

# Administrador de Menús Para Sistemas de Información Orientados a la Web

Autor Jorge Tancara Aguilar

*Universidad Católica Boliviana “San Pablo”*

*Facultad de Ingeniería y Ciencias Exactas*

*Carrera Ingeniería de Sistemas*

*La Paz – Bolivia*

2010

jtancara@ucb.edu.bo

**Resumen:** Todo “Sistema de Información Orientado a la Web” requiere de un medio de interacción mas ordenada que un “Sitio Web”, precisamente debido a que los usuarios requieren de certeza a la hora de elegir sus opciones, razón por lo que presentamos esta alternativa para administrar opciones de Sistemas de Información orientados a la Web.

## INTRODUCCIÓN

Una de las debilidades en los Sistemas de Información que funcionan en el Internet, es el control y la administración del menú de opciones, debido a que la mayoría de las aplicaciones basan sus menús en generadores o en ejemplos donde ajustan sus requerimientos con las dificultades que conlleva tener un menú poco flexible y para peor en la mayoría de los casos los generadores de menús ofrecen el código de fuente de manera “ofuscada” en cuyo caso es casi imposible hacer algún cambio.

Sin embargo no es nada complicado realizar un administrador de menús, con las ventajas de: crecer indefinidamente, darle la forma que uno desee, tener acceso diferenciado según el nivel del usuario y hasta hacerlo multi-idioma.

Precisamente este trabajo esta dirigido a demostrar la construcción de un administrador de menús jerárquico “sin limite de profundidad” y además de ser multi-idioma, lo cual significa que el usuario puede cambiar al idioma que más lo entienda; Esta basado en conceptos de la Web 2.0, es altamente dinámico, debido a que todas las acciones las realiza en el cliente dejando en libertad plena al servidor para que realice otras acciones más importantes.

## ENTORNO DE TRABAJO

Para el desarrollo de esta aplicación se utilizó las siguiente herramientas, para el lado cliente: JavaScript, CSS, JSON, “XMLHTTP” que son parte esencial de AJAX, y para

procesos en el servidor PHP (en el ejemplo no se encuentra esta parte).

## PATRON DE DESARROLLO

Se utilizo el MVC (Model View Controller), que es un patrón de diseño que separa en capas: datos, Interfaz del usuario, y la lógica de control del producto, cualquier aplicación para lograr un menor acoplamiento entre el código. Es una solución que sirve a todo tipo de aplicación, no solo las de tipo ricas de Internet.

A continuación vemos un diagrama sencillo que muestra la relación entre el modelo, la vista y el controlador. Nota: las líneas sólidas indican una asociación directa, y las punteadas una indirecta.

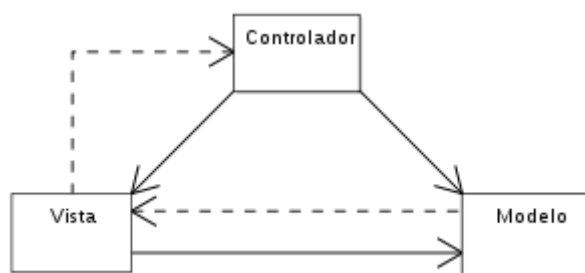


Fig. 1 Relación Modelo Vista Controlador

## CONCEPTOS

### PROCESOS EN EL CLIENTE

Son aquellas que se ejecutan en el cliente, obviamente luego de ser descargado desde algún sitio Web, se caracterizan por

ser muy veloces ya que todos sus procesos los realiza en la misma maquina del cliente, invocando solamente servicios del servidor cuando se requiere algún dato y este proceso lo efectúa en segundo plano.

PROCESOS EN EL SERVIDOR

Se caracterizan por que son efectuados en el mismo Servidor, por ejemplo “triguers”, o funciones incorporadas en las bases de datos, sin embargo también se puede realizar es tos procesos mediante lenguajes de Servidor tales como: PHP, JAVA o ASP. Si desea construir una página veloz y altamente dinámica le recomiendo no utilizar estos lenguajes para hacer proceso en el cliente, ya que por cualquier proceso trivial acudirá al servidor (esto no se nota cuando esta trabajando en “localhost”).

