

Introducción a HTML

Introducción a las App. Web

La idea fundamental de los navegadores, es que muestren documentos escritos en HTML que han obtenido de un servidor Web. Estos documentos HTML habitualmente presentan información de forma estática, sin más posibilidad de interacción con ellos.

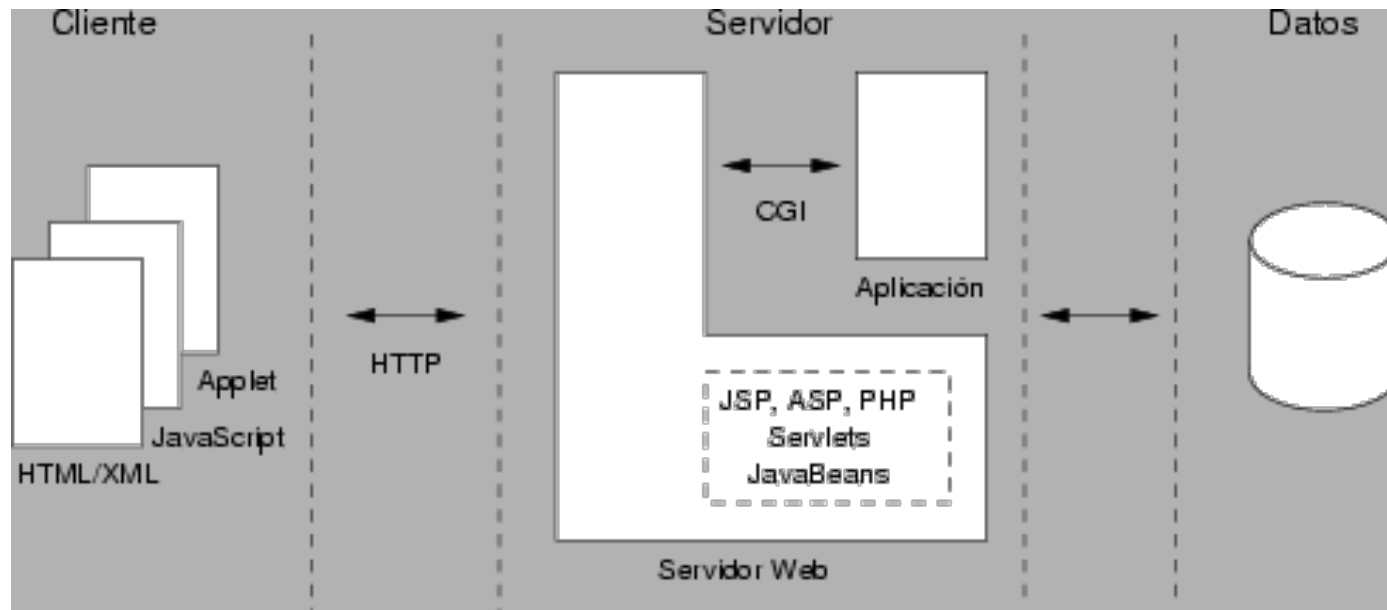
El modo de crear los documentos HTML ha variado a lo largo de la corta vida de las tecnologías Web pasando desde las primeras páginas escritas en HTML almacenadas en un archivo en el servidor Web hasta aquellas que se generan en tiempo de ejecución como respuesta a una acción del cliente y cuyo contenido varía según las circunstancias.

Introducción a las App. Web

Además, el modo de generar páginas dinámicas ha evolucionado, desde la utilización del CGI ,*Common Gateway Interface*, hasta los *servlets* pasando por tecnologías tipo *JavaServer Pages*. Todas estas tecnologías se encuadran dentro de aquellas conocidas como *Server Side*, ya que se ejecutan en el servidor web.

Otro aspecto que completa (¿**complica**?) el panorama son las inclusiones del lado del cliente, *Client Side*, que se refieren a las posibilidades de que las páginas lleven incrustado código que se ejecuta en el cliente, como por ejemplo JavaScript.

