

Toc²: Manual del Programador

Jorge Hortelano Otero.

17 de febrero de 2006

1. Introducción.

Este documento tiene como finalidad ayudar a un programador a ampliar la arquitectura Toc-Toc. Pretende ser una guía para aquel que quiera continuar el desarrollo del mismo, explicando detalladamente que partes de la arquitectura se encargan de cada función del sistema.

2. Estructura general del sistema.

El sistema esta dividido en tres partes: Un conjunto de *scripts* que controlan el funcionamiento del servidor, una interfaz visual generada mediante distintos lenguajes de desarrollo web y un sistema de almacenamiento de datos, compuestos por dos bases de datos.

2.1. Estructura de los *scripts*.

Aunque no es necesario, los distintos *scripts* y archivos que genera el sistema están dividido en diversas carpetas para una mayor claridad. Estos *scripts* se encargan del funcionamiento interno del servidor y del control de los usuarios. Están divididos por defecto en:

- TocToc_*scripts*: contiene los *scripts* principales del sistema, junto con el archivo *starstop.txt* que indica si el sistema está o no activo.
- TocToc_*log*: contiene el *log* del sistema y también la copia de aquellas entradas anteriores a dos meses.
- TocToc_*clientes*: aquí el sistema genera los archivos de control de cada uno de los clientes, indicando en cada caso si están o no activos en el sistema y han salido a Internet.

2.2. Estructura de la interfaz web.

La interfaz web esta contenida en la carpeta que sirve el servidor Apache. A esta carpeta la llamaremos carpeta raíz de la interfaz web. La jerarquía de los ficheros de esta estructura queda representada en la figura 1.

Como se puede ver en dicha figura, se distribuye de la siguiente forma:

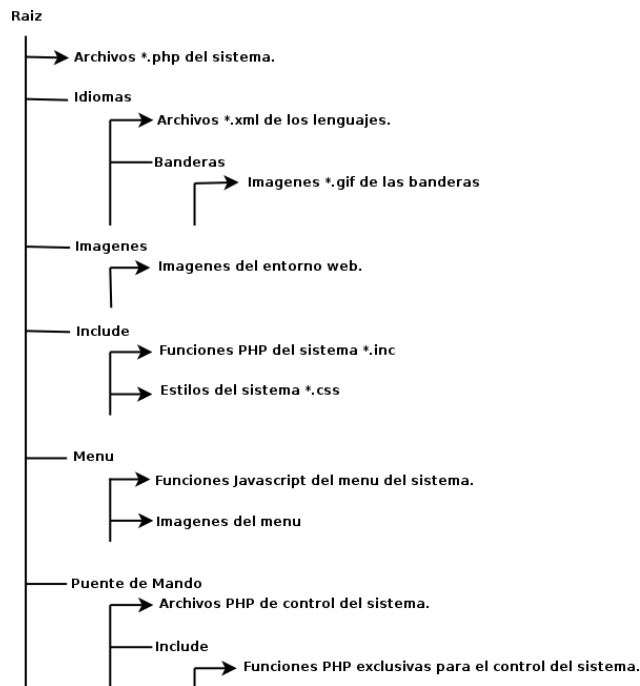


Figura 1: Estructura de la interfaz web.

1. Los archivos principales de la web están depositados en esta carpeta raíz directamente. Como en todas las webs, el archivo inicial será aquel llamado *index.php*.
2. La carpeta “Idiomas” contiene todos los archivos utilizados en el multilingüismo. Estos archivos son los XML que contienen las diversas traducciones de la interfaz web.
 - a) La carpeta “Banderas” contiene todas las banderas que permitirán a un usuario escoger idioma.
3. “Imágenes” es la carpeta que contiene todas las imágenes del sistema relativas a su diseño. En el caso más simple, solamente contiene el logo del sistema.
4. La carpeta “Include” contiene todas las funciones PHP que utilizan el resto de archivos para mantener el control del sistema. Dentro de esta carpeta cabe destacar los siguientes archivos:
 - a) FuncionesPHP.inc: contiene todas las funciones de sesiones, estructuras de datos, etc.
 - b) FuncionesBD.inc: incluye todas las funciones de acceso a la base de datos del sistema.
 - c) FuncionesBD_USUARIO.inc: aquí están todas las funciones que acceden a la base de datos para el control de usuarios.

