

CLASE 08

CONCEPTOS INICIALES

- Bases de datos con MySQL
 - Creación de base y tablas en MySQL
 - Tipos de datos en MySQL
 - Exportación / Importación de bases entre distintos equipos
 - Conexión con PHP
- Procesos iniciales de consultas a la base de datos:
 - Registro de datos tomando valores de formularios (INSERT)
 - Mostrar listados de registros (SELECT)

OBJETIVOS

- Aprender a usar la función de conexión a una base de datos MySQL.
- Aprender a usar las funciones de consultas contra una base de datos MySQL.
- Usar los scripts de ejemplo y ejercicios ubicados dentro del archivo Clase_08_EjemplosBD.zip
- Aplicar conocimientos de clases anteriores para integrar con estas estructuras.
- Desarrollar código para ser visualizado en el navegador.

DESARROLLO

Introducción

Hemos llegado a un tema **muy importante** en todo curso de lenguaje de programación. Cómo **conectarnos a una base de datos**, y cómo manejar sus **datos mediante consultas**. En nuestras próximas clases, estaremos trabajando con **MySQL**.



Al hablar de Bases de Datos debemos tener presentes y bastante conocido el tema de cómo generar un **Diagrama de Entidad Relación**, mediante el cual vamos a estructurar de una manera organizada todos los datos de nuestra aplicación o sistema web.

Una vez analizados estos datos, ya podremos crear la base de datos, teniendo en cuenta esas relaciones entre las distintas entidades.

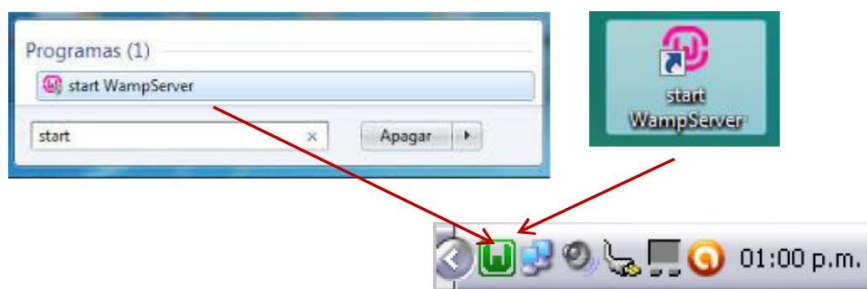
Como todas las clases, ya sabemos cómo empezar a trabajar:



Recordemos los pasos previos para todas nuestras prácticas:

a) Debemos tener activo el WampServer:

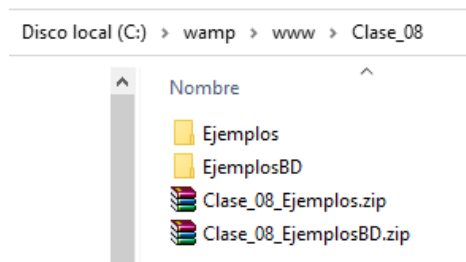
Para comenzar a probar recuerden que el servidor local debe estar activado [en verde], tras doble click en el icono del Wamp del escritorio, o tras buscarlo en los Programas instalados:



b) Abriremos nuestro editor de texto, para trabajar con el código de los ejemplos que iremos viendo a continuación.



c) Tendremos que descargar el archivo **Clase_08_EjemplosBD.zip** y ubicar su contenido en el directorio de alojamiento, creando las carpetas correspondientes para que quede de esta manera:



d) Abriremos el navegador [Chrome recomendado] y nos ubicaremos en nuestro server y la carpeta recién creada. De allí podremos ir revisando los ejemplos.

http://localhost/Clase_08/EjemplosBD

Recordemos nuevamente:



La **ruta física a los archivos** o scripts de nuestra **carpeta de alojamiento de nuestro servidor** (local), será **C:\wamp\www**

La **ruta para ejecutar** esos archivos o scripts, alojadas en nuestro **servidor** (local), será **desde el navegador**, indicando primero el **servidor a donde apuntamos**, y luego las carpetas hasta llegar al script: http://localhost/Clase_xx/carpeta....

