**PHP con Bases de Datos MySQL**

**Conectando PHP con Bases de Datos**

**Definición**

PHP utiliza un conjunto de funciones para acceder a los datos de la base. Dichas funciones forman parte de las DLLS utilizadas en la instalación.

PHP permite al programador relacionarse con las bases de datos más importantes como con otras menos conocidas. Ej: PostgreSQL, MySql, Oracle, ODBC, etc.

**ODBC**

En particular, la conexión ODBC permite relacionarse con bases de datos tipo Access. La utilización de las instrucciones para la conexión a las bases de datos no tiene nada que ver con el conocimiento de las mismas. El lenguaje SQL sirve para formular las preguntas y para saber el significado de índices, claves, relaciones, etc.

**MySQL**

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario (desde enero de 2008 una subsidiaria de Sun Microsystems desarrolla MySQL como software libre en un esquema de licenciamiento dual).

Soporta multiplataforma al igual que PHP, por eso son una combinación típica para desarrollo de aplicaciones en Internet.

Las funciones utilizadas desde PHP comenzarán todas con la palabra “mysql\_” (esta es una característica común dentro de la sintaxis PHP, la cual busca agrupar conjuntos de funciones relacionadas)

**Funciones de PHP para interactuar con MySQL**

**mysqli\_connect()**

Se utiliza para abrir una conexión con un servidor MySQL.

Espera como parámetros la IP (o nombre) del servidor, el usuario, la contraseña y el nombre de la base de datos (opcional).

Ejemplo de uso:

<?php

$laConexion = mysqli\_connect('localhost', 'root', 'root', 'phpdb');

if (!$laConexion) {

echo "No se pudo conectar con el servidor";

}

echo "Conexion satisfactoria";

?>

**mysqli\_close()**

Se utiliza para cerrar una conexión abierta y libera los recursos utilizados por la conexión.

Luego de abrir y utilizar una conexión, se deberá cerrar.

Recibe como parámetro optativo el identificador de la conexión.

Si el identificador no está definido, la función cerrará la última conexión abierta.

Ejemplo de uso:

<?php

mysqli\_close($laConexion);

?>

**mysqli\_select\_db()**

Se utiliza para seleccionar la base de datos a utilizar.

Nota: Esta función debería ser utilizada sólo para cambiar la base de datos por defecto para la conexión. Se puede seleccionar la base de datos por defecto en el cuarto parámetro de la función mysqli\_connect().

Ejemplo de uso:

<?php

$laConexion = mysqli\_connect('localhost', 'root', 'root');

if (!$laConexion) {

echo "No se pudo conectar con el servidor";

}

echo "Conexion satisfactoria";

//Selecciono la base de datos Clase

mysqli\_select\_db($laConexion, 'phpdb');

//Cierro la conexion

mysqli\_close($laConexion);

?>

**mysqli\_query()**

Utilizada para hacer consultas a una base de datos.

Ejemplo de uso:

<?php

$laConexion = mysqli\_connect('localhost', 'root', 'root', 'phpdb');

if (!$laConexion) {

echo "No se pudo conectar con el servidor";

}

echo "Conexion satisfactoria";

//armo la consulta en una variable string

$query = "SELECT \* FROM ALUMNOS";

//ejecuto la consulta

$rowset = mysqli\_query($laConexion, $query);

//Cierro la conexion

mysqli\_close($laConexion);

?>

**mysqli\_num\_rows ()**

Esta función devuelve la cantidad de registros que tiene una tabla.

//ejecuto la consulta

$rowset = mysqli\_query($laConexion, $query);

$num\_rows = mysqli\_num\_rows($rowset);

echo "Registros: $num\_rows" . "<br>\n";

**mysqli\_fetch\_array()**

Esta función devuelve el contenido de una consulta en un array asociativo, identificando a cada elemento (índice) en un campo de la base de datos.

Recibe como parámetros el identificador que se utiliza al ejecutar la consulta mysql\_query ().

<?php

$laConexion = mysqli\_connect('localhost', 'root', 'root', 'phpdb');

if (!$laConexion) {

echo "No se pudo conectar con el servidor";

}

echo "Conexion satisfactoria";

//armo la consulta en una variable string

$query = "SELECT \* FROM usuarios";

//ejecuto la consulta

$rowset = mysqli\_query($laConexion, $query);

//ejecuto la entara en forma de tabla

echo "<table border=1 cellpadding=0 cellspacing=0 style='border-collapse: collapse' width=80%>

<TR>

<TH bgcolor=#00FFFF WIDTH=10%>nombre</TH>

<TH bgcolor=#00FFFF WIDTH=10%>apellido</TH>

<TH bgcolor=#00FFFF WIDTH=8%>sexo</TH>

<TH bgcolor=#00FFFF WIDTH=20%>mail</TH>

<TH bgcolor=#00FFFF WIDTH=10%>pais</TH>

</TR>";

while($row = mysqli\_fetch\_array ($rowset))

{

echo "<TR>

<TD WIDTH=10% ALIGN=CENTER>&nbsp;$row[nombre]</TD>

<TD WIDTH=10% ALIGN=CENTER>&nbsp;$row[apellido]</TD>

<TD WIDTH=08% ALIGN=CENTER>&nbsp;$row[sexo]</TD>

<TD WIDTH=20% ALIGN=CENTER>&nbsp;$row[mail]</TD>

<TD WIDTH=10% ALIGN=CENTER>&nbsp;$row[pais]</TD>

</TR>";

}

echo "</TABLE>";

//Cierro la conexion

mysqli\_close($laConexion);

?>

**mysqli\_error()**

Se utiliza para determinar si existió un error al ejecutar una sentencia SQL.

Ejemplo de uso:

<?php

$laConexion = mysqli\_connect('localhost', 'root', 'root', 'phpdb');

$result = mysqli\_query($laConexion , "SELECT \* FROM usuariosXXX",);

if (@$row = mysqli\_num\_rows($result)) {

return $row;

} else {

print (mysqli\_error($laConexion ));

}

?>

**Modificaciones de Datos**

**Insertar un Registro**

Dentro de PHP se utiliza la sentencia INSERT perteneciente al lenguaje SQL Script de MySQL.

$nombre = "xxx";

$apellido = "xxx";

$telefono = "xxx-xxx";

$email = "xxx@xxx.com";

//$alumnoID = 4;

$consulta=" insert into Alumnos (nombre, apellido, telefono, email, fechanac) values ( '$nombre', '$apellido', '$telefono', '$email', '$fechaNac' )";

$resultados=mysqli\_query ($laConexion, $consulta) ;

**Modificar un Registro**

Dentro de PHP se utiliza la sentencia UPDATE perteneciente al lenguaje SQL Script de MySQL.

$nombre = "Ernesto";

$apellido = "Usuriaga";

$telefono = "4574-4574";

$email = "Ernesto@Usuriaga.com";

$alumnoID = 8;

$consulta=" update Alumnos

set

Nombre = '$nombre',

Apellido = '$apellido',

Telefono = '$telefono',

Email = '$email',

FechaNac = '$fechaNac'

Where alumnoID = $alumnoID ";

$resultados=mysqli\_query ($laConexion, $consulta) ;

**Borrar un Registro**

Dentro de PHP se utiliza la Sentencia DELETE perteneciente al lenguaje SQL Script de MySQL.

$alumnoID = 8;

$consulta=" delete from Alumnos Where alumnoID = $alumnoID ";

$resultados=mysqli\_query ($laConexion, $consulta) ;

**Registros afectados**

**La función mysqli\_affected\_rows()**

Obtiene la cantidad de filas afectadas en la última operación. Se utiliza para determinar cuantos registros afecto la ejecución de la ultima sentencia SQL.que se realizo. Si la misma fallo entonces esta función devuelve -1.

Ejemplo de uso:

<?php

$laConexion = mysqli\_connect('localhost', 'root', '', 'phpdb');

$usuario = "1xxx";

$password = "1xxx";

$nombre = "1xxx";

$apellido = "1xxx";

$email = "1xxx@xxx.com";

//$alumnoID = 4;

$consulta=" insert into usuarios (usuario, psw, nombre, apellido, mail ) values ( '$usuario', '$password', '$nombre', '$apellido', '$email' )";

$resultados=mysqli\_query ($laConexion, $consulta) ;

echo mysqli\_affected\_rows($laConexion);

?>

**Listado de Productos**

**Armar el Reporte de Productos**

El objetivo de este reporte es extraer la información de los productos, la cual está distribuida en tres tablas:

A continuación, se muestra la pantalla con el resultado del reporte:

**Armar la consulta SQL**

A continuación, se arma la consulta SQL con dos INNER JOINS desde la tabla producto a las tablas Marcas y Categorías donde se obtienen las descripciones.

if (empty($\_GET[orden]))

$\_GET[orden] = 'p.cod\_producto';

$sql = "SELECT p.cod\_producto,

p.nombre,

p.precio,

p.stock,

p.cod\_categoria,

c.descripcion,

p.cod\_marca,

m.nombre\_marca

FROM productos p

INNER JOIN categorias c ON p.cod\_categoria = c.cod\_categoria

INNER JOIN marcas m ON p.cod\_marca = m.cod\_marca

ORDER BY $\_GET[orden]";

$rs = mysqli\_query($link, $sql);

**Reporte**

En el siguiente código se recorre el resultado de la consulta utilizando la función mysqli\_fetch\_array() dentro de un bucle repetitivo While. Se tabula el resultado de la consulta en una tabla HTML.

<table width="860px" cellpadding="5" cellspacing="0" border="1" align="center">

<tr bgcolor="#efc789">

<td><a target="\_blank" href="?orden=cod\_producto">cod\_producto</a></td>

<td><a target="\_blank" href="?orden=nombre">nombre</a></td>

<td><a target="\_blank" href="?orden=precio">precio</a></td>

<td><a target="\_blank" href="?orden=stock">stock</a></td>

<td><a target="\_blank" href="?orden=cod\_categoria">cod\_categoria</a></td>

<td><a target="\_blank" href="?orden=descripcion">descripcion</a></td>

<td><a target="\_blank" href="?orden=cod\_marca">cod\_marca</a></td>

<td><a target="\_blank" href="?orden=nombre\_marca">nombre\_marca</a></td>

</tr>

<?php

while($v=mysqli\_fetch\_array($rs)) {

?>

<tr bgcolor='<?php echo $col ?>'>

<td><?php echo $v[cod\_producto] ?></td>

<td><?php echo $v[nombre] ?></td>

<td><?php echo $v[precio] ?></td>

<td><?php echo $v[stock] ?></td>

<td><?php echo $v[cod\_categoria] ?></td>

<td><?php echo $v[descripcion] ?></td>

<td><?php echo $v[cod\_marca] ?></td>

<td><?php echo $v[nombre\_marca] ?></td>

</tr>

<?php

}

?>

<tr bgcolor='<?php echo $col ?>'>

<td colspan="13"><font face="arial" size="2">Total registros: <?php echo mysqli\_num\_rows($rs) ?></font></td>

</tr>

</table>

**Colores Interlineados**

Agreguen los colores interlineados en el reporte anterior.

Mostrar al final del reporte la cantidad de registros encontrados

Dentro del bucle repetitivo While se coloca una variable $i como contador de filas y si el número de fila es impar se modificará el color de la variable $col.

<?php

while($v=mysqli\_fetch\_array($rs)) {

$i++;

if($i%2==0)

$col = '#b7b6ba';

else

$col = '#939197';

}

?>

**Búsquedas internas en el sitio**

Agregar la funcionalidad al sitio para poder hacer búsquedas internas.

**Paginación de los resultados del Reporte**

Paginar los resultados del reporte anterior en fracciones de 10 registros por páginas.

El objetivo de esta práctica es brindar la posibilidad de fraccionar los resultados obtenidos de una base de datos en cantidades de registros inferiores a 10 por página, para que los mismos puedan ser fácilmente leídos por el usuario y a su vez, reducir el tráfico de información en la web.