



VERACRUZ  
GOBIERNO  
DEL ESTADO



SEV  
Secretaría  
de Educación



DET  
Dirección de Educación  
Tecnológica del Estado  
de Veracruz



# **Instituto Tecnológico Superior de Chicontepec**

**Ingeniería en Sistemas Computacionales**

**Nombre:**

**Camelia Bautista Hernández**

**Docente:**

**Ing. Efrén Flores Cruz**

**Asignatura:**

**Programación Web**

**Practica:**

**Conexión y Registro MySQL con PHP.**

**Fecha de entrega**

**4 de Abril del 2020**



# Contenido

<b>INTRODUCCION .....</b>	<b>3</b>
<b>DESARROLLO .....</b>	<b>4</b>
<b>1.-Creacion de base de datos .....</b>	<b>4</b>
<b>2.- Procesando los Datos (Conexión a Base de Datos php).....</b>	<b>5</b>
<b>3.- Filtrando los Datos (ProcesoDatos.php).....</b>	<b>6</b>
<b>4.- Recibir Datos. Php.....</b>	<b>7</b>
<b>5.- Index .HTML .....</b>	<b>8</b>
<b>6.- Localhost.....</b>	<b>9</b>
<b>7.- Verificar Resultados en Base de Datos .....</b>	<b>11</b>
<b>CONCLUSION .....</b>	<b>12</b>

## INTRODUCCION

En el presente trabajo se realiza la práctica de Conexión y Registro MySQL con PHP. La cual consiste en crear un login para poder registrar la cuenta de un usuario la cual está conformado por una conexión a base de datos a través de un formulario en php para poder realizar registros de los usuarios Para guardar datos en MySQL es necesario primero crear una conexión con MySQL, también necesitamos una base de datos.

Para poder continuar con la siguiente práctica de la conexión de datos necesitamos crear los siguiente:

1. Creación de una nueva base de datos en MySQL.
2. Registros en la base de datos en php.
3. Proceso datos en php.
4. Conexión en php.
5. Realizar un archivo en php donde va a recibir los datos.
6. Index en donde estará conformado la estructura de nuestro diseño
7. Verificar en la base de datos si se registran los datos

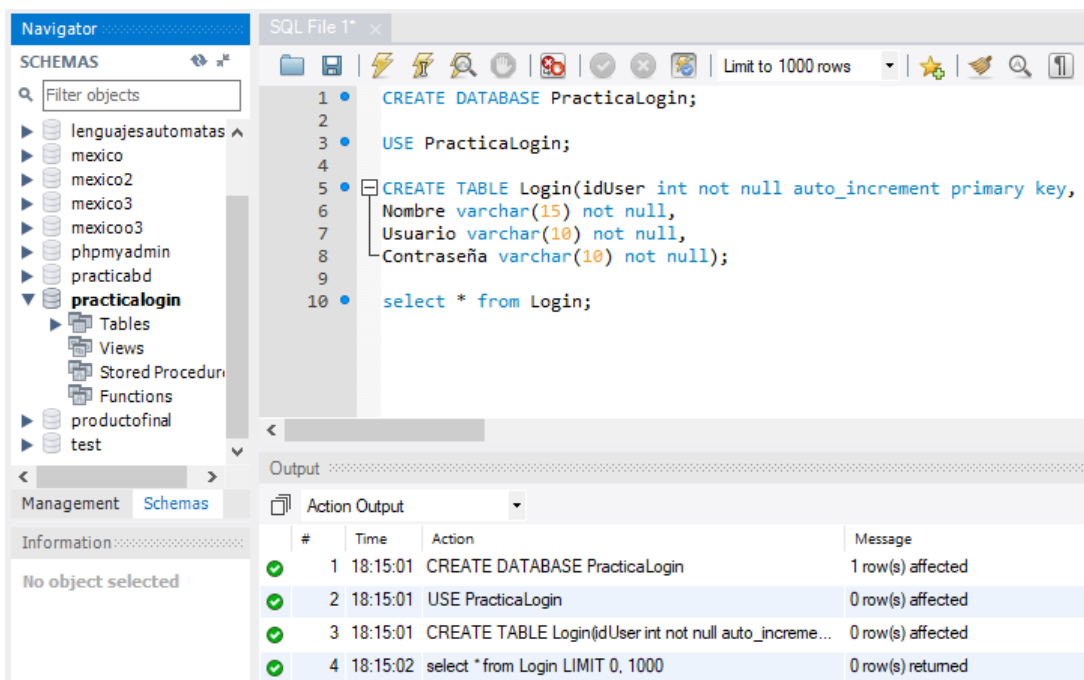
PHP provee funciones nativas que permiten interactuar con MySQL, para el proceso de insertar registros utilizaremos las siguientes funciones.