- Noten que envolvemos nuestro código con las etiquetas **<?php** y **?>**
- La sentencia **echo** es la que muestra en pantalla el mensaje que se escribe entre comillas [dobles o simples].
- Recordar siempre terminar las sentencias con punto y coma ;
- Cada **cambio** que se genere en el **script** desde el editor de texto, **guardarlo**. Para ver el cambio en el **navegador**, **refrescar la página** con F5.¹
- Cada script se **guarda con la extensión .php** para que pueda ser interpretado correctamente.

Ahora si, seguimos adelante!

¹ Deshabiliten el caché para que el navegador fuerce a traer los cambios recientes en cada cambio que hagan:
<https://amerikanistik.org/es/software/930-how-to-completely-disable-cache-in-google-chrome.html>

Análisis inicial

Como venimos trabajando con formularios, ya hemos visto cómo tomar los datos ingresados por el navegante. Ya aprendimos a validarlos, y ya estamos en condiciones de manipularlos como para poder grabarlos en alguna base, y luego utilizarlos.

Para esto, podemos comenzar pensando en registrar los datos que ya vimos en el Registro inicial de Usuarios de la clase anterior, pensemos cómo diseñar la estructura para guardarlos.

La idea es que vayamos armando un **panel**, donde puedan ir viendo cómo registrar datos, consultarlos, modificarlos, eliminarlos, con **consultas desde y hacia una Base de Datos**. Vamos a ver paso a paso, cómo ir generando las estructuras necesarias para todas estas transacciones.

Creación de la Base de Datos

En nuestra materia, usaremos **MySQL** como nuestro Sistema de Gestión de Bases de Datos Relacional.

Si quieren conocer mas detalles de MySQL, pueden googlear pues existe muchísima información al respecto. Les dejo un link para que unifiquemos conceptos inciales, y tras revisarlo, ya podemos seguir con nuestros temas:



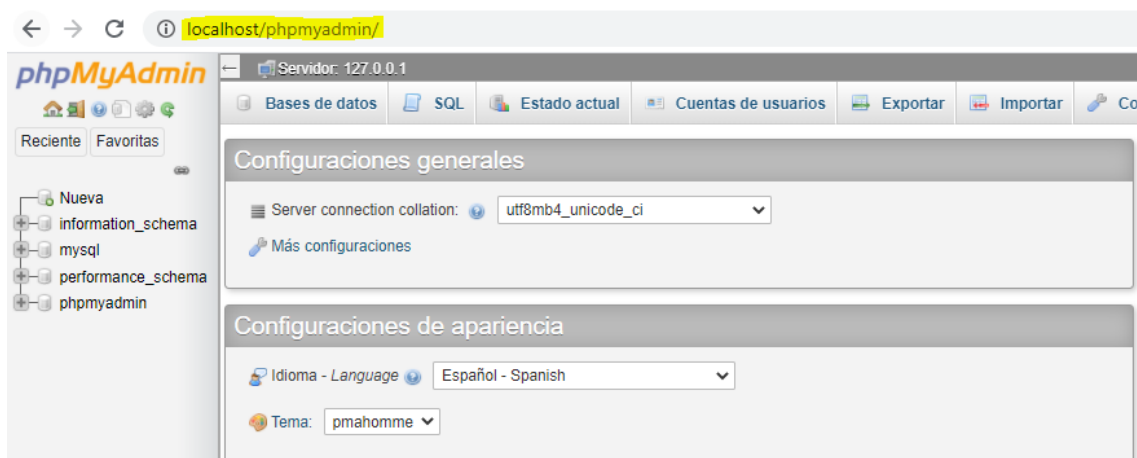
<https://www.hostinger.com.ar/tutoriales/que-es-mysql>

<https://www.hn.cl/blog/las-ventajas-de-mysql-por-sobre-otras-bases-de-datos/>

Cuando en la primer clase instalamos las herramientas para trabajar, habíamos visto de instalar el Apache, PHP y también MySQL. Así que ya contamos con ella para poder usarla.

