



VICERRECTORIA ACADEMICA  
DIRECCION DE EXTENSION UNIVERSITARIA  
AREA DE COMUNICACION Y TECNOLOGIA



2017

# Unidad No5. MySQL y PHP

70688 – Programación en PHP

Área de comunicación y tecnología  
UNIVERSIDAD ESTATAL A DISTANCIA

## **Tabla de contenidos**

Introducción .....	3
Objetivos de Aprendizaje .....	3
Objetivo general .....	3
Objetivos específicos.....	3
5.  MySQL y PHP .....	4
5.1  Fundamentos de MySQL .....	4
5.1.1  ¿Qué es MySQL? .....	4
5.2  Crear una base de datos en MySQL .....	4
5.2.1  Crear tablas en MySQL .....	7
5.2.2  Exportación e importación del script de base de datos.....	10
5.3  MySQL y PHP .....	11
5.3.1  Conexión a MySQL desde PHP .....	11
5.3.2  Interacción formularios PHP con MySQL .....	12
Conclusión .....	19
Bibliografía.....	19

## **Introducción**

Las aplicaciones web por definición son dinámicas y un componente de gran importancia es la base de datos, ya que no solo brinda el servicio de persistencia de la información sino que puede servir para administrar la configuración de una aplicación.

En esta unidad se estudiará MySQL, un motor de base de datos muy relacionado con PHP y la interacción entre estos dos componentes de software.

## **Objetivos de Aprendizaje**

### **Objetivo general**

- Construir bases de datos en MySQL con el fin de unirla con páginas web dinámicas creadas con PHP.

### **Objetivos específicos**

- Describir qué es MySQL y sus principales características.
- Crear una base de datos.
- Identificar el procedimiento de respaldo de estructura de base de datos.
- Conectar una aplicación PHP con la base de datos para la recuperación de información desde la base de datos.

## 5. MySQL y PHP

### 5.1 Fundamentos de MySQL

#### 5.1.1 ¿Qué es MySQL?

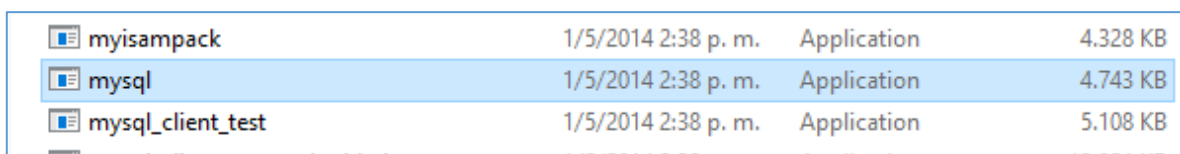
MySQL es un motor de base de datos que presenta dos clases principales de distribución:

- Una distribución gratuita
- Una versión con soporte llamada “Enterprise”.

El motor en sí es una aplicación relativamente pequeña en comparación con otros motores, como por ejemplo Oracle, pero que ofrece buenas prestaciones a nivel de rendimiento y funciones a realizar, además de la compatibilidad en una cantidad mayor de sistemas operativos [1].

### 5.2 Crear una base de datos en MySQL

Si usted realizó la instalación sin usar WAMP diríjase a la carpeta de instalación de MySQL y busque la carpeta “bin”, en ella se encuentran los ejecutables necesarios para MySQL y ejecute el archivo “mysql”, mostrado en la figura 5-1.



A screenshot of a file explorer window showing a list of files and folders. The 'mysql' folder is selected and highlighted in blue. The list includes 'myisampack', 'mysql', and 'mysql\_client\_test', all with a date of 1/5/2014 2:38 p. m. and a size of 4.328 KB, 4.743 KB, and 5.108 KB respectively.

File Name	Date Modified	Type	Size
myisampack	1/5/2014 2:38 p. m.	Application	4.328 KB
mysql	1/5/2014 2:38 p. m.	Application	4.743 KB
mysql_client_test	1/5/2014 2:38 p. m.	Application	5.108 KB

Figura 5-1. Ejecutable de MySQL.

Si usted utilizó el “WAMP” entonces inicie la aplicación WAMP, cuando este el widget de la “W” (ubicado en la parte del reloj) en color verde (figura 5-2) haga clic en el widget y seleccione "MySQL Console" (figura 5-3), esto abre la consola de MySQL.