- `mysql_query` permite enviar cualquier consulta SQL (SELECT, INSERT, UPDATE, etc.) desde PHP.
- `mysql_insert_id` devuelve el identificador del último registro ingresado.

## DESARROLLO

### 1.-Creacion de base de datos

MySQL es el sistema gestor de bases de datos de código abierto más popular en la actualidad. MySQL está disponible para distintos sistemas operativos, como Linux, Mac OS X, Solaris, Windows y otros más. MySQL es muy popular en el desarrollo de aplicaciones web, ya que forma parte como sistema gestor de bases de datos. En la creación de la base de datos utilizamos MySQL Workbench, crearemos nuestra base de datos la cual primeramente es crear la base de datos la cual la llame PracticaLogin, a continuación la ponemos en uso, ya que contendrá una tabla de Login con los campos de idUser, Nombre, Usuario y Contraseña.



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'practicaLogin' selected. The main editor shows the following SQL script:

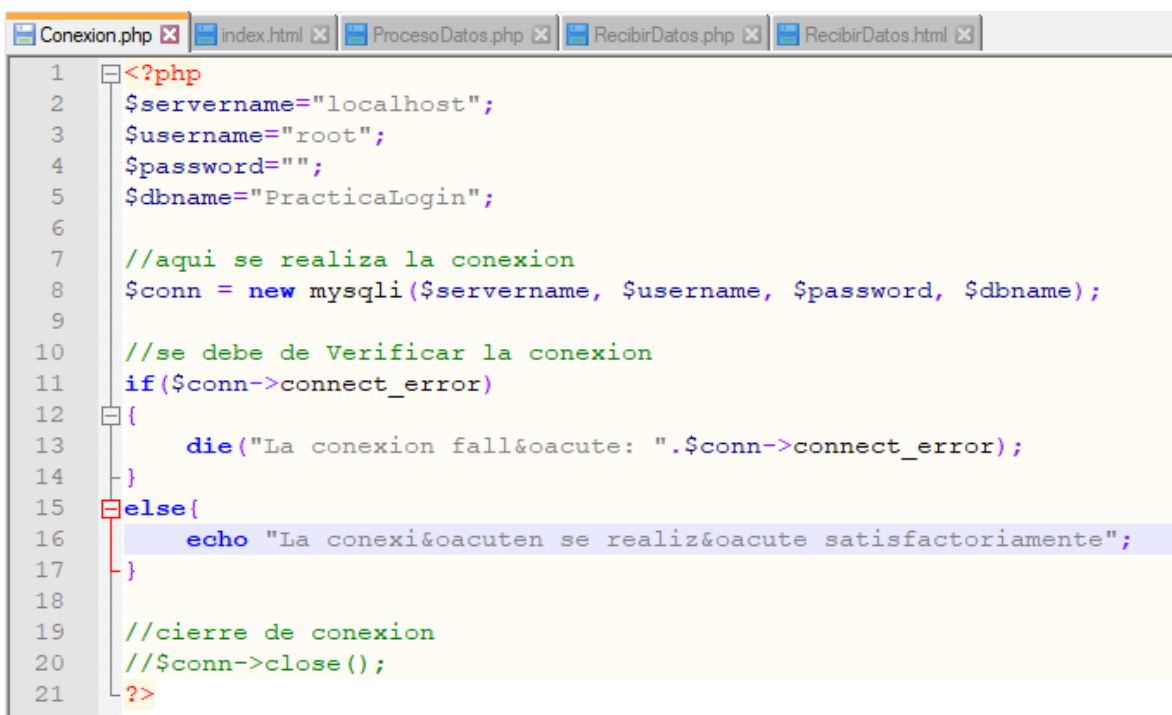
```
1 CREATE DATABASE PracticaLogin;
2
3 USE PracticaLogin;
4
5 CREATE TABLE Login(idUser int not null auto_increment primary key,
6 Nombre varchar(15) not null,
7 Usuario varchar(10) not null,
8 Contraseña varchar(10) not null);
9
10 select * from Login;
```