TRANSACCIONES EN SEGUNDO PLANO

Este concepto apareció por la necesidad de “Evitar la constante recarga de paginas”, lo cual es un perjuicio tanto para el usuario como para el servidor. Es por eso que se debe manejar adecuadamente los criterios de: Procesos en el cliente y Procesos en el servidor.  
Concretamente una transacción en segundo plano es realizado por el objeto : “XMLHTTP” que es parte esencial del “AJAX”, cuya función es realizar una transacción “asincrona”, lo cual significa que el cliente luego de solicitar datos al servidor, puede continuar su trabajo y los datos le llegará “sin que se de cuenta”, apareciendo en al algún lugar de la pantalla sin necesidad de recara de página.

ARQUITECTURA DEL MENÚ

Como se mencionó en el título, el menú que genera la aplicación es jerárquico y no tiene límite en los niveles de profundidad como podemos verlo gráficamente:

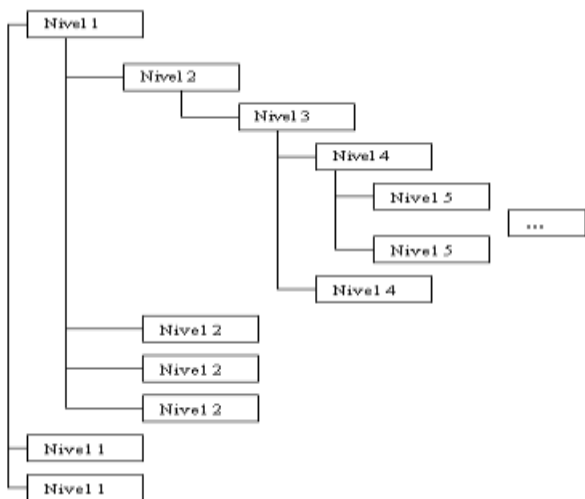


Fig. 2 Arquitectura del menú de opciones

ESTRUCTURAS DE ALMACENAMIENTO DEL MENÚ

En el ejemplo, las opciones se encuentran en vectores, sin embargo en aplicaciones seguras, estas deben estar almacenadas en la base de datos y mediante transacciones en segundo plano bajar a los vectores.

Los vectores utilizados en el programa son los siguientes:

- Voping: Contiene los mensajes en ingles.
- Vop: Contiene los mensajes en español
- Vniv: Contiene el nivel de profundidad de la opción
- Vtam: Indica la cantidad de hijos que tiene una opción, si es cero carga un URL
- Vini: Indica el lugar donde empiezan los hijos de una opción.
- Vurl: Contiene las URL’s que serán cargados según la opción
- Ace: Flag, que indica si una opción esta abierta o no
- Vus: Contiene las opciones del menú según el tipo de usuario (En el ejemplo esta abierto para manejar todas las opciones)

DISPOSICIÓN DE LOS DATOS DENTRO LOS VECTORES:

Ind	VopIng	Vop	Vniv	Vtam	Vini	Vurl
0				4	1	
1	File	Archivo	1	4	5	
2	Manage	Gestion	1	2	9	
3	Reprots	Reportes	1	3	11	
4	Out	Salir	1	0		Salir.html
5	Producto	Producto	2	0		Prod.html
6	Suplier	Proveedor	2	0		Prove.html
7	Storage	Almacen	2	0		Almacen.html
8	Inv.Inicial	Inv. Inicial	2	0		Inveini.html
9	Buys	Compras	2	0		Compra.html
10	Sales	Ventas	2	0		Venta.html
11	Lists	Listados	2	0		Lista.html
12	Estadistics	Estadisticas	2	2	14	
13	Executive	Ejecutivo	2	0		Liseje.html
14	Grid	Cuadros	3	3	16	
15	Grafics	Graficos	3	4	19	
16	Buys	Compras	4	0		Liscompra.html
17	Sales	Ventas	4	0		Lisventa.html
18	Exists	Existencia	4	0		Existe.html
19	Pie	Tortas	4	0		Torta.html
20	Line	Linea	4	0		Linea.html
21	Area	Area	4	0		Area.html
22	Bar	Barra	4	0		Barra.html

Fig. 1 Disposición de las opciones del menú dentro los vectores

DESCRIPCIÓN DE LA DISPOSICIÓN DE DATOS

En la Fig. 3, se ve una disposición matricial, que en realidad sería una tabla de una base de datos que luego bajarán en los

vectores como se muestra en el código fuente. Respecto a como intervienen en la generación del menú, podemos decir que los datos de las posiciones cero indican en que fila empieza el menú y por cuantos elementos esta formado, en el ejemplo nos dice que las opciones empiezan en la fila uno vini[0] y que tiene 4 opciones vtam[0]. Luego todas las opciones con nivel "1" son menús principales de quienes dependen las opciones de nivel "2" y así sucesivamente. En caso de tener valor "0" vtam[i] esto significa que invocara a un "url" que es un archivo HTML, es caso de que el valor sea mayor a cero, de inmediato genera un submenú con el criterio de un acordeón.

