIPT* EN EL SERVIDOR.

# Limitaciones de HTML

- Falta de interactividad:
  - Las páginas deben ser más que simple información:
    - Aplicaciones empotradas en WWW para tareas interactivas con los usuarios, y con la ventaja de ser sistemas universales (como web).
    - Pequeñas tareas como comprobación de datos, inserción de información (hora, lenguaje, etc), deberían ser posibles, y sin tener que generar carga adicional hacia los servidores.
  - SOLUCIÓN → LENGUAJES EMPOTRADOS (JAVA), Y LENGUAJES “LIGEROS” PARA SCRIPT EN LADO DEL CLIENTE: JAVASCRIPT, JSCRIPT, ETC.
  - AJAX (Asynchronous JavaScript and XML)

# HTML 4.0

- <http://www.w3.org/TR/html4/cover.html>
- Actualización cuantitativa de HTML 3.2
- Con HTML 4.0 se intenta separar el “contenido” de la “presentación”, usando un mecanismo generalizado de hojas de estilo.
- Introduce nuevas etiquetas:
  - Facilidades para discapacitados
  - Internacionalización mediante el atributo *lang*
  - Nuevos elementos: <FRAME>, <OBJECT>, etc.

# HTML 4.0

- Actualización cuantitativa de HTML 3.2
- Con HTML 4.0 se intenta separar el “contenido” de la “presentación”, usando un mecanismo generalizado de hojas de estilo.
- Facilidades para discapacitados
- Internacionalización mediante el atributo *lang*
- Nuevos elementos: <FRAME>, <OBJECT>, etc.

# HTML 4.0

- Es posible usar lenguajes de *script* de modo más flexible y útil.
- Además, se proporciona un mecanismo consistente para insertar objetos empotrados (como un *applet* Java o una animación Flash).
- Ciertos elementos se declaran en desuso *deprecated* (todavía pueden usarse, pero no está recomendado): `CENTER`, `FONT`
- Otros elementos se declaran obsoletos (ya no pueden usarse): `LISTING`, `PLAINTEXT`, `XMP`

# HTML 4.0

```
<html>
<!-- Ejemplo 18 -->

<frameset cols="*,*">
  <frame src="ejemplo1.html">
  <frame src="ejemplo2.html">
</frameset>

</html>
```

- Los nuevos elementos FRAME y FRAMESET proporcionan un medio para dividir la pantalla en páginas independientes.
- Cada cuadro contiene una página totalmente independiente (con su separador, barra de desplazamiento, etc).

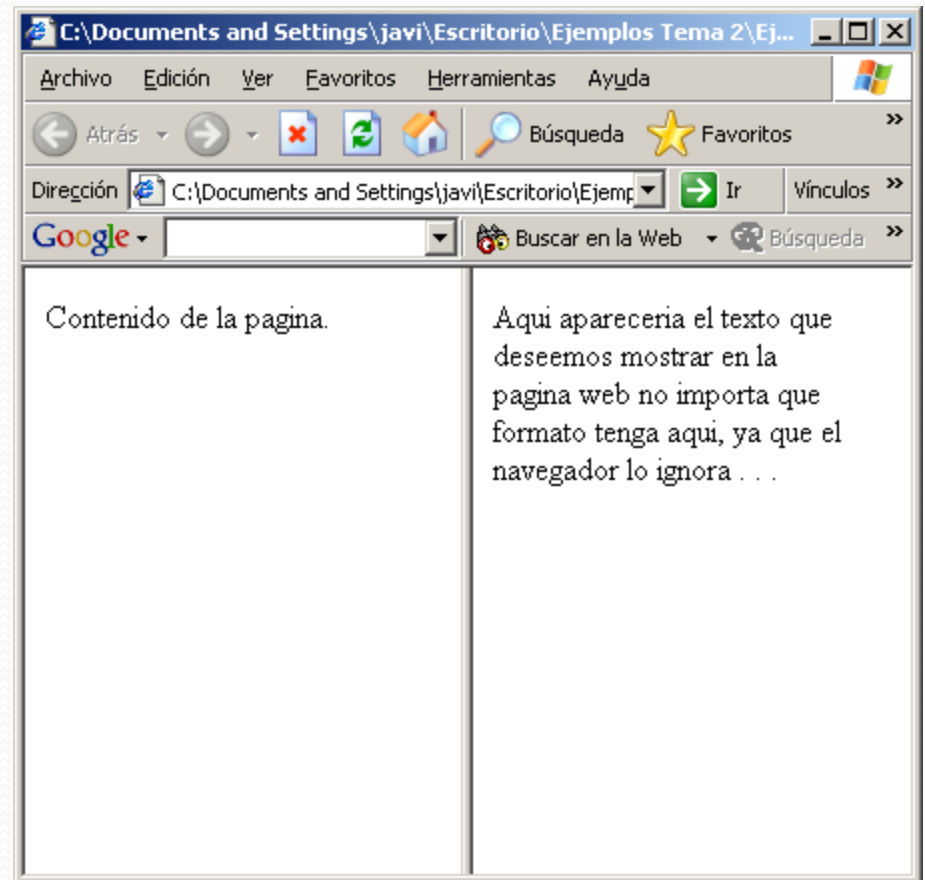


# HTML 4.0

```
<html>
<!-- Ejemplo 18 -->

<frameset cols="*,*">
  <frame src="ejemplo1.html">
  <frame src="ejemplo2.html">
</frameset>

</html>
```