- d) XML.inc: incluye todas las funciones que permiten el multilingüismo y generan el XML que después se enviará al cliente.
 - e) RedWireless.inc: Todas las funciones de acceso a Internet del sistema. Incluye el registro del *log* de usuarios.
5. “Menu” es la carpeta donde se guardan todos los archivos relativos al menú que permite a un usuario elegir las opciones disponibles en el sistema. Como esta generado en DHTML, aquí se incluyen las funciones Javascript que lo componen y las imágenes que aparecerán después en él.
6. “Puente de mando” es la carpeta que contiene todos los archivos de administración del sistema. La decisión de crear una carpeta aparte para esto reside en que así se permite darle algunos permisos más restrictivos si así se desea. Esta carpeta imita la estructura global, pero limitándola a la parte de control:
- a) Los archivos PHP de esta carpeta componen las páginas que permiten al administrador modificar el sistema.
 - b) La carpeta “Include”, al igual que su homóloga en el nivel superior, contiene las funciones PHP exclusivas para esto.

2.3. Estructura de las bases de datos.

El sistema esta compuesto por dos bases de datos. La primera de ellas, llamada *toctoc_database* contiene todos los datos relativos al sistema. La segunda, llamada *toctoc_users* contiene todas las tablas de los usuarios. El motivo de la división del sistema en dos bases de datos, consiste en proporcionar una mayor seguridad a los datos y facilitar la tarea al programador para la ampliación del sistema sin riesgo a perder los usuarios dados ya de alta en el mismo. El sistema esta compuesto por dos bases de datos. La primera de ellas, llamada *toctoc_database* contiene todos los datos relativos al sistema. La segunda, llamada *toctoc_users* contiene todas las tablas de los usuarios. El motivo de la división del sistema en dos bases de datos, consiste en proporcionar una mayor seguridad a los datos y facilitar la tarea al programador para la ampliación del sistema sin riesgo a perder los usuarios dados ya de alta en el mismo. A continuación se presentan las tablas que componen ambas bases de datos con una pequeña explicación de lo que significa cada campo.

3. Ampliación del sistema.

En los puntos sucesivos, se pretende explicar que estrategia hay que seguir para ampliar el sistema.

3.1. Ampliación del menú principal.

Por menú principal se hace referencia al menú que aparece una vez un usuario entra vía web en el sistema. Este menú enumera las distintas opciones a las que puede acceder un usuario, tales como la modificación de su perfil o el acceso a Internet. La estructura de este menú fue generado por un programa especializado

Campo	Tipo	Significado
nick	VARCHAR(45)	El nombre de usuario.
pwd	VARCHAR(45)	La contraseña de acceso (encriptada).
cityticket	INTEGER	El nivel de acceso a Internet.
nombre	VARCHAR(45)	El nombre del cliente.
apellidos	VARCHAR(45)	Los apellidos del cliente.
pais	VARCHAR(45)	El país de procedencia del cliente.
email	VARCHAR(45)	Una dirección de contacto del cliente.
idioma	VARCHAR(10)	El idioma preferido de la página.
tmp_pwd	VARCHAR(45)	Contraseña temporal (encriptada).
tiempo_tmp_pwd	TIMESTAMP	Fecha de la contraseña temporal.
dispositivo	VARCHAR(15)	Clave ajena a la tabla dispositivos.
color	INTEGER	Clave ajena a la tabla colores.
resolucion	INTEGER	Clave ajena a la tabla resoluciones.
vel_conexion	INTEGER	Velocidad de descarga del cliente.
vel_subida	INTEGER	Velocidad de subida del cliente.
IP	VARCHAR(15)	IP que está usando el cliente .
hora_conexion	TIMESTAMP	Hora a la que ha iniciado la conexión.
admin	INT(1)	Los derechos de administración.
fecha_fin_conexion	TIMESTAMP	Hora de la ultima desconexión.