The 'Output' tab at the bottom shows the execution results:

#	Time	Action	Message
✓ 1	18:15:01	CREATE DATABASE PracticaLogin	1 row(s) affected
✓ 2	18:15:01	USE PracticaLogin	0 row(s) affected
✓ 3	18:15:01	CREATE TABLE Login(idUser int not null auto_increme...	0 row(s) affected
✓ 4	18:15:02	select * from Login LIMIT 0, 1000	0 row(s) returned

## 2.- Procesando los Datos (Conexión a Base de Datos php)

El siguiente paso, es verificar si se están enviando datos desde un formulario para luego crear la consulta concatenando los datos enviados para finalmente guardar los datos con la función *mysql\_query*. La cual contiene el nombre del servidor, nombre de usuario, contraseña, y el nombre de la base de datos que se allá creado ,también debemos de verificar la conexión de la base de datos.

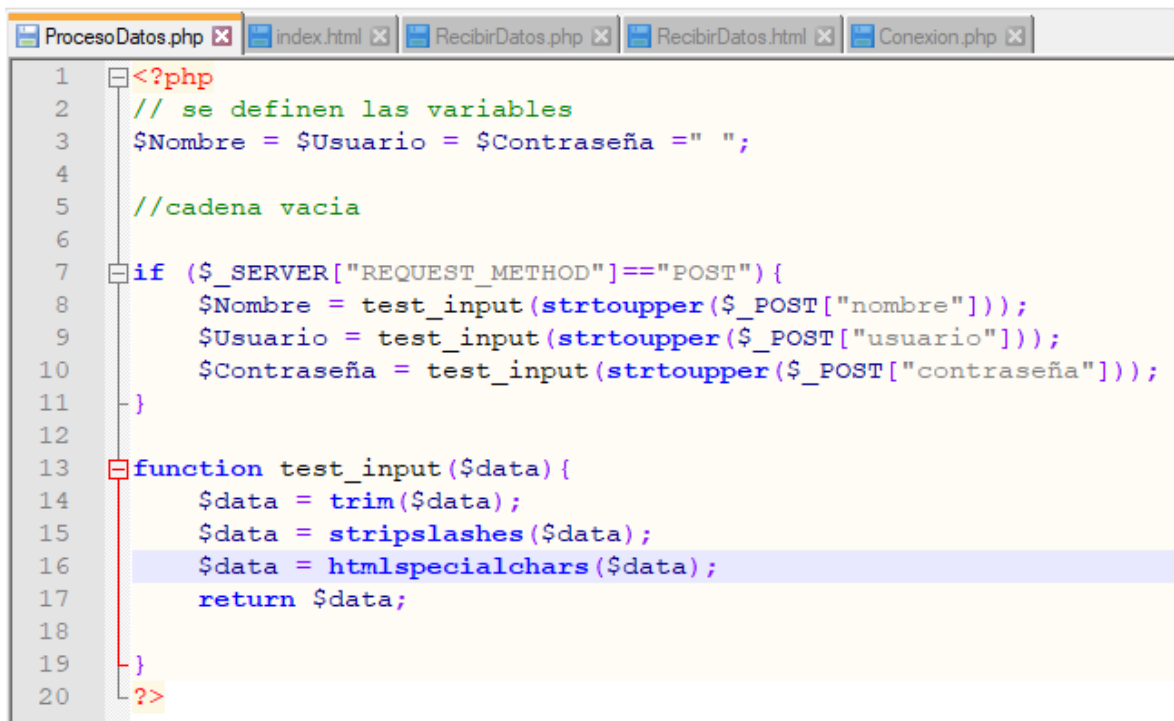


```
1  <?php
2  $servername="localhost";
3  $username="root";
4  $password="";
5  $dbname="PracticaLogin";
6
7  //aqui se realiza la conexion
8  $conn = new mysqli($servername, $username, $password, $dbname);
9
10 //se debe de Verificar la conexion
11 if($conn->connect_error)
12 {
13     die("La conexion falló: ".$conn->connect_error);
14 }
15 else{
16     echo "La conexión se realizó satisfactoriamente";
17 }
18
19 //cierre de conexion
20 // $conn->close();
21 ?>
```

### 3.- Filtrando los Datos (ProcesoDatos.php)

A continuación, una vez creado la conexión en php realizamos lo siguiente el procesoDatos.php en la cual esta conexión nos sirve para definir las variables la cual son llamadas Nombre, Usuario, Contraseña, así como también nos muestra los requerimientos del método y el nombre de las variables

Por ejemplo, texto donde solo se aceptan números o también enviar código SQL para atacar a nuestra base de datos. Por ello antes de utilizar los datos enviados por el usuario es necesario antes normalizar estos datos.



```
1 <?php
2 // se definen las variables
3 $Nombre = $Usuario = $Contraseña = " ";
4
5 //cadena vacia
6
7 if ($_SERVER["REQUEST_METHOD"]=="POST") {
8     $Nombre = test_input(strtoupper($_POST["nombre"]));
9     $Usuario = test_input(strtoupper($_POST["usuario"]));
10    $Contraseña = test_input(strtoupper($_POST["contraseña"]));
11 }
12
13 function test_input($data) {
14     $data = trim($data);
15     $data = stripslashes($data);
16     $data = htmlspecialchars($data);
17     return $data;
18 }
19
20 ?>
```

#### 4.- Recibir Datos. Php

Esta función recibe dos parámetros, el primero el dato que deseamos normalizar. Luego vamos a modificar nuestro primer script para verificar los datos antes de guardarlos. Realizamos un formulario llamado RecibirDatos.php y vamos a requerir los formularios de Conexión. Php y ProcesoDatos.php y se van a declarar nuestra tabla con sus campos ingresados en la base de datos para poder realizar los registros.



```
1 <?php
2 require("Conexion.php");
3 require("ProcesoDatos.php");
4
5 $sql = "INSERT Login(Nombre, Usuario, Contraseña)
6 VALUES ('".$Nombre."', '".$Usuario."', '".$Contraseña."')";
7
8 if($conn-> query($sql) === TRUE)
9 {
10     echo "Registro Exitoso";
11 }
12 else
13 {
14     echo "Error ".$sql. "<br>" . $conn->connect_error;
15 }
16
17 // $conn->close();
18
19 ?>
```

