

MANUAL TECNICO



INDICE DEL MANUAL TECNICO:

| | |
|---|-----------|
| 1. INTRODUCCION | 2 |
| 2. INSTALACION EN SERVIDOR | 2 |
| 3. PROCESO INICIAL | 3 |
| 4. ARCHIVOS GLOBALES | 3 |
| 6. MANEJO DE ENTIDADES..... | 13 |
| 6.1 USUARIOS | 13 |
| <i>Tabla 1 - Codificación de etiquetas por tipo de usuario.....</i> | <i>13</i> |
| 6.2 ARTICULOS | 14 |
| <i>Tabla 2. Estados de Artículos.....</i> | <i>15</i> |
| 6.3 REVISTA, SECCIONES Y POLITICAS..... | 16 |
| 6.4 COMENTARIOS..... | 16 |
| 7. USO DE LA TABLA 'CONTENTS' | 16 |
| <i>Tabla 3. Estructura de la tabla 'contents'.....</i> | <i>17</i> |
| <i>Tabla 4. Etiquetas por tipo de usuario para tabla 'contents'.....</i> | <i>17</i> |
| <i>Tabla 5. Registros necesarios para que funcione el Sistema [RECs].....</i> | <i>18</i> |
| 8. MANEJO DE MENÚ E HIPERVÍNCULOS | 19 |
| 9. DICCIONARIO DE ARCHIVOS | 20 |
| 9.1 MODULO ADMINISTRADOR..... | 20 |
| 9.2 MODULO AUTOR..... | 22 |
| 9.3 MODULO EDITOR | 23 |
| 9.4 MODULO LECTOR..... | 25 |
| 10. DISEÑO: EDICION DE BANNERS | 27 |
| <i>Figura 1. Banners actuales en [RECs]</i> | <i>27</i> |
| 11. DISEÑO: EDICION DE BANNER PUBLICITARIO | 28 |
| <i>Figura 2. Banner Publicitario</i> | <i>28</i> |

1. INTRODUCCION

El propósito de este manual es dar a conocer el proceso de instalación, procesos iniciales, funciones específicas e implementación para hacer funcionar este Sistema de forma correcta.

Para ello se tratará de forma amena y concisa un repaso de algunos puntos específicos, con el fin de que el usuario pueda modificar a su gusto algunos de los valores de las variables y formularios.

Es recomendable que para seguir estos pasos sean personas con conocimientos en el área de programación, Computación.

2. INSTALACION EN SERVIDOR

Para hacer funcionar [RECs] por primera vez es necesario tomar en cuenta varios parámetros.

Primero, tener programas o gestores, software de distribución libres, como **Servidor Apache y Gestor de base de datos MySQL** instalados en el Servidor. Y antes evaluar los requisitos mínimos respecto a las características del Servidor a utilizar.

Segundo, adjunto a este documento existe una carpeta llamada "recs", es la carpeta general donde se contiene el Sistema. Esta carpeta se deberá copiar y pegar en la carpeta del Servidor Apache instalado. Buscar archivo "conexion.php" para especificar el usuario y contraseña respecto al Gestor MySQL.

Tercero, adjunto a este documento también se agrego un archivo recs.sql, el cual contiene la base de datos completa, junto a los Inserts necesarios para hacer funcionar el Sistema. Se debe de importar este archivo en el Gestor de base de datos MySQL instalado.

Seguido de estos pasos, es posible ingresar al sistema [RECs]. A continuación se presenta un usuario y contraseña con privilegios de Administrador para realizar el Proceso Inicial.

| |
|-----------------------|
| Usuario: admin |
|-----------------------|

| |
|---------------------------|
| Contraseña: admin. |
|---------------------------|

3. PROCESO INICIAL

Llamamos proceso Inicial a un conjunto de procedimientos que debe seguirse de formar jerárquica. Con el usuario administrador.

A continuación se presenta el orden a seguir:

- Edición de Usuario, en caso sea necesario.
- Creación del nuevo Sitio.
- Creación de Usuarios (Autor y Editor), desde un Usuario Administrador.
- Creación de Auditor(es).
- Creación de Revista(s).
- Creación o Edición de Sección(es).
- Creación o Edición de Políticas

4. ARCHIVOS GLOBALES

Dentro del Sistema se utilizan archivos globales, los cuales son llamados por distintos archivos dentro de los módulos. A continuación se describen cada uno de estos Archivos.

conexión.php

Dentro de este archivo se establece la conexión a la base de datos utilizada, con los respectivos parámetros de la función Conectarse para establecerla:

```
<?php
function conexion(){
if(!($link=mysql_connect("localhost","root","root"))){
echo "Error conectando al localhost";
exit();
}
if(!mysql_select_db("recs",$link)){
echo "Error al seleccionar la Base de Datos";
exit();
}
return($link);
```

```
}  
?>
```

Tomar en cuenta la modificación de este archivo de acuerdo a la configuración del gestor de MySQL instalada en el equipo a implementarse¹. (Usuario y Contraseña).

menu.php

Este archivo se utiliza junto a load.php, el cual por medio del tipo de usuario autenticado, muestra un menú de acuerdo a los privilegios establecidos en la tabla “contents” de la base de datos “recs”.²

En este archivo se especifica la función “Mi Perfil”, al momento de ingresar como Autor, Editor y Lector aparezca la función Mi Perfil, para poder editar la información del usuario.

