



e s t r a t e g i a • c r e a t i v i d a d • t e c n o l o g í a

Sistema de administración de contenido de la Taurus

# **CMS TAURUS**

**Manual Técnico**



## INDICE

Indice .....	2
Manual Técnico .....	3
1 Datos Generales .....	3
1.1 Sistema CMS Taurus.....	3
1.2 Requerimientos del sistema .....	3
2 Arquitectura del CMS Taurus.....	4
2.1 Estructura de directorios del site .....	5
2.1.1 Modelo de acceso a base de datos.....	6
2.1.2 Las Clases Entidad ( Entity ) .....	7
2.2 Lógica del Administrador de Contenidos.....	12
2.2.1 Estructura de tablas.....	12
2.2.2 Mantenimiento y optimizacion de tablas.....	13
3 Descripción de tablas del Web Site .....	14
3.1 T_Administrador .....	14
3.1 T_Configuracion .....	14
3.1 T_Email.....	14
3.1 T_HomePage.....	15
3.1 T_Idioma.....	15
3.1 T_IdiomaNivel.....	15
3.1 T_IdiomaPais.....	15
3.1 T_Menu.....	16
3.1 T_OportunidadLaboralMensaje .....	16
3.1 T_OportunidadesDetalle.....	16
3.1 T_PáginaWeb .....	16
3.1 T_Permission .....	17
3.1 T_Plantilla .....	17
3.1 T_Plantilla_x_SeccionWeb.....	17
3.1 T_Rol .....	18
3.1 T_Seccion.....	18
3.1 T_SeccionWeb .....	18
3.1 T_SeccionWeb_x_Idioma.....	18
3.1 T_StringTable .....	19
3.1 T_StringTable_x_Idioma .....	19
3.1 T_StringTableCategoria .....	19
3.1 T_Submenu .....	19



## MANUAL TÉCNICO

El presente manual tiene como objetivo proporcionar la orientación para el mantenimiento del CMS de TAURUS. A lo largo del documento se explicará:

- Los datos generales.
- Los estándares utilizados en el CMS que permiten de manera rápida e intuitiva comprender el funcionamiento del sitio web en su integridad.
- La arquitectura técnica del CMS.
- El diseño de la base de datos.

### 1 DATOS GENERALES

#### 1.1 SISTEMA CMS TAURUS

El sistema es una interfaz que permite administrar el contenido de un sitio web almacenada en una base de datos, dicha administración se realiza en función a los permisos asignados a los diversos roles en el sistema.

#### 1.2 REQUERIMIENTOS DEL SISTEMA

El sistema requiere la previa instalación de los siguientes programas:

- Servidor Web Apache versión 1.2 o superior ([www.apache.org](http://www.apache.org))
- Base de datos mysql versión 4.0.\* ([www.mysql.org](http://www.mysql.org))
- PHP versión 4.2\* ([www.php.org](http://www.php.org))
- Servidor de correo

## 2 ARQUITECTURA DEL CMS TAURUS

El CMS de TAURUS fue desarrollado bajo la plataforma LAMP esto implica el siguiente conjunto de software: el sistema operativo Linux, “Servidor” web Apache, la base de datos MySQL y el lenguaje de programación PHP. El sistema también puede ser ejecutado en un entorno Windows con las versiones de software para esta plataforma. El sitio web de TAURUS ha sido testeado en los siguientes navegadores: Internet Explorer 6.0, 7.0 y Firefox 2.0.

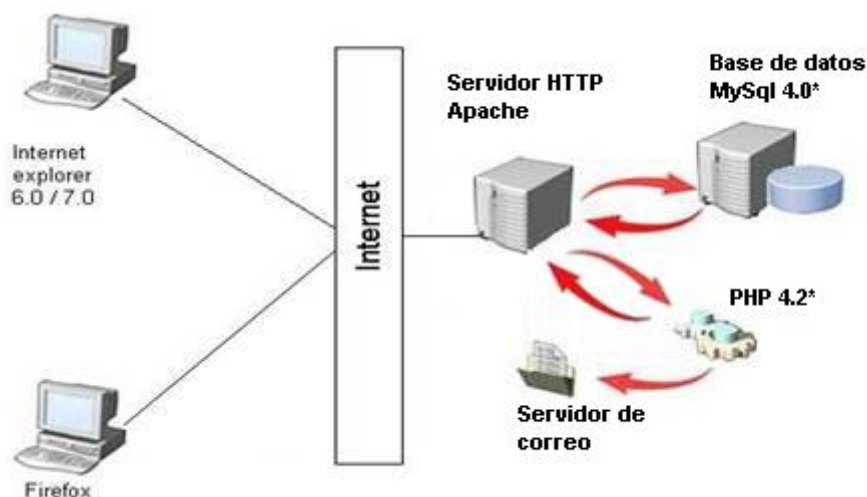


Figura 1: Arquitectura del CMS TAURUS

- Es escalable pues permite la adición de nuevos módulos sin afectar los módulos existentes.
- Es modular pues contiene una infraestructura que permite activar y desactivar opciones de administración y front-end.
- Es portable pues a pesar de estar escrito en un lenguaje de programación en específico, puede ser reprogramado en varios lenguajes de programación sin afectar la lógica y obteniendo el mismo resultado.