Cuadro 1: Tabla de clientes.

Campo	Tipo	Significado
dispositivo	VARCHAR(15)	Los dispositivos válidos en el sistema.

Cuadro 2: Tabla de dispositivos.

Campo	Tipo	Significado
dispositivo	VARCHAR(15)	Clave ajena a dispositivos.
tipo	VARCHAR(15)	El nombre de la resolución.
id	INTEGER	Un identificador autoincrementable.

Cuadro 3: Tabla de resoluciones.

Campo	Tipo	Significado
dispositivo	VARCHAR(15)	Clave ajena a dispositivos.
profundidad	VARCHAR(25)	El color aplicado.
id	INTEGER	Un identificador autoincrementable.

Cuadro 4: Tabla de colores.

en la generación de menús dinámicos en lenguaje DHTML. Sin embargo, como el sistema está basado en PHP, las entradas tienen que ser realizadas a mano.

La estructura descrita por el programa define un objeto principal que contiene

Campo	Tipo	Significado
nombre	VARCHAR(10)	Nombre del servidor
puerto	VARCHAR(5)	Puerto (si se gestiona por puertos).
direccion	VARCHAR(50)	URL
ip	VARCHAR(17)	Dirección IP del servidor.
dest_port	VARCHAR(5)	Puerto de destino del servidor.

Cuadro 5: Tabla de servidores.

Campo	Tipo	Significado
idioma	VARCHAR(15)	Nombre del idioma.
bandera	VARCHAR (15)	Dirección a la imagen.
id	INTEGER	La etiqueta XML que se usará.

Cuadro 6: Tabla de idiomas.

ne alguna información general para todas las opciones, como las imágenes que indican que existe un submenú. La primera entrada del menú debe contener, además, la información del formato y colores visuales de cada opción del menú:

- La primera línea de todas, indica el comienzo del menú con el objeto principal. Esta línea es la siguiente:

```
stm_bm(["menu4bb3", 430, "<?php echo MENU_NO_IMG_1
?>", "<?php echo MENU_IMG_2 ?>", 0, "", "", 0, 0, 250, 0,
1000, 1, 0, 0, "", "", 0], this);
```

Esta línea contiene el nombre del menú y la estructura genérica del mismo. Aquí se indican las imágenes que se usan cuando es un menú desplegable o no lo es. Las variable `MENU_IMG_2` y `MENU_IMG_2` indican que imágenes se tienen que mostrar al lado del texto (cuando está seleccionado y no seleccionado respectivamente). Ambas están definidas en la página `colores.inc`.

- La última línea debe de todo el menú debe de ser:

```
stm_em();
```

- La primera línea de los campos del menú contiene el formato. Es decir, los colores del mismo. Esta línea es:

```
stm_bp("p0", [1, 4, 0, 0, 2, 3, 0, 7, 100, "", -2, "", -2, 50, 0,
0, $COLOR1, $COLOR2, "", 3, 1, 1, $COLOR_RECUADRO_ -
MENU]);
```

El primer campo indica que es el objeto inicial del menú, del que colgarán las demás opciones del mismo. Las variables `$COLORX` y `$COLOR_RECUADRO_MENU`, contienen los colores que se utilizarán en la creación del menú. Los colores, como es el campo principal, se heredarán para el resto de opciones.

