

Cómo crear una tabla editable con PHP, MySQL y jQuery

Nombre	Autor
Los Pilares de la Tierra	Ken Follet
Inferno2	Dan Brown
<input type="text" value="Cincuenta sombras de Grey"/>	E. L. James
La catedral del mar	Ildefonso Falcones

El uso de tablas en las páginas es un recurso muy utilizado para mostrar información a los usuarios. Suelen tratarse de tablas estáticas donde el usuario no puede modificar los datos que aparecen en ellas. Que esto sea lo habitual, no significa que no podamos crear tablas que nos permitan modificar los datos de sus celdas. Como creemos que puede tratarse de un recurso muy útil para cualquier **proyecto web**, en nuestro White Paper de este mes os explicaremos cómo podemos crear una tabla editable utilizando el lenguaje de programación PHP, **base de datos MySQL** y jQuery. También utilizaremos el plugin jQuery Tabledit que nos ayudará a hacer este proceso mucho más sencillo.

Serán tres los archivos que vamos a utilizar en nuestro ejemplo, además del plugin comentado anteriormente:

- **index.php**: archivo donde pintaremos la tabla con la información traída de la base de datos
- **table_edit.js**: aquí estará el código encargado de hacer editable la tabla
- **editarCelda.php**: el código encargado de guardar en base de datos las modificaciones realizadas en cada una de las celdas
- **db_connect.php**: aquí colocaremos el código encargado de conectarse a la base de datos

Conocidos los archivos que vamos a utilizar, es el momento de explicar el proceso.

Creación de la base de datos

Lo primero que tendremos que hacer será crearnos la tabla donde recuperaremos los datos que se mostrarán en la tabla editable que vamos a crear. Nosotros hemos creado una tabla de nombre “**libro**” donde se almacena el nombre, el autor, su ISBN y su categoría, además de un ID que actúa como identificador.

SQL

```
CREATE TABLE `libro` (  
  `id` INT NOT NULL AUTO_INCREMENT ,  
  `nombre` VARCHAR(255) NOT NULL ,  
  `autor` VARCHAR(255) NOT NULL ,  
  `isbn` VARCHAR(50) NOT NULL ,  
  `categoria` VARCHAR(255) NOT NULL ,  
  PRIMARY KEY (`id`)) ENGINE = InnoDB;
```

Para el ejemplo hemos tomado una tabla con pocos campos, pero ésta podría tener todos los que queramos.

Archivo db_connect.php

Este archivo será donde tendremos el código que utilizaremos para conectarnos a la base de datos. El código sería el siguiente:

PHP

```
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "libros";
$conn = mysqli_connect($servername, $username, $password, $dbname) or
die("Connection failed: " . mysqli_connect_error());
if (mysqli_connect_errno()) {
    printf("Connect failed: %s\n", mysqli_connect_error());
    exit();
}
```

Aquí lo que tenemos son una serie de variables donde almacenaremos los datos de nuestra conexión. Para establecerla, utilizaremos la función “**mysqli_connect**” a la que le pasaremos nuestros datos. En caso de producirse algún tipo de error, lo lanzamos por pantalla.

Código archivo index.php

Una vez creada la base de datos, empezaremos con el código que tendremos en el archivo **index.php**. En este archivo, lo primero que haremos será incluir el archivo que realiza la conexión.

PHP

```
include_once("db_connect.php");
```

El siguiente paso será añadir llamada a la librería jQuery que necesitamos para crear nuestra tabla editable.

HTML

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
```

Lo siguiente será añadir el código que se encargará de pintar la tabla con los datos. El código sería el siguiente.

HTML

```
<div class="container home">
  <h2>Ejemplo tabla editable con PHP, MySQL y jQuery</h2>
  <table id="data_table" class="table table-striped">
    <thead>
      <tr>
        <th>Id</th>
        <th>Nombre</th>
        <th>Autor</th>
        <th>ISBN</th>
        <th>Categoría</th>
      </tr>
    </thead>
    <tbody>
      <?php
        $sql_query = "SELECT id, nombre, autor, isbn, categoria FROM libro";
        $resultset = mysqli_query($conn, $sql_query) or die("error base de
datos:". mysqli_error($conn));
        while( $libro = mysqli_fetch_assoc($resultset) ) {
          ?>
          <tr id="<?php echo $libro ['id']; ?>">
            <td><?php echo $libro ['id']; ?></td>
            <td><?php echo $libro ['nombre']; ?></td>
            <td><?php echo $libro ['autor']; ?></td>
            <td><?php echo $libro ['isbn']; ?></td>
            <td><?php echo $libro ['categoria']; ?></td>
          </tr>
          <?php } ?>
        </tbody>
      </table>
    </div>
```

Como podéis ver, se trata del código habitual que utilizamos para crear las tablas con la diferencia de que al tag `<table>` le hemos indicado como `id` el valor `data-table`. Este `id` será el que utilizemos para hacer la tabla editable por medio del plugin `Tabledit`.

Si nos fijamos en el código anterior, nos daremos cuenta de que realizamos una consulta a **base de datos** para traernos los registros de la tabla `libro`.

PHP