## 5.- Index .HTML

Se crea el index.html en el siguiente formulario nos sirve para poder crear el diseño de la presentación del login la cual agregamos estilos, colores, imágenes, así como también se le agrega las cajas de texto para poder ingresar el nombre, usuario y contraseña por ultimo agregar un botón de guardar.

```

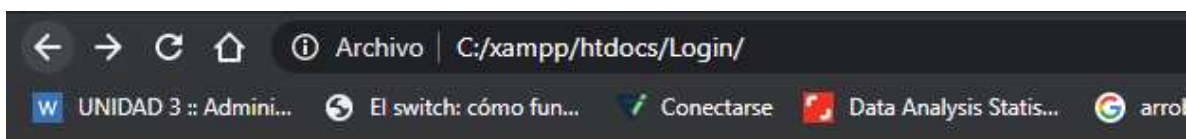
1  C:\xampp\htdocs\Login\index.html
2
3  <html>
4  <head>
5    <meta charset="utf-8">
6    <title> PRACTICA PROGRAMACION WEB</title>
7    <link rel="stylesheet" type="text/css" href="Estilos.css">
8
9  <style>
10   body{background-color:#990033}
11
12   p {
13     font: 150% cursive, font-family: "ARIAL", serif;
14   }
15
16   b {
17     font: 150% cursive;
18   }
19
20 </style>
21 </head>
22 <body>
23
24   <center> <h1> <p style="color:#CC00CC;">BIENVENIDO </p> <h1> </center>
25
26   <div id="img1">
27     <center> <h1> </h1></center>
28   <form action="RecibirDatos.php" method="POST">
29
30     <fieldset>
31       <center>
32         <font face="impact" size=2.5 color="orange">
33         <b>
34
35         <label for="Nombre">Nombre:</label>
36         <input type="text" name="nombre" placeholder="Escribir tu nombre"> </br> </br>
37
38         <label for="Usuario">Usuario:</label>
39         <input type="text" name="usuario" placeholder="Escribir tu usuario"> </br> </br>
40
41         <label for="Contraseña">Contraseña:</label>
42         <input type="text" name="contraseña" placeholder="Escribir tu contraseña">
43         </br> </br>
44
45         <center> <input class="form-btn" name="Guardar" type="submit" value="Guardar" /></center>
46       </b>
47     </fieldset>
48
49   </form>
50 </body>
51 </html>

```










## 6.- Localhost

Una vez que hayamos terminado los formularios abrimos el navegador y escribimos la ruta C:/xampp/htdocs/Login y nos muestra el nombre de los archivos que Conexión. Php, index.html, ProcesoDatos.php, RecibirDatos.php así como también se agrega una carpeta de estilos y una imagen la cual también nos muestra el tamaño y la fecha de modificación.

