



# **EXPERTO UNIVERISTARIO EN MySQL Y PHP NIVEL INTERMEDIO**





## Módulo 2

### Mas coceptos sobre Mysql y PHP



## Integrando conceptos.



## **Presentación de la Unidad:**

**En esta unidad trataremos de integrar los conceptos vistos hasta ahora de PHP y Mysql.**

**Sobre todo poniendo en práctica las consultas sql que vimos al comienzo del módulo y utilizándolas desde PHP a través de las funciones que venimos empleando.**



## Objetivos:

- ❖ Integrar consultas que involucren mas de una tabla desde PHP a través de las funciones de la extensión mysqli.



## Temario:

*Repaso de funciones de PHP para acceso a las bases de datos*

*Ejemplo práctico: Panel de administración*



## CONSIGNAS PARA EL APRENDIZAJE COLABORATIVO

En esta Unidad los participantes se encontrarán con diferentes tipos de consignas que, en el marco de los fundamentos del MEC\*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los de aprovecharlas pedagógicamente:



efectos

1. Los foros asociados a cada una de las unidades.
2. La Web 2.0.
3. Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen las actividades sugeridas y compartan en los foros los resultados obtenidos.



## PHP Y MYSQL

Como ya empezáramos a ver, en la parte inicial del curso, desde PHP es posible invocar funciones y ejecutar sentencias específicas de MySQL.

### *Para entender*

La interconexión entre ambos se realiza mediante una serie de instrucciones, a saber `mysqli_query` (`$conexion`, consulta) envía una consulta a la base activa asociada al identificador de enlace `$conexion`.

Devuelve TRUE si la sentencia se ha ejecutado correctamente y FALSE en caso contrario.

La ejecución de esta instrucción requiere, como vimos, que previamente hayamos creado una conexión a un servidor MySQL con la función `mysqli_connect()`, estableciendo en los tres primeros parámetros de la función el host, usuario y password de conexión, lo cual nos permitirá luego interactuar con una base de datos y las tablas que la compongan, y que se haya asociado justamente, una base de datos al identificador de enlace mediante la misma instrucción asignando como cuarto parámetro el nombre de la base.

Vamos a plantear una sencillísima base de datos, para ir ejemplificando cada sentencia, para ello debemos importar la base ejemplo.sql desde nuestro PHPmyAdmin. Allí nos encontraremos con una tabla llamada "clientes" con sólo cuatro campos, id, nombre, actividad y teléfono.

### *Selección y lectura de registros con PHP*

Una vez creada la tabla clientes en nuestra base de datos ejemplo, haremos la consulta de los datos que se encuentran en la tabla.

Utilizamos el comando SELECT de SQL para crear una selección de nuestra tabla y mostrar todos los datos en pantalla por medio de un bucle. Con PHP.

Dentro de una base de datos, organizada por tablas, la selección de una tabla entera o de un cierto número de registros resulta una operación rutinaria.

La siguiente es una forma bastante clásica de mostrar en pantalla a partir de un bucle los registros seleccionados por una sentencia SQL:





```
<html>
<head>
<title>Lectura de datos</title>
</head>
<body>
<caption><h1><div align="center">lectura de la tabla</div></h1></caption>

<?php
//conexion con la base y seleccion de la basede datos
$conexion = mysqli_connect("localhost","root","","ejemplo");

//ejecutamos la sentencia sql
$result=mysqli_query($conexion, "select * from clientes");
?>
<table align="center">
<tr>
<th>Nombre</th>
<th>Actividad</th>
<th>Web</th>
</tr>
<?php
//mostramos los registros
while ($row=mysqli_fetch_array($result))
{
echo '<tr><td>'.$row["nombre"].'</td>';
echo '<td>'.$row["actividad"].'</td>';
echo '<td>'.$row["web"].'</td></tr>';
}
mysqli_free_result($result)
?>
</table>

<div align="center">
<br>
<br>
<a href="insertar.html">añadir un nuevo registro</a><br>
<a href="actualizar1.php">actualizar un registro existente</a><br>
<a href="borrar1.php">borrar un registro</a><br>
</div>

</body>
</html>
```

Los pasos a realizar son, en un principio: Conexión a la base y ejecución de la sentencia.

