

Insertar desde el front end. La tabla histórico. De verdad que se puede

Escrito por Administrator

Lunes, 24 de Agosto de 2015 18:37 - Actualizado Viernes, 01 de Marzo de 2019 08:28

Es el momento de trabajar con el front end. No existen muchas diferencias entre ambos, algún pequeño detalle. Por tanto prácticamente vamos a hacer lo mismo que hicimos en el último paso: Vamos a asociar a un jugador con un equipo en la tabla para insertarla en una tabla, pero no en la tabla juegan, sino en la tabla histórico que además del equipo y del jugador tendrá otros campos que no se verán en el formulario. Digamos que la tabla histórico se diferencia de la tabla fichajes en que guarda más datos (a lo mejor no es el mejor diseño de la base de datos de la historia, pero nos servirá para introducir conceptos importantes y necesarios, como el trabajo con usuarios Joomla o fechas).

En concreto la estructura de esta tabla será la siguiente (a continuación de cada línea se ofrece una breve explicación del sentido del campo):

```
CREATE TABLE IF NOT EXISTS `#_pre_comp_historico` (
```

```
`id` int(11) NOT NULL AUTO_INCREMENT, //identificador
```

```
`id_operador` int(11) NOT NULL, //operador que realiza la operación de añadir un elemento a la tabla
```

```
`id_fichaje` int(11) NOT NULL, //combinación de jugador - equipo
```

```
`id_oficina` int(11) NOT NULL, //Lugar físico dónde se realiza la acción de añadir el elemento a la tabla
```

Insertar desde el front end. La tabla histórico. De verdad que se puede

Escrito por Administrator

Lunes, 24 de Agosto de 2015 18:37 - Actualizado Viernes, 01 de Marzo de 2019 08:28

```
`fecha` datetime NOT NULL, //fecha de la operación
```

```
`observaciones` varchar(300) DEFAULT NULL, //observaciones
```

```
`importe` float NOT NULL DEFAULT '0', //precio del fichaje
```

```
PRIMARY KEY (`id`),
```

```
KEY `id_operador` (`id_operador`),
```

```
KEY `id_oficina` (`id_oficina`),
```

```
KEY `id_fichaje` (`id_fichaje`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

Esta vez vamos a trabajar con 5 ficheros

Como siempre empezamos con la vista principal, la cual, al contrario que en el back end, no contiene ningún tipo de botón especial (los megautilizados JToolBarHelper). En esta ocasión, si queremos usar un botón (lo más lógico) tendrá que ser un botón html como el de toda la vida.

En realidad en la vista lo único que vamos a hacer es llamar al modelo para asignar a las variables las listas que sacaremos de la base de datos. Veámoslo:

Insertar desde el front end. La tabla histórico. De verdad que se puede

Escrito por Administrator

Lunes, 24 de Agosto de 2015 18:37 - Actualizado Viernes, 01 de Marzo de 2019 08:28

/views/presupuestos/view.html.php

```
defined('_JEXEC') or die('Restricted access');
```

```
jimport('joomla.application.component.view');
```

```
class gessegViewhistorico extends JViewLegacy {
```

```
function display($tpl=null) {
```

```
    $model = $this->getModel( 'historico' );
```

```
    $this->ts = $model->getJugadores();
```

```
    $this->cs = $model->getEquipos();
```

```
    $this->oficina = $model->getOficina();
```

```
    parent::display($tpl);
```

```
}
```

Insertar desde el front end. La tabla histórico. De verdad que se puede

Escrito por Administrator

Lunes, 24 de Agosto de 2015 18:37 - Actualizado Viernes, 01 de Marzo de 2019 08:28

```
}
```

Hay 3 llamadas al modelo, que como dijimos anteriormente nos devolverá (en sus respectivas variables) la lista de elementos que formaran las listas desplegadas en html. En el modelo (**/models/historico.php**

) tendremos esas 3 funciones. Pondremos una función sólo y sin extendernos en los comentarios que al fin y al cabo lo hemos hecho varias veces.