Introducción a las App. Web



HTTP

Aunque entender el modo en que funciona HTTP no es estrictamente necesario para desarrollar aplicaciones Web, algunas nociones sobre lo que esconden esas siglas puede ayudar a desarrollarlas con más facilidad y confianza.

HTTP es un protocolo del nivel de aplicación para el intercambio de información. Es un protocolo no orientado a estado que puede ser utilizado para más propósitos que para manejar archivos HTML.

HTTP

Entre las propiedades de HTTP se pueden destacar las siguientes:

- Un esquema de direccionamiento comprensible.

Utiliza el Universal Resource Identifier (URI) para localizar sitios (URL) o nombres (URN) sobre los que hay que aplicar un método. La forma general de un URL es `servicio://host/archivo.ext`.

- Arquitectura Cliente-Servidor.

HTTP se asienta en el paradigma solicitud/respuesta. La comunicación se asienta sobre TCP/IP. El puerto por defecto es el 80, pero se pueden utilizar otros. (Socket - ServerSocket)

- Es un protocolo sin conexión y sin estado.

Después de que el servidor ha respondido la petición del cliente, se rompe la conexión entre ambos. Además no se guarda memoria del contexto de la conexión para siguientes conexiones.

HTTP

- Está abierto a nuevos tipos de datos.

HTTP utiliza tipos MIME (Multipart Internet Mail Extension) para la determinación del tipo de los datos que transporta. Cuando un servidor HTTP transmite información de vuelta a un cliente, incluye una cabecera que le indica al cliente sobre los tipos de datos que componen el documento. De la gestión de esos datos se encargan las utilidades que tenga el cliente (imágenes, vídeo, etc.)

Una transacción HTTP está compuesta por una cabecera, y opcionalmente, por una línea en blanco seguida de los datos. En la cabecera se especifica tanto la acción solicitada en el servidor, como los tipos de datos devueltos o un código de estado.

Interacción entre el Servidor y el Cliente

Durante la comunicación entre el cliente y el servidor HTTP en el que el cliente solicita el documento doc1.html al servidor se intercambian la siguiente transacción HTTP:

```
GET /doc1.html HTTP/1.1
```

```
Accept: www/source
```

```
Accept: text/html
```

```
Accept: image/gif
```

```
User-Agent: firefox 1.02
```

```
From:200.45.124.200
```

El método GET indica el archivo que el cliente solicita y la versión de HTTP. El cliente también muestra una lista de los tipos MIME que puede aceptar como retorno, además de identificar el *browser* que utiliza (para que el servidor pueda optimizar la salida para el tipo particular de navegador) y su dirección.

Interacción entre el Servidor y el Cliente

El servidor responde mandando la siguiente transacción HTTP:

```
HTTP/1.0 200 OK  
Date: Friday, 23-Feb-01 16:30:00 GMT  
Server: Apache/1.1.1  
Content-type: text/html  
Content-length: 230
```

```
<HTML><HEAD><TITLE> ..... </HTML>
```

En este mensaje el servidor utiliza la versión 1.1 de HTTP, y manda el código de estado 200 para indicar que la petición del cliente ha sido procesada satisfactoriamente. También se identifica como un servidor Apache. Indica al cliente que el contenido del documento es texto en formato HTML y que tiene una longitud de 230 bytes.

Métodos de petición (GET)

La primera línea de una petición contiene los comandos HTTP, conocidos como métodos. Existen varios, pero los más conocidos y utilizados son tres: GET y POST.

El método GET se utiliza para recuperar información identificada por un URI por parte de los navegadores. El método GET se utiliza para pasar una pequeña cantidad de información al servidor en forma de pares atributo-valor añadidos al final del URI detrás de un símbolo de interrogación, ?.

GET /cgi/saludar.pl?nombre=pepe&email=pepe@xxfoxx.xx.ar HTTP/1.1

La longitud de la petición GET está limitada por el espacio libre en los *buffers* de entrada (ej 256 bytes). Por lo que para mandar una gran cantidad de información al servidor ha de utilizarse el método POST.

Métodos de petición (POST)

El método POST se refiere normalmente a la invocación de procesos que generan datos que serán devueltos como respuesta a la petición. Además se utiliza para aportar datos de entrada a esos programas. En este caso los pares atributo-valor son incluidos en el cuerpo de la petición.

```
POST /pedidos/alta.jsp HTTP/1.1  
Accept: */*
```

```
nombre=pepe  
email=pepe@xxxx.xx.ar
```

De este modo el método POST no sufre de las limitaciones de espacio y puede enviar mucha más información al servidor.

Repaso general HTML

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Mi primera pagina</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<CENTER><H1>Mi Primera pagina</H1></CENTER>
```

```
<P>Esta es mi primera pagina. Por el  
momento no se que tendra, pero dentro de poco  
pondre aqui muchas cosas interesantes.</P>
```

```
</BODY>
```

```
</HTML>
```

Etiquetas de uso general

Formateo de párrafo

<P>....</P>

Sirve para delimitar un párrafo. Inserta una línea en blanco antes del texto.

<CENTER> ... </CENTER>

Permite centrar todo el texto del párrafo.

<DIV ALIGN='LEFT'> ... </DIV>

Permite justificar el texto del párrafo a la izquierda (ALIGN='LEFT'), derecha (RIGHT), al centro (CENTER) o a ambos márgenes (JUSTIFY)

Etiquetas de uso general

Cambiando el tipo de letra

<code> ... </code>	Pone el texto en negrita.
<code><I> ... </I></code>	Representa el texto en cursiva.
<code><U> ... </U></code>	Para subrayar algo.

Otros elementos comunes

<code><HR></code>	Inserta una barra horizontal.
<code>
</code>	Salto de línea.
<code><!-- ... --></code>	Comentarios.

Links

`Noticias`

Etiquetas de uso general

Imágenes

```
<IMG SRC='archivo_grafico' ALT="descripcion">
```

```
<IMG SRC=../graficos/dwnldns.gif" ALT="Ejemplo" WIDTH='88' HEIGHT='31'>
```

Imagen con link

```
<A HREF='http://www.google.com'><IMG SRC='/img/logo.jpg' BORDER='0'></A>
```

Etiquetas de uso general

Tipo de letras, tamaño y color

Color

`Estoy en rojo`

`Tengo Color`

Tamaño

`Tamaño 2`

Tipo

`No sé como voy a salir exactamente`

Etiquetas de uso general

Tablas

```
<TABLE>  
  <TR>  
    <TD>1,1</TD>  
    <TD>1,2</TD>  
    <TD>1,3</TD>  
  </TR>  
  <TR>  
    <TD>2,1</TD>  
    <TD>2,2</TD>  
    <TD>2,3</TD>  
  </TR>  
</TABLE>
```

1,1	1,2	1,3
2,1	2,2	2,3

Etiquetas de uso general

<TABLE BORDER='1' WIDTH='50%' ALIGN='CENTER'>

BORDER: Especifica el grosor del borde que se dibujará alrededor de las celdas. Por defecto es cero, lo que significa que no dibujará borde alguno.

CELLSPACING: Define el número de pixels que separarán las celdas.

CELLPADDING: Especifica el número de pixels que habrá entre el borde de una celda y su contenido.

WIDTH: Especifica la anchura de la tabla. Puede estar tanto en pixels como en porcentaje de la anchura total disponible para él (pondremos "100%" si queremos que ocupe todo el ancho de la ventana del navegador).

ALIGN: Alinea la tabla a izquierda (LEFT), derecha (RIGHT) o centro (CENTER).

Etiquetas de uso general