Además hace la función de que en el menú aparezca Iniciar Sesión al momento de no estar autenticado y Cerrar Sesión cuando ya el usuario este autenticado.

Y además funciones como agregar funciones extras para cada tipo de usuario como Vistas de Artículos y Mi Perfil.

```
if (ereg (basename (__FILE__), $_SERVER['PHP_SELF'])) {  
    header ("HTTP/1.0 404 Not Found");  
    exit;  
}  
function draw_menu($link){  
    $str_menu = "";  
    if(isset($_SESSION['LOG'])){  
  
$str_menu = "SELECT * FROM contents WHERE (permissions = '[ALL]' OR  
permissions LIKE '%$_SESSION[GROUP]%' ) AND clogin = '1' AND visible = '1'  
ORDER BY orden ASC";  
  
    }  
    else{  
$str_menu = "SELECT * FROM contents WHERE permissions LIKE '%[ALL]%' AND  
visible = '1' ORDER BY orden ASC";  
    }  
    $consulta = mysql_query($str_menu,$link);  
    $echo = "<div align=\"center\">|\n";  
    while($menu = mysql_fetch_assoc($consulta)){
```

¹ Ver Proceso Inicial en este documento.

² Ver Manejo de la Tabla de base de datos ‘contents’

```

        $href = "";
        $nombre = "";
if($menu['nombre'] == "Vistas Articulos" && !isset($_SESSION['LOG'])){
$href = "javascript:popUp('index.php?content=log');";
        $nombre = "Iniciar Sesion";
        }

else{
        $href = "index.php?content=$menu[nombre]";
        $nombre = $menu['nombre'];
        }
        $echo.="<a href=\"\$href\" class=\"bot\">$nombre</a>|\n";

    }
if(isset($_SESSION['LOG'])){
if($_SESSION['GROUP'] == "[LEC]" || $_SESSION['GROUP'] == "[AUT]" ||
$_SESSION['GROUP'] == "[EDT]") {

$echo .="<a class=\"bot\" href=\"index.php?content=Nuevo
Lector&id_usuario=$_SESSION[LOG]&d1=2\">Mi Perfil</a>|";
        }
$echo .="<a href=\"index.php?content=logout\" class=\"bot\">Cerrar
Sesi&oacute;n</a>|\n";
        }
        $echo .="</div>\n";
        return $echo;
}

?>

```

load.php

Dentro de este archivo, se especifica varias funciones.

Es donde por cada tipo de usuario carga una página de inicio diferente. A continuación se presenta el código donde se especifica la página de inicio por tipo de usuario:³

```

if($_SESSION['GROUP'] == "[ADM]") $inicio = "mtorevista.php";
if($_SESSION['GROUP'] == "[AUT]") $inicio= "misaportaciones.php";
if($_SESSION['GROUP'] == "[EDT]") $inicio= "larticulosE.php";
if($_SESSION['GROUP'] == "[LEC]") $inicio= "vistalector.php";

```

Además, este es el archivo que por medio de la tabla 'contents', se gestionan los sitios de acuerdo al 'contents' y da el acceso o error.

Es donde se da la Bienvenida al usuario por medio de su Nick de la base. Y además en caso ser usuarios de tipo Editor, da la cantidad de artículos con estado pendiente.

³ Ver 9 -Diccionario de Archivos para la descripción de cada página.

```

<?php
if(isset($_SESSION['LOG'])){
    $articulos_pendientes = pendientes();
    $link_articulos = "";
    if($articulos_pendientes > 0){
$link_articulos = " | <a href=\"index.php?content=Editar Articulos\"
class=\"bot\">\n".
        "Tiene $articulos_pendientes </a><span
class=\"call\">articulo(s) pendiente(s)</span>\n";
    }
    if(!isset($_GET['content'])){
        $menu = draw_menu($LINK);

echo "<center><span class=\"call\">Bienvenido
<strong>$_SESSION[USR]</strong>$link_articulos</span></center>\n". $menu."<br
/>\n";

    }
    else{
        if($_GET['content'] != "logout"){
            if($_GET['content'] == "log"){
                $menu = draw_menu($LINK);
echo "<center><span class=\"call\">Bienvenido
<strong>$_SESSION[USR]</strong>$link_articulos</span></center>". $menu." \n";
if($_SESSION['GROUP'] == "[ADM]") $inicio = "mtorevista.php";
if($_SESSION['GROUP'] == "[AUT]") $inicio= "misaportaciones.php";
if($_SESSION['GROUP'] == "[EDT]") $inicio= "larticulosE.php";
if($_SESSION['GROUP'] == "[LEC]") $inicio= "vistalector.php"
            include("$inicio");
        }else{
$estr_load = "SELECT * FROM contents WHERE nombre = '$_GET[content]'";
$consulta = mysql_query($estr_load,$LINK);
$exist = 0;
$exist = mysql_num_rows($consulta);
if($exist == 0){
echo "<h2 style=\"color:red;\">[RECs] Error: Enlace no encontrado</h2>\n";
            }
            else {
$file = mysql_fetch_assoc($consulta);
$menu = draw_menu($LINK);
echo "<center><span class=\"call\">Bienvenido
<strong>$_SESSION[USR]</strong></span>$link_articulos</center>". $menu." \n";
include("$file[url]");
            }
        }
    }
    else{
        include('logout.php');
    }
}