```
function getJugadores(){ // Get the database connection
```

```
$db =& $this->_db;
```

```
if( empty($this->_jugadores) ){
```

Comprobamos primero si en la variable que vamos a devolver hay algo, porque si lo hay es que ya hemos llamado a esta función y no hay que revisarlo otra vez, sino devolverlo directamente.

```
// Build query and get the limits from current state
```

```
$query = 'SELECT * FROM #__gesseg_jugadores ORDER BY nombre';
```

```
$limitstart = $this->getState('limitstart');
```

Insertar desde el front end. La tabla histórico. De verdad que se puede

Escrito por Administrator

Lunes, 24 de Agosto de 2015 18:37 - Actualizado Viernes, 01 de Marzo de 2019 08:28

```
$limit = $this->getState('limit');
```

```
$this->_getJugadores = $this->_getList($query,$limitstart,$limit);
```

```
}
```

```
// Return the list of articulos
```

```
return $this->_jugadores;
```

```
}
```

Al final de la vista, tenemos la llamada al display, es decir al fichero **/views/presupuestos/tmpl/default.php**. Se trata de código html en su mayoría así que comentaré solamente los aspectos que me parecen más destacables.

```
<?php defined('_JEXEC') or die('Restricted access');
```

```
$view = 'historico';
```

```
$task = 'edithistorico';
```

```
$usuario = JFactory::getUser();
```

Insertar desde el front end. La tabla histórico. De verdad que se puede

Escrito por Administrator

Lunes, 24 de Agosto de 2015 18:37 - Actualizado Viernes, 01 de Marzo de 2019 08:28

Con la función `JFactory::getUser()` guardamos en la variable `$usuario` la información del usuario que está logueado en la página. Si hay, obviamente, se cargaran los datos en `$usuario` si no hubiera nadie en `$usuario` habrá un usuario vacío con `id = 0`. Si ese `id` es 0 (no hay nadie logueado) no mostraremos el formulario. Estamos hablando de desarrollar algo en el

Front End a lo que solo accederán los usuarios registrados. Esta parte nos la podríamos saltar si lo hiciéramos público.

```
if ($usuario->id!=0){
```

```
?>
```

```
<form method="post" action="<?php echo JRoute::_('index.php'); ?>" name="historico" id="historico">
```

En el action del formulario hemos puesto una función `JRoute` que nos enviará a la misma página en la que estamos.

```
<table class=adminlist>
```

```
<tr>
```

```
<td>
```

```
<label for="jugadores">Jugador:</label>
```

Insertar desde el front end. La tabla histórico. De verdad que se puede

Escrito por Administrator

Lunes, 24 de Agosto de 2015 18:37 - Actualizado Viernes, 01 de Marzo de 2019 08:28

</td>

<td>

<select name="jugador">

<option value="0" selected> Selecciona un jugador </option>

<?php foreach (\$this->jugadores as \$jug): ?>

<option value="<?php echo \$jug->id;?>"><?php echo \$jug->tipo; ?> </option>

<?php endforeach; ?>

</select>

</td>

</tr>

<tr>

Insertar desde el front end. La tabla histórico. De verdad que se puede

Escrito por Administrator

Lunes, 24 de Agosto de 2015 18:37 - Actualizado Viernes, 01 de Marzo de 2019 08:28

<td>

<label for="equipos">Equipo:</label>

</td>

<td>

<select name="equipo">

<option value="0" selected> Selecciona un equipo

<?php foreach (\$this->equipos as \$eq): ?>

<option value="<?php echo \$eq->id;?>"><?php echo \$eq->nombre; ?> </option>

<?php endforeach; ?>

</select>

</td>

</tr>

Insertar desde el front end. La tabla histórico. De verdad que se puede

Escrito por Administrator

Lunes, 24 de Agosto de 2015 18:37 - Actualizado Viernes, 01 de Marzo de 2019 08:28

