

## PHP y bases de datos MySQL

### I. OBSERVACIONES PREVIAS

---

La extensión `mysql` de PHP es la extensión diseñada originalmente para desarrollar aplicaciones PHP que interactúan con bases de datos MySQL utilizando una interfaz procedural. Esta extensión está pensada para usarla sólo con versiones de MySQL anteriores a la 4.1.3 y, aunque también se puede utilizar con versiones de MySQL 4.1.3 o posteriores, en estos casos no estarán disponibles todas las nuevas funcionalidades del servidor MySQL.

La extensión `mysqli` (`mysql improved`, o MySQL mejorada) permite acceder a la funcionalidad proporcionada por MySQL 4.1 y versiones posteriores, y se encuentra incluida en las versiones PHP 5 y posteriores. Algunas de las características que presenta respecto a la extensión `mysql` son:

- o Interfaz orientada a objetos
- o Soporte para Declaraciones Preparadas
- o Soporte para Múltiples Declaraciones
- o Soporte para Transacciones
- o Mejoras en las opciones de depuración
- o Soporte para servidor empujado

### II. CONEXIÓN DESDE PHP CON BASES DE DATOS MySQL

---

La tarea de obtener registros de una base de datos utilizando PHP se basa en cuatro pasos:

- Conectar con el servidor y seleccionar la base de datos
- Ejecutar una instrucción de SQL
- Procesar la información
- Cerrar la conexión con el servidor

#### Paso 1: Conectar con el servidor y seleccionar la base de datos MySQL

Dado que en un mismo servidor pueden coexistir varias bases de datos, es necesario además de conectar con el servidor MySQL decir a qué BD nos vamos a conectar en concreto:

**Sintaxis:**

```
mysql_connect (servidor, usuario, clave);
```

Si la conexión tiene éxito devuelve un identificador, en otro caso devuelve false.

```
mysql_select_db (baseDeDatos, identificadorDeConexión);
```

Si la selección tiene éxito devuelve true, en otro caso devuelve false

Ejemplo:

```
$conexion = mysql_connect ("localhost", "usuario", "clave");  
$abreBD = mysql_select_db ("SIBW_bd", $conexion);
```

Podría omitirse la variable \$abreBD, pero no se aconseja:

```
mysql_select_db ("SIBW_bd", $conexion);
```

**Sintaxis:**

```
mysql_query (instrucciónSQL, identificadorDeConexión);
```

Si la instrucción se ejecuta correctamente devuelve true o las filas afectadas por la instrucción, dependiendo de la instrucción ejecutada, en otro caso devuelve false

## Paso 2: Ejecutar una instrucción de SQL

Ejemplo:

```
$seleccion = "SELECT * FROM productos";  
$resultado = mysql_query ($seleccion, $conexion);
```

## Paso 3: Procesar la información

Los contenidos de la variable \$resultado son de lo más diverso, pues depende del tipo de orden SQL que se haya ejecutado y de si ésta devuelve una única fila o varias. Por ello hay varias funciones asociadas.

**Sintaxis:**

```
mysql_num_rows ($resultado);
```

# Devuelve el número de filas afectadas por la instrucción ejecutada anteriormente

```
mysql_fetch_array ($resultado);
```

# Cada vez que se ejecuta esta instrucción extrae una fila del resultado como un array asociativo, o como un array numérico o como ambos; la referencia a los campos podría ser de una de las dos formas siguientes:

```
$fila["nombreDeCampo"] // siendo un array asociativo  
$fila[$indice]         // siendo $indice el índice del campo  
desde 0
```

```
mysql_fetch_assoc ($resultado);
```

# Cada vez que se ejecuta esta instrucción extrae una fila del resultado como un array asociativo

```
mysql_fetch_row ($resultado);
```

# Cada vez que se ejecuta esta instrucción extrae una fila del resultado como un array numérico

```
mysql_fetch_object ($resultado);
```

# Cada vez que se ejecuta esta instrucción extrae una fila del resultado como un objeto

```
mysql_data_seek ($resultado, númeroDeFila);
```

# Mueve el puntero de filas interno del resultado de MySQL asociado con el identificador de resultado especificado para apuntar al número de fila especificada mediante un valor o una variable; la siguiente llamada a una función de obtención de MySQL, tal como mysql\_fetch\_array() , devolverá esa fila

Ejemplo utilizando como índices valores numéricos en lugar de como array asociativo:

```
$numFilas = mysql_num_rows ($resultado);

if ($numFilas> 0) {
    while ( $fila = mysql_fetch_array ($resultado) ) {
        //procesamos una fila del resultado:
        echo " Nombre de producto: " . $fila[0] . " <br>";
        echo " Origen: " . $fila[1] . " <br>";
    }
}
```