- La siguiente líneas es ya la entrada que aparecerá visualmente en la página:

```
stm_ai("p0i0", [0, $TEXTO, "", "", -1, -1, 0, $LINK, $DESTINO, "", "", "", "", 0, 0, 0, "", "", 0, 0, 0, 1, 2, $COLOR_MENU, 0, $COLOR_TEXTO_MENU, 0, "", "", 3, 3, 1, 4, "$COLOR_LINK", $COLOR_ALINK, $COLOR_VLINK, $COLOR_TEXTO_SELEC", "bold 10pt Verdana", "bold 10pt Verdana", 0, 0]);
```

Esta entrada del menú se ha llamado “*p0i0*”. Mostrará el texto, que aparezca en \$TEXTO. El campo \$LINK indica hacia donde se navegará al pulsar este enlace. El campo relleno con \$DESTINO (“_self”, “_parent”, “_blank”) indica donde se abrirá el enlace al igual que en HTML. Las variables *COLOR_X* contienen los diversos colores con los que se mostrará el menú. Los dos campos siguientes indican el tamaño de letra y el tipo. La repetición de la información es debido a que permite modificar el tipo de letra según si la opción está seleccionada o no.

- Una vez se ha definido el primer botón del menú, todos los botones que aparecen detrás de este heredan las opciones que no se definan. Por ejemplo:

```
stm_aix("p0i2", "p0i0", [0, $TEXTO, "", "", -1, -1, 0, $LINK]);
```

Significa que la siguiente opción, llamada “*p0i2*” está al mismo nivel que la “*p0i0*” (por tanto tendrán los mismos colores y tipos de letra. Aparecerá el texto \$TEXTO en el botón del menú y al pulsarlo, el navegador irá a \$LINK. Las opciones no definidas las hereda del objeto “*p0i0*”.

- Para generar un submenú, hay que ampliar los campos a rellenar añadiendo las opciones que indican que se muestre la imagen que representa la presencia de un submenú (en este caso el dibujo de una flecha).

```
stm_aix("p0i3", "p0i0", [0, $TEXTO, "", "", -1, -1, 0, $URL, $DESTINO, "", "", "", "", 0, 0, 0, "menu/arrow_rw.gif", "menu/arrow_rw.gif", 7, 7]);
```

En este caso, las variables \$URL y \$DESTINO suelen estar vacías. Inmediatamente después de esto, se incluye esta instrucción que indica que las siguientes opciones pertenecen a este submenú.

```
stm_bpx("p1", "p0", [1, 2, 0, 0, 2, 3, 0, 0]);
```

Y se van incluyendo las nuevas opciones del menú:

```
stm_aix("p1iX", "p0i0", [0, $TEXTO, "", "", -1, -1, 0, $URL]);
```

La X representa un número creciente que es el identificador de la siguiente opción.

Y se finaliza el menú:

```
stm_ep(); stm_ep();
```

De esta forma, todas las opciones incluidas dentro de las instrucciones de inicio y fin del submenú aparecerán solamente cuando el usuario selecciona la opción adecuada. Esto debe hacerse para cada submenú.

En el Ejemplo 1, se muestra la creación de un menú completo.

Algorithm 1 Ejemplo de un menú.

```
stm_bm(["menu4bb3", 430, "", "", 0, "", "", 0, 0, 250, 0, 1000, 1, 0, 0, "", "",
0], this);
stm_bp("p0", [1, 4, 0, 0, 2, 3, 0, 7, 100, "", -2, "", -2, 50, 0, 0, "#000000",
"#000000", "", 3, 1, 1, "#FFFFFF"]);

stm_ai("p0i0", [0, "Opcion1", "", "", -1, -1, 0, "opcion1.html", "_self", "", "",
"", "", 0, 0, 0, "", "", 0, 0, 0, 1, 2, "#000000", 0, "#FFFFFF", 0, "", "", 3,
3, 1, 4, "#FFFFFF", "#CCCCCC", "#FFFFFF", "#CCCCCC", "bold 10pt
Verdana", "bold 10pt Verdana", 0, 0]);

stm_aix("p0i1", "p0i0", [0, "Submenu2", "", "", -1, -1, 0, "", "", "", "", "",
0, 0, 0, "flecha.gif", "flecha.gif", 7, 7]);

stm_bpx("p1", "p0", [1, 2, 0, 0, 2, 3, 0, 0]);
stm_aix("pli1", "p0i0", [0, "Opcion21", "", "", -1, -1, 0, "opcion21.html"]);
stm_aix("pli2", "p0i0", [0, "Opcion22", "", "", -1, -1, 0, "opcion22.html"]);
stm_aix("pli3", "p0i0", [0, "Opcion23", "", "", -1, -1, 0, "opcion23.html"]);
stm_ep();

stm_ep();
stm_em();
```