# HTML 4.0

```
<html>
<!-- Ejemplo 19 -->

<frameset rows="35%,*">
  <frameset cols="*,*" frameborder=0 >
    <frame src="ejemplo1.html">
    <frame src="ejemplo2.html">
  </frameset>
  <frame src="ejemplo3.html">
</frameset>

</html>
```

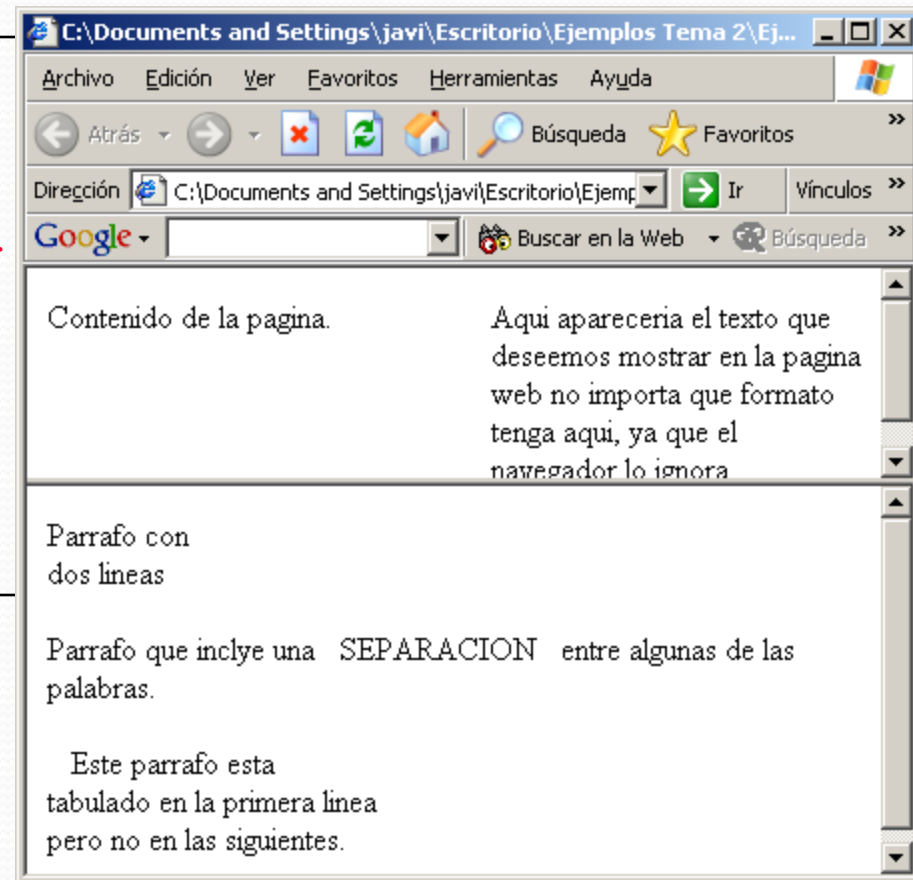
- Es posible usar cuadros anidados, si aparece o no la barra de separación, la barra de *scroll*, etc.
- El uso de los cuadros es una característica controvertida en HTML.

# HTML 4.0

```
<html>
<!-- Ejemplo 19 -->

<frameset rows="35%,*">
  <frameset cols="*,*" frameborder=0 >
    <frame src="ejemplo1.html">
    <frame src="ejemplo2.html">
  </frameset>
  <frame src="ejemplo3.html">
</frameset>

</html>
```



# HTML 4.0

- Mejoras en tablas y formularios.
- Cambios significativos a nivel de parámetros: todos aquellos que hacían referencia a la “presentación” se eliminan a favor de las hojas de estilo.
- Problema de los cuadros (*frames*). No es recomendable usarlos

# HTML 4.0

- Varias versiones adicionales de HTML 4.01 (cada una con su correspondiente DTD):
  - **Estricto**: describe estructuras del HTML, pero no proporciona marcado relacionado con *frames*.
  - **En transición**: incluye elementos de marcado para formato (como `<B>` o `<FONT>`), pero sin elementos relacionados con *frames*.
  - **Trabajo con cuadros**: como el DTD de transición pero con los elementos relacionados con *frames*.