El sitio web se compone básicamente de dos capas virtuales:

- La capa de datos que se encuentra formada por clases en PHP y cuya función es la de almacenamiento de data.
- La capa de lógica y vista cuya función es proporcionar la lógica de la aplicación con una adecuada lógica de presentación.

El sitio web está compuesto por dos interfaces de usuario:

- El front-end para el usuario. En este caso el usuario de Internet.
- El módulo administrador que permite el cambio de casi el 90% del sitio web. Los casos que no son posibles de administrar son los siguientes:
  1. Los Flash principales del home. Esto se debe a que estos flashes hacen la conexión de la presentación con el contenido, las cadenas de conexión se encuentran en los archivos \*.xml que se localizan en la raíz.
  2. La adición de nuevos módulos. Esto se hace de forma manual a través de la tabla T\_Seccion.



## 2.1 ESTRUCTURA DE DIRECTORIOS DEL SITE

Actualmente, la estructura de directorios del sitio web se compone de las siguientes partes:

- admin: Es el directorio donde reside el Módulo Administrador de Contenidos. Este directorio a su vez se compone de:
  - js: Archivos Javascript
  - Carpetas para cada módulo. Cada carpeta significa un módulo de mantenimiento del Módulo de Administración de Contenidos.
- classes: Es el directorio donde se encuentra las clases utilizadas en todo el sitio web.
- website: Es el directorio que aloja dos directorios.
  - content: Es el directorio que aloja las plantillas que manipulan el contenido de la base de datos.
  - includes: que código común a varias páginas PHP.
- hoja\_vida: Es el directorio en donde se depositan los currículos del formulario “Trabajos & Carreras.”
- images: Es el directorio donde se guardan las imágenes y recursos del sitio web. Esta carpeta no debe ser usada para la carga de archivos.
- mail: Contiene la plantilla que se le enviara al destinatario de contáctenos.
- js: Archivos Javascript.
- mail: Es el directorio donde se encuentran todas las plantillas de presentación para el envío de mails.
- templates: Plantillas de Macromedia Dreamweaver para la creación de nuevos módulos de administración y páginas.
- includes: Es el directorio donde se encuentran los estilos del sitio web
- swf: Es el directorio en el cual se guardan los archivos flash de la web.

La estructura de este web site debe ser respetado pues un cambio en la estructura significa el malfuncionamiento de todo el sitio web.

## 2.1.1 MODELO DE ACCESO A BASE DE DATOS

El modelo de acceso a base de datos esta compuesto por un conjunto de clases que nos facilitan el trabajo al interactuar con la base de datos estas clases se encuentra en el archivo (/classes/base/Database.php).

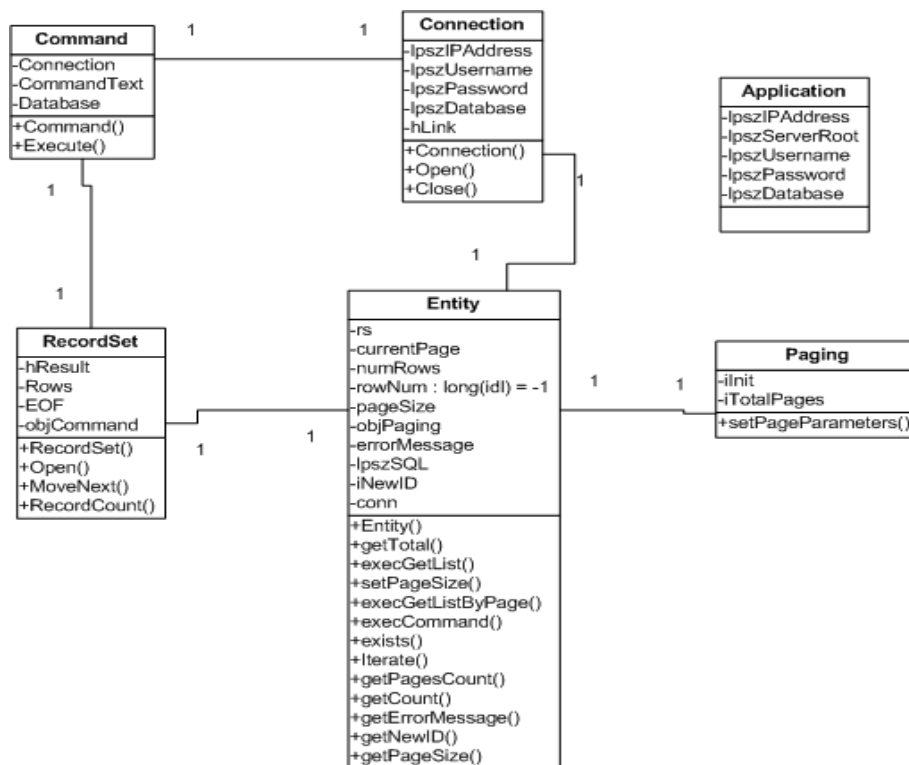


Diagrama 1: Diagrama de clases del modelo de acceso a base de datos.

- **Application:** en sus atributos fijaremos los valores para efectuar la conexión con la base de datos.

**IpszIPAddress** almacena la dirección IP y el puerto al servidor de base de datos.

**IpszUsername** almacena el nombre del usuario de la base de la base de datos.

**IpszPassword** almacena el password de usuario de la base de datos.

**IpszDatabase** almacena el nombre de la base de datos a acceder.

- **Connection:** hace uso del objeto Application que posee los valores de acceso a la base de datos. Esta encargado de abrir y cerrar la conexión a la base de datos.

**IpszIPAddress** almacena la dirección IP y el puerto al servidor de base de datos.

**IpszUsername** almacena el nombre del usuario de la base de la base de datos.

**IpszPassword** almacena el password de usuario de la base de datos.

**IpszDatabase** almacena el nombre de la base de datos a acceder.

**Open** abre una nueva conexión.

**Close** cierra una conexión.

- **Command**: encargado de ejecutar las instrucciones SQL.

**Connection**: objeto que proporciona la conexión a la base de datos.

**Execute**: ejecuta instrucciones SQL.

- **RecordSet**: encargado de manipular la data recuperada.

**hResult**: en este array se almacenan los resultados de la consulta.

**Rows**: almacena el número actual del registro.

**EOF**: si es false indica que ha finalizado el recorrido del array.

**objCommand**: objeto que nos permite ejecutar consultas.

**Open**: ejecuta un query y el resultado lo deposita en el hResult.

**MoveNext**: se mueve al siguiente registro.

**RecordCount**: devuelve el número de registros devueltos en el recordset.

- **Paging**: encargada de la función de paginación.

**iInit**: pagina inicial.

**iTotalPages**: total de paginas.

**setPageParameters**: hace el cálculo de paginado del registro inicial al registro final de cada página.

- **Entity**: encargada de administrar las clases anteriores así como proveer los valores adecuados esta clase de detalla en la siguiente sección.

**Nota**: cada uno de los objetos mencionados anteriormente son creados por cada objeto entidad (Entity) es decir una relación de uno a uno con cada objeto del diagrama, pero solamente se abre la conexión cuando se ejecuta alguna instrucción SQL y luego se cierra inmediatamente lo que impide que existan conexiones abiertas.

## 2.1.2 LAS CLASES ENTIDAD ( ENTITY )

Las clases entidades, al estar directamente relacionadas a una tabla de la base de datos, requieren de una estructura que permita, con poco cambio de líneas de código, mantener las tablas, las clases y los objetos en la página web.

Es por este motivo que todas las clases entidades del CMS se componen en cinco partes:

- La declaración de atributos de la clase.
- Los métodos básicos de toda entidad.
- Los métodos del módulo de administración.
- Los métodos del front-end.
- Los métodos de recuperación de datos.



Todas las clases entidad implican un mantenimiento en el módulo administrador y la presentación de datos en el front-end. Por ello, los métodos se separan en dos secciones: métodos del administrador y métodos del front-end.

Para una explicación más profunda de la estructura de las clases, se tomará una clase de ejemplo. Entity, localizada en la carpeta /classes/base/ Database.php.

### 2.1.2.1 DECLARACIÓN DE LA CLASE

La declaración de la clase se realiza a través de la sentencia:

```
Class [Nombre de la clase]
```

En este caso, el nombre de la clase es Proyecto:

```
Class Entity
```

### 2.1.2.2 ATRIBUTOS BÁSICOS

A continuación, los atributos básicos de la clase son definidos debajo del comentario

```
* ***** ATRIBUTOS
```

Los atributos básicos de toda clase entidad son:

```
var $rs;  
var $currentPage;  
var $numRows;  
var $rowNum=-1;  
var $pageSize=10;  
var $objPaging;  
var $errorMessage="";  
var $lpszSQL="";  
var $iNewID;  
var $con;
```

- El atributo \$rs es utilizado para el recordset que la entidad va a manejar. Dicho atributo es inicializado en los métodos del módulo de administración y de front-end. Por ejemplo en una consulta en la cual se obtiene todas los proyectos, estos serán guardados en \$rs.
- El atributo \$currentPage contiene la página actual de la consulta; sólo es usado cuando se utiliza paginación de datos.
- El atributo \$numRows contiene el número total de registros de la consulta actual.
- El atributo \$rowNum contiene el número de orden del registro actual.
- El atributo \$pageSize contiene el número de filas máximo por cada página, sólo es utilizado cuando se utiliza paginación de datos.
- El atributo \$objPaging contiene métodos que participan en el proceso de paginación, como calcular el número de páginas y el registro inicial y final de la página actual.
- El atributo \$errorMessage indica el mensaje de error actual.
- El atributo \$iNewID contiene el último ID nuevo generado a partir de una inserción en la entidad de la base de datos.
- El atributo \$con contiene el objeto que facilita la conexión a la base de datos;
- El atributo \$lpszSQL contiene el código SQL de la consulta actual.



### 2.1.2.3 MÉTODOS BÁSICO

Esta sección es la misma para todas las clases entidad. Dado que son propiedades extendidas de la clase Entity por medio de lo que se denomina herencia de clases.