La información de dicha ejecución será almacenada en una variable (\$result).

El siguiente paso será plasmar en pantalla la información recogida en \$result. Esto lo haremos mediante la función `mysqli_fetch_array` que devuelve una variable array con los contenidos de un registro a la vez que se posiciona sobre el siguiente. El bucle `while` nos permite leer e imprimir secuencialmente cada uno de los registros.

La función `mysqli_free_result` se encarga de liberar la memoria utilizada para llevar a cabo la consulta. Aunque no es necesaria su utilización, resulta altamente aconsejable.

### *Introducción de nuevos registros con PHP*

Para ello usaremos un formulario y en el ACTION del FORM indicaremos que debe ser procesado una página PHP, llamada "insertar.php", esta página lo que hará será introducir los datos del formulario en la base de datos.

```
<form action= "insertar.php" >
```

Entonces, los datos del registro son recogidos a partir de un formulario que recoge los datos y los envía a una página PHP que se encarga de procesarlos.

```
<html>
<head>
<title>Insertar datos</title>
</head>
<body>
<div align="center">
<h1>insertar un registro</h1>
<br/>
<form method="post" action="insertar.php">
nombre<br>
<input type="text" name="nombre"><br>
actividad<br>
<input type="text" name="actividad"><br>
sitio web<br>
<input type="text" name="web"><br>
<input type="submit" value="insertar">
</form>
<a href="lectura.php">Volver al listado de Clientes</a></div>
</body>
```

</html>

Llegados a la página destino del formulario (insertar.php), lo primero que habrá es establecer un vínculo entre el programa y la base de datos. Esta conexión se lleva a cabo como dijimos, con la función `mysqli_connect`. Luego de seleccionar la base a utilizar deberemos generar una orden de inserción del registro en lenguaje SQL.

Esta orden será ejecutada por medio de la función `mysqli_query`:

```
<html>
<head>
<title>Insertar datos</title>
</head>
<body>
<?php
//conexion con la base y seleccion de la base de datos
$conexion = mysqli_connect("localhost","root","","ejemplo");

//traigo los datos del formulario
$nombre = $_POST["nombre"];
$actividad = $_POST["actividad"];
$web = $_POST["web"];

//ejecución de la sentencia sql
mysqli_query($conexion, "insert into clientes (nombre,actividad,web) values
('$nombre','$actividad','$web')");
?>
<h1><div align="center">registro insertado</div></h1>
<div align="center">
<a href="lectura.php">visualizar el contenido de la base</a></div>
</body>
</html>
```

### ***Borrado de un registro con PHP***

Otra de las operaciones elementales que se pueden realizar sobre una base de datos es borrar un registro. Para hacerlo, SQL nos propone sentencias del tipo DELETE.



Cabe señalar que primero debemos seleccionar el registro que se desea borrar y luego realizar el borrado propiamente dicho. Para ello crearemos un menú desplegable dinámico, donde se podrá seleccionar el elemento que se desea borrar. Luego se pasará a una página PHP una referencia al elemento seleccionado, para borrarlo de la base de datos.

```
<html>
<head>
<title>Borrar Datos</title>
</head>
<body>
<div align="center">
<h1>Borrar un registro</h1>

<br>
<?php
//conexion con la base y seleccion de la base de datos
$conexion = mysqli_connect("localhost","root","","ejemplo");

echo '<form method="post" action="borrar2.php">nombre<br>';

//creamos la sentencia sql y la ejecutamos
$sql="select id, nombre from clientes order by nombre";
$result=mysqli_query($conexion, $sql);

echo '<select name="nombre">';

//mostramos los registros en forma de menú desplegable
while ($row=mysqli_fetch_array($result)){
    echo '<option value="'. $row['id']. ">'. $row['nombre']. '</option>';
}
mysqli_free_result($result)
?>

</select>
<br>
<input type="submit" value="borrar">
</form>
<a href="lectura.php">Volver al listado de Clientes</a>
</div>
</body>
</html>
```