```
<tr>
```

```
<td>
```

```
<label for="oficina">oficina:</label>
```

```
</td>
```

```
<td>
```

```
<select name="oficinas">
```

```
<option value="0" selected> Selecciona una oficina
```

```
<?php foreach ($this->oficinas as $of): ?>
```

```
<option value="<?php echo $of->id;?>"> <?php echo $of->nombre; ?> </option>
```

```
<?php endforeach; ?>
```

```
</select>
```

Insertar desde el front end. La tabla histórico. De verdad que se puede

Escrito por Administrator

Lunes, 24 de Agosto de 2015 18:37 - Actualizado Viernes, 01 de Marzo de 2019 08:28

</td>

</tr>

<tr>

<td>

<label for="cantidad">Importe:</label>

</td>

<td>

<input type="text" name="cantidad" id="cantidad" maxlength="300" />

</td>

</tr>

<tr>

Insertar desde el front end. La tabla histórico. De verdad que se puede

Escrito por Administrator

Lunes, 24 de Agosto de 2015 18:37 - Actualizado Viernes, 01 de Marzo de 2019 08:28

```
<td>
```

```
<label for="Observaciones">Observaciones:</label>
```

```
</td>
```

```
<td>
```

```
<input type="text" name="observaciones" id="observaciones" maxlength="600" />□
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td></td>
```

```
<td align="right"> <input id="loginSubmit" type="submit" value="insertar"
name="botonInsertar"> </td>
```

```
</tr>
```

Insertar desde el front end. La tabla histórico. De verdad que se puede

Escrito por Administrator

Lunes, 24 de Agosto de 2015 18:37 - Actualizado Viernes, 01 de Marzo de 2019 08:28

```
</table>
```

```
<input type="hidden" name="task" value="insertarhistorico">
```

Al establecer la tarea, al apretar en el botón de la vista, el controlador buscará la función insertarhistórico() que se encuentra en el controlador.

```
<input type="hidden" name="option" value="com_gesseg">
```

```
<input type="hidden" name="checked" value="0" />
```

```
<input type="hidden" name="view" value="<?php echo $view; ?>" />
```

```
</form
```

```
<?php } ?>
```

Lo único que hace esta función es llamar a la función insertarhistórico que se encuentra en el controlador (**controller.php**)

```
public function insertarhistorico(){
```

Insertar desde el front end. La tabla histórico. De verdad que se puede

Escrito por Administrator

Lunes, 24 de Agosto de 2015 18:37 - Actualizado Viernes, 01 de Marzo de 2019 08:28

```
//definimos el modelo y la vista y los asociamos
```

```
$vista = $this->getView('historico','html');
```

```
$modelo = $this->getModel(' historico ', 'gessegModel');
```

```
$vista->setModel($modelo, true);
```

```
// Tomamos las variables de la vista
```

```
$jugador = JRequest::getVar('jugador');
```

```
$equipo = JRequest::getVar('equipo');
```

```
$this->fichaje = $modelo->getFichaje($jugador, $equipo);
```

Con esta función lo que hacemos es obtener el id del fichaje del jugador por el equipo, por un lado para asegurarnos de que efectivamente existe el fichaje y por otro porque en la tabla histórico insertamos ese id. La veremos después

```
$oficina = JRequest::getVar('oficina');
```

Insertar desde el front end. La tabla histórico. De verdad que se puede

Escrito por Administrator

Lunes, 24 de Agosto de 2015 18:37 - Actualizado Viernes, 01 de Marzo de 2019 08:28

```
$cantidad = JRequest::getVar('cantidad');
```

```
$observaciones = JRequest::getVar('observaciones');
```

```
foreach ($this->fichaje as $fich){
```

```
$correcto = $modelo->insertarHistorico($fich->id, $sucursales, $cantidad, $observaciones);
```

```
}
```

```
if ($correcto) $this->setRedirect(JRoute::_('index.php'),'Inserción correcta');
```