## COMO AGREGAR UNA NUEVA OPCIÓN

Dependiendo del lugar donde quiera ingresar, por ejemplo en caso de ser opción de menú principal, debe ubicarse luego de la fila 4 en insertar una nueva posición obviamente, mediante un algoritmo o manualmente debe redistribuir los punteros de inicio de los submenús.

## LIBRERIAS UTILIZADAS EN LA PROGRAMACIÓN

Se esta utilizando funciones de animación del "framework JQuery.js", (esto se lo puede realizar fácilmente mediante un timer), todo lo demás se encuentra en la capa de la lógica "libreJTA.js", que controla toda la gestión del menú de opciones. También se incluya una hoja de estilos que contiene la configuración de la presentación, esta se encuentra en el archivo: "hoja.css".

## CODIGO FUENTE DEL PROGRAMA:

La programación del proyecto esta desarrollado bajo los principios del patrón de programación MVC, dividido en capas.

## CAPA DE ENTRADA:

Esta formado por un archivo HTML preparado para ejecutarse en cualquier navegador.

main.html

```
<html>
<head>
  <title>Administrador de menu By Jorge Tancara
  Aguilar</title>
  <script type="text/javascript" src="jquery.js"></script>
  <script type="text/javascript"
src="libreJTA.js"></script>
  <style type="text/css">@import "hoja.css";</style>
```

```
</head>
```

```
<body>
```

```
  <div id="boton"></div>
```

```
  <div id="cara1"></div>
```

```
  <iframe id="idhuesped" name="huesped" width="0"
height="0" scrolling="no" frameborder="0" src=""
style="border: 0px solid #123456;"></iframe>
```

```
  <div id="idIng">Ing</div>
```

```
  <div id="idEsp">Esp</div>
```

```
</body>
```

```
</html>
```

## CAPA DE LA ARQUITECTURA (hoja de estilos)

Las hojas de estilo permiten configurar de forma genérica las características visuales de una aplicación orientado a la Web, en el ejemplo práctico tenemos toda esta descripción en el archivo: "hoja.css", en la parte de abajo vemos un pequeño ejemplo de notación del mismo.

```
body
{
  background:#152;
}
```

```
li
{
  position:relative;
  text-decoration:none;
  list-style-type:none;
  display:block;
}
```

CAPA DE LA LOGICA

Esta desarrollado en javaScript , para que ejecute cualquier cambio en el cliente de manera rápida; la secuencia principal haber i cierra el menú de la siguiente manera:

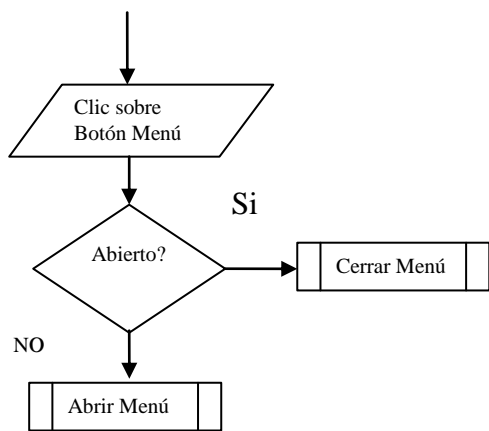


Fig. 4 Acción clic sobre el botón Menú

Una vez abierto el menú, las acciones que pueden suceder sobre una opción son las siguientes:

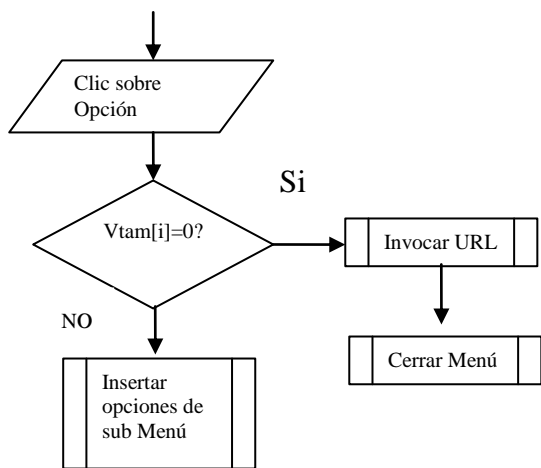


Fig. 5 Acción clic sobre una opción del menú

La lógica completa en código fuente puede verla en el archivo: “libreJTA.js”.