```

```

    }
}
else{
$str_load = "SELECT * FROM contents WHERE nombre = '$_GET[content]'";
    $consulta = mysql_query($str_load,$LINK);
    $exist = 0;
    $exist = mysql_num_rows($consulta);
    if($exist == 0)
    {
echo "<h2 style=\"color:red;\">[RECs] Error: Enlace no encontrado</h2>\n";
    }
    else {
        $file = mysql_fetch_assoc($consulta);
        $menu = draw_menu($LINK);
        echo $menu;
        include("$file[url]");
    }
}
//echo "</td></tr></table>\n";
?>

```

validator.php

Archivo utilizado para validaciones en todos los formularios.

Las validaciones que se manejan son los campos vacíos, datos numéricos, fechas validas y correos. El fin de esta implementación es que se tengan los datos necesarios dependiendo de la información requerido en cada campo de la base de datos.

Para usar este archivo es necesario antes utilizar la siguiente sentencia:

include('validator.php');

Para el uso de cada función específica se debe hacer de la siguiente manera:

\$error .= vacio(\$_POST[campobase],"etiqueta");

Donde vacío es el nombre de la función, campobase es el nombre del campo en la base de datos, y etiqueta es el nombre que representa el campo de base en el formulario. Esta función verifica si la entrada es vacía.

**\$error .= valid_date("Fecha De Nacimiento",
\$_POST['day'],\$_POST['month'],\$_POST['year']);**

Se detalla en esta sentencia fechas validas. Sentencia utilizada en Formulario de Usuarios.

\$error .= comprobar_email(\$_POST['us_email']);

Sentencia relacionada a la comprobación de una estructura de email, en base al símbolo “@”, comprobación de dominio,

Estos son algunos ejemplos de la forma de implementación de la validación en cada campo de los formularios. A continuación se presenta el código de este archivo:

```
<?php
if (eregi (basename (__FILE__), $_SERVER['PHP_SELF'])) {
    header ("HTTP/1.0 404 Not Found");
    exit;
}

function vacio($value,$nom_campo){
    $error = "";
    if(empty($value)){
        $error = "<span class=\"forWrong\">El campo
<strong>\">$nom_campo\"</strong> no debe estar vacio</span><br />\n";
    }
    return $error;
}

function numerico($value,$nom_campo){
    $error = "";
    if(!is_numeric($value)){
        $error = "<span class=\"forWrong\">El campo
<strong>\ \"$nom_campo\"</strong> debe ser num&eacute;rico</span><br />\n";
    }
    return $error;
}

function valid_date ($label,$day,$month,$year) {
    $error="";
    $dif_annio = date('Y') - $year;
    if($dif_annio < 18) $error .= "<span class=\"forWrong\">Deber mayor de 18
&ntilde;os</span><br />\n";
    if (! is_numeric($year) ) $year = 0;
    if (! checkdate($month,$day,$year)) $error .= "<span class=\"forWrong\">El
valor para el campo <strong>$label</strong> no son validos.</span><br />\n";
    return $error;
}

function valid_pass($pass,$confirm){
    $error = "";
    if($pass != $confirm){
        $error = "<span class=\"forWrong\">Las contrase&ntilde;as deben
ser iguales</span><br />\n";
    }
    return $error;
}
}
```

```

function in_db($campo,$table,$valor,$link,$label){
    $error = "";
    $sql = "SELECT $campo FROM $table WHERE $campo = $valor";
    $consulta = mysql_query($sql,$link);
    $resultado = 0;
    $resultado = mysql_num_rows($consulta);
    if($resultado > 0){
        $error = "<span class=\"forWrong\">El $label ya existe</span><br
/>\n";
    }
    return $error;
}

function comprobar_email($email){
    $mail_correcto = 0;
    $error="";
    //compruebo unas cosas primeras
    if ((strlen($email) >= 6) && (substr_count($email,"@") == 1) &&
(substr($email,0,1) != "@") && (substr($email,strlen($email)-1,1) != "@")){
        if ((!strstr($email,"'")) && (!strstr($email,"\"")) &&
(!strstr($email,"\\")) && (!strstr($email,"\$")) && (!strstr($email," "))) {
            //miro si tiene caracter .
            if (substr_count($email,".")>= 1){
                //obtengo la terminacion del dominio
                $term_dom = substr(strrchr ($email, '.'),1);
                //compruebo que la terminación del dominio sea correcta
                if (strlen($term_dom)>1 && strlen($term_dom)<5 &&
(!strstr($term_dom,"@")) ){
                    //compruebo que lo de antes del dominio sea correcto
                    $antes_dom = substr($email,0,strlen($email) -
strlen($term_dom) - 1);
                    $caracter_ult = substr($antes_dom,strlen($antes_dom)-1,1);
                    if ($caracter_ult != "@" && $caracter_ult != "."){
                        $mail_correcto = 1;
                    }
                }
            }
        }
    }
    if ($mail_correcto == 0) $error = "<span class=\"forWrong\">La
direcci&oacute;n de correo eletr&oacute;nico no es v&aacute;lida</span>\n";

    return $error;
}
?>