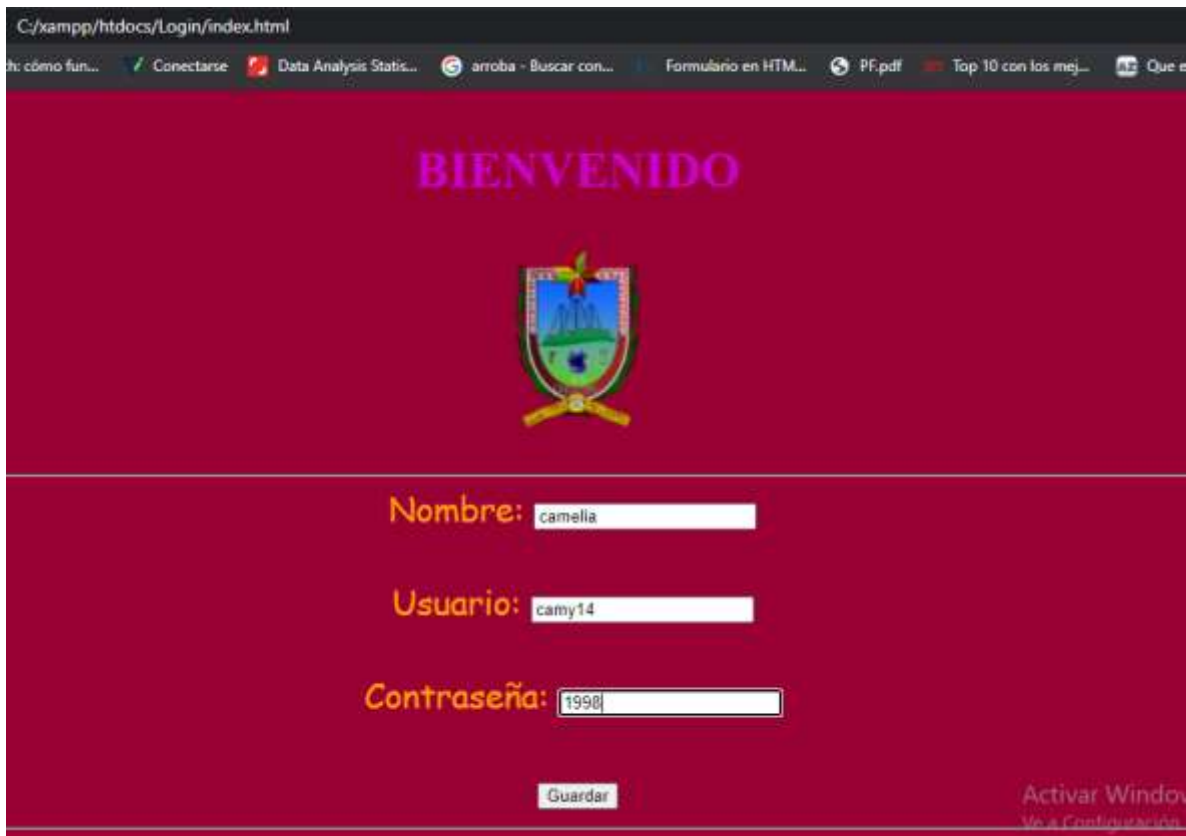


## Índice de C:\xampp\htdocs>Login\

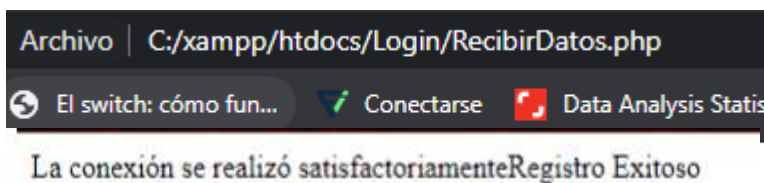
 [\[directorio principal\]](#)

Nombre	Tamaño	Fecha de modificación
 <a href="#">css/</a>		29/6/20 19:18:49
 <a href="#">Conexion.php</a>	437 B	29/6/20 18:46:19
 <a href="#">index.html</a>	1.1 kB	29/6/20 23:57:07
 <a href="#">logo.png</a>	13.2 kB	29/6/20 23:44:28
 <a href="#">ProcesoDatos.php</a>	460 B	29/6/20 18:53:09
 <a href="#">RecibirDatos.html</a>	361 B	29/6/20 23:59:37
 <a href="#">RecibirDatos.php</a>	361 B	29/6/20 19:01:06

En esta imagen nos muestra el diseño del índice como podemos observar nos muestra el logo el nombre, usuario y la contraseña ya que en este formulario ingresamos los datos para poder guardar en nuestra base de datos.

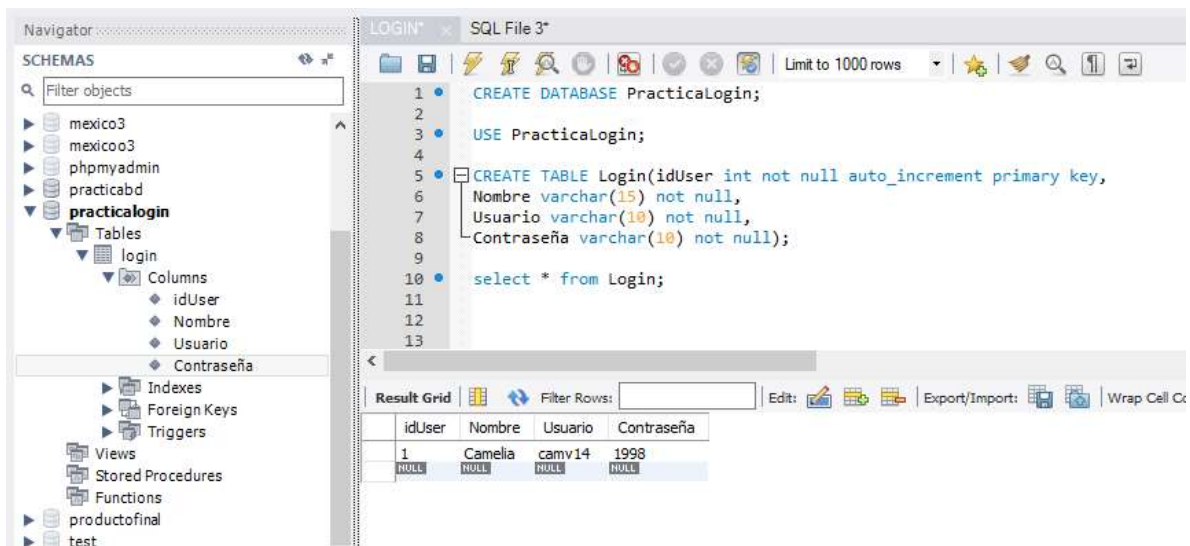


Una vez agregado los datos correspondientes y al guardar los datos nos muestra que el registro de la conexión se realizó satisfactoriamente el registro con éxito.



## 7.- Verificar Resultados en Base de Datos

Para verificar el resultado de lo agregado ingresamos a mysql en donde se realizó la base de datos como podemos observar en la imagen nos muestra la base de datos creado la cual a continuación la ponemos en uso y por ultimo ejecutamos en donde nos muestra select \* from Login la cual es el nombre de la tabla y así como nos muestra en la imagen podemos verificar los datos guardados en la base de datos.



The screenshot displays the MySQL Workbench interface. On the left, the 'Navigator' pane shows the 'practicallogin' database selected, with its tables, columns, indexes, foreign keys, triggers, views, stored procedures, and functions listed. The 'Columns' pane for the 'login' table shows the following structure:

Column	Type	Constraints
idUser	int	not null, auto_increment, primary key
Nombre	varchar(15)	not null
Usuario	varchar(10)	not null
Contraseña	varchar(10)	not null

The 'SQL File 3\*' pane shows the following SQL commands:

```
1 CREATE DATABASE PracticalLogin;
2
3 USE PracticalLogin;
4
5 CREATE TABLE Login(idUser int not null auto_increment primary key,
6 Nombre varchar(15) not null,
7 Usuario varchar(10) not null,
8 Contraseña varchar(10) not null);
9
10 select * from Login;
```

The 'Result Grid' pane shows the results of the 'select \* from Login;' query:

idUser	Nombre	Usuario	Contraseña
1	Camelia	camv14	1998

## CONCLUSION

En la siguiente actividad, lo que se necesitaba realizar era: establecer una conexión con una base de datos para después poder registrar los datos de usuario, que contienen un nombre, usuario y una contraseña la cual se utilizó el editor de notepad para crear los formularios y mysql WorkBeanch para crear la base de datos.

PHP proporciona diferentes maneras para conectarse a un servidor de bases de datos MySQL. Hasta PHP 5.5, una de las formas más populares era a través de la extensión MySQL: esta proporcionaba una manera "procedimental" para conectarse al servidor MySQL.

Los parámetros nombre, usuario y contraseña son definidos por el creador de la base. Es conveniente en un principio, al crear nuestras bases, trabajar sin ellos con lo cual dejaremos las cadenas correspondientes vacías: "".

Además de la propia inserción, el programa avisa de la introducción del registro y ofrece un enlace hacia una página de lectura.