3.2. Añadir un nuevo campo para los usuarios.

Si se quiere añadir alguna nueva opción que se debe de guardar en la base de datos de los usuarios, lo primero sería obtener los datos del mismo. Esto se puede hacer en las páginas *registrarse.php* y *perfil.php* que es donde un usuario rellena las opciones del mismo. Para ello hay que modificar el formulario y añadir los cambios pertinentes. Para que estos datos se guarden en la base de datos, habrá que modificar las páginas *crea_usuario.php* y *actualiza_usuario.php* y modificar las funciones PHP incluidas para obtener y guardar los campos en la base de datos.

Posiblemente sea interesante mantener esos datos en la sesión de usuario, para utilizarlos más tarde. Para ello, hay que revisar las funciones *lee_sesion* y *guarda_sesion* y añadir los campos adecuados.

3.3. Multilingüismo.

Modificación del XML. El multilingüismo esta basado en la utilización de un formato XML que permite la diferenciación de los idiomas que componen los párrafos. Este XML contiene una estructura que asocia a una etiqueta determinada el significado correspondiente a cada uno de los idiomas. Por ejemplo, en

el Algoritmo 2 se muestra el ejemplo más simple de XML aplicado al multilingüismo.

Algorithm 2 Ejemplo de una página XML multilingüe.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<toctoc>
  //Etiqueta de párrafo.
  <hola_mundo>
    //Etiquetas de idioma junto con el significado.
    <esp>Hola mundo</esp>
    <eng>Hello Word</eng>
  </hola_mundo>
</toctoc>
```

La ampliación de una página ya creada, resulta sencilla gracias a este formato. Se realiza dependiendo de como se quiera ampliar:

- Si la ampliación es un nuevo idioma, solamente hay que añadir entre dos nuevas etiquetas que lo diferencien, la traducción correspondiente.
- Si se quiere añadir un nuevo párrafo, habrá que crear una etiqueta de párrafo que lo diferencie. Dentro habrá que escribir las diferentes versiones en cada uno de los idiomas deseados.

Modificación de los archivos PHP. Los archivos de PHP se encargan de mostrar el contenido adecuado que se recoge en los archivos XML. Para ello, PHP tiene que hacer dos pasos:

- Abrir el archivo XML adecuado y obtener el texto. Esto se hace mediante:

```
include("include/XML.inc");
$fichero="idiomas/ejemplo.xml";
abreXML($fichero,$arr_vals, $arr_index);
```

La primera línea incluye las funciones que se usan más tarde. La variable *\$fichero* contiene la ruta al XML relacionado. La función *abreXML()* abre el fichero indicado y guarda el XML en forma de matriz dentro de la variable *\$arr_vals* que será utilizada más tarde.

- Imprimir el párrafo adecuado en el idioma deseado. Esto se hace con la función:

```
obtieneCampoXML($arr_vals, $PARRAFO, $IDIOMA, $SALIDA); echo $SALIDA;
```

Esta función obtiene el párrafo etiquetado por la variable *\$PARRAFO*, escogiendo el idioma *\$IDIOMA* y lo guarda en la variable *\$SALIDA*. Imprimir esta última variable mostrará el texto.

Estos pasos tendrán que repetirse para cada página PHP que contengan texto y por cada uno de los párrafos escritos.

Añadir un nuevo idioma. Para añadir un nuevo idioma, aparte de las traducciones pertinentes en cada página, hay que añadir una nueva bandera en la página principal. Cada bandera recarga la página enviando una variable que contiene el código del idioma, que es el que se utilizará a su vez en el XML para diferenciarlos.

3.4. Conexión a Internet.

La conexión a Internet viene definida dentro de la página *redWireless.inc* que contiene todas las funciones necesarias para su manejo. Primero se presenta una explicación de cada una de ellas:

- *escribe_log()*: guarda una cadena dentro del log del sistema.
- *reduce_log()*: deja en el log solamente aquellas entradas que tienen menos de un periodo de tiempo definido por la constante *TIEMPO_LOG*. El resto de entradas se guardan en la copia de seguridad del log.
- *da_internet_usuario()*: se encarga de generar las colas de TC específicas para el cliente que invoca a esta función. De esta manera se permite el acceso de este a Internet.
- *mantiene_internet_usuario()*: esta función se encarga de que el ordenador principal no dé de baja al cliente que la invoca. Debe ser llamada periódicamente o el sistema pensará que el cliente a perdido su conexión.
- *obtiene_estado_cliente()*: obtiene del sistema el estado del cliente, es decir, comprueba si el sistema da por conectado o desconectado este cliente.
- *obtiene_estado_sistema()*: indica si TocToc está encendido o apagado.
- *cierra_internet_usuario()*: cierra correctamente la conexión de un usuario.

De todas estas funciones la más importante es *da_internet_usuario()*. Esta es la función que hay que revisar si se pretende cambiar la funcionalidad de la conexión a Internet. Aquí habrá que indicar la nueva cola si se quiere usar algún otro tipo de cola o es donde se añadirá el IMQ en caso de que se quiera utilizar.

3.5. Generación de la ayuda del sistema.

La generación de la página de ayuda viene facilitada por dos funciones de PHP. Estas funciones se encargan de formar la estructura del índice de ayuda, los enlaces entre el índice y la explicación, y la propia explicación. Estas funciones:

- *ayudamenu(\$arr_vals, \$ANCLA, \$TEXTO)*: Esta función genera el menú de ayuda, mostrando el texto indicado en la página XML por la etiqueta incluida en la variable *\$TEXTO* y que llevará al párrafo que tenga el enlace *\$ANCLA*. La variable *\$arr_vals* es, como se ha explicado anteriormente, la que contiene el texto obtenido de la página XML.

- *ayudaexplicacion(\$arr_vals, \$TITULO, \$EXPLICACION, \$ANCLA)*: genera el párrafo explicativo al que se dirigirá el enlace del menú con el enlace *\$ANCLA* y que se compondrá por un título y su explicación compuestos por el texto que devuelva el XML a pasarle las variables *\$TITULO* y *\$EXPLICACION* respectivamente.

Por tanto, para cada entrada en la ayuda habrá que incluir cada una de las funciones anteriores, generando así un índice principal y una explicación que aparecerá más abajo.

3.6. Ampliación general de las páginas PHP.

Por motivos de facilidad de desarrollo futuro, cada página que se utiliza en el sistema incluyen las páginas *cabecera.inc* y *pie.inc*. La primera de ellas se importa justo al principio del documento, mientras que la segunda se incluye justo al final. La finalidad de estas dos páginas es la aplicación de un elemento común en todas las páginas de forma sencilla. Por ejemplo, en *cabecera.inc* se incluyen todos los META-TAGs que se utilizan en todas las páginas, de forma que el mantenimiento de estos resulta muy sencillo en todas las páginas. El archivo *pie.inc* está en este momento vacío y se puede utilizar para añadir elementos al final de la página, como por ejemplo, contadores de visitas y otros elementos comunes como logotipos.

3.7. Archivos de protección.

El arquitectura TocToc incluye algunas funciones de protección para que no existan accesos no autorizados en el sistema. Estas funciones están en los archivos *proteccion.inc* y *proteccion_internet.inc*. El primero de los dos archivos, contienen unas funciones que impiden el acceso a una página a aquellos navegadores que no hayan generado una sesión de PHP. Como la sesión solamente se genera si el usuario entra en el sistema introduciendo su nombre de usuario y su contraseña, solamente los usuarios dados de alta pueden tener acceso al sistema. El segundo archivo, *proteccion_internet.inc*, es similar al primero salvo que impide el acceso a ciertas páginas a los usuarios que no tienen permitida la conexión a Internet.