```

index.php

Archivo el cual hace llamadas a archivos de conexión, menú, load, y sys_function, en este archivo se hace la configuración del login o autenticación respecto a los archivos login.php e inicio.html.

Aquí es donde se ejecuta el archivo load.php o si no se ha iniciado sesión el archivo login.php

```
<?php
session_start();

require_once("conexion.php");
$LINK = conexion();
require_once("config.php");
require_once("menu.php");
require_once("sys_function.php");

/* Aqui es donde se ejecuta el archivo load.php o si no se ha logeado el
archivo login.php
*/
    if(! isset($_SESSION['LOG'])){
        if(isset($_GET['content'])){
            if($_GET['content'] == "log"){
                include ('login.php');
            }
            else{
                include('load.php');
            }
        }
        else{
            echo draw_menu($LINK);
            include('inicio.html');
        }
    }
    else{
        include('load.php');
    }
    //echo "AQUI TERMINA\n";
?>
```

articulos.css

Este archivo presenta el conjunto de formatos o estilos utilizados para la generación del artículo en HTML y formato pdf por cada campo del artículo. Con el fin de realizar dicha acción y salida de forma configurable por medio de este archivo.

```
. encabezado{
    font-family: Arial, Times New Roman;
    font-size: 10pt;
    font-weight: bold;
    text-align: right;
    color: #454545;
}
.titulo{
    font-family: Arial, Times New Roman;
    font-size: 24pt;
```

```

        text-align: center;
        color: #000000;
    }
.autor{
    font-family: Arial, Times New Roman;
    font-size: 10pt;
    text-align: center;
    color: #000000;
    font-weight: bold;
    font-style: italic;
}
.resumen{
    font-family: Arial, Times New Roman;
    font-size: 10pt;
    text-align: justify;
    font-weight: bold;
    color: #000000;
}
.palabraclave{
    font-family: Arial, Times New Roman;
    font-size: 10pt;
    text-align: left;
    color: #000000;
    font-weight: bold;
    font-style: italic;
}
.introduccion{
    font-family: Arial, Times New Roman;
    font-size: 10pt;
    text-align: justify;
    color: #000000;
}
.introduccion{
    font-family: Arial, Times New Roman;
    font-size: 10pt;
    text-align: justify;
    color: #000000;
}
.contenido{
    font-family: Arial, Times New Roman;
    font-size: 10pt;
    text-align: justify;
    color: #000000;
}
.titulocaps{
    font-family: Arial, Times New Roman;
    font-size: 12pt;
    text-align: center;
    color: #000000;
    font-weight: bold;
}
.agradecimientos{
    font-family: Arial, Times New Roman;
    font-size: 10pt;
    text-align: justify;
    color: #000000;
    font-weight: italic;
}

```

```
.conclusiones{
    font-family: Arial, Times New Roman;
    font-size: 10pt;
    text-align: justify;
    color: #000000;
}
.referencias{
    font-family: Arial, Times New Roman;
    font-size: 9pt;
    text-align: justify;
    color: #000000;
}
```

5. VARIABLES GLOBALES

Se utilizan variables globales en este sistema con el fin de relacionar y dar accesos a los usuarios por medio de su tipo de usuario, a continuación se presentan las variables globales que contiene este Sistema:

- **\$_SESSION['LOG']**

Guarda el id del usuario de la tabla usuario (id_usuario) de forma global, luego de la respectiva autenticación.

Esta sesión se ocupa en varios formularios con el fin de relacionar a los usuarios con varias actividades o bien modificar su propio perfil.

- **\$_SESSION[USR]**

Variable global que guarda el nick del usuario de forma global, guardado en la tabla usuario (us_nick), luego de la respectiva autenticación.

Esta variable nos sirve para presentarla en Bienvenido 'Nick de Usuario', en todos los sitios.

- **\$_SESSION['GROUP']**

Esta Variable global guarda el tipo de usuario de la tabla usuario (us_tusuario) de forma global, luego de la respectiva autenticación.