Esta sección se compone de 10 métodos:

- El método Entity es el constructor de toda la clase entidad y donde se crea una conexión a la base de datos.
- El método getCount obtiene el número de registros de la consulta actual por página.
- El método getTotal obtiene el número total de registros de la consulta actual.
- El método exists verifica si existen más registros en el recordset.
- El método getPagesCount obtiene el número de páginas de la consulta actual.
- El método Iterate mueve el puntero del recordset al siguiente registro.
- El método getErrorMessage obtiene el último mensaje de error.
- El método execGetList ejecuta la consulta actual; retorna verdadero si tuvo éxito, en caso contrario retornará falso.
- El método execGetListByPage ejecuta la consulta actual con un determinado número de páginas; retorna verdadero si tuvo éxito, en caso contrario retornará falso.
- El método execCommand ejecuta las consultas de mantenimiento (insert, update, delete) retorna verdadero si tuvo éxito, en caso contrario retornará falso.

### 2.1.2.4 MÉTODOS DEL MÓDULO ADMINISTRADOR

Posterior a la declaración de los atributos se encuentran los métodos del módulo de administración.

Todos los métodos de administración son nombrados bajo la siguiente estructura:

- Para consultas: get[Item | List][[Not]ByCriterio1.....ByCriterioN][ByPage][Ordered]
  - Item: utilizado para la carga de un solo registro. Generalmente para obtener un solo registro es necesario pasar como parámetro el valor de la llave primaria del registro deseado. Por ello, no se especifica en el nombre el criterio ByID
  - List: utilizado para la carga de un grupo de registros
  - ByCriterioN define los criterios pasados como parámetros. La palabra Not indica la negación del criterio.
  - ByPage define la paginación del resultado
  - Ordered define el criterio de ordenamiento.
- Para actualización de datos: [Insert | Update | Delete][[Not]ByCriterio1....ByCriterioN]
  - Insert: utilizado para la inserción de datos.
  - Update: utilizado para la actualización de datos.
  - Delete: utilizado para la eliminación de datos



- ByCriterioN: define los criterios pasados como parámetros. La palabra Not indica la negación del criterio.

Por ejemplo:

- Para obtener el registro mediante un ID:  
Function getItem(ID)
- Para obtener la lista de áreas académicas:  
Function getList
- Para obtener la misma lista ordenado por un criterio (los parámetros definen el criterio de orden)  
Function getListOrdered(Field, Asc)
- Para obtener la misma lista paginada  
Function getListByPageOrdered(Pg, Field, Asc)
- Para obtener la misma lista filtrada por el estado activo  
Function getListByActivoByPageOrdered(bActivo, Pg, Field, Asc)

Nótese que los parámetros se encuentran en el mismo orden indicado en el nombre. En el último ejemplo:

- ByActivo → bActivo
- ByPage → Pg
- Ordered → Field, Asc

De esta manera, al programar un nuevo método es más intuitivo ubicar el orden de los parámetros a través del nombre del método.

Todas las funciones del administrador retornan verdadero si es que la ejecución de la consulta fue satisfactoria, o en caso contrario, falso.

La estructura de los métodos de consulta sin paginación es:

```
Class Proyecto extends Entity{  
    function getListOrdered( $parametros ){  
        1)//inicialización de variables  
        2)//desarrollo de la consulta  
        $this->lpszSQL="SELECT idproyecto,titulo,orden,activo FROM proyecto";  
        3)//ejecución de la consulta  
        return $this->execGetList();  
    }  
}
```

La función getListOrdered es un método que pertenece a una clase entidad (Entity) llamada Proyecto.

En la primera parte podemos realizar la inicialización de las variables y asignarles valores por defecto si fuera el caso.



En la segunda parte se desarrolla la consulta con código SQL estándar que posteriormente es asignada al atributo `lpszSQL` de la clase.

La tercera parte efectuamos la ejecución de la consulta asignada al atributo `lpszSQL` la cual nos retorna verdadero si tuvo éxito y falso en caso contrario.

Para el caso de las consultas paginadas la secuencia sería idéntica salvo los parámetros de paginación y la consulta la efectuaremos a través de la función `execGetListByPage` como en ejemplo siguiente.

```
Function getWebListByCriteriaByPageOrdered($parametros..,$numeroPágina , páginaActual ){  
    //inicialización de variables ejemplo el número de registros por página  
    $this->setPageSize($numeroPágina);  
    //desarrollo de la consulta  
    $this->lpszSQL="SELECT * FROM proyecto";  
    //ejecución de la consulta con el número de la página actual como parámetro.  
    return $this->execGetListByPage($Pg);  
}
```

#### 2.1.2.5 MÉTODOS DEL FRONT-END

Luego de la declaración de los métodos del administrador, se encuentran los métodos front-end. Los métodos front-end utilizan la misma nomenclatura que los métodos del administrador. La diferencia en la nomenclatura radica en la palabra `Web` posterior a la palabra `get`.

Así por ejemplo:

- Para obtener todos los registros para la web de los proyectos:

```
Function getWebList
```

Los métodos `web` presuponen en su lógica que los registros deben tener el atributo `activo`, si lo hubiese, en verdadero. De esta manera sólo se presentan los registros activados por el administrador. Dicho criterio, por lo tanto, no es especificado en el nombre del método. Lo mismo es indicado para el orden por defecto.

#### 2.1.2.6 MÉTODOS DE RECUPERACIÓN DE DATOS

Esta sección contiene los métodos para recuperar los datos de la consulta actual.