Pueden conectarse con la herramienta que deseen, aquí usaremos el **PHPMyAdmin**, que es la interfaz que nos provee el Wamp. Accedemos mediante **nuestro servidor local**, indicando:

<http://localhost/phpmyadmin/> dado que no tenemos configurado usuarios ² de acceso, ingresamos directamente. Al ingresar a esta dirección, ya podrán ver algo así:

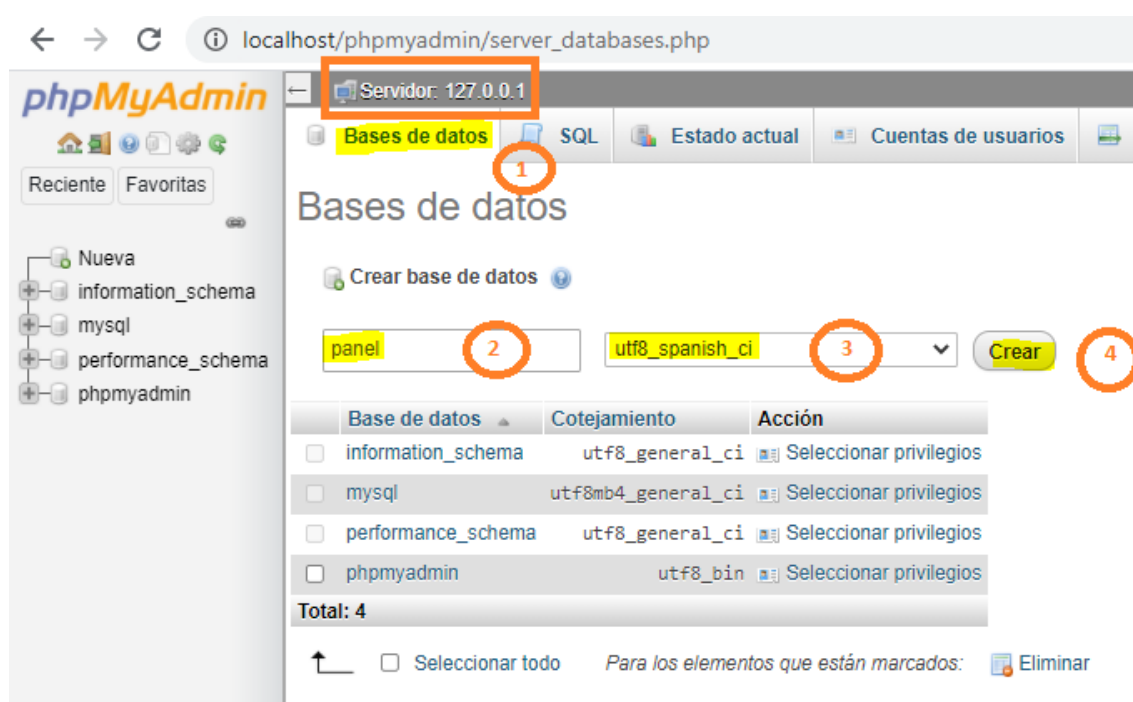


² Si alguien desea configurarlo puede hacerlo, pero no se exige en este curso.

Noten que **estamos trabajando en el servidor**, local, IP **127.0.0.1**. En otra ocasión cuando les toque hacerlo, verán que ese IP será la dirección del servidor remoto donde tengan alojada la base, lo cual es otro equipo, ya no su equipo local.

Vamos a **crear nuestra primer base de datos**, que será la que contenga todas las estructuras de tablas que vayamos creando para lograr hacer funcional nuestro Panel.

- 1) Pulsamos en **Base de datos**
- 2) En esta pantalla vamos a ingresar el **Nombre** de la misma: le llamaremos *Panel*.
- 3) Elegimos esta colección de caracteres, **UTF8**, para que tome caracteres especiales.
- 4) Pulsamos en **Crear**



Nos informa que se ha creado, y ya la vemos en el lateral izquierdo:



Listo! Ya tenemos lugar para empezar a guardar nuestros datos.

Seguimos!

Creación de Tablas

Para esta primera estructura, vamos a querer almacenar los datos del **Registro de Usuarios**, los cuales vemos en el siguiente link: http://localhost/Clase_08/Panel/registro.php

Este formulario es el que veníamos trabajando, ahora con los campos necesarios:

← → ↻ localhost/Clase_08/Panel/registro.php

Página inicial

Form Registro

Formulario de Registración

Ingresa tus datos

REGISTRATE GRATIS

Nombre:

Apellido:

Email:

Clave:

Reingresa la clave:

País:

Sexo: ☐ Femenino ☐ Masculino

Condiciones del sitio: ☐ Acepto los Términos y Condiciones.

[o Volver al inicio](#)

Analicemos: vemos entonces que debemos almacenar datos de **Usuarios**:

- ✓ Nombre: caracteres, un valor string, que podría ser de hasta 50 caracteres.
- ✓ Apellido: caracteres, un valor string, que podría ser de hasta 50 caracteres.
- ✓ Email: idem anterior
- ✓ Clave: caracteres, un valor string, que podría ser de hasta 15 caracteres.
- ✓ Obviamente que el reingreso de clave no se guardará, solo es para validación
- ✓ País: aquí podemos contar con un dato Numérico, el cual será una referencia a otra tabla de **Países**.
- ✓ Sexo: aquí podemos contar con un caracter, para hacerlo inicialmente mas facil. (F, M, O)
- ✓ Aceptar términos: podría guardarse, aquí solamente lo usaremos como validación.
- ✓ Además podríamos guardar la fecha de creación del usuario.
- ✓ También podríamos generar un nivel de acceso al sistema, por defecto, que sea "Nivel Básico". Esto a los fines que luego en el panel, tenga restringidos ciertos accesos, y no vea todas las funcionalidades, solo algunas. (Esto corresponde a un manejo de niveles de usuarios). Por lo tanto tendríamos otra tabla con **Niveles**, y este dato se referenciará a ella.
- ✓ Recuerden que cada registro debe ser identificado de una manera única e irrepetible: deberemos agregar un valor de clave primaria para cada usuario que tengamos.

Debemos crear entonces 3 estructuras:

- ❖ Países: con un número identificatorio y un nombre o denominación.
- ❖ Niveles: con un número identificatorio y un nombre o denominación
- ❖ Usuarios: con los datos ya planteados, una clave única, y las referencias a las otras tablas.



Creación de Tablas Países, Niveles, Usuarios

Les dejo un video donde podemos ver cómo seguimos trabajando con la creación de las estructuras necesarias: <https://share.vidyard.com/watch/cJ7RCEZxjpAKY82MnhszZk?>

Ya visto el video, les dejo que ustedes generen la tabla grande de Usuarios. Igualmente pueden revisar cómo generarla desde aquí: <https://share.vidyard.com/watch/T9xkirq52qrCNuL1qc9MbQ?>

Si ya vieron los videos, y pudieron ir creando la base con las tablas, ya estamos en condiciones de poder conectarnos desde PHP.

Conexión desde PHP a MySQL

Para ver estos ejemplos, deberán ubicar desde la carpeta de alojamiento el siguiente script:

C:\wamp\www\Clase_08\Ejemplos_BD\EJEMPLOS_CONEXION\conexion.php

Aquí tenemos un simple ejemplo de conexión:

```
Clase_08 > EjemplosBD > EJEMPLOS_CONEXION >  conexion.php > ...
1  <?php
2  //para el mysql instalado con Wamp, estos son los datos de conexio
3  $Host = 'localhost'; //127.0.0.1
4  $User = 'root';
5  $Password = '';
6  $BaseDeDatos='panel';
7
8  //procedo al intento de conexion con esos parametros
9  $linkConexion = mysqli_connect($Host, $User, $Password, $BaseDeDatos);
10 if ($linkConexion!=false) {
11     echo '<h3>Acceso al Mysql del Localhost: La conexion es correcta!</h3>';
12 }else {
13     echo '<h3>Hubo algun error al intentar conectarse...</h3>';
14 }
15 ?>
```


Al trabajar con el Wamp o Xampp, los **datos** por defecto de **conexión** a la base de datos **Mysql** son los que se ven allí:

- ✓ **Host:** es el server donde se encuentra nuestra base, donde queremos conectarnos. Pueden usar el IP de equipo también. En nuestro caso, al ser el equipo local que hace de servidor, será "localhost" o el IP 127.0.0.1.
- ✓ **User:** es el usuario de conexión a la base de datos. Aquí el usuario por defecto es *root*.
- ✓ **Password:** es la clave de conexión a la base de datos. Aquí por defecto el valor está vacío.
- ✓ La **BaseDeDatos:** la que creamos en el punto anterior, con ayuda de los videos tutoriales.
- ✓ Recordar que en todos los casos los valores deben ser exactos, **respetando mayúsculas y minúsculas**, sino arrojará error de conexión.

Para conectarnos a la base, usaremos la **función de PHP `mysqli_connect()`**.

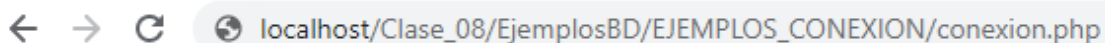
Toma los **parámetros** que vemos allí, en ese **orden**: el host primero, luego usuario y clave, por último la base. Para versiones mas antiguas de PHP se usaba `mysql_connect()`, pero la evolución del lenguaje nos obliga a actualizarnos y usar esta nueva con la "i" **`mysqli_connect()`**.

Esta función devuelve *false*, si no pudo conectarse.

Prueben el script desde el navegador:

http://localhost/Clase_08/Ejemplos_BD/EJEMPLOS_CONEXION/conexion.php

Si todo va bien, debieran poder ver el mensaje de ok:



← → ↻ 🌐 localhost/Clase_08/EjemplosBD/EJEMPLOS_CONEXION/conexion.php

Acceso al Mysql del Localhost: La conexion es correcta!

Si surgen errores, por favor verificar cada dato de conexión.

Insertando valores

Las tablas que hemos creado se encuentran casi vacías por ahora. Si bien agregamos datos para que vean cómo hacerlo, no lo hicimos desde nuestro PHP. Vamos a cargarles desde PHP algunos valores, estos ejemplos son a modo inicial para que vayan comprendiendo cómo se envían los datos y las funciones que intervienen.

Deberán ubicar el script: `C:\wamp\www\Clase_08\Ejemplos_BD\EJEMPLOS_INSERT\insert_niveles.php`

```
Clase_08 > EjemplosBD > EJEMPLOS_INSERT > insert_niveles.php > ...
1  <?php
2  $Host = 'localhost';
3  $User = 'root';
4  $Password = '';
5  $BaseDeDatos = 'panel';
6
7
8  //procedo al intento de conexion con esos parametros
9  $linkConexion = mysqli_connect($Host, $User, $Password, $BaseDeDatos);
10
11 if ($linkConexion != false) {
12     //si la conexion es correcta...
13     echo '<h2>Acceso al Mysql del Localhost... ok</h3>';
14
15     //en la variable $SQL Insert asigno la CONSULTA de INSERCIÓN con el valor a agregar:
16     $SQL_Insert = "INSERT INTO niveles (Denominacion) VALUES ('Nivel -----')";
17
18     //procedo a ejecutar la consulta
19     if (!mysqli_query($linkConexion, $SQL_Insert)) {
20         //si surge un error, finalizo la ejecucion del script con un mensaje
21         die('<h4>Error al intentar insertar el Nivel.</h4>');
22     }
23     echo "<h4>El Nivel se ha insertado correctamente!</h4>";
24 }
25
26
```

Este script se va a conectar a la base recientemente creada, y tratará de insertar un registro en la tabla de **Niveles**. Sigamos cada línea y los comentarios:

- ❖ La inserción de un nuevo registro se realiza con la sentencia SQL: **INSERT INTO**. Esta consulta se guarda en una variable de PHP `$SQL_Insert` (Línea 16)
- ❖ La función **mysqli_query()** es otra función de PHP, toma la conexión actual y la consulta generada, para procesarlas y llegar hasta la tabla especificada para realizar la consulta SQL. Devuelve un *false* en caso de no poder realizar la operación.

Aquí las funciones de PHP usadas, con sus respectivos parámetros en ese orden, son:

- **mysqli_connect**(*Host, usuario, clave, BD*) Para la conexión con la BD (línea 9)
- **mysqli_query**(*linkConexion, laConsulta*) para enviar la SQL a la BD (línea 19, se interpreta como "si no pudo ejecutar la consulta con la conexión dada, que muestre mensaje de error. Si todo fue bien, mostrará el mensaje de Nivel insertado correctamente").

En el navegador al ejecutar verán:

← → ↻ localhost/Clase_08/EjemplosBD/EJEMPLOS_INSERT/insert_niveles.php

Acceso al Mysql del Localhost... ok

El Nivel se ha insertado correctamente!

Excelente! Desde PHP ya enviamos datos para la base! Si van a la base verán el nuevo nivel agregado.

← → ↻ localhost/phpmyadmin/sql.php?server=1&db=panel&table=niveles&pos=0

phpMyAdmin

Reciente Favoritas

Nueva

- information_schema
- mysql
- panel
 - Nueva
 - niveles
 - países
 - usuarios
- performance_schema
- phpmyadmin

Servidor: 127.0.0.1 » Base de datos: panel » Tabla: niveles

Examinar Estructura SQL Buscar In

✓ Mostrando filas 0 - 2 (total de 3, La consulta tardó 0,0021 segundos.)

SELECT * FROM `niveles`

☐ Mostrar todo | Número de filas: 25 ▼ Filtrar filas: E

+ Opciones

					Id	Denominacion
<input type="checkbox"/>	Editar	Copiar	Borrar		1	Admin
<input type="checkbox"/>	Editar	Copiar	Borrar		2	Basico
<input type="checkbox"/>	Editar	Copiar	Borrar		3	Nivel xx

Cabe aclarar que por ahora el navegante no ingresa valores a mano, sino que los valores insertados se programan para comenzar con ejemplos básicos.

Tarea #1



Desde el siguiente script:

C:\wamp\www\Clase_08\Ejemplos_BD\EJEMPLOS_INSERT\insert_paises.php

Analicen siguiendo los comentarios del script, ejecuten en el navegador y verifiquen en la base de datos los países insertados.

Prueben de cambiar y generar nuevos valores de Países y Niveles.

Listado de registros

Ahora veremos cómo traer registros DESDE nuestra base. Con el *INSERT* alojamos nuevos valores, con el *SELECT* tomamos y mostramos lo que necesitamos.

Deberán ubicar el script: C:\wamp\www\Clase_08\Ejemplos_BD\EJEMPLOS_SELECT\listado_paises.php

```

Clase_08 > EjemplosBD > EJEMPLOS_SELECT > listado_paises.php > ...
/
$password = '';
8 $BaseDeDatos = 'panel';
9
10 //procedo al intento de conexión con esos parametros
11 $linkConexion = mysqli_connect($Host, $User, $Password, $BaseDeDatos);
12
13 if ($linkConexion != false) {
14     //quiero traer un listado de los PAISES cargados
15     //1) genero la consulta que deseo
16     $SQL = "SELECT Id, Denominacion FROM paises ORDER BY 1";
17
18     //2) a la conexión actual le brindo mi consulta, y el resultado lo entrego a variable $rs
19     $rs = mysqli_query($linkConexion, $SQL);
20
21     //3) el resultado deberá organizarse en una matriz, entonces lo recorro
22     //mientras haya resultados, voy tomando cada dato del resultado en el array $data
23     //cada elemento se corresponde a un campo de mi consulta
24     //estructuro mi propio array $Listado
25     $i=0;
26     while ($data = mysqli_fetch_array($rs)) {
27         $Listado[$i]['ID'] = $data['Id'];
28         $Listado[$i]['NOMBRE'] = $data['Denominacion'];
29         $i++;
30     }
31 }