#### Paso 4: Cerrar la conexión con el servidor

**Sintaxis:**

```
mysql_close (identificadorDeConexión);
```

Ejemplo:

```
mysql_close ($conexion);
```

#### EJEMPLO COMPLETO

Crear un archivo denominado PruebaMySQL.php, poner en él el código siguiente y ejecutarlo en el navegador.

```
<html>
<head>
<title>Ejemplo de acceso a MySQL con PHP</title>
</head>
<body>
    <?php
        // Conectamos con el servidor
        $conexion = mysql_connect ('localhost', 'root', '')
            or exit('No se pudo conectar con el servidor');

        // Abrimos la base de datos
        $abreBD = mysql_select_db ('SIBW_bd', $conexion);
        if (!$abreBD) {
            die('No se pudo abrir la base de datos.Error: ' .
mysql_error());
        }

        // Ejecutamos una consulta
        $seleccion = 'SELECT * FROM Productos';
        $resultado = mysql_query ($seleccion, $conexion);

        // Averiguamos cuántas filas ha devuelto la consulta
        $numFilas = mysql_num_rows ($resultado);

        // Si la consulta ha devuelto filas, las procesamos
        if ($numFilas> 0) {
```

```
// Creamos una tabla donde se pone una fila con los nombres de
los campos
echo '<center>'; // Así centramos la tabla>
echo '<table border=1>';
echo '<tr><th>Producto</th><th>Origen</th></tr>';

// Añadimos a la tabla las filas de datos que haya devuelto la
consulta
while ( $fila = mysql_fetch_array ($resultado) ) {
    echo "<tr><td>$fila[0]</td><td>$fila[1] </td></tr>";
}

// Cerramos la tabla y quitamos el centrado de la misma
echo '</table>';
echo '<center>';
}

// Cerramos la conexión con el servidor
mysql_close ($conexion);
?>
</body>
</html>
```

### III. PHPMYADMIN

#### 1. Características generales

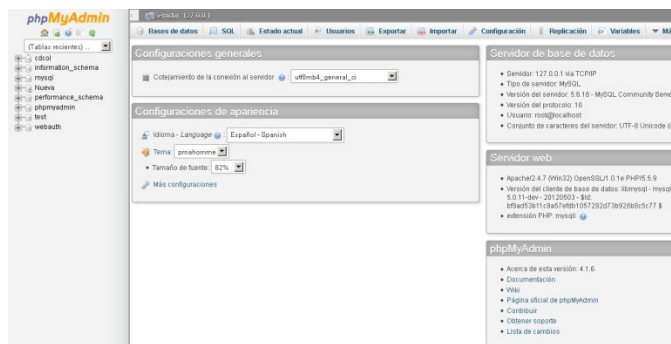
Es una herramienta de libre distribución para administrar el servidor de bases de datos MySQL que proporciona una interfaz gráfica que se ejecuta desde el navegador y permite realizar operaciones de administración de bases de datos locales y remotas:

- o Creación, borrado y modificación de tablas
- o Inserción, modificación, consulta y eliminación de datos
- o Definición de usuarios y asignación de permisos
- o Realización de copias de seguridad, ...

Está incluida en el paquete de software xampp, aunque hay versiones independientes de este paquete. En la versión incluida en xampp se puede ejecutar escribiendo en el navegador:

<http://localhost/phpmyadmin/>

La imagen que se tiene inicialmente es similar a esta:



## 2. Algunas operaciones básicas de gestión de bases de datos

### 2.1 Creación de una base de datos

Seleccionar la ficha Bases de datos, introducir el nombre de la base de datos en el campo reservado para ello, dejar el idioma con la opción por defecto (Cotejamiento) y pulsar Crear. En nuestro ejemplo, la base de datos será SIBW\_bd:

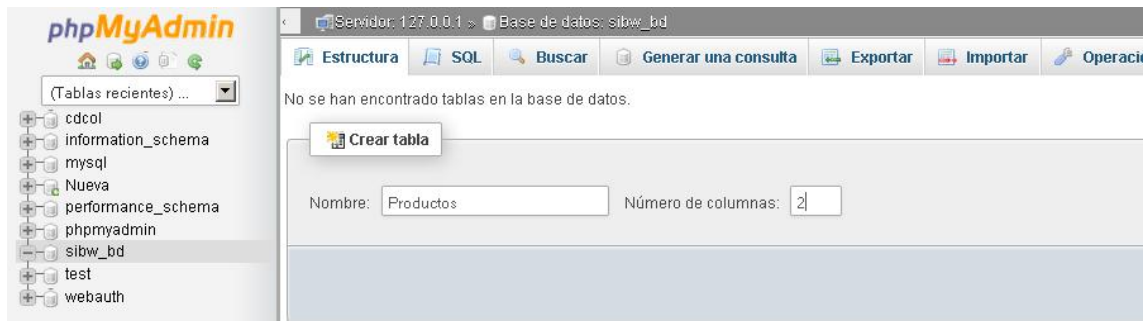


La nueva base de datos aparecerá en la lista de la izquierda y en la de debajo:

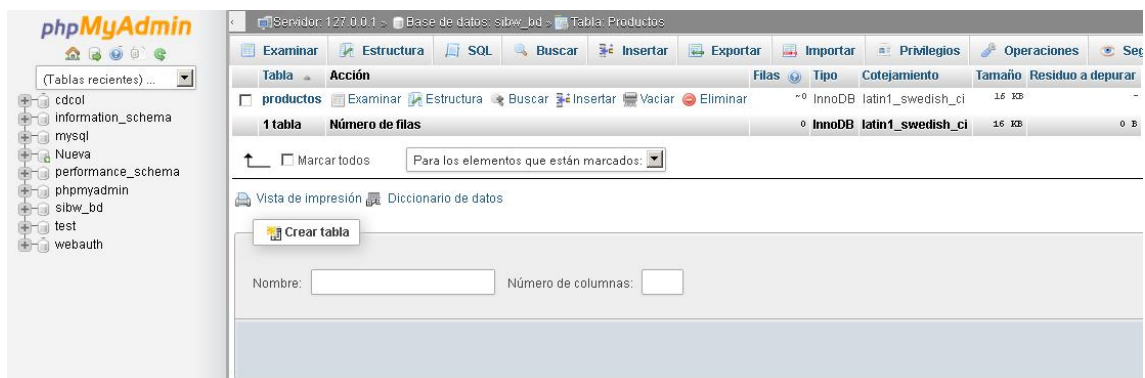


### 2.2 Creación de una tabla

Pulsar sobre el nombre de la base de datos en la lista de la izquierda. Se abrirá la ficha Estructura con un aviso advirtiéndolo que no se han encontrado tablas en la base de datos y mostrará la opción Crear tabla. Introducir entonces el nombre de la tabla y el número de columnas de la misma, y pulsar Continuar. En nuestro ejemplo, Productos y 2:

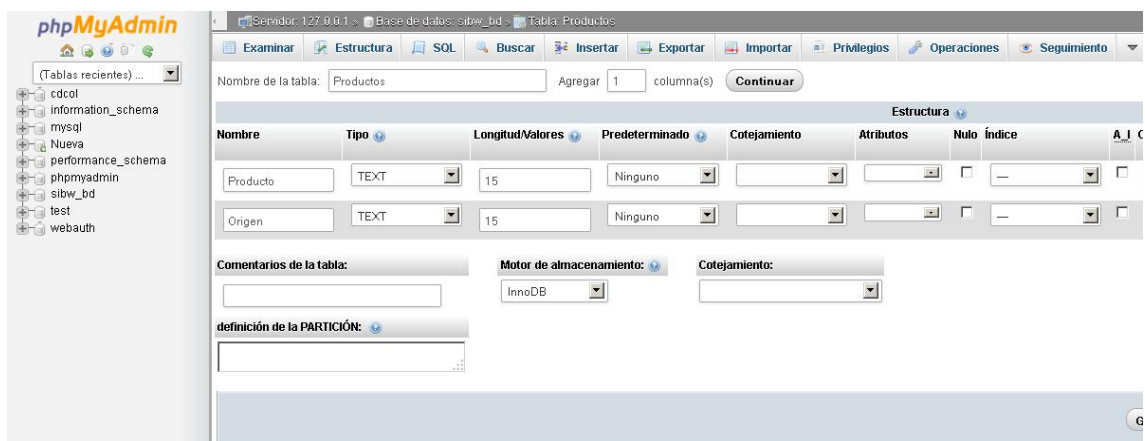


Tras la creación aparecerá una ventana con el nombre de la tabla y posibles acciones sobre ella:



## 2.3 Creación de campos en una tabla

Después de crear la tabla se muestra una nueva ventana donde se pueden introducir todos los campos de la tabla con las características que se deseen. También se pueden añadir más columnas a la tabla. En nuestro ejemplo bastará con añadir dos campos, Producto y Origen, de tipo TEXT y longitud 15, y pulsar Guardar.



## 2.4 Insertar filas en una tabla

Junto al nombre de la tabla creada aparecen varias opciones. Seleccionar Insertar para introducir filas en la tabla. Se abre una nueva ventana donde se pueden introducir valores para los diferentes campos y también se puede elegir entre Volver o Insertar un nuevo registro, según se desee volver a la pantalla de opciones sobre la tabla o se quiera seguir introduciendo registros en la tabla. La inserción se efectúa pulsando Continuar; si esto se hace con la opción Volver, también nos mostrará el código SQL correspondiente a la acción de inserción.

