

19 - Base de datos (MySQL)

Uno de los empleos principales de PHP es el acceso a una base de datos en el servidor. Las operaciones básicas se hacen empleando como lenguaje el SQL.

PHP implementa distintas funciones según la base de datos a emplear. Existen funciones actualmente para acceder a las siguientes servidores de bases de datos:

- MySQL
- Microsoft SQL Server
- Oracle
- PostgreSQL
- SysBase
- FrontBase
- Informix
- InterBase
- Ingres
- mSQL
- dBase
- SQLite

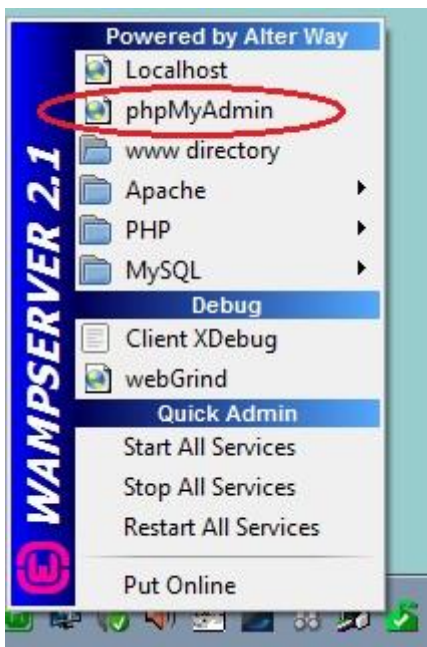
El más empleado en la actualidad en la web es el gestor de base de datos MySQL.

Cuando instaló el WampServer en un principio para trabajar con PHP, se instaló el MySQL (recordemos que las letras que forman el WampServer son : W(Windows)A(Apache)M(MySQL)P(PHP)Server).

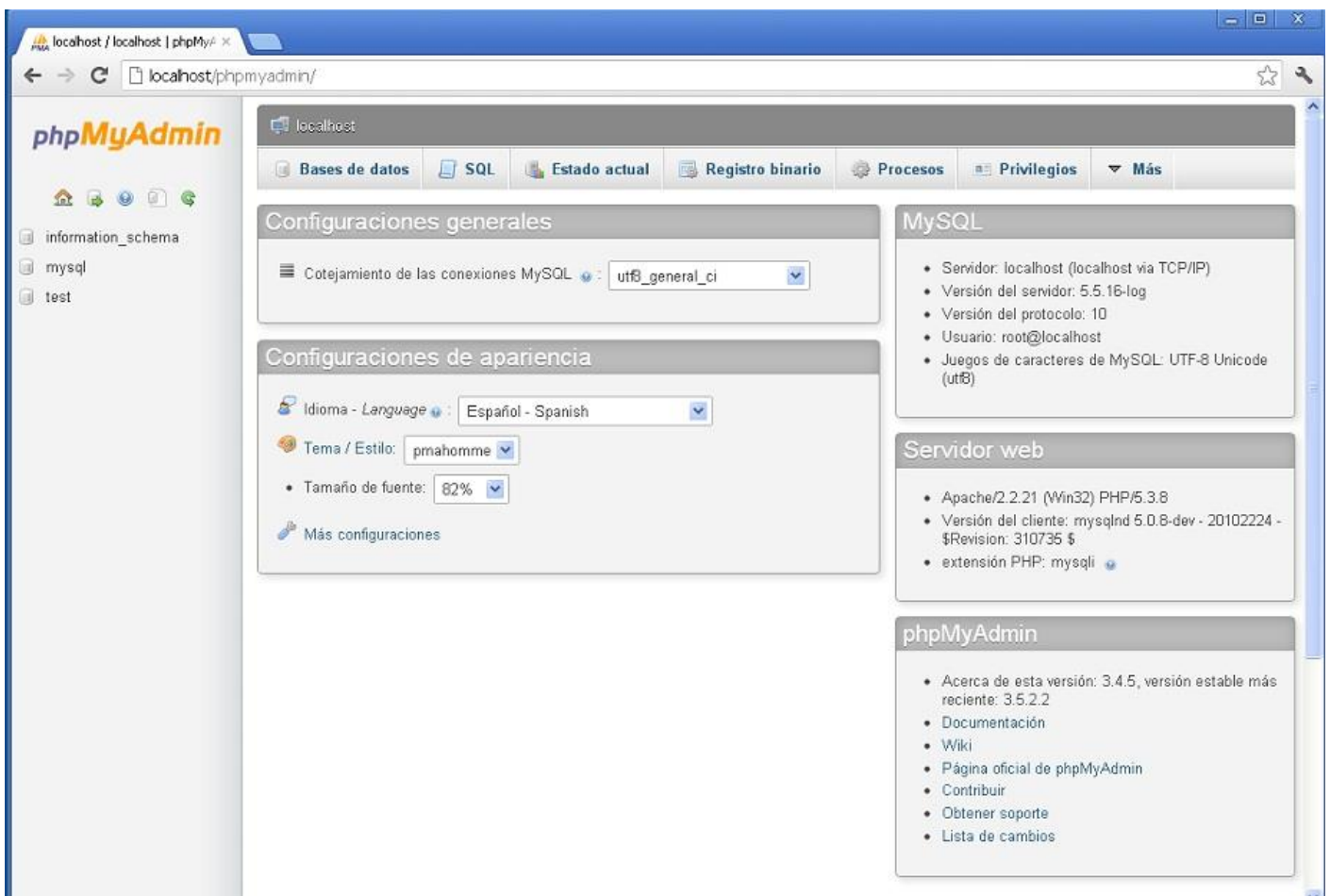
Para crear una base de datos el WampServer instala también un programa codificado en PHP que nos permite interactuar con el MySQL.

Este programa se llama PHPMyAdmin (como veremos nos permite crear las bases de datos, tablas, índices, usuarios etc.)

Para iniciar el PHPMyAdmin debemos presionar el botón izquierdo del mouse sobre el ícono del WampServer que se encuentra en la bandeja del sistema:



Como podemos ver la interfaz del PHPMyAdmin es un programa que se ejecuta en la web:



Para crear una base de datos procedemos a seleccionar la pestaña "Base de datos" e ingresamos como nombre "base1" y presionamos el botón crear:

localhost / localhost | phpMyAdmin

localhost/phpmyadmin/

phpMyAdmin

information_schema
mysql
test

localhost

Bases de datos SQL Estado actual Registro binario Proceso

Bases de datos

Crear nueva base de datos

base1 Cotejamiento **Crear**

Base de datos	Replicación maestra	
<input type="checkbox"/> information_schema	✓ Replicado/a	Comprobar los privilegios
<input type="checkbox"/> mysql	✓ Replicado/a	Comprobar los privilegios
<input type="checkbox"/> test	✓ Replicado/a	Comprobar los privilegios
Total: 3		

Marcar todos / Desmarcar todos Para los elementos que están marcados: Eliminar

Activar las estadísticas

Luego de crear la base de datos podemos ver que aparece en el lado izquierdo:

localhost / localhost | phpMyAdmin

localhost/phpmyadmin/

phpMyAdmin

base1
information_schema
mysql
test

localhost

Bases de datos SQL Estado actual Registro binario Proceso

Bases de datos

Crear nueva base de datos

base1 Cotejamiento **Crear**

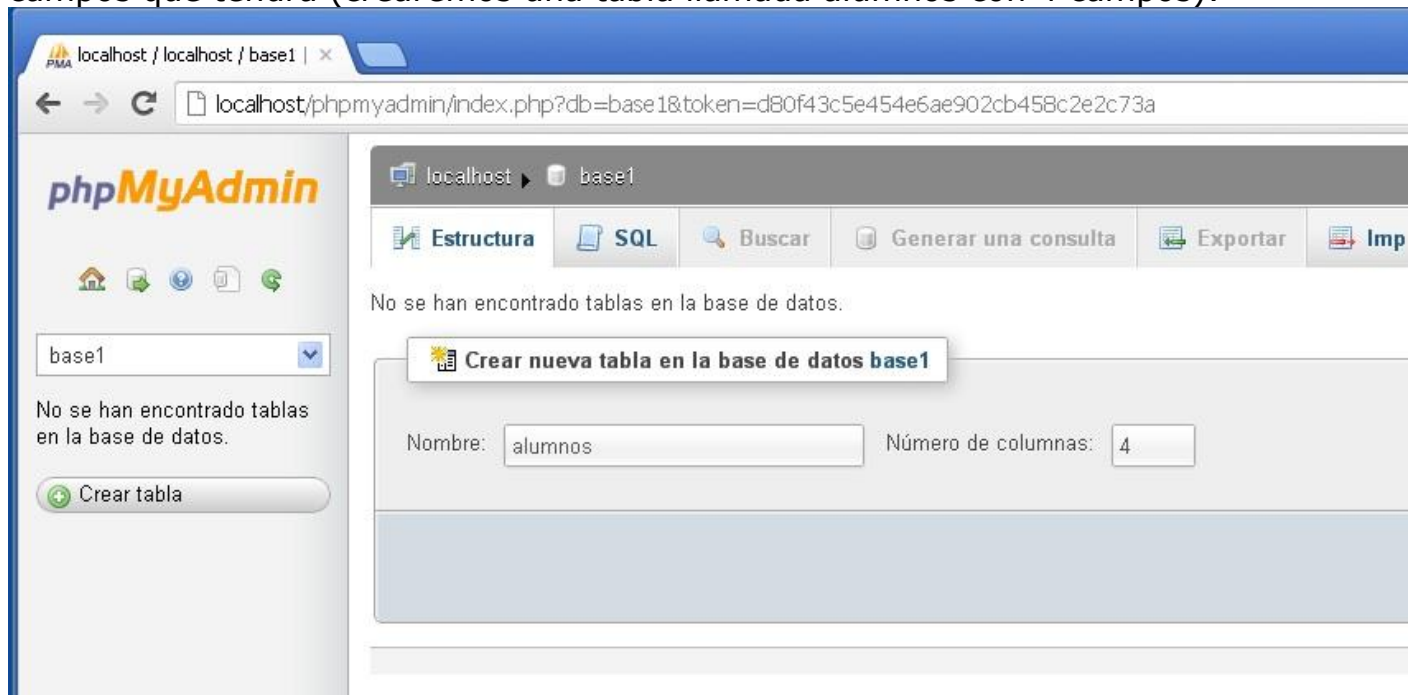
Base de datos	Replicación maestra	
<input type="checkbox"/> base1	✓ Replicado/a	Comprobar los privilegios
<input type="checkbox"/> information_schema	✓ Replicado/a	Comprobar los privilegios
<input type="checkbox"/> mysql	✓ Replicado/a	Comprobar los privilegios
<input type="checkbox"/> test	✓ Replicado/a	Comprobar los privilegios
Total: 4		

Marcar todos / Desmarcar todos Para los elementos que están marcados: Eliminar

Activar las estadísticas

Seleccionamos el nombre de la base de datos "base1" y se actualiza la interfaz de pantalla para que ingresemos el nombre de una tabla y la cantidad de

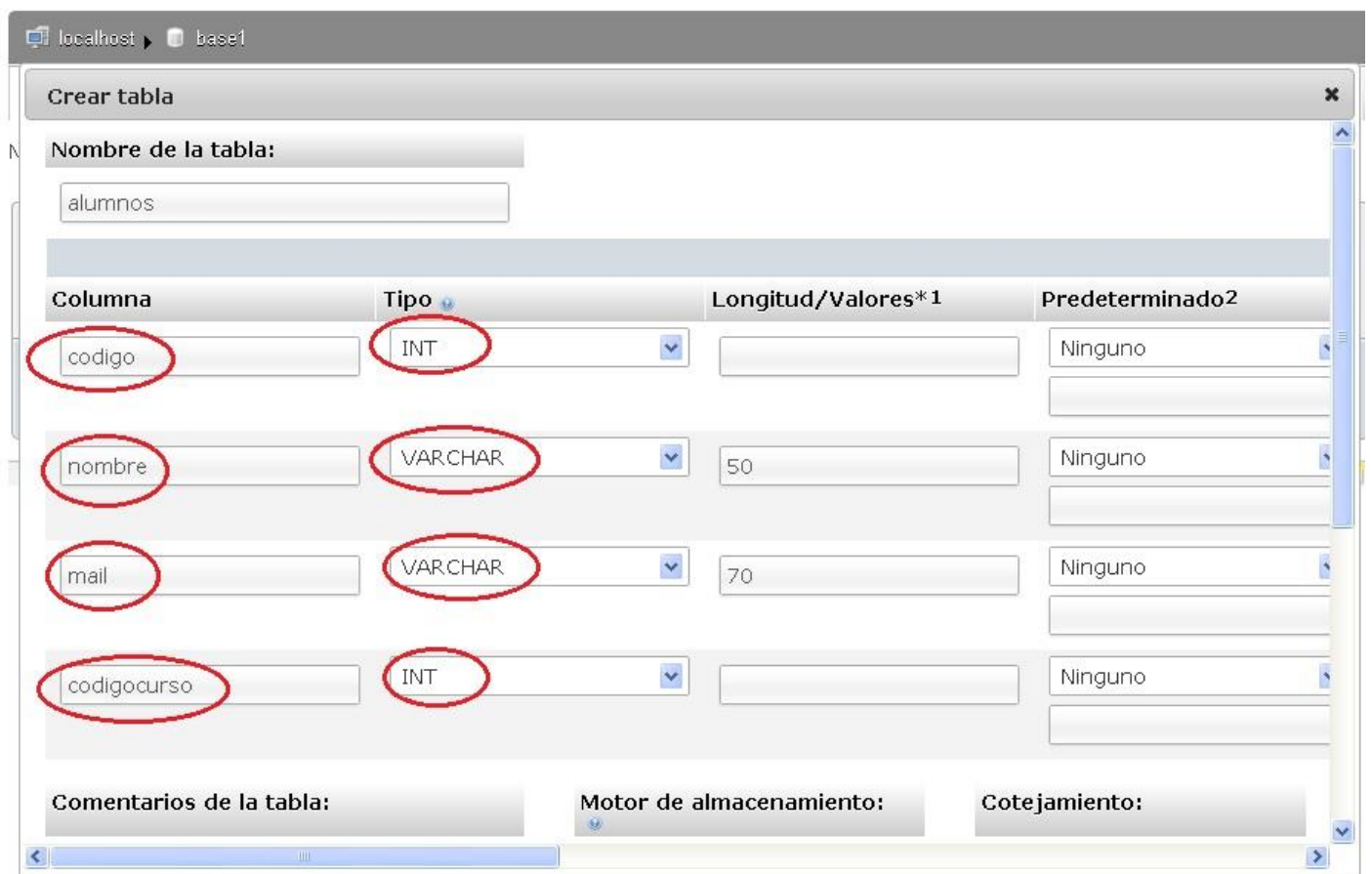
campos que tendrá (crearemos una tabla llamada alumnos con 4 campos):



La estructura de la tabla es:

```
codigo int auto_increment primary key
nombre varchar(50)
mail varchar(70)
codigocurso int
```

En el PHPMyAdmin ingresamos:



Es importante también hacer notar que en el campo código debemos marcar en Índice el valor "primary" y tildar la columna A_I:

Nulo	Índice	A_I	Comentarios
<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	---	<input type="checkbox"/>	
<input type="checkbox"/>	---	<input type="checkbox"/>	
<input type="checkbox"/>	---	<input type="checkbox"/>	

Por último presionamos el botón guardar y ya tenemos la tabla "alumnos" creada en la base de datos "base1":

Crear tabla

Comentarios de la tabla:

Motor de almacenamiento:

InnoDB

Cotejamie

definición de la PARTICIÓN:

Guardar

o Agregar

1

columna(s)

Continuar

Cancel

La tabla almacenará datos de alumnos que desarrollarán cursos de programación en PHP, ASP y JSP.

El código del alumno es de tipo numérico (int) y al indicar que es auto_increment se generará automáticamente por el gestor de base de datos.

Los campos nombre y mail son de tipo varchar (podemos almacenar cualquier carácter) y por último el campo codigocurso representa el curso a tomar por el alumno (1=PHP, 2=ASP y 3=JSP)

El campo clave de esta tabla es el código de alumno (es decir no podemos tener dos

alumnos con el mismo código, no así el nombre del alumno que puede eventualmente repetirse)

En los próximos conceptos comenzaremos a ver como desde PHP podemos comunicarnos con la base de datos "base1" y acceder a la tabla "alumnos" para ejecutar los comandos SQL más comunes como pueden ser: select, insert, delete, update etc.

20 - INSERT (Alta de registros en una tabla)

Luego de crear una base de datos y sus tablas (Vamos a trabajar con la base de datos ya creada: base1, que contiene la tabla alumnos), veremos como agregar registros.

Para añadir datos en la tabla empleamos el comando SQL llamado insert.

Necesitamos dos páginas para este proceso, una será el formulario de carga de datos y la siguiente será la que efectúe la inserción en la tabla.

Formulario de carga de datos:

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<h1>Alta de Alumnos</h1>
<form action="pagina2.php" method="post">
Ingrese nombre:
<input type="text" name="nombre"><br>
Ingrese mail:
<input type="text" name="mail"><br>
Seleccione el curso:
<select name="codigocurso">
<option value="1">PHP</option>
<option value="2">ASP</option>
<option value="3">JSP</option>
</select>
<br>
<input type="submit" value="Registrar">
</form>
</body>
</html>
```

El formulario es bastante similar a los que venimos desarrollando en puntos anteriores, tal vez lo distinto es cómo emplearemos el control "select" del curso a desarrollar:

```
<select name="codigocurso">
<option value="1">PHP</option>
<option value="2">ASP</option>
<option value="3">JSP</option>
</select>
```


Cada opción tiene su respectivo valor (en este caso los números 1,2 y 3) y los textos a mostrar PHP, ASP y JSP. El dato que se envía a la otra página es el código de curso (esto debido a que definimos la propiedad value).

Ahora veremos como realizar la registración de los datos cargados en el formulario, en la tabla alumnos de la base de datos base1:

```
<html>
<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

mysqli_query($conexion,"insert into alumnos(nombre,mail,codigocurso) values
('$REQUEST[nombre]','$REQUEST[mail]','$REQUEST[codigocurso])")
    or die("Problemas en el select".mysqli_error($conexion));

mysqli_close($conexion);

echo "El alumno fue dado de alta.";
?>
</body>
</html>
```

Veamos los pasos para efectuar el alta en la tabla alumnos:

```
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");
```

La función `mysqli_connect` se conecta a una base de datos de tipo **MySQL**, el primer parámetro es la dirección donde se encuentra el gestor de base de datos (en este caso en el mismo servidor por lo que indicamos esto con "localhost"), el segundo parámetro es el nombre de usuario de la base de datos ("root" en nuestro caso, que es el usuario por defecto que crea MySQL para el administrador), seguidamente indicamos la clave del usuario root (por defecto al instalar el Wamp se crea con clave vacía) y por último indicamos el nombre de la base de datos a conectarnos (en nuestro ejemplo ya creamos la base de datos llamada: base1 que tiene la tabla alumnos)

En caso de haber algún error en la llamada a la función la misma retorna false por lo que se ejecuta la instrucción seguida del operador `or`, en nuestro caso llamamos a la función `die` que detiene la ejecución del programa y muestra el mensaje por pantalla.

El paso más importante es la codificación del comando SQL insert(debemos llamar a la función **`mysqli_query`** pasando como primer parámetro la referencia a la conexion y el segundo parámetro es un string con el comando insert):

```
mysqli_query($conexion,"insert into alumnos(nombre,mail,codigocurso) values
('$REQUEST[nombre]','$REQUEST[mail]','$REQUEST[codigocurso])")
    or die("Problemas en el select".mysqli_error($conexion));
```

La sintaxis del comando insert es bastante sencilla, indicamos el nombre de la tabla y los campos de la tabla a cargar. Luego debemos indicar en el mismo orden los valores a cargar en cada campo (dichos valores los rescatamos del formulario anterior).

Los campos de tipo varchar SQL requiere que encerremos entre comillas simples, esto sucede para el nombre y el mail; en cambio, para el codigocurso esto no debe ser así.

Otra cosa a tener en cuenta es que los subíndices de los vectores asociativos no deben ir entre simples comillas ya que se encuentran dentro de un string, sino se producirá un error.

En caso que MySql detecte un error, retorna false esta función, por lo que se ejecuta la instrucción posterior al or, es decir la función die que mostrará el error generado por MySql llamando a la función mysqli_error().

Por último cerramos la conexión con la base de datos y mostramos un mensaje indicando que la carga se efectuó en forma correcta.

Tener en cuenta que el campo código se generó en forma automática.

Si queremos controlar que el insert se efectuó en forma correcta podemos ingresar al PHPMyAdmin y seleccionar la tabla "alumnos", y en la pestaña "examinar" podremos ver los datos ingresados desde la página que creamos:

The screenshot shows the PHPMyAdmin interface for a database named 'base1'. The 'alumnos' table is selected, and the 'Examinar' (Browse) tab is active. The table structure is shown as follows:

codigo	nombre	mail	codigocurso
1	Martinez Luis	martinezluis@gmail.com	1

The interface also displays the SQL query used to retrieve the data: `SELECT * FROM `alumnos` LIMIT 0, 30`. The 'Examinar' tab and the 'alumnos' table name in the sidebar are circled in red.

Problema propuesto.

Crear en la base de datos "base1" otra tabla llamada "cursos".

La estructura de esta segunda tabla debe ser:

```
codigo int auto_increment primery_key  
  
nombrecurso varchar(40)
```

Utilizar el PHPMyAdmin para la creación de esta tabla.

Implementar las dos páginas necesarias para efectuar el alta de cursos. Un formulario para ingresar el nombre del curso y otra página donde se efectuará el insert.

```
<html>  
<head>  
<title>Problema</title>  
</head>  
<body>  
  
<form action="pagina20_1_resposta.php" method="post">  
Ingrese nombre del curso:  
<input type="text" name="nombrecurso" size="40">  
<br>  
<input type="submit" value="Confirmar">  
</form>  
  
</body>  
</html>
```

```
<html>  
<head>  
<title>Problema</title>  
</head>  
<body>  
<?php  
    $conexion=mysqli_connect("localhost","root","","base1") or  
        die("Problemas con la conexión");  
  
    mysqli_query($conexion,"insert into cursos(nombrecurso) values  
( '$_REQUEST[nombrecurso]' )" ) or  
        die("Problemas en el select".mysqli_error($conexion));  
  
    mysqli_close($conexion);  
  
    echo "El curso fue dado de alta."  
?>  
</body>  
</html>
```

21 - Listado (selección de registros de una tabla)

Para recuperar datos desde MySQL debemos emplear el comando select:

```
select codigo,nombre,mail,codigocurso from alumnos
```

Debemos pasar desde PHP un string con este comando para que MySQL lo ejecute y retorne todas las filas de la tabla alumnos.

Veremos entonces como recuperar los datos almacenados en la tabla alumnos de la base de datos "base1".

El programa que muestra los registros en una página es:

```
<html>
<head>
<title>Problema</title>
</head>
<body>

<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select codigo,nombre, mail, codigocurso
                                from alumnos") or
    die("Problemas en el select:".mysqli_error($conexion));

while ($reg=mysqli_fetch_array($registros))
{
    echo "Codigo:".$reg['codigo']."<br>";
    echo "Nombre:".$reg['nombre']."<br>";
    echo "Mail:".$reg['mail']."<br>";
    echo "Curso:";
    switch ($reg['codigocurso']) {
        case 1:echo "PHP";
            break;
        case 2:echo "ASP";
            break;
        case 3:echo "JSP";
            break;
    }
    echo "<br>";
    echo "<hr>";
}

mysqli_close($conexion);
?>

</body>
</html>
```

La primera parte es similar a lo visto hasta ahora, es decir nos conectamos con el servidor de base de datos y seleