

Curso de PHP con MySQL Gratis

Introducción

Este mini curso o mini tutorial de PHP le ayudará a realizar cualquier sistema para que pueda insertar uno o varios registros a una base de datos con MySQL, este manual lo guiará desde la creación de la base de datos, la creación del formulario y la explicación del código de PHP paso a paso para que pueda realizar la inserción a la base de datos.

Le aseguramos que al término de este pequeño curso de PHP gratis usted podrá realizar su propio sistema para que pueda insertar registros en la base de datos. Este curso de PHP gratuito está explicado ampliamente de forma práctica y paso a paso para que usted pueda seguirlo y realizarlo en su computadora.

Si usted sigue este curso de PHP sin costo, en menos de una hora podrá realizar todo lo que corresponde a este manual. Este curso gratuito de PHP está dirigido a aquellas personas que quieren aprender a insertar información a una base de datos y/o adentrarse al mundo de la programación web mejor conocida como programación del lado del servidor utilizando el lenguaje PHP.

Este curso usted podrá descargarlo en un formato pdf para tenerlo en su equipo de cómputo y así poder revisarlo en el momento en que usted lo desee sin tener que estar conectado a internet.

¿Qué es PHP y para que me sirve?

PHP es un lenguaje de scripting que permite la generación dinámica de contenidos en un servidor web. El significado de sus siglas son: *HyperText Preprocessor*.

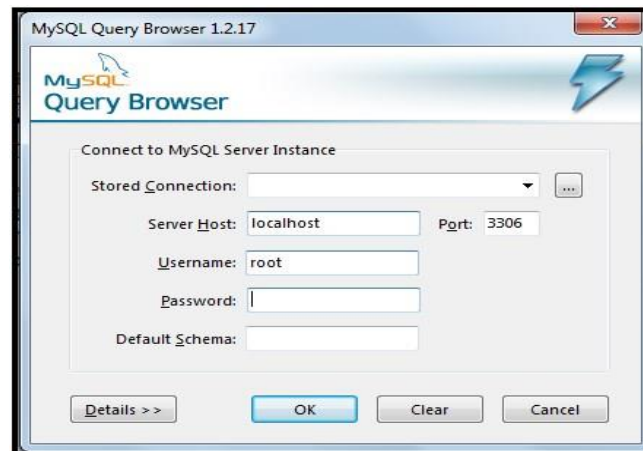
Entre sus principales características se destacan su potencia, su alto rendimiento, su facilidad de aprendizaje y su escasez de consumo de recursos.

Además PHP es multiplataforma, esto quiere decir que podemos realizarlo para diversos sistemas operativos como son: Windows, Linux, Mac, etc., Es uno de los lenguajes de programación más fácil de aprender ya que sus sintaxis no es tan compleja como otros programas para desarrollar sistemas.

Una de las ventajas principales de PHP es que no necesita ser compilado, esto se debe a que PHP es un lenguaje interpretado por los navegadores, esta es una de sus ventajas principales ya que si usted necesita una modificación de cualquier sistema que este desarrollado en este poderoso lenguaje usted solamente tendría que analizar su problema modificar el código y subir sus cambios a su servidor sin necesidad de compilar toda la aplicación para que se reflejen los cambios que usted realizó.

Creación de una base de datos con MySQL

Para poder realizar este ejercicio necesitamos la base de datos, para esto abriremos el gestor de bases de datos MySQL. Podemos utilizar cualquier editor de MySQL en nuestro caso utilizaremos MySQL Query Browser.

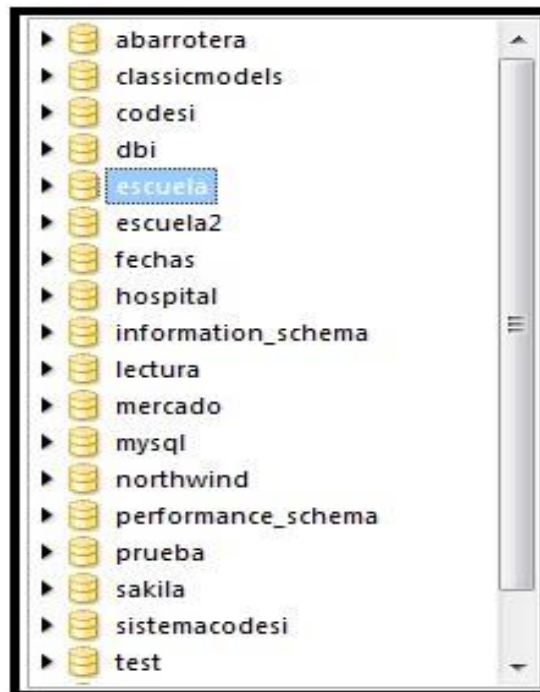


Al principio nos pide usuario y contraseña debemos ingresar el usuario y contraseña que configuramos en la instalación de MySQL en este caso el usuario es el root de MySQL. Ya dentro de Query Browser necesitamos crear nuestra base que nos servirá para este ejemplo y la hacemos con el comando de sql **create database**.

```
1 create database escuela ;
```

Después de nuestro comando necesitamos poner un nombre a nuestra base ya que si no lo ponen nos marcará error.