El siguiente paso es hacer efectiva la operación a partir de la ejecución de la sentencia SQL que construimos a partir de los datos del formulario:

```
<html>
<head>
<title>Borrar datos</title>
</head>
<body>
<?php
//pasamos los datos del formulario
$id = $_POST["nombre"];

//conexion con la base y seleccion de la base de datos
$conexion = mysqli_connect("localhost","root","","ejemplo");

//creamos la sentencia sql y la ejecutamos
$sql="delete from clientes where id='$id';
mysqli_query($conexion,$sql);
?>
<h1><div align="center">Registro borrado</div></h1>
<div align="center"><a href="lectura.php">Visualizar el contenido de la base</a></div>
</body>
</html>
```

### **Actualización de un registro de base de datos con PHP**

Para mostrar cómo se actualiza un registro presente en nuestra base de datos, vamos a hacerlo también a partir de un formulario a través del cual establecemos el registro que vamos a modificar y luego el script definido en el action del formulario se encargará de ejecutar la sentencia sql encargada de hacer las modificaciones en la tabla.

El archivo del formulario va a ser esta vez un script PHP en el que efectuaremos una llamada a nuestra base de datos para construir un menú desplegable donde aparezcan todos los nombres. Quedaría así:

```
<html>
<head>
<title>Actualizar datos</title>
</head>
<body>
<div align="center">
```

```
<h1>actualizar un registro</h1>
<br>
<?php
//conexion con la base y seleccion de la basede datos
$conexion = mysqli_connect("localhost","root","","ejemplo");

echo '<form method="post" action="actualizar2.php">nombre<br>';

//creamos la sentencia sql y la ejecutamos
$sql="select id, nombre from clientes order by nombre";
$result=mysqli_query($conexion,$sql);

echo '<select name="nombre">';

//generamos el menu desplegable
while ($row=mysqli_fetch_array($result)){
    echo '<option value="'. $row['id']. "'>'. $row['nombre']. '</option>';
}
?>
</select>
<br>
Actividad<br>
<input type="text" name="actividad"><br>
Web<br>
<input type="text" name="web"><br>
<input type="submit" value="actualizar">
</form>
<a href="lectura.php">Volver al listado de Clientes</a>
</div>

</body>
</html>
```

La manera de operar para construir el menú desplegable es la misma que para visualizar la tabla. De nuevo empleamos un bucle while en combinación con la función `mysqli_fetch_array` lo que nos permite mostrar cada una de las opciones.

El script de actualización será muy parecido al de inserción:

```
<html>
<head>
<title>Actualizar datos</title>
</head>
<body>
<?php
```



```
//pasamos los datos del formulario
$id = $_POST["nombre"];
$actividad = $_POST["actividad"];
$web = $_POST["web"];

//conexion con la base y seleccion de la basede datos
$conexion = mysqli_connect("localhost","root","","ejemplo");

//creamos la sentencia sql y la ejecutamos
$sql="update clientes set actividad='$actividad', web='$web' where id='$id'";
mysqli_query($conexion, $sql);
?>
<h1><div align="center">registro actualizado</div></h1>
<div align="center"><a href="lectura.php">visualizar el contenido de la base</a></div>
</body>
</html>
```



### *Panel de Administración*

Adjunto a los ejemplos un panel de administración similar en funcionamiento al aquí realizado con un poco más de diseño y desarrollo, se adjunta también la base de datos necesaria para su prueba.

El panel de administración tiene un diseño basado en el popular framework de HTML, CSS y Javascript, Bootstrap, que seguramente muchos de ustedes deben conocer y del cual pueden encontrar información adicional en los siguientes links:

[getbootstrap.com/](http://getbootstrap.com/)

[www.w3schools.com/bootstrap/](http://www.w3schools.com/bootstrap/)

[librosweb.es/bootstrap\\_3/](http://librosweb.es/bootstrap_3/)

Y está desarrollado en HTML con PHP utilizando la extensión Mysql en su versión procedural que es la que acabamos de ver en el ejemplo.