Esta variable funciona a la par de la tabla 'contents' ya que por medio de los tipos de usuario y accediendo a esta variable son distribuidos los accesos al sitio, es decir se ofrecen privilegios.

- **\$_GET[content]**

Esta variable global guarda los nombres de los accesos de acuerdo a la tabla 'contents' y campo nombre de dicha tabla.⁴

6. MANEJO DE ENTIDADES

Cada una de las entidades se maneja de forma diferente los estados en la mayoría de los casos. A continuación se describen los estados por entidad.

6.1 USUARIOS

Se manejan 4 tipos de usuario dentro de este Sistema:

| Código | Descripción |
|--------|---------------|
| [ADM] | Administrador |
| [AUT] | Autor |
| [EDT] | Editor |
| [LEC] | Lector |

Tabla 1 – Codificación de etiquetas por tipo de usuario.

Se presentan privilegios diferentes por lo que es necesario realizar los accesos necesarios para operaciones específicas por usuario.

⁴ Ver Manejo de la Tabla 'contents' en este Manual Técnico.

Cada uno de los mantenimientos de usuario, tienen propósitos de guardado de tipo de usuario diferente y específico.

Por ejemplo: Un mantenimiento de Administrador, es el encargado de ingresar un nuevo administrador, nuevo autor o bien nuevo editor, esto se hace seleccionándolo de acuerdo a las opciones que presenta el tipo de usuario dentro del mantenimiento de usuario, automáticamente se ocupa por tipo de usuario las etiquetas mostradas en la tabla 1, respectivamente. Es aquí también posible, que en este mantenimiento posea la capacidad de activar o desactivar por medio del estado.

Los estados en esta entidad se manejan de la misma forma de numeración binaria, 1-Activo y 0- Desactivo.

Siguiendo, para un tipo de usuario lector, ya que por ser un tipo de usuario con menores privilegios, el tipo de usuario es un estado default a lector [LEC], sin posibilidad de crearse como otro tipo de usuario. Este usuario representa a la etiqueta [ALL] de la tabla 'contents'.⁵

Y para terminar, los autores y editores presentan edición de perfil sin posibilidad de poder cambiar su tipo de usuario. Solo es posible por medio del administrador, poder realizar dicho cambio o creación de un nuevo usuario.

6.2 ARTICULOS

Esta entidad varía respecto a cada tipo de usuario, ya que se maneja de forma diferente, en base a función y a estado.

A continuación se describe el manejo de los estados por artículos dentro de este sistema:

| Estado | Nombre |
|--------|-------------|
| 0 | Desactivado |
| 1 | Aprobado, |
| 2 | Pendiente, |

⁵ Ver manejo de la tabla de base de datos 'contents' en este Manual Técnico.

| | |
|---|-------------|
| 3 | En Proceso, |
| 4 | Rechazado, |
| 5 | Activado |

Tabla 2. Estados de Artículos.

A continuación se explicaran los Estados en Artículos:

0-DESACTIVADO, 5-ACTIVADO

Este estado lo utiliza únicamente el Administrador, de acuerdo al mantenimiento de artículo para Administrador.

Estados= 1-3, Son estados establecidos por medio de algunos procesos del tipo de usuario: Editor.

1-APROBADO

Este estado únicamente lo puede asignar un tipo de usuario: Editor, el cual al terminar la revisión y corrección del artículo, procede a una Aprobación.

2-PENDIENTE

Al momento de Crear un Artículo (Nuevo Articulo) desde Usuario: Autor, al guardarse de forma automática, se asigna este estado de forma automática por medio del mantenimiento.

3-EN PROCESO

Este estado únicamente lo puede asignar un tipo de usuario: Editor, este proceso se asigna automáticamente, mientras el Editor lo tiene en proceso de revisión.

4-RECHAZADO

Este estado únicamente lo puede asignar un tipo de usuario: Editor, el cual al terminar la revisión y corrección del artículo, procede a un Rechazo, debido a muchas razones. Este proceso se hace de forma manual por medio de un botón del mantenimiento de revisión de artículo.

6.3 REVISTA, SECCIONES Y POLITICAS

Los estados en esta entidad se manejan de la misma forma de numeración binaria, **1-Activo y 0- Desactivo**. Y solo el tipo de usuario Administrador, tendrá la capacidad de afectar dicho estado.

6.4 COMENTARIOS

Los estados en esta entidad se manejan de la misma forma de numeración binaria, **1-Aprobado y 0- Sin Publicación**. Y solo el tipo de usuario Administrador, tendrá la capacidad de afectar dicho estado, es decir, solo el administrador podrá aprobarlo para que sea posible la publicación dentro de los comentarios.

7. USO DE LA TABLA ‘CONTENTS’

Esta tabla permite administrar el sitio dando privilegios a cada tipo de usuario, es aquí donde se debe modificar cualquier cambio o agregación de enlace para afectar los menús o bien dar otros accesos dentro del sitio.