CARACTERISTICAS DE FUNCIONAMIENTO

Puede correr en cualquier navegador, el administrador en principio únicamente muestra un Botón en la parte superior izquierda, a partir de ella puede abrir el menú cuyo funcionamiento tiene la característica de un "acordeón" es decir cuando una opción principal se abre automáticamente cerrara la que estuviera abierta, de esta manera puede administrar un sistema. También en tiempo real puede cambiar el idioma de las opciones de menú haciendo clic sobre las etiquetas: “Esp” para ver en español y en “Ing” para ver en Ingles.

FORMA DEL MENU EN FUNCIONAMIENTO

Pantalla inicial del administrador:

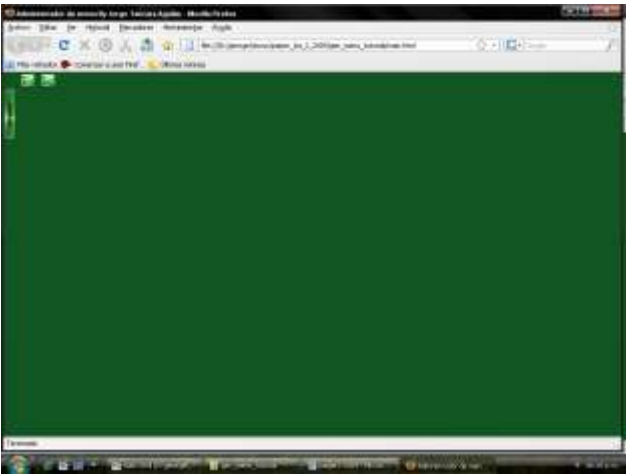


Fig. 6 Pantalla inicial del menú

Pantalla con el menú abierto

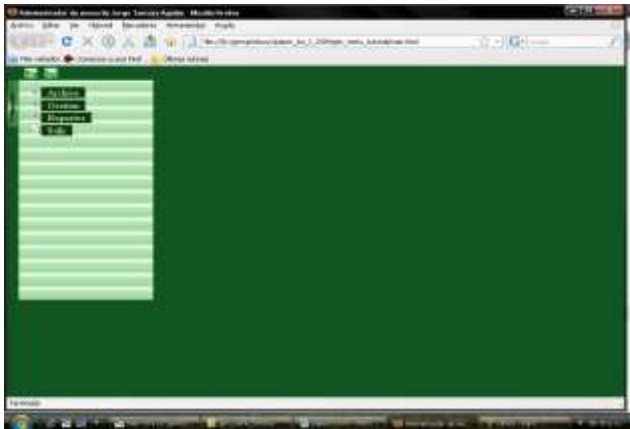


Fig. 7 Pantalla luego de hacer clic sobre el botón Menú

## DEMOSTRACION

Para ver en funcionamiento la aplicación, ingrese al sitio:  
[www.jorge-tancara.com](http://www.jorge-tancara.com)

Para descargar la aplicación, ingrese al sitio:  
[www.jtamenugen.com](http://www.jtamenugen.com)

## CONCLUSIONES

Si bien existen “FrameWorks” que facilitan la elaboración de menús para Sitios Web y Sistemas de Información Web, no siempre se ajustan a los requerimientos “complejos” de los usuarios, este pequeño trabajo demuestra que es posible también crearse “Frameworks” propios, que hacen más personalizados y fundamentalmente más seguras las aplicaciones Web.

## REFERENCIAS

- [1] Maximiliano Fritman , Ajax Web 2.0 para profesionales, Buenos Aires: Alfaomega Grupo editor Argentino, 2008, p. 421
- [2] Javier Eguíluz Pérez, Introducción a JavaScript, 2008, p.140, [www.librosweb.es](http://www.librosweb.es)
- [3] Javier Eguíluz Pérez, Introducción a Ajax, 2008, p.255, [www.librosweb.es](http://www.librosweb.es)
- [4] Javier Eguíluz Pérez, CSS Avanzado, 2008, p. 155 [www.librosweb.es](http://www.librosweb.es)