La nomenclatura de dichos métodos es sencilla:

```
get[Format][Nombre descriptivo del campo]
```

Por ejemplo:

- Para obtener el estado activo de una entidad:  
Function getActivo
- Para obtener el mismo estado pero como texto descriptivo:  
Function getFormatActivo

## 2.2 LÓGICA DEL ADMINISTRADOR DE CONTENIDOS

El administrador de contenidos basa su funcionamiento en:

- La estructura de tablas de configuración del módulo.
- Las páginas php que conforman cada módulo.

### 2.2.1 ESTRUCTURA DE TABLAS

La estructura de tablas básicas del Módulo de Administración de Contenidos almacena datos que sirven para la ubicación de cada módulo dentro del administrador, así como la activación y desactivación de cada módulo. En el siguiente diagrama se grafica la estructura de dichas tablas:

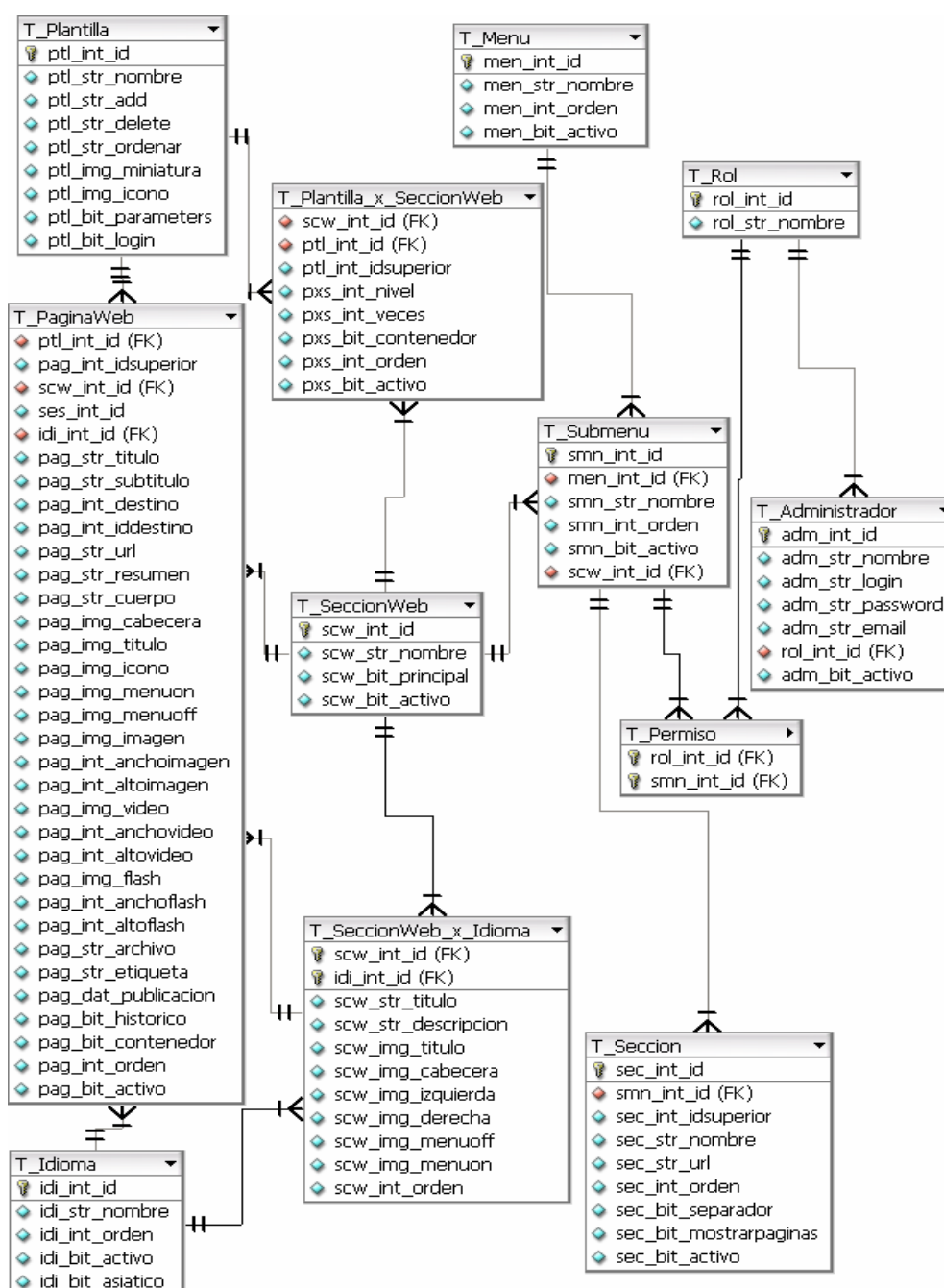


Diagrama 2: Estructura de las tablas básicas del administrador



## 2.2.2 MANTENIMIENTO Y OPTIMIZACION DE TABLAS

Para optimizar la base de datos podemos hacer uso del siguiente aplicativo mysqlcheck comprueba, repara, optimiza, y analiza las tablas.

El mysqlcheck es similar en la función al myisamchk. La diferencia operacional principal es que el mysqlcheck debe ser utilizado cuando el servidor del mysqld está funcionando, mientras que el myisamchk debe ser utilizado cuando no lo esta. La ventaja de usar el mysqlcheck es que no tienes que parar el servidor para comprobar o reparar las tablas.