La tabla ‘contents’ presenta la siguiente estructura:

| Campo | Tipo |
|--------------|-------------|
| nombre | varchar(30) |
| visible | char(1) |
| available | char(1) |
| permissions | text |
| url | text |
| orden | varchar(2) |
| cllogin | varchar(2) |

Tabla 3. Estructura de la tabla 'contents'.

Al momento de ingresar los campos se debe tomar las siguientes consideraciones:

Nombre, es el distintivo de cada enlace o página dentro del sistema [RECs].

Visible, Available, Orden y clogin son campos que se aprovechan para la autocreación de menú superior.

Permissions, se ocupa para establecer el tipo de usuario el cual va usar dicho enlace. Dentro de este campo se han establecido 4 tipos de etiquetas:

| Nombre | Descripción |
|--------|---|
| [ALL] | Al etiquetar el enlace como [ALL], se está especificando que cualquier tipo de usuario podrá usar el enlace. Se usa para Lector y todos los demás usuarios. |
| [ADM] | Etiqueta para enlaces de Administrador. |
| [EDT] | Etiqueta para enlaces de Editor. |
| [AUT] | Etiqueta para enlaces de Autor. |

Tabla 4. Etiquetas por tipo de usuario para tabla 'contents'

A continuación se presentará una tabla la cual muestra los datos necesarios para que el sistema trabaje. (Se han tomado datos relevantes en esta tabla)

| Nombre | Visible | Permisos | URL |
|----------------------------------|---------|----------|---------------------|
| Inicio | 1 | [ALL] | inicio.html |
| Acerca de... | 1 | [ALL] | acercade.html |
| Mapa del Sitio, Administrador | 1 | [ADM] | mapasitioadmin.html |

| | | | |
|-------------------------|---|-------------|--------------------------|
| Mapa del Sitio, Autor | 1 | [AUT] | mapasitioautor.html |
| Mapa del Sitio, Editor | 1 | [EDT] | mapasitioeditor.html |
| Mapa del Sitio | 1 | [LEC] | mapasitiolector.html |
| Mantenimiento Usuarios | 1 | [ADM] | formusuarioadmin.php |
| Nuevo Lector | 1 | [ALL] | nuevousuario.php |
| Nuevo Articulo | 1 | [AUT] | apropoliticass.php |
| Envio de Trabajos | 1 | [ALL] | enviotrabajos.html |
| Terminos y Politicas | 1 | [ALL] | termcondiciones.html |
| Edicion de Sitios | 1 | [ADM] | formsitios.php |
| Mantenimientos Admin | 1 | [ADM] | mantenimientosadmin.html |
| Mantenimiento Politicas | 1 | [ADM] | formpoliticas.php |
| Busqueda | 1 | [ALL] | busqueda.php |
| Mantenimiento Secciones | 1 | [ADM] | formsecciones.php |
| Vistas Articulos | 1 | [ALL] | vp_articulos.php |
| Mantenimiento Revistas | 1 | [ADM] | revistas.php |
| Mantenimiento Auditor | 1 | [ADM] | formauditor.php |
| Contactos | 1 | [ALL] | contactos.html |
| Articulos | 1 | [ADM] | formarticulo.php |
| MtoRevista | 1 | [ADM] | mtorevista.php |
| seleccionrevista | 1 | [ADM],[AUT] | selrevista.php |
| seleccionseccion | 1 | [ADM],[AUT] | selseccion.php |
| Vista HTML | 1 | [ALL] | vp_ahtml.php |
| Vista PDF | 1 | [ALL] | vp_apdf.php |
| nuevoarticulo | 1 | [AUT] | articulos.php |
| Nuevo Capitulo | 1 | [AUT] | capitulos.php |
| Editar Articulo | 1 | [AUT] | articulosE.php |
| Editar Articulos | 1 | [EDT] | larticulosE.php |
| Mis Aportaciones | 1 | [AUT] | misaportaciones.php |

Tabla 5. Registros necesarios para que funcione el Sistema [RECs].

8. MANEJO DE MENÚ E HIPERVÍNCULOS

De acuerdo a la tabla 'contents' se aprovecha también para la autocreación de menús. La información que se toma es **Nombre, Visible, Orden y clogin**. Siendo el '**nombre**' una descripción pequeña del enlace, '**visible**' se ocupa de forma binaria, '1' para que cada enlace se muestre y '0' para que no se muestre dentro del menú, el '**orden**' es el campo donde se especifica antes o después de que enlace se desea poner un nuevo enlace, esto se hace de forma numérica, y '**clogin**' es un campo el cual sirve para mostrar u ocultar enlaces cuando el usuario esta autenticado o no, se usa también de forma binaria.

Es decir, cada tipo de usuario tendrá diferentes menús, debido a sus privilegios, al seguir estos pasos es posible combinar la seguridad con procesos automáticos como estos.