Verificamos que nuestra base se ha creado correctamente en nuestra parte derecha, ahí se mostrarán todas nuestras base de datos que tengamos en nuestra computadora.



Después de crear nuestra base necesitamos crear nuestra tabla la que va a contener nuestros datos, pero antes de crear nuestra tabla necesitamos poner en uso nuestra base, hacemos esto porque no estamos trabajando con ninguna base hasta este momento.

```
use escuela;
```

Con el comando **use** de SQL nos referimos que vamos a usar la base de datos escuela.

Después necesitamos crear la tabla junto con sus campos para poder insertar registros desde PHP. Para crear la tabla necesitamos el comando **create table**.

```
use escuela;
create table alumnos (id int (4) primary key auto_increment,
                      nombre varchar(25) not null,
                      apellidopat varchar(25) not null,
                      apellidomat varchar(25) not null,
                      edad int (3) not null);
```

Creamos una tabla llamada alumnos con sus campos y sus diferentes tipos de datos, también con sus restricciones de campos.

Creación de un formulario HTML

Hasta ahora solo hemos utilizado MySQL para crear la base de datos y una tabla llamada alumnos a partir de aquí dejaremos un lado MySQL y nos adentraremos con PHP.

Primero necesitamos crear el formulario desde PHP para el ingreso de datos desde el usuario y después insertarlo en la base de datos correspondientemente creada anteriormente.

```
1 <html>
2 <head>
3 </head>
4 <body>
5
6 <center>
7 <form action = "Ingresadatos.php" method = "post">
8 <table>
9 <tr><td>Nombre </td><td><input type = "text" name = "nom" </td></tr>
10 <tr><td>Apellido Paterno</td><td><input type = "text" name = "apellidopa"></td></tr>
11 <tr><td>Apellido Materno</td><td><input type = "text" name = "apellidoma"></td></tr>
12 <tr><td>Edad </td><td><input type = "text" name = "edad" </td></tr>
13 <tr><td><input type = "submit" value = "Guardar"></td></tr>
14 </table>
15 </form>
16 </center>
17 </body>
18 </html>
```

Con el código anterior creamos un formulario para el manejo de datos observe que el código es puro HTML esto se debe que PHP trabaja desde HTML, la salida del código anterior sería:

Nombre	<input type="text"/>
Apellido Paterno	<input type="text"/>
Apellido Materno	<input type="text"/>
Edad	<input type="text"/>
<input type="submit" value="Guardar"/>	

Ahora explicaremos línea por línea el código:

```
1 <html>
2 <head>
3 </head>
4 <body>
5
6 <center>
7 <form action = "Ingresadatos.php" method = "post">
8 <table>
9 <tr><td>Nombre </td><td><input type = "text" name = "nom" </td></tr>
10 <tr><td>Apellido Paterno</td><td><input type = "text" name = "apellidopa"></td></tr>
11 <tr><td>Apellido Materno</td><td><input type = "text" name = "apellidoma"></td></tr>
12 <tr><td>Edad </td><td><input type = "text" name = "edad" </td></tr>
13 <tr><td><input type = "submit" value = "Guardar"></td></tr>
14 </table>
15 </form>
16 </center>
17 </body>
18 </html>
```

- ✓ Empezaremos en la línea 1 hasta la 5 son etiquetas de HTML que nos sirve para indicarle al navegador que crearemos un archivo HTML.
- ✓ En la línea 6 etiqueta HTML de centrar a partir de la línea 7 crearemos el formulario de datos.
- ✓ En la línea 7 indicamos que crearemos un formulario con la etiqueta <form> con el atributo action le indicamos al formulario va tomar una acción hacia una página de destino llamada ingresadatos.php es importante indicarle el nombre de la página y su extensión así con el método que utilizaremos que en este caso utilizaremos el método POST de PHP.
- ✓ A partir de la línea 9 a la 13 creamos los diferentes tipos de control que contiene el formulario que son las cajas de textos y los botones, así como sus variables para poder pasar los datos al archivo PHP.