Ejemplo: mysqlcheck

```
# mysqlcheck -a -e --auto-repair -o -A -u root --password=my_password
```

Ejemplo: myisamchk

```
#usr/local/etc/rc.d/mysql-server.sh stop
```

```
#usr/local/bin/myisamchk -re /var/db/mysql/*/*.MYI
```

```
#usr/local/bin/myisamchk -oe /var/db/mysql/*/*.MYI
```

```
#usr/local/etc/rc.d/mysql-server.sh start
```

```
#sleep 5
```

```
#usr/local/etc/rc.d/mysql-server.sh start
```

### 3 DESCRIPCIÓN DE TABLAS DEL WEB SITE

#### 3.1 T\_ADMINISTRADOR

Usuario del sistema.

Campo	Tipo de dato	No nulo	Descripción
adm_int_id	smallint(5)	Si	Llave primaria
adm_str_nombre	varchar(255)	Si	Nombre del administrador
adm_str_login	varchar(255)	Si	Usuario
adm_str_password	varchar(255)	Si	Password
adm_str_email	varchar(255)		Cuenta de correo
rol_int_id	smallint(5)	Si	referencia al rol
adm_bit_activo	tinyint(1)		1 activo, 0 inactivo

#### 3.2 T\_CONFIGURACION

Contiene la información general correspondiente al proyecto (web site)

Campo	Tipo de dato	No nulo	Descripción
cfg_str_proyecto	varchar(255)	Si	Título del proyecto
cfg_str_host	varchar(255)	Si	IP del servidor
cfg_int_meslanzamiento	tinyint(3)	Si	Mes del lanzamiento
cfg_int_aniolanzamiento	smallint(5)	Si	Año de lanzamiento
cfg_int_mesrediseño	tinyint(3)	Si	Mes rediseño
cfg_int_anioediseño	smallint(5)	Si	Año de rediseño
cfg_str_root	varchar(255)	Si	Directorio root
cfg_bit_activo	tinyint(1)	Si	1 activo, 0 inactivo
fechaEventoIni	date		No aplica
fechaEventoFin	date		No aplica
fechaCambioMontos	date		No aplica
montoAsociadoAntes	decimal(10,0)		No aplica
montoAsociadoDespues	decimal(10,0)		No aplica
montoNoAsociadoAntes	decimal(10,0)		No aplica
montoNoAsociadoDespues	decimal(10,0)		No aplica
montoConyugeAntes	decimal(10,0)		No aplica
montoConyugeDespues	decimal(10,0)		No aplica

#### 3.3 T\_DIRECTORIO

Directorios (rutas relativas) de web site, donde se alojan los diversos archivos tipo imágenes, documentos entre otros

Campo	Tipo de dato	No nulo	Descripción
dir_int_id	tinyint(3)	Si	Llave primaria
dir_str_nombre	varchar(255)	Si	Nombre del directorio
dir_str_pathadmin	varchar(255)	Si	Ruta del directorio para el administrador
dir_str_pathweb	varchar(255)	Si	Ruta del directorio para el front-end

#### 3.4 T\_EMAIL

Contiene los correos de los administradores.

Campo	Tipo de dato	No nulo	Descripción
eml_int_id	tinyint(3)	Si	Llave primaria

eml_str_nombre	varchar(255)	Si	Nombre del e_mail
eml_int_cantidad	tinyint(3)	Si	cantidad
eml_str_email1	varchar(255)	Si	Cuenta de correo 1
eml_str_email2	varchar(255)	Si	Cuenta de correo 2
eml_str_email3	varchar(255)	Si	Cuenta de correo 3

### 3.5 T\_HOMEPAGE

Datos de la página principal.

Campo	Tipo de dato	No nulo	Descripción
idi_int_id	tinyint(3)	Si	Llave foránea a Idioma
hmp_str_titulo	varchar(255)	Si	Nombre de home page
hmp_img_titulo	varchar(255)	Si	No aplica
hmp_img_logo	varchar(255)	Si	No aplica
hmp_img_logohome	varchar(255)	Si	No aplica
hmp_img_principal	varchar(255)	Si	No aplica
hmp_swf_principal	varchar(255)	Si	No aplica
hmp_str_copyright	text	Si	Texto de copyright
hmp_str_terminosetiqueta	varchar(255)	Si	Etiqueta de términos
hmp_str_terminos	text	Si	Texto de términos

### 3.6 T\_IDIOMA

Contiene los idiomas que maneja el sistema

Campo	Tipo de dato	No nulo	Descripción
idi_int_id	tinyint(3)	Si	Llave primaria
idi_str_nombre	varchar(255)	Si	Nombre del idioma
idi_int_orden	tinyint(3)	Si	Orden del registro
idi_bit_activo	tinyint(1)	Si	1 Activo, 0 inactivo
idi_bit_asiatico	tinyint(1)		No aplica

### 3.7 T\_IDIOMANIVEL

Contiene los datos del nivel de idioma del objeto "select" del campo Nivel de idioma del formulario "Trabajo & Carreras".