```
<TABLE BORDER='1'>
  <TR>
    <TD COLSPAN=2>1,1 y 1,2</TD>
    <TD>1,3</TD>
  </TR>
  <TR>
    <TD ROWSPAN=2>2,1 y 3,1</TD>
    <TD>2,2</TD>
    <TD>2,3</TD>
  </TR>
  <TR>
    <TD>3,2</TD>
    <TD>3,3</TD>
  </TR>
</TABLE>
```

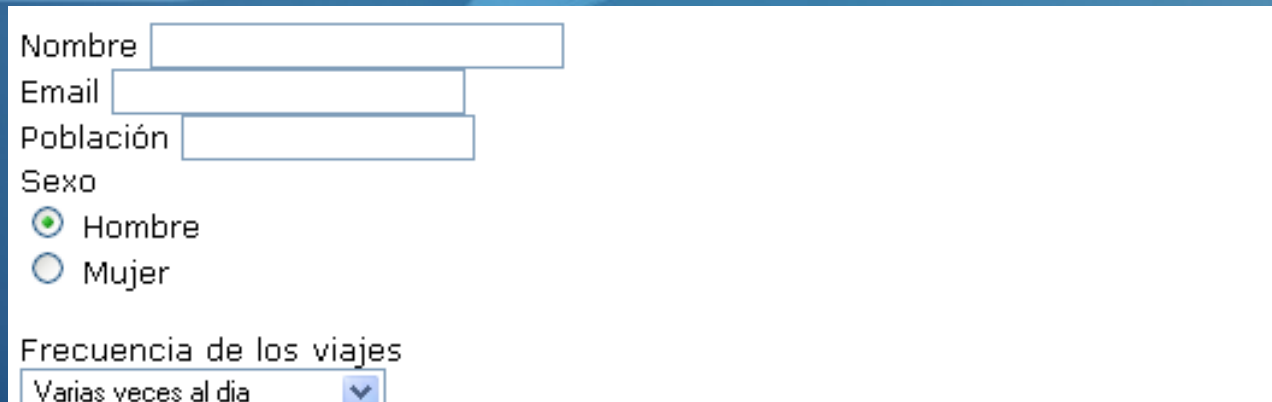
1,1 y 1,2		1,3
2,1 y 3,1	2,2	2,3
	3,2	3,3

HTML– Introducción

Formularios

Hasta ahora hemos visto la forma en la que el HTML gestiona y muestra la información, esencialmente mediante texto, imágenes y enlaces. Nos queda por ver de qué forma podemos intercambiar información entre el servidor y el cliente.

Los formularios son esas famosas cajas de texto y botones que podemos encontrar en muchas páginas web. Mediante los mismo podemos permitir al usuario cargar datos para luego estos sean enviados al servidor y sean procesados o almacenados en una base de datos.



Nombre

Email

Población

Sexo

☒ Hombre

☐ Mujer

Frecuencia de los viajes

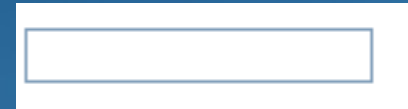
HTML - Formularios

Texto corto

Las cajas de texto son colocadas por medio de la etiqueta `<input>`. Dentro de esta etiqueta hemos de especificar el valor de dos atributos: `type` y `name`.

La etiqueta es de la siguiente forma:

```
<input type="text" name="nombre">
```



De este modo expresamos nuestro deseo de crear una caja de texto cuyo contenido será llamado `nombre` (por ejemplo).

El nombre del elemento del formulario es de gran importancia para poder identificarlo en nuestro programa de procesamiento. Por otra parte, es importante indicar el atributo `type`, ya que, como veremos, existen otras modalidades de formulario que usan esta misma etiqueta.

El empleo de estas cajas esta fundamentalmente destinado a la toma de datos breves: palabras o conjuntos de palabras de longitud relativamente corta.

HTML - Formularios

Además de estos dos atributos, esenciales para el correcto funcionamiento de nuestra etiqueta, existen otra serie de atributos que pueden resultarnos de utilidad pero que no son imprescindibles:

Size: Define el tamaño de la caja en número de caracteres.

Maxlength: Indica el tamaño máximo del texto que puede ser tomado por el formulario.

Value: En algunos casos puede resultarnos interesante asignar un valor definido al campo en cuestión. Esto puede ayudar al usuario a rellenar más rápidamente el formulario o darle alguna idea sobre la naturaleza de datos que se requieren. Este valor inicial del campo puede ser expresado mediante el atributo value.