Hay tres archivos que ayudan a poder enlazar todos estos procedimientos, llamados **menu.php y load.php**.⁶

En el archivo menu.php, es el encargado de hacer la consulta a la base de acuerdo a cuales tienen q ser visibles, permisos a que tipo de usuario, orden ascendente por medio del campo orden, es donde se crea el enlace Cerrar sesión dentro del menú y además ahí se especifica que en el caso del tipo de usuario Lector, al momento de entrar al sistema aparezca el enlace Nuevo Lector, y luego al momento ya de haberse autenticado cambie a llamarse el enlace Mi Perfil.

En load.php, en este archivo se encuentra funciones para "Bienvenido usuario", siendo usuario una variable de acuerdo al nombre o nick del usuario autenticado, además es donde se redirecciona la pagina luego de la autenticación y es donde se manejan los errores de enlaces dentro del sistema.

Es necesario mencionar y advertir, es necesario ingresar a [RECs] a cualquier pagina de la siguiente manera <http://localhost/recs/index.php?content=nombre>, siendo **localhost** variable de acuerdo al servidor local o remoto de donde se este corriendo

⁶ Ver 4-Archivos Globales en este Manual Tecnico.

y el **nombre**, referenciando al campo nombre de la tabla 'contents'. De otra forma en archivos con extensión .php no será posible ingresar al sistema.

Todos los sitios poseen **index.php?content=nombre**, esto servirá para conservar el menú en la parte superior y para poder ingresar a los enlaces respectivos a cada usuario dentro del Sistema [RECs]. Estos criterios son solo tomados en caso se ingrese la dirección manualmente en el explorador Web, ya que el sistema cuenta con enlaces accesibles en todo su entorno. La asociación de todo lo explicado da como resultado el menú en [RECs].

9. DICCIONARIO DE ARCHIVOS

9.1 Modulo Administrador

Los archivos que se utilizan en este perfil con sus pertinentes funciones y acciones se describen en la siguiente tabla:

| Nombre Archivo | Descripción | Archivos que utiliza o vínculos que contiene |
|-------------------|---|--|
| formarticulo.php | Formulario que envía la información de un artículo con el método POST a la base de datos para la actualización de estado de este que será únicamente para activarlo o desactivarlo. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |
| formauditor.php | Formulario que envía la información de un auditor con el método POST para su registro en a la base de datos, además de incluir las acciones de edición y eliminación del registro. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |
| formpoliticas.php | Formulario que envía la información de las políticas con el método POST para su registro en a la base de datos, además de incluir las acciones de edición y eliminación del registro. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |

| | | |
|-------------------------|---|--|
| formsecciones.php | Formulario que envía la información de una sección con el método POST para su registro en a la base de datos, además de incluir las acciones de edición y eliminación del registro. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |
| formsitios.php | Formulario que envía la información de un sitio con el método POST a la base de datos para la actualización de los datos de este. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |
| formusuarioadmin.php | Formulario que envía la información de un usuario con el método POST para su registro en a la base de datos, además de incluir las acciones de edición y eliminación del registro. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |
| mantenimientosadmin.php | Formulario que envía la información de un vínculo con el método GET para el acceso de los mantenimientos de usuarios y revista. | index.php?content=edición de Sitio, index.php?content=Mantenimiento Revistas, index.php?content=Mantenimiento Secciones, index.php?content=Artículos, index.php?content=Mantenimiento Auditor y index.php?content=Mantenimiento políticas |

| | | |
|--------------------|--|--|
| mapasitioadmin.php | Formulario que envía la información de un vínculo con el método GET para el acceso a inicio, acerca de, mantenimiento de usuarios, mantenimientos de revista, contáctanos. | index.php?content=Inicio, index.php?content=Acerca de..., index.php?content=edición de Sitio, index.php?content=Mantenimiento Revistas, index.php?content=Mantenimiento Secciones, index.php?content=Artículos, index.php?content=Mantenimiento Auditor y index.php?content=Mantenimiento políticas |
| mtorevista.php | Formulario que envía la información de las revistas, secciones y artículos con el método POST para su auditoria en a la base de datos, además de incluir la acción de realizar comentarios a las revistas. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |
| revistas.php | Formulario que envía la información de las revistas con el método POST para su registro en a la base de datos, además de incluir las acciones de edición y eliminación del registro. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |

9.2 Modulo Autor

Los archivos que se utilizan en este perfil con sus pertinentes funciones y acciones se describen en la siguiente tabla:

| Nombre Archivo | Descripción | Archivos que utiliza o vínculos que contiene |
|--------------------|--|--|
| aprobpoliticas.php | Formulario que envía la información de las políticas aprobadas con el método POST a la lista de revistas activas del sitio [RECs]. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |

| | | |
|---------------------|--|--|
| articulos.php | Formulario que envía la información de un artículo con el método POST para su registro en a la base de datos, además de incluir la llamada al formulario de capítulos. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |
| articulosE.php | Formulario que envía la información de un artículo con el método POST para su edición en a la base de datos. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |
| capitulos.php | Formulario que envía la información de un capítulo con el método POST para su registro en a la base de datos. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |
| mapasitioautor.php | Formulario que envía la información de un vínculo con el método GET para el acceso a inicio, acerca de, crear un nuevo artículo, contáctanos. | index.php?content=Inicio, index.php?content=Acerca de..., index.php?content=Nuevo Artículo |
| misaportaciones.php | Formulario que envía la información de un vínculo con el método GET para el acceso a la edición de un artículo que ya haya ingresado el autor. | sys_function.php, functions.js, style.css y index.php?content=Editar Artículo |