Figura 5-2. Indicador de ejecución de WAMP

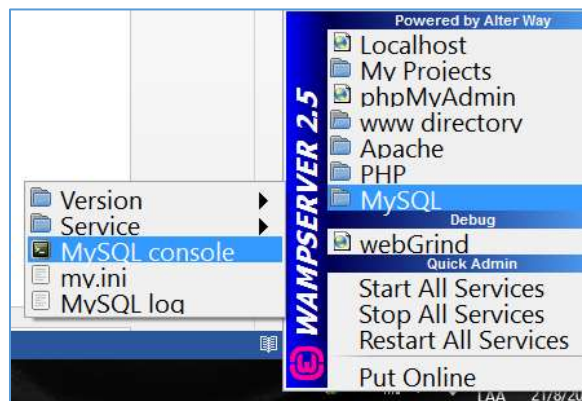


Figura 5-3. Inicio de consola de MySQL con WAMP.

De cualquiera de las dos maneras anteriores, se abre una terminal o consola con el “prompt” en escucha de MySQL (figura 5-4), desde ahí se pueden crear bases de datos, gestionarlas y eliminarlas.

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql>
```

*Figura 5-4. Terminal de gestión MySQL*

Si usted colocó una clave en el proceso de instalación de MySQL se le pedirá, de lo contrario el prompt para MySQL aparecerá de manera inmediata.

Para crear la base de datos se debe ejecutar el comando “create database” con el nombre de la base de datos (en este caso **inventario**) que deseamos crear, es decir, al ejecutar el comando:

```
create database inventario;
```

Obtendríamos en MySQL el resultado de “database created” tal y como se muestra en la figura 5-5, el cual quiere decir que la base de datos se creó correctamente.

```
mysql> create database inventario;  
Query OK, 1 row affected (0.00 sec)
```

*Figura 5-5. Ejecución de creación de base de datos MySQL.*

Como puede observar al igual que PHP las sentencias en MySQL terminan con punto y coma.

Con este simple comando ya se tiene la base de datos creada.

### 5.2.1 Crear tablas en MySQL

Una vez creada la base de datos es necesario indicar a MySQL sobre cual base de datos se desea trabajar, ya que en MySQL pueden existir muchas bases de datos. Para conocer cuales bases de datos existen en un servidor MySQL se ejecuta el comando

*Show databases;*

Un ejemplo de la ejecución de ese comando puede ser visto en la figura 5-6. Donde se muestran los nombres de las bases de datos creadas actualmente.

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| inventario |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.03 sec)
```

*Figura 5-6. Ejemplo de ejecución de consulta sobre base de datos existentes.*

Para seleccionar sobre cuál base de datos se trabajará se utiliza el comando **use** (figura 5-7) y ya con eso todo lo que se haga, hasta no replicar el comando con otra base de datos se hará sobre la base de datos elegida.

```
mysql> use inventario;
Database changed
```

*Figura 5-7. Selección de base de datos MySQL*

Para crear las tablas en la base de datos se usa el comando `create table` acompañado de los campos que compondrán la tabla y sus argumentos, tal como se presenta a continuación:

```
Create table (  
  
    campo1 tipo_de_dato(longitud) argumentos,  
  
    campo2 tipo_de_dato(longitud) argumentos,  
  
    campo3 tipo_de_dato(longitud) argumentos);
```

Los tipos de dato que puede tener un campo en la tabla pueden ser los indicados en la tabla 5-1 [2] y los argumentos son mostrados en la tabla 5-2.

TIPO DE DATO	DESCRIPCION
BIT	Indica un valor de 1 a 64
TINYINT	Valor de -128 a 127
SMALLINT	Valor de - 32768 a 32767
MEDIUMINT	Valor de -8388608 a 8388607
INTEGER o INT	Valor de -2147483648 a 2147483647
DECIMAL (M, D)	Numérico con decimales, M la longitud del entero, D la cantidad de decimales.
FLOAT (M, D)	Igual comportamiento con DECIMAL pero con rango mayor de -3.402823466E+38 a 3.402823466E+38.
DATETIME	Fecha y hora
YEAR	Almacena solo un año.
CHAR(N)	Cadena de caracteres de longitud "N" aunque no se utilicen todos los caracteres.
VARCHAR (N)	Cadena de caracteres con codificación UTF-8. Como máximo usa "N" caracteres pero si se usa menos se ajusta al tamaño.
BLOB	Columna con tamaño de 255 (28 – 1) bytes.

Tabla 5-1 Tipos de dato MySQL



ARGUMENTOS	DESCRIPCION
ENGINE	Indica un valor de 1 a 64
AUTO-INCREMENT	Aumenta automáticamente el valor del campo que lo tiene como una secuencia. Ideal para llaves primarias numerales.
COMMENT	Es el comentario del campo.
NULL	Acepta que el campo no tenga valor.
NOT NULL	El campo debe tener obligatoriamente un valor.

*Tabla 5-2. Tipos de dato MySQL*

Además de los atributos se puede usar otras cualidades de la tabla, como definición de llave primaria y llaves foráneas por medio de constraints.

Por ejemplo en los siguientes scripts (script 5-1 y 5-2) permiten crear las tablas para completar la base de datos.

En el script 5-1, se crea la tabla categoriaProducto, con los campos idCategoria de tipo int y el argumento es que no sea nulo, posterior se agrega otro campo llamado descripción tipo varchar y al final se indica que el campo idCategoria es una llave primaria.

```
create table categoriaProducto(  
    idCategoria int(4) not null auto_increment,  
    descripcion VARCHAR(35) not null,  
    PRIMARY KEY (idCategoria));
```

*Script 5-1. Creación tabla categoriaProducto*

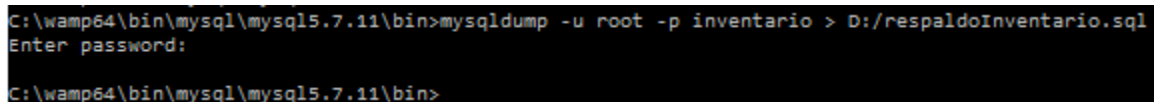
En el script 5-2 se crea la tabla producto, con los campos idProducto, idCategoria, descripción y precioUnitario, con la llave primaria idProducto, la llave foránea idCategoria que hace referencia a la tabla categoriaProducto, específicamente conectado con el campo idCategoria.

```
create table producto(  
  idProducto int(5) not null,  
  idCategoria int(4) not null,  
  descripcion varchar(25) not null,  
  precioUnitario float(5,2) not null,  
  PRIMARY KEY (idProducto),  
  INDEX idxIdCat (idCategoria),  
  FOREIGN KEY (idCategoria)  
  REFERENCES categoriaProducto(idCategoria)  
  ON DELETE CASCADE);
```

*Script 5-2. Creación tabla producto*

### 5.2.2 Exportación e importación del script de base de datos

En una terminal o consola, navegue hasta la carpeta de instalación de MySQL y dentro de ella ingrese a la carpeta bin. Ejecute el programa “mysqldump” seguido del usuario root, el indicador de password y la base de datos a respaldar (figura 5-8), luego de esto se coloca la ruta y nombre del archivo que va a contener el volcado de la base de datos.



```
C:\wamp64\bin\mysql\mysql5.7.11\bin>mysqldump -u root -p inventario > D:/respaldoInventario.sql  
Enter password:  
C:\wamp64\bin\mysql\mysql5.7.11\bin>
```

*Figura 5-8. Volcado de base de datos MySQL.*

Esto creará un archivo con las sentencias SQL para creación de base de datos y registros en la misma.

Para efectos de replicar la información de nuevo solamente se ingresa a MySQL, se copia el contenido del archivo y se pega en la consola de MySQL.

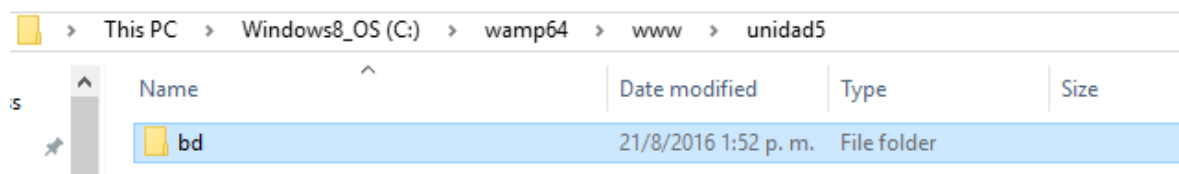
## 5.3 MySQL y PHP

### 5.3.1 Conexión a MySQL desde PHP

La conexión a MySQL con PHP no requiere complementos especiales, dado que el núcleo de PHP ya contiene el driver necesario para conectarse con bases de datos en MySQL.