```
<input type="text" name="nombre" value="Perico Juan">
```

HTTP - Formularios

Texto oculto

Podemos esconder el texto escrito por medio asteriscos de manera a aportar una cierta confidencialidad. Este tipo de campos son análogos a los de texto con una sola diferencia: remplazamos el atributo `type="text"` por `type="password"`:

```
<input type="password" name="nombre">
```



HTTP - Formularios

Texto largo

Si deseamos poner a la disposición de usuario un campo de texto donde pueda escribir cómodamente sobre un espacio compuesto de varias líneas, hemos de invocar una nueva etiqueta: `<textarea>`.

Este tipo de campos son prácticos cuando el contenido a enviar no es un nombre teléfono o cualquier otro dato breve, sino más bien, un comentario, opinión, etc. Dentro de la etiqueta `textarea` deberemos indicar, el atributo `name`. Además, podemos definir las dimensiones del campo a partir de los atributos siguientes:



```
Escribe tu comentario....
```

HTTP - Formularios

Texto largo

Si deseamos poner a la disposición de usuario un campo de texto donde pueda escribir cómodamente sobre un espacio compuesto de varias líneas, hemos de invocar una nueva etiqueta: `<textarea>`.

Este tipo de campos son prácticos cuando el contenido a enviar no es un nombre teléfono o cualquier otro dato breve, sino más bien, un comentario, opinión, etc. Dentro de la etiqueta `textarea` deberemos indicar, el atributo `name`. Además, podemos definir las dimensiones del campo a partir de los atributos siguientes:

Rows: Define el número de líneas del campo de texto.

Cols: Define el número de columnas del campo de texto.

La etiqueta queda por tanto de esta forma:

```
<textarea name="comentario" rows="10" cols="40">comentario..</textarea>
```

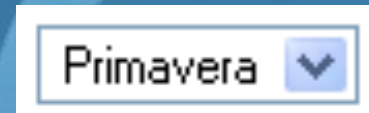
HTML - Formularios

Listas de opciones

Las listas de opciones son ese tipo de menús desplegables que nos permiten elegir una (o varias) de las múltiples opciones que nos proponen. Para construirlas emplearemos una etiqueta con su respectivo cierre: `<select>`

Dentro de esta etiqueta definiremos su nombre por medio del atributo `name`. Cada opción será incluida en una línea precedida de la etiqueta `<option>`. Podemos ver, a partir de estas directivas, la forma más típica y sencilla de esta etiqueta:

```
<select name="estacion">  
  <option>Primavera</option>  
  <option>Verano</option>  
  <option>Otoño</option>  
  <option>Invierno</option>  
</select>
```



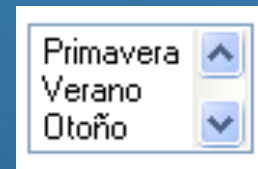
HTML - Formularios

Esta estructura puede verse modificada principalmente a partir de otros dos atributos:

Size: Indica el número de valores mostrados de la lista. El resto pueden ser vistos por medio de la barra lateral de desplazamiento.

Multiple: Permite la selección de más varios elementos de la lista. La elección de más de un elemento se hace como con el explorador de Windows, a partir de las teclas ctrl o shift. Este atributo se expresa sin valor alguno, es decir, no se utiliza con el igual: simplemente se pone para conseguir el efecto, o no se pone si queremos una lista desplegable común.

```
<select name="estacion" multiple>  
  <option>Primavera</option>  
  <option>Verano</option>  
  <option>Otoño</option>  
  <option>Invierno</option>  
</select>
```



HTML - Formularios

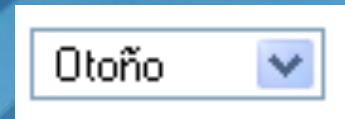
La etiqueta `<option>` puede asimismo seleccionada por medio de otros atributos

selected: Del mismo modo que `multiple`, este atributo no toma ningún valor sino que simplemente indica que la opción que lo presenta esta elegida por defecto.

Así, si cambiamos la línea del código anterior:

`<option>Otoño</option>`

por:



```
<select name="estacion" multiple>  
  <option>Primavera</option>  
  <option>Verano</option>  
  <option selected>Otoño</option>  
  <option>Invierno</option>  
</select>
```

HTML - Formularios

Value: Define el valor de la opción que será enviado al programa si el usuario elige esa opción. Este atributo puede resultar muy útil si el formulario es enviado a un programa puesto que a cada opción se le puede asociar un número o letra, lo cual es más fácilmente manipulable que una palabra o texto.