Campo	Tipo de dato	No nulo	Descripción
inp_int_id	tinyint(6)	Si	Llave primaria
idi_int_id	tinyint(3)	Si	Llave foránea a Idioma
inp_str_nivel	varchar(100)		Nombre del nivel
inp_bit_activo	tinyint(1)	Si	1 activo, 0 inactivo
inp_int_orden	tinyint(3)	Si	Orden del registro

### 3.8 T\_IDIOMAPAIS

Contiene los datos de los idiomas del objeto "select" del campo Lenguas del formulario de la sección web "Trabajo & Carreras".

Campo	Tipo de dato	No nulo	Descripción
len_int_id	tinyint(6)	Si	Llave primaria
idi_int_id	tinyint(3)	Si	Llave foránea a Idioma
len_str_idioma	varchar(100)		Nombre de la lengua
len_bit_activo	tinyint(1)	Si	1 activo, 0 inactivo

len_int_orden	tinyint(6)		Orden del registro
---------------	------------	--	--------------------

### 3.9 T\_MENU

Opción del menú de nivel 1 del administrador.

Campo	Tipo de dato	No nulo	Descripción
men_int_id	tinyint(3)	Si	Llave primaria
men_str_nombre	varchar(255)	Si	Nombre del menú
men_int_orden	tinyint(3)	Si	Número de orden
men_bit_activo	tinyint(1)	Si	1 activo, 0 inactivo

### 3.10 T\_OPORTUNIDADLABORALMENSAJE

Contiene los datos formulario de de la sección web “Trabajo & Carreras”.

Campo	Tipo de dato	No nulo	Descripción
opl_int_id	tinyint(6)	Si	Llave primaria
idi_int_id	tinyint(3)	Si	Llave foránea a la tabla idioma
opl_str_carrera	varchar(100)		Carrera
opl_str_experiencia	varchar(100)		Experiencia
opl_str_nacionalidad	varchar(100)		Nacionalidad
opl_bit_maestria	tinyint(1)		0 sin maestría , 1 con maestría
opl_str_email	varchar(100)		Email
opl_str_documento	varchar(100)		Archivo
opl_bit_activo	tinyint(1)	Si	1 activo, 0 inactivo
opl_dat_registro	datetime		Fecha de registro

### 3.11 T\_OPORTUNIDADESDETALLE

Contiene el detalle de las lenguas habladas del registrado y sus niveles de aquellas lenguas.

Campo	Tipo de dato	No nulo	Descripción
odi_int_id	tinyint(6)	Si	Llave primaria
opl_int_id	tinyint(6)	Si	Llave foránea a la tabla OportunidadLaboralMensaje
len_int_id	tinyint(6)		Id lengua
inp_int_id	tinyint(6)		Id nivel de lengua

### 3.12 T\_PÁGINAWEB

Contenido general administrado por plantillas.

Campo	Tipo de dato	No nulo	Descripción
pag_int_id	int(10)	Si	Llave primaria
pag_int_idsuperior	int(10)		página superior
scw_int_id	tinyint(3)	Si	sección web
ptl_int_id	tinyint(3)		Plantilla
ses_int_id	tinyint(3)		No aplica
idi_int_id	tinyint(3)	Si	Llave foránea a la tabla idioma
pag_str_titulo	varchar(255)	Si	Título de la página
pag_str_subtitulo	varchar(255)		Subtitulo de la página (texto alternativo)



pag_int_destino	tinyint(3)	Si	1 enlace interno, 2 enlace externo, 3 sin enlace
pag_int_iddestino	int(10)		enlace a una página interna
pag_str_url	varchar(255)		enlace a una página externa
app_int_id	tinyint(3)		No aplica
pag_str_resumen	text	Si	Texto de resumen
pag_str_cuerpo	text	Si	Texto de fondo
pag_img_cabecera	varchar(255)	Si	Imagen de cabecera
pag_img_titulo	varchar(255)		No aplica
pag_img_icono	varchar(255)		No aplica
pag_img_menuon	varchar(255)	Si	No aplica
pag_img_menuoff	varchar(255)	Si	No aplica
pag_img_imagen	varchar(255)	Si	Imagen
pag_int_anchoidimagen	int(10)		Ancho de imagen
pag_int_altoimagen	int(10)		Alto de imagen
pag_img_video	varchar(255)	Si	No aplica
pag_int_anchovideo	int(10)		No aplica
pag_int_altovideo	int(10)		No aplica
pag_img_flash	varchar(255)	Si	Flash
pag_int_anchoflash	int(10)		Ancho de Flash
pag_int_altoflash	int(10)		Ancho de Flash
pag_str_archivo	varchar(255)	Si	No aplica
pag_str_etiqueta	varchar(255)	Si	No aplica
pag_dat_publicacion	datetime		No aplica
pag_bit_historico	tinyint(1)	Si	No aplica
pag_bit_contenedor	tinyint(1)	Si	1 contenedor, 0 contenido
pag_int_orden	int(10)	Si	Orden
pag_bit_activo	tinyint(1)	Si	1 activo, 0 inactivo

### 3.13 T\_PERMISO

Item que se pueden acceder con un rol.