Existen distintos diseños o maneras en que se administra la conexión hacia MySQL, una manera es mediante conexión / desconexión que es la que se verá en esta unidad.

Bajo la estructura del servidor Apache, en la carpeta “www” genere una nueva carpeta con el nombre del sitio web que desee, y bajo este, una carpeta llamada “bd” (figura 5-9).



*Figura 5-9. Estructura del sitio web a trabajar*

Bajo esta carpeta “bd” se alojará un archivo que contendrá la conexión a la base de datos y las funciones relacionadas.

La manera de conectarse a una base de datos MySQL (figura 5-10) es mediante el uso de la función `mysqli_connect` enviando la dirección IP del servidor, usuario, clave y nombre de base de datos.

```
function conectarBD(){  
    $conexion = mysqli_connect('127.0.0.1', 'root', '', 'inventario');  
    if (!$conexion) {  
        return;  
    }  
  
    return $conexion;  
}
```

*Figura 5-10. Conexión a MySQL desde PHP.*

La conexión se maneja siempre en una variable. De la misma manera que se hizo la conexión es posible desconectarse de la base de datos (figura 5-11) mediante el uso de la función `mysqli_close`.

```
function desconectarBD($conexion){  
    mysqli_close($conexion);  
}
```

*Figura 5-11. Desconexión a MySQL desde PHP.*

Ambas funciones están provistas como se dijo por el núcleo de PHP por lo que solo es necesario llamarlos no se deben desarrollar.

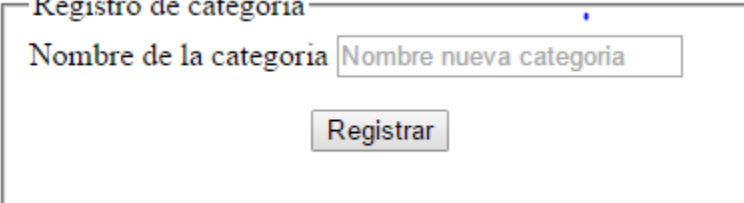
### 5.3.2 Interacción formularios PHP con MySQL

Una vez realizada la conexión y desconexión a la base de datos, se puede usar esta desde funciones PHP para mediante sentencias de SQL hacer las operaciones necesarias en la base de datos.

El formulario de la imagen 5-12 y 5-13 contiene solamente el nombre de una categoría nueva a ser registrada en la base de datos, pero la acción la realiza una página php distinta:

```
<html>
<head>
</head>
<body>
  <fieldset style="width:350px">
    <legend>Registro de categoria</legend>
    <form action="procesoRegistroCatProducto.php" method="post">
      Nombre de la categoria <input type="text" name="nombreCat" placeholder="Nombre nueva categoria"/>
    <p/>
    <center><input type="submit" value="Registrar"></center>
  </form>
</fieldset>
</body>
</html>
```

Figura 5-12 Script del formulario



Registro de categoria

Nombre de la categoria

Figura 5-13 Ejecución del formulario

La función para almacenamiento de la categoría viene a conectar la base de datos, hacer la sentencia “sql” de insertar y cierra la conexión como se muestra en la imagen 5-14.

```
function registrarCategoria($descripcion){
    $conexion = conectarBD();

    if ($conexion){
        $sql = "insert into categoriaproducto (descripcion) values ('".$descripcion."')";

        $resultado = mysqli_query($conexion, $sql);

        desconectarBD();

        return $resultado;
    }else{
        desconectarBD();
        return false;
    }
}
```

Figura 5-14 Función de registro

Como se puede observar en la imagen 5-14 lo primero que se hace es realizar la conexión a la base de datos, si esta se da, se procede a crear el “sql” de inserción y a ejecutarlo con el método `mysqli_query` el cual devuelve “true” si el registro fue exitoso, “false” de lo contrario. Luego se desconecta de la base de datos y se retorna el resultado.