```
<option value="1">Primavera</option>
```

De este modo, si el usuario elige primavera, lo que le llegara al programa es una variable llamada estacion que tendrá como valor 1.

```
estacion=1
```

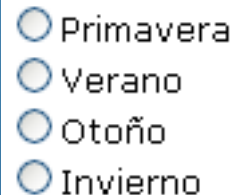
HTML - Formularios

Botones de radio

Existe otra alternativa para plantear una elección, en este caso, obligamos al usuario a elegir únicamente una de las opciones que se le proponen.

La etiqueta empleada en este caso es `<input>` en la cual tendremos el atributo `type` ha de tomar el valor `radio`. Veamos un ejemplo:

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

- 
- A visual representation of the HTML code above, showing four radio buttons stacked vertically. Each button is an empty circle followed by its corresponding season name: Primavera, Verano, Otoño, and Invierno.
- ☐ Primavera
 - ☐ Verano
 - ☐ Otoño
 - ☐ Invierno

HTML - Formularios

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
```

HTML - Formularios

Cajas de validación

Este tipo de elementos pueden ser activados o desactivados por el usuario por un simple clic sobre la caja en cuestión. La sintaxis utilizada es muy similar a las vistas anteriormente:

```
<input type="checkbox" name="comida" value="4507">Me gusta la paella
```

☐ Me gusta la paella

HTML - Formularios

Datos ocultos

En algunos casos, aparte de los propios datos enviados por el usuario, puede resultar práctico enviar datos definidos por nosotros mismos que ayuden al programa en su procesamiento del formulario. Este tipo de datos, que no se muestran en la página pero si pueden ser detectados solicitando el código fuente.

```
<input type="hidden" name="idPedido" value="451774568">
```

Esta etiqueta, incluida dentro de nuestro formulario, enviara un dato adicional al programa encargado de la gestión del formulario. podríamos, a partir de este dato, dar a conocer al programa el origen del formulario o algún tipo de acción a llevar a cabo (una redirección por ejemplo, un id de usuario, etc).

¿Preguntas?

HTML – Formularios (agrupación)

Los formularios son definidos por medio de las etiquetas `<form>` y `</form>`. Entre estas dos etiquetas colocaremos todos los campos y botones que componen el formulario. Dentro de esta etiqueta `<form>` debemos especificar algunos atributos:

Action: Define el tipo de acción a llevar a cabo con el formulario, y hacia donde se desea enviar los datos existentes dentro del formulario. Todo lo que se encuentre dentro del tag form será enviado al servidor. Si lo que queremos es que el formulario sea procesado por una pagina jsp o un servlet, tenemos q especificar la url donde se encuentra el recurso

`<form action="/clientes/alta.jsp" ...>`

HTML - Formularios

Method: Este atributo se encarga de especificar la forma en la que el formulario es enviado. Los dos valores posibles que puede tomar este atributo son **post** y **get**.

```
<form action="/clientes/alta.jsp" method="post" >  
  Nombre:<input type="text" name="nombre"><br>  
  Edad:<input type="text" name="edad"><br>  
</form>
```

HTML - Formularios

botón de envío

Para dar por finalizado el proceso de relleno de un formulario y hacerlo llegar al servidor, el navegante ha de enviar los datos mediante botón previsto a tal efecto. La construcción de dicho botón no reviste ninguna dificultad una vez familiarizados con las etiquetas input ya vistas

```
<form action="/clientes/alta.jsp" method="post" >  
  Nombre:<input type="text" name="nombre"><br>  
  Edad:<input type="text" name="edad"><br>  
  <input type="submit" value="Enviar">  
</form>
```