9.3 Modulo Editor

Los archivos que se utilizan en este perfil con sus pertinentes funciones y acciones se describen en la siguiente tabla:

| Nombre Archivo | Descripción | Archivos que utiliza o vínculos que contiene |
|--|--|--|
| capitulosE.php | Formulario que envía la información de un capítulo con el método POST para su edición en a la base de datos. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |
| articulosE.php Final del formulario | Formulario que envía la información de un vínculo con el método GET para el acceso a la revisión de artículos en el cual se pueden actualizar, aprobar o rechazar. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |

| | | |
|--|---|---|
| <p>lcapitulosE.php</p> <p>Final del formulario</p> | <p>Formulario que envía la información de un vínculo con el método GET para el acceso a la revisión de capítulos en el cual se pueden actualizar y aprobar.</p> | <p>sys_function.php, validator.php, functions.js, browsersniffer.js y style.css</p> |
| <p>mapasitioeditor.php</p> | <p>Formulario que envía la información de un vínculo con el método GET para el acceso a inicio, acerca de, revisión de artículo, contáctanos.</p> | <p>index.php?content=Inicio, index.php?content=Ac erca de..., index.php?content=E ditar Articulos</p> |

9.4 Modulo Lector

Los archivos que se utilizan en este perfil con sus pertinentes funciones y acciones se describen en la siguiente tabla:

| Nombre Archivo | Descripción | Archivos que utiliza o vínculos que contiene |
|---|--|--|
| acercade.php | Formulario que contiene la información de lo se puede realizar en [RECs]. | functions.js, browsersniffer.js y style.css |
| busqueda.php Final del formulario | Formulario que envía la información de una palabra clave con el método GET para realizar la búsqueda de artículos que ya hayan sido aprobados listos para publicar. | functions.js y style.css |
| contactanos.php Final del formulario | Formulario que contiene la información de el soporte técnico que conforma [RECs]. | functions.js y style.css |
| enviotrabajos.php Final del formulario | Formulario que contiene la información de cómo puede ser enviado el artículo por el usuario y el formato en el que se envía a [RECs] por medio de una dirección de correo electrónico. | functions.js y style.css |
| inicio.html | Página que contiene la información que se esta actualizando según una agenda en la cual puede informarse el lector y en la cual también puede iniciar sesión el usuario. | functions.js y style.css |
| nuevousuario.php Final del formulario | Formulario que envía la información de un usuario lector con el método POST para su registro en a la base de datos. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |

| | | |
|---|--|--|
| rbusquedas.php Final del formulario | Formulario que envía la información de una palabra clave con el método GET para realizar la búsqueda de artículos que ya hayan sido aprobados listos para publicar y también utiliza otro formulario con el método POST para listar el resultado de la búsqueda. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |
| termcondiciones.php | Formulario que contiene la información de los términos bajo los cuales se pueden aportar contenidos a [RECs]. | functions.js y style.css |
| vistalector.php Final del formulario | Formulario que envía la información de un vínculo con el método GET para visualizar las secciones de una revista, los artículos de las secciones siempre y cuando estos ya hayan sido aprobados por el equipo publicista. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |
| vp_ahtml.php Final del formulario | Formulario que genera la información de un artículo aprobado con el método GET para visualizar los artículos en formato HTML. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |
| vp_apdf.php | Formulario que genera la información de un artículo aprobado con el método GET para visualizar los artículos en formato PDF. | sys_function.php, validator.php, functions.js, browsersniffer.js y style.css |

10. Diseño: EDICION DE BANNERS

Si es necesario cambiar los banners representativos con logos [RECs], es necesario modificar los archivos adjuntos en la carpeta recs, luego carpeta 'images', llamados MAIN y TOP, ambos modificables en formato .psd (modificables en Adobe Photoshop).



Figura 1. Banners actuales en [RECs]

La Figura 1, muestra las dos figuras antes mencionadas.

11. Diseño: EDICION DE BANNER PUBLICITARIO

Al hacer funcionar el sitio por primera vez, se considera el banner vertical como un prototipo para justificar el espacio libre en la parte derecha, por lo que se brinda en la carpeta recs, el archivo original editable con extensión .fla (Adobe/Macromedia Flash) para la edición e inserción de publicidad, respecto a los patrocinadores de [RECs].



Figura 2. Banner Publicitario

La Figura 2, muestra el .swf ubicado en todas las paginas del sitio.