Campo	Tipo de dato	No nulo	Descripción
rol_int_id	smallint(5)	Si	Llave primaria, Llave foránea a un rol
smn_int_id	tinyint(3)	Si	Llave primaria, Llave foránea a un submenú

### 3.14 T\_PLANTILLA

Plantillas para las páginas web

Campo	Tipo de dato	No nulo	Descripción
ptl_int_id	tinyint(3)	Si	Llave primaria
ptl_str_nombre	varchar(255)	Si	Nombre de la plantilla
ptl_str_add	varchar(255)	Si	Título añadir
ptl_str_delete	varchar(255)	Si	Título eliminar
ptl_str_ordenar	varchar(255)	Si	Título ordenar
ptl_img_miniatura	varchar(255)	Si	Imagen
ptl_img_icono	varchar(255)	Si	Icono asociado a la plantilla
ptl_bit_parameters	tinyint(1)	Si	No aplica
ptl_bit_login	tinyint(1)	Si	No aplica

### 3.15 T\_PLANTILLA\_X\_SECCIONWEB

Plantillas definidas por cada sección web.

Campo	Tipo de dato	No nulo	Descripción
scw_int_id	tinyint(3)	Si	Sección web
ptl_int_id	tinyint(3)	Si	Plantilla
ptl_int_idsuperior	tinyint(3)		Plantilla en el nivel superior
pxs_int_nivel	tinyint(3)	Si	Nivel de profundidad
pxs_int_veces	tinyint(3)	Si	número veces que se podrá utilizar la plantilla en este nivel
pxs_bit_contenedor	tinyint(1)	Si	1 contenedor, 0 contenido
pxs_int_orden	tinyint(3)	Si	Número de orden
pxs_bit_activo	tinyint(1)	Si	1 activo, 0 inactivo

### 3.16 T\_ROL

Rol de los usuarios del sistema.

Campo	Tipo de dato	No nulo	Descripción
rol_int_id	smallint(5)	Si	Llave primaria
rol_str_nombre	varchar(255)	Si	Nombre del rol

### 3.17 T\_SECCION

Opción de menú de tercer nivel.

Campo	Tipo de dato	No nulo	Descripción
sec_int_id		Si	Llave primaria
smn_int_id		Si	Llave foránea a un submenú
sec_int_idsuperior			Llave foránea a sección superior
sec_str_nombre		Si	Nombre de la sección
sec_str_url		Si	Ruta de la página a la que estará enlazado
sec_int_orden		Si	Número de orden de la sección
sec_bit_separador		Si	1 si la sección actúa como separador
sec_bit_mostrarpaginas		Si	1 si mostrara contenido( páginas web )
sec_bit_activo		Si	1 activo, 0 desactivo

### 3.18 T\_SECCIONWEB

Sección Web opción de primer nivel en el front-end.

Campo	Tipo de dato	No nulo	Descripción
scw_int_id	tinyint(3)	Si	Llave primaria
scw_str_nombre	varchar(255)	Si	Nombre de la sección web
scw_bit_principal	tinyint(1)	Si	1 si es sección principal
scw_bit_activo	tinyint(1)	Si	1 activo, 0 inactivo

### 3.19 T\_SECCIONWEB\_X\_IDIOMA

Atributos de la sección web por idioma.

Campo	Tipo de dato	No nulo	Descripción
-------	--------------	---------	-------------

scw_int_id	tinyint(3)	Si	Llave principal, Llave foránea a Sección
idi_int_id	tinyint(3)	Si	Llave principal, Llave foránea a idioma
scw_str_titulo	varchar(255)	Si	Número de orden
scw_str_descripcion	text	Si	Título de sección por idioma
scw_img_titulo	varchar(255)	Si	Imagen de la sección web
scw_img_cabecera	varchar(255)	Si	Flash cabecera
scw_img_izquierda	varchar(255)	Si	No aplica
scw_img_derecha	varchar(255)	Si	No aplica
scw_img_menuoff	varchar(255)	Si	No aplica
scw_img_menuon	varchar(255)	Si	No aplica
scw_int_orden	tinyint(3)	Si	No aplica

### 3.20 T\_STRINGTABLE

Campo	Tipo de dato	No nulo	Descripción
sst_int_id	tinyint(3)	Si	Llave primaria
stc_int_id	tinyint(3)	Si	Llave foránea a StringtableCategoria
sst_str_nombre	varchar(255)	Si	Nombre del registro
sst_str_clave	varchar(255)		Nombre clave

### 3.21 T\_STRINGTABLE\_X\_IDIOMA

Campo	Tipo de dato	No nulo	Descripción
sst_int_id	tinyint(3)	Si	Llave foránea a T_StringTable
idi_int_id	tinyint(3)	Si	Llave primaria
sxi_str_texto	varchar(255)	Si	Nombre del registro

### 3.22 T\_STRINGTABLECATEGORIA

Campo	Tipo de dato	No nulo	Descripción
stc_int_id	tinyint(3)	Si	Llave primaria
stc_str_nombre	varchar(255)	Si	Nombre del registro

### 3.23 T\_SUBMENU

Opción del menú de nivel 2 del administrador.

Campo	Tipo de dato	No nulo	Descripción
smn_int_id	tinyint(3)	Si	Llave primaria
men_int_id	tinyint(3)	Si	menú relacionado
smn_str_nombre	varchar(255)	Si	Nombre de menú
smn_int_orden	tinyint(4)	Si	Número de orden
smn_bit_activo	tinyint(1)	Si	1 activo, 0 inactivo
scw_int_id	tinyint(3)		sección web que administrará