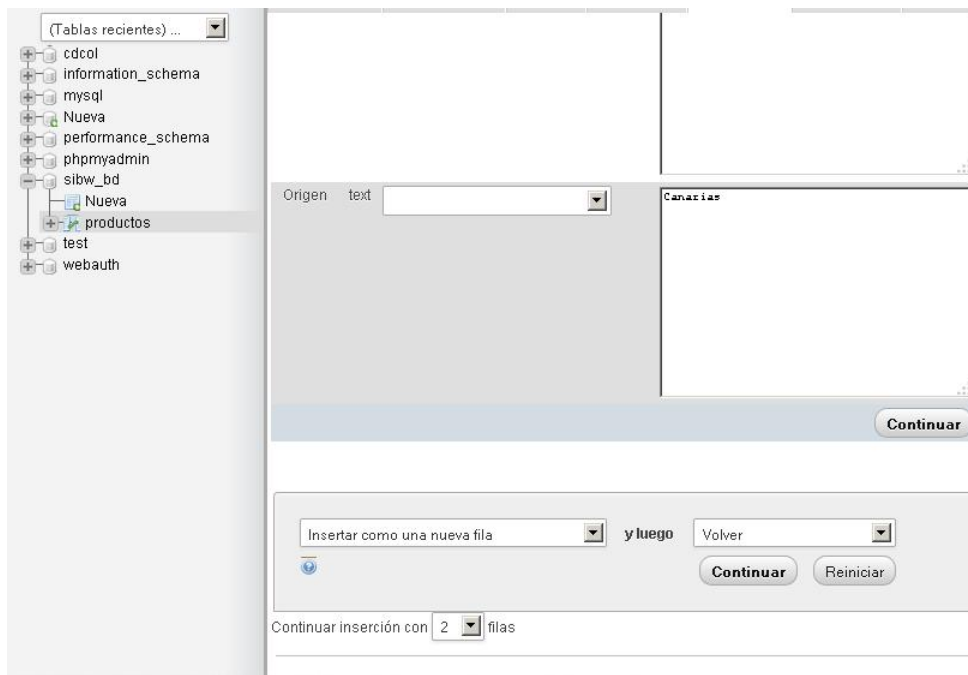
### Advertencias:

- Antes de confirmar la inserción hay que asegurarse de haber marcado correctamente, en la parte de abajo, el número de filas que se desea insertar para evitar que se introduzcan filas con campos vacíos; por defecto, las filas de una tabla se introducen de dos en dos.
- Si se han introducido filas con valores nulos, se pueden eliminar utilizando la opción SQL que aparece junto al nombre de la tabla y ejecutando allí una instrucción del tipo DELETE. En el caso de nuestro ejemplo sería:

```
DELETE FROM `productos` WHERE producto = ''
```

Un ejemplo de inserción de filas en la tabla Productos es el que se ve en las imágenes siguientes:

The screenshot shows the phpMyAdmin interface for the 'productos' table. The 'Insertar' (Insert) form is displayed, allowing users to add new records. The form is divided into two main sections. The top section contains the 'Producto' field (a dropdown menu) and the 'Origen' field (a text input). The bottom section contains the 'Producto' field (a dropdown menu). The 'Continuar' (Continue) button is located at the bottom right of the form. The left sidebar shows the database structure, including the 'productos' table.



## 2.5 Otras opciones para una tabla

- **Examinar:** muestra los registros insertados
- **Estructura:** presenta los diferentes campos de la tabla con opciones de manipulación, así como información general de la misma.
- **SQL:** permite realizar operaciones de SQL con un asistente.
- **Buscar:** permite utilizar diversos filtros para buscar datos en la tabla.
- **Exportar:** exporta filas de la tabla en diferentes formatos.
- **Importar:** inserta filas en la tabla desde archivos de diversos tipos.
- **Privilegios:** muestra los privilegios de los usuarios para acceder a la tabla y permite hacer modificaciones.
- **Operaciones:** permite ordenar la tabla, moverla o copiarla, así como otras acciones de mantenimiento.
- **Seguimiento:** deja elegir cuáles son las sentencias de definición y de manipulación de datos para las que se hará un seguimiento, permitiendo crear una nueva versión de la tabla para volver a ella posteriormente si se desea.
- **Disparadores:** muestra los disparadores asociados a la tabla; también permite crearlos.

## IV. ENLACES INTERESANTES

<http://es.php.net/manual/es/book.mysql.php>

<http://es.php.net/manual/en/book.mysql.php>

<http://es.php.net/manual/es/set.mysqlinfo.php>

[http://www.phpmyadmin.net/home\\_page/docs.php](http://www.phpmyadmin.net/home_page/docs.php)