La página que procesa la solicitud (`procesoRegistroCatProducto.php`) solamente recibe el argumento por el método POST y lo envía a la función (figura 5-15) y según el resultado imprime en pantalla si el registro fue o no exitoso.

```
<?php
include 'bd/funcionesBD.php';

$exitoRegistro = registrarCategoria($_POST["nombreCat"]);

if ($exitoRegistro){
    echo 'Registro exitoso de la categoría '.$_POST["nombreCat"];
}else{
    echo 'Fallo en registro de la categoría '.$_POST["nombreCat"];
}

?>
```

Figura 5-15. `procesoRegistroCatProducto.php`

La ejecución del formulario daría como resultado una pantalla similar a la mostrada en la figura 5-16

Registro exitoso de la categoría Granos basicos

*Figura 5-16. Resultado del registro de categoría de producto.*

Y si se hace un “select” de la tabla en base de datos (figura 5-17) se verá el registro ya persistido en base de datos.

```
mysql> select * from categoriaproducto;
+-----+-----+
| idCategoria | descripcion |
+-----+-----+
|          1 | Granos basicos |
+-----+-----+
1 row in set (0.00 sec)
```

*Figura 5-17. Revisión en base de datos del registro.*

En la figura 5-17 se hace el select de base de datos, la misma sentencia se puede replicar en PHP de manera que se pueda tener la misma información en una página web. Para esto existen diversas maneras, una de ellas es creando el html dentro del mismo método y retornándolo, tal y como se muestra en la imagen 5-18

```
function buscarTodasCategoriasProd(){
    $conexion = conectarBD();
    $datos = "";

    if ($conexion){
        $sql = "select idCategoria, descripcion from categoriaproducto order by descripcion asc";

        $result = $conexion->query($sql);

        if ($result->num_rows > 0) {
            while($row = $result->fetch_assoc()) {
                $datos = $datos.'<tr><td>'.$row["idCategoria"].'</td><td>'.$row["descripcion"].'</td></tr>';
            }
        }else{
            $datos = $datos.'<tr><td colspan="2">No hay datos</td></tr>';
        }
    }else{
        desconectarBD();
    }
    return $datos;
}
```

Figura 5-18. Consulta de base de datos.

La diferencia con respecto a la inserción de registros radica en que no se llama a ningún método para enviar el script de consulta sino que es un puntero de la propia conexión y que retorna los registros que coinciden con lo consultado.

Posteriormente se pregunta si el objeto en la propiedad número de filas es mayor que cero (trae información), de ser así se recorre cada fila en un ciclo y se crea la estructura de filas de una tabla html.

Para completar la presentación se crea una página que llame a la función de la figura 5-18 y complete la estructura de la tabla que hace falta (figuras 5-19 y 5-20).



```
<html>
<head>
</head>
<body>
    <?php
        include 'bd/funcionesBD.php';
    ?>
    <table>
        <tr>
            <th>Codigo</th><th>Descripcion</th>
        </tr>
        <?php echo buscarTodasCategoriasProd();?>
    </table>
</body>
</html>
```

Figura 5-19. Página de resultado.

Codigo	Descripcion
1	Granos basicos

Figura 5-20 Resultado de consulta a base de datos.

Las formas de actualización (update) y eliminación de registros en PHP trabajan de manera similar al de registro, solo que varía la sentencia de SQL a ejecutar (figuras 5-21 y 5-22).

```
function actualizarCategoria($codigo, $descripcion){
    $conexion = conectarBD();

    if ($conexion){
        $sql = "update categoriaproducto set descripcion = '". $descripcion.'" where idCategoria = ".$codigo;

        $resultado = mysqli_query($conexion, $sql);

        desconectarBD();

        return $resultado;
    }else{
        desconectarBD();
        return false;
    }
}
```

Figura 5-20. Actualización de base de datos

```
function eliminarCategoria($codigo, $descripcion){  
    $conexion = conectarBD();  
  
    if ($conexion){  
        $sql = "delete from categoriaproducto where idCategoria = ".$codigo;  
  
        $resultado = mysqli_query($conexion, $sql);  
  
        desconectarBD();  
  
        return $resultado;  
    }else{  
        desconectarBD();  
        return false;  
    }  
}
```

*Figura 5-21. Eliminación de base de datos*

## **Conclusión**

Las bases de datos son hoy por hoy partes fundamentales de cualquier aplicación web, y MySQL brinda una buena opción para la creación de los sitios además de una forma de comunicación con PHP bastante natural.

Algo importante es que se haga uso de todas las características que ofrecen las bases de datos sean cual sean para optimizar y mejorar el rendimiento y la experiencia de usuario.

## **Bibliografía**

[1] N.D. (n.d.). MySQL vs Oracle. Retrieved August 21, 2016, from <https://itxdesign.com/mysql-vs-oracle/>

[2] Chapter 12 Data Types. (n.d.). Retrieved August 21, 2016, from <http://dev.mysql.com/doc/refman/5.7/en/data-types.html>