```
else $this->setRedirect(JRoute::_('index.php'),'No existe ese fichaje (combinación jugador -  
equipo incorrecta) o la oficina es incorrecta. Póngase en contacto con el  
administrador','ERROR');
```

```
}
```

Nos quedan por ver dos funciones del modelo: Por un lado `getFichajes()` que como dijimos devuelve la lista de los fichajes que tienen al jugador y el equipo (lo llamo lista porque es lo que devuelve, pero obviamente solo devuelve uno). Este es el código en el modelo **/models/histori**

Insertar desde el front end. La tabla histórico. De verdad que se puede

Escrito por Administrator

Lunes, 24 de Agosto de 2015 18:37 - Actualizado Viernes, 01 de Marzo de 2019 08:28

co.php

```
function getFichaje($j, $e){
```

```
    $db =& $this->_db;
```

```
    if( empty($this->_fichaje) ) {
```

si no se ha hecho la consulta antes creamos el query y lo ejecutamos

```
    // Build query and get the limits from current state
```

```
    $query = 'SELECT id FROM #__gesseg_seguros WHERE id_tipo_de_seguro = '.$ts.' AND  
id_comp_de_seguros = '.$cs;
```

```
    $limitstart =$this->getState('limitstart');
```

```
    $limit = $this->getState('limit');
```

```
    $this->_seguro = $this->_getList($query,$limitstart,$limit);
```

```
}
```

Insertar desde el front end. La tabla histórico. De verdad que se puede

Escrito por Administrator

Lunes, 24 de Agosto de 2015 18:37 - Actualizado Viernes, 01 de Marzo de 2019 08:28

```
return $this->_seguro;
```

Al final lo devolvemos, tanto si se acaba de hacer como si ya estaba hecho.

```
}
```

La función insertarHistorico(...) (con mayúsculas, perteneciente al modelo, no confundir con insertarhistorico() que pertenece al controlador) tendrá las instrucciones mysql para insertar.

```
function insertarHistorico($fichaje, $oficina, $cantidad, $observaciones){
```

```
$db =& $this->_db;
```

```
if( empty($seguro) || ($sucursal == 0) ) return false;
```

```
else {
```

```
$nuevoPrep = new stdClass();
```

```
$usuario = JFactory::getUser();
```

```
$nuevoPrep->id_operador = $usuario->id;
```

```
$nuevoPrep->id_fichaje = $fichaje;
```


Insertar desde el front end. La tabla histórico. De verdad que se puede

Escrito por Administrator

Lunes, 24 de Agosto de 2015 18:37 - Actualizado Viernes, 01 de Marzo de 2019 08:28

```
$nuevoPrep->id_oficina = $oficina;
```

```
$nuevoPrep->observaciones = $observaciones;
```

```
$nuevoPrep->importe = $cantidad;
```

```
$nuevoPrep->fecha = date("Y-m-d h:i:s");
```

La función date nos devuelve la hora en el formato que le pasemos como argumento. En nuestro caso año-mes-día hora:minuto:segundos que es el formato de Mysql. Además nótese que la Y del año es mayúscula por lo que devuelve 4 dígitos.

```
$result = JFactory::getDbo()->insertObject ('#__gesseg_historico', $nuevoPrep);
```

Finalmente nos falta por ver un archivo que, si bien no es fundamental a la hora de poder utilizar el módulo, nos permitirá crear un enlace en un menú Joomla a nuestra vista del front end. Ese archivo se encuentra en /views/historico/tmpl/**default.xml**.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<metadata>
```

```
<layout title="historico">
```

Insertar desde el front end. La tabla histórico. De verdad que se puede

Escrito por Administrator

Lunes, 24 de Agosto de 2015 18:37 - Actualizado Viernes, 01 de Marzo de 2019 08:28

Nombre con el que queremos que aparezca en la selección del tipo de menú

<message>

<![CDATA[COM_GESSEG_HISTORICO_VIEW_DEFAULT_DESC]]>

Nombre de la vista

</message>

</layout>

</metadata>