```
<?php
  $sql_query = "SELECT id, nombre, autor, isbn, categoria FROM libro";
  $resultset = mysqli_query($conn, $sql_query) or die("error base de datos:".
mysqli_error($conn));
```

Mediante un bucle “**while**”, los iremos recorriendo para ir pintando cada una de las filas que forman parte de la tabla.

PHP

```
while( $libro = mysqli_fetch_assoc($resultset) ) {
?>
    <tr id="<?php echo $libro ['id']; ?>">
    <td><?php echo $libro ['id']; ?></td>
    <td><?php echo $libro ['nombre']; ?></td>
    <td><?php echo $libro ['autor']; ?></td>
    <td><?php echo $libro ['isbn']; ?></td>
    <td><?php echo $libro ['categoria']; ?></td>
    </tr>
<?php } ?>
```

Código archivo table_edit.js

Lo siguiente que veremos será el código del archivo “**table_edit.js**” que se encargará de hacer editable nuestra tabla. Para ello, recordemos que utilizaremos el plugin para jQuery Tabledit. El código de este archivo será el siguiente.

JavaScript

```
$(document).ready(function() {
    $('#data_table').Tabledit({
        deleteButton: false,
        editButton: false,
        columns: {
            identifier: [0, 'id'],
            editable: [[1, 'nombre'], [2, 'autor'], [3, 'isbn'], [4, 'categoria']]
        },
        hideIdentifier: true,
        url: 'editarCelda.php'
    });
});
```

Lo primero que vemos es que este código no se ejecutará hasta que no se haya cargado todo el DOM de la página. Esto es controlado por la primera línea de código.

Luego, lo que se hace es recuperar la referencia a nuestra tabla (recordemos que le habíamos indicado como “id” el valor “data_table”). A esa referencia será a la que se le aplique las funcionalidades del plugin Tabledit. Este plugin tiene una serie de características que pasamos a detallar:

- **deleteButton**: indica si queremos que se muestre un botón de borrado o no
- **editButton**: similar al caso anterior pero con el botón de editar
- **columns**: sección donde se indica las columnas que tendrá la tabla. Aquí hay que diferencia de la que actuará como identificadas y aquellas que son editables

- **hideIdentifier:** Si le asignamos el valor “true”, no se mostrará en la tabla la columna que hacer referencia al identificador.
- **url:** Archivo PHP que se ejecutará para guardar en base de datos la información modificada.

Como podéis ver, es un código muy sencillo de implementar.

Código archivo editarCelda.php

Por último tenemos el archivo editarCelda.php que será el encargado de guardar en base de datos los cambios realizados en las celdas de la tabla. El código es el siguiente.

PHP

```
include_once("db_connect.php");
$input = filter_input_array(INPUT_POST);
if ($input['action'] == 'edit') {
    $update_field='';
    if(isset($input['nombre'])) {
        $update_field.= "nombre='". $input['nombre']. "'";
    } else if(isset($input['autor'])) {
        $update_field.= "autor='". $input['autor']. "'";
    } else if(isset($input['isbn'])) {
        $update_field.= "isbn='". $input['isbn']. "'";
    } else if(isset($input['categoria'])) {
        $update_field.= "categoria='". $input['categoria']. "'";
    }
    if($update_field && $input['id']) {
        $sql_query = "UPDATE libro SET $update_field WHERE id='". $input['id']
        . "'";
        mysqli_query($conn, $sql_query) or die("database error:".
        mysqli_error($conn));
    }
}
```

Aquí, lo primero que nos encontramos, es la llamada al archivo de conexión a la base de datos que hemos explicado anteriormente. Sin esta llamada no podremos actualizar la información en base de datos.

Lo siguiente es recuperar la información que hay que modificar. Para ello hacemos uso de la función de PHP “**filter_input_array**” que se encarga de recuperar todos los valores que se envían por GET, POST, COOKIE... En nuestro caso la información se envía mediante el método POST, de ahí que se utilice el valor “INPUT_POST”. Los valores recuperados son almacenados en un array de nombre “**\$input**”.

PHP

```
$input = filter_input_array(INPUT_POST);
```

El siguiente paso será comprobar que la acción que se ha enviado es la de editar. Si no fuera así, no se ejecutaría el resto del código.

PHP

```
if ($input['action'] == 'edit') {  
    .....  
}
```

Si realmente la acción a realizar es la de editar la información, lo que haremos será detectar el nombre del campo a modificar.

PHP

```
$update_field='';  
if(isset($input['nombre'])) {  
    $update_field.= "nombre='".$input['nombre']."'";  
} else if(isset($input['autor'])) {  
    $update_field.= "autor='".$input['autor']."'";  
} else if(isset($input['isbn'])) {  
    $update_field.= "isbn='".$input['isbn']."'";  
} else if(isset($input['categoria'])) {  
    $update_field.= "categoria='".$input['categoria']."'";  
}
```

Por último, se modifica la información en base de datos.

PHP

```
if($update_field && $input['id']) {  
    $sql_query = "UPDATE libro SET $update_field WHERE id='" . $input['id'] .  
    "'";  
    mysqli_query($conn, $sql_query) or die("database error:".  
mysqli_error($conn));  
}
```

Como hemos podido ver a lo largo de nuestro [White Paper](#), disponer de una tabla editable en cualquier proyecto web es sencillo gracias al uso del plugin Tabledit para jQuery.