```

- ❖ Noten que se conecta a la base, y si todo va bien, creo mi consulta **SQL** para **SELECCIONAR** todos los registros de la tabla de Países. (Línea 18)
SELECT campo1, campo2, FROM tabla.
- ❖ Esa consulta también pasa a una variable, y esta variable junto a la de conexión son brindadas como parámetro a la función **mysqli_query()**. (línea 21) Esta función de PHP ejecutará la consulta de selección SQL, y al resultado de esa consulta se lo damos a una variable de PHP llamada **\$rs** [llamada así por *recordset*, conjunto de registros.]
- ❖ Ese **resultado** será organizado de una manera especial, para poder contar con los datos.
- ❖ Deberemos recorrer todo ese conjunto de registros obtenido, mediante el **while**, con la condición de hacerlo mientras haya resultados en el **\$rs**. Aquí iremos tomando cada fila de resultados y generando nuestro propio array **\$Listado**, para luego usarlo según lo necesitemos. (Líneas 27 a 32).
- ❖ Noten que en líneas 29 y 30, tomo los nombres de los campos de la tabla, y se los asigno a mi array **\$Listado**. Este array contiene ID y NOMBRE como elementos con los valores que les estamos asignando.
- ❖ La función de PHP aquí utilizada es **mysql_fetch_array()**, la cual irá procesando los registros traídos como un array con **\$data**. Cada elemento de ese array tiene como **identificador** al **nombre del campo** brindado en la consulta **SQL**, recuerden respetar mayúsculas y minúsculas.

```

25     $i=0;
26     while ($data = mysqli_fetch_array($rs)) {
27         $Listado[$i]['ID'] = $data['Id'];
28         $Listado[$i]['NOMBRE'] = $data['Denominacion'];
29         $i++;
30     }

```

Por cada valor en el \$data, abriremos un elemento en nuestro array **\$Listado**, el cual será una **matriz**. [Recuerden incrementar \$i].

Noten que al **\$Listado** ya puedo **nombrar** cada elemento **como quiera**, en este caso cada subelemento tiene el nombre del campo, pero en mayúscula.

Cuando la condición deje de cumplirse, es porque no hay más registros traídos en esa SQL. Ya tendremos nuestro \$Listado listo para ser usado.

Debajo simplemente pregunto si tiene valores, es decir, no esté vacío, y puedo recorrerlo con un for, como ya veníamos trabajando con matrices y vectores.

```

33     //si $Listado contiene valores, ya dispongo de los datos
34     if (!empty($Listado)) {
35         echo '<h3>Listado de Países</h3>';
36         $cantidadRegistros = count($Listado);
37         for ($i = 0; $i < $cantidadRegistros; $i++) {
38             echo "El elemento $i de mi listado contiene: <br />
39                 El ID --> {$Listado[$i]['ID']} <br />
40                 Nombre --> {$Listado[$i]['NOMBRE']} <br />
41                 <hr />
42                 ";
43         }
44     }

```

Noten que al usar comillas dobles en el *echo*, pongo entre llaves los valores del array multidimensional \$Listado, para brindar precisión a PHP que todo esos caracteres conforman el elemento del array.

Prueben en el navegador y verán el listado:

http://localhost/Clase_08/EjemplosBD/EJEMPLOS_SELECT/listado_paises.php

← → ↻ 🌐 localhost/Clase_08/EjemplosBD/EJEMPLOS_SELECT/listado_paises.php

Listado de Países

El elemento 0 de mi listado contiene:

El ID --> 1

Nombre --> Argentina

El elemento 1 de mi listado contiene:

El ID --> 2

Nombre --> Chile

El elemento 2 de mi listado contiene:

El ID --> 3

Nombre --> Brasil

Tarea #2



Desde el siguiente script:

C:\wamp\www\Clase_08\Ejemplos_BD\EJEMPLOS_SELECT\listado_niveles.php

Analicen siguiendo los comentarios del script, ejecuten en el navegador y verifiquen el listado.

Noten que en la consulta ahora tengo un asterisco, indica que voy a usar TODOS los campos de esa tabla.

Tarea #3



Aquí tenemos un listado con un filtro LIKE para un listado.

Desde el siguiente script:

C:\wamp\www\Clase_08\Ejemplos_BD\EJEMPLOS_SELECT\listado_paises_like.php

Analicen siguiendo los comentarios del script, ejecuten en el navegador y verifiquen el listado.

Prueben de insertar nuevos países (con el script que vimos primero), que comiencen con "A", y ejecutan nuevamente este script, así ven el filtro en acción.

Cómo harían si quisieran ver el listado ordenado alfabéticamente?

Continuamos!!