

WebCoordinador

Estructura de carpetas

En el proyecto dentro de sus carpetas siempre vamos encontrar el mismo sistema de archivos con su nombramiento específico. Dicho nombramiento posee un archivo con la extensión **.php** y **.html** trabajando en conjunto.

Nombramiento estándar:

- **bsq**: Encargado generalmente de la vista del menú.
- **brw**: Encargado generalmente de cargar el contenido.
- **del**: Este se encarga de eliminar los registros que se requiera de la base de datos.
- **grb**: Encargado de agregar registros a la base de datos.
- **mst**:

Manejo de Sigma

Sigma es el template utilizado en webcoordinador. En grandes rasgos con sigma operamos la lógica en el archivo **.php** y le pasamos las variables al archivo **.html** para ser visualizadas en el frontend.

Ejemplo: `<h1>{MiVariable}</h1>`

Uso básico (**.php**):

- Creamos un objeto de tipo **HTML_Template_Sigma**:

```
$tmp1= new HTML_Template_Sigma();
```

- Cargamos el archivo **.html** donde queremos mostrar los datos:

```
$tmp1->loadTemplateFile('brw.html');
```

- Utilizamos el metodo **setVariable()**:

```
$tmp1->setVariable('clave' , valor);
```

Para mostrar en un bloque específico el contenido debemos recurrir al método **setCurrentBlock('Bloque')** y utilizar esta estructura:

- En PHP:

```
$tmp1->setCurrentBlock('bloque');
```

```
$tmpl->setVariable('clave' , valor);
```

```
$tmpl->parse('bloque');
```

- En HTML

```
<!-- BEGIN browser -->
```

```
<H1>{variable}</H1>
```

```
<!-- END browser -->
```

Cuando terminamos toda la lógica y los datos están listos para ser enviados al frontend activamos el "Interruptor"

```
$tmpl->show();
```

Base de datos

Para nombrar bases de datos debemos utilizar el patrón de caracteres ABC_ABC / ABC_ABCD / ABC_ABCDE, los primeros tres caracteres hacen referencia a titular de la tabla y los siguientes cuatro o cinco caracteres a la razón de ser de la tabla. Ejemplo para la tabla persona.

PER_: Referencia al titular de la tabla. **PER_CATE**: Referencia a las categorías de personas. **PER_FAV**: Referencia a los favoritos de la persona.

En varios titulares como por ejemplo persona (**PER_**) vamos a encontrar una razón de ser llamada **_MAEST**, esta reúne todas las tablas del titular con sus claves primarias siempre y cuando la instancia lo requiera.

Funciones

Las funciones están almacenadas en la carpeta de utilidades llamada func.

zdatabase.php

zdatabase.php contiene todas las funciones encargadas de la gestión de la base de datos.

- Abrir conexión con la base de datos:

```
`$conn = sql_conectar()`
```

- Abrir una transacción

```
trans = sql_begin_trans($conn);
```

- Ejecutar Querys Consultas: `$Table = sql_query($query,$conn);`

Update:

Chequeamos si la query retorna error:

```
$err = sql_execute($query,$conn,$trans);
```

Si todo sale bien ejecutamos el commit:

```
if($err == 'SQLACCEPT' && $errcod==0){sql_commit_trans($trans);}
```

En caso de error:

```
sql_rollback_trans($trans);
```

- Cerrar Conexion:

```
sql_close($conn);
```

zfvarias.php

zfvarias seria el archivo "utils" que lo componen varias funciones de uso común dentro de todo el proyecto. Como por ejemplo: `valDatoVacio($dato)` Que valida un dato comprobando si esta o no vacío. Otra no menos importante es: `logError($value)` **logError()** nos ayuda a debuggear los errores de base de datos, generando un archivo **.err** en la siguiente ruta: `/func/errdb` Hay muchas mas funciones en el archivo pero poseen comentarios indicando su comportamiento,

zglobals.php