Insertar datos con PHP y MySQL

Solo nos queda crear el código PHP que es el que va insertar los datos del formulario a nuestra base de datos, el código sería:

```
1 <?php
2 $dbserveridor = 'localhost';
3 $dbusuario = 'root';
4 $dbpass = 'toor';
5 $dbbase = 'escuela';
6 $nombre = $_POST['nom'];
7 $apellidoopa = $_POST['apellidoopa'];
8 $apellidoma = $_POST['apellidooma'];
9 $edad = $_POST['edad'];
10
11 mysql_connect("$dbserveridor", "$dbusuario", "$dbpass") or die ("Problemas con la conexion a mysql");
12 mysql_select_db("$dbbase") or die ("no se encuentra la base");
13 mysql_query("insert into alumnos (nombre , apellidoopat , apellidoomat , edad) values ('$nombre' , '$apellidoopa' ,
14 '$apellidoma' , $edad )") or die ("problemas con la sentencia de mysql ");
15 print("Los datos se han ingresado correctamente ");
16 mysql_close();
17
18 ?>
```

Analizaremos el código anterior a detalle:

Este código es puro PHP no estamos ingresando código HTML ya que no nos servirá de nada, puesto PHP hace toda el trabajo pesado. Tenemos que indicarle al sistema que vamos a empezar a trabajar con código PHP por eso abrimos las etiquetas de "<?php" todo lo que este dentro de la etiqueta será código PHP.

Empezamos declarando 4 variables y las inicializamos con diferentes valores estas variables contendrán el valor del servidor, el usuario de MySQL, la contraseña del mismo y la base de datos que vamos a utilizar esto para poder conectarnos desde PHP a MySQL.

```
$dbserveridor = 'localhost';
$dbusuario = 'root';
$dbpass = 'toor';
$dbbase = 'escuela';
```

Lo siguiente se declara 4 variables, que nos van a servir para guardar los datos del formulario, con el vector asociativo `$_POST []` se pasan los valores del formulario hacia las variables declaradas anteriormente.

```
$nombre = $_POST['nom'];  
$apellidopa = $_POST['apellidopa'];  
$apellidoma = $_POST['apellidoma'];  
$edad = $_POST['edad'];
```

Ahora que ya tenemos los datos en las variables anteriores empezaremos con la conexión a MySQL empezaremos utilizando una función de PHP que es: `mysql_connect()`;

```
mysql_connect("$dbserver" , "$dbusuario", "$dbpass") or die ("Problemas con la conexion a mysql");
```

Esta función hace la conexión hacia la base de datos MySQL pero para esto necesitamos pasar 3 parámetros que son el servidor donde se encuentra en este caso estamos trabajando de manera local, le tenemos que indicar que va hacer localhost, después le indicamos el usuario y al último la contraseña del usuario todo esto ya lo habíamos hecho al pasar estos datos a las variables de conexión. Entonces en la función que conecta solo pasamos las variables como parámetros.

Otra función importante es la de `or die ()` ya que esta se activará mandando un mensaje a pantalla cuando la conexión marque un error.

Los errores más comunes son:

- ✓ El servidor no se encuentra o está apagado
- ✓ EL usuario no existe en la administración de MySQL
- ✓ La contraseña es incorrecta para este usuario

Ya que hicimos la conexión con éxito necesitamos usar la base de datos correspondiente para esto mandamos a llamar la función `mysql_select_db()`;

```
mysql_select_db("$dbbase") or die ("no se encuentra la base");
```


Anteriormente creamos la variable que contendrá la base de datos que se va utilizar para este ejercicio, solo hay que pasar esa misma variable hacia la función, es importante que no se les vaya olvidar la sintaxis que debe llevar con las comillas aunque es una variable necesita llevarlas. De igual forma lleva la función `or die ()`; que es la que se encarga de mostrar el error en caso que no se seleccione la base y ese sería el único error de esta línea, el no encontrar la base de datos.

Y por último seguimos con funciones de MySQL desde PHP y esta va hacer el query correspondiente para insertar `mysql_query()`; el **insert into** es el encargado de insertar registros a nuestra tabla.

```
mysql_query("insert into alumnos (nombre , apellidopat , apellidomat , edad) values ('$nombre' , '$apellidopa' , '$apellidoma' , $edad )" ) or die ("problemas con la sentencia de mysql ");
```

La función `query` de PHP hace un `insert into` hacia la tabla `alumnos` llenando todos los campos de la tabla con los valores que contienen las variables del formulario.

De igual forma se manda a llamar la función `or die` por si contiene algún error.

Por último mandamos a imprimir un mensaje a la página para avisarle al usuario que se insertó correctamente el dato desde el formulario con la sentencia `print()`;

También necesitamos cerrar la base de datos que ocupamos anteriormente y con eso hace la función de `mysql_close()`.

Con esto terminamos nuestro ejercicio de insertar un dato hacia una base de datos desde PHP y MySQL todo esto y más podrás aprender a utilizar en el curso presencial de PHP y MySQL con excelentes profesores capacitados en un ambiente profesional no te quedes fuera tenemos costo accesible para ti no dejes pasar esta oportunidad de aprender unos de los lenguajes más usados y mejor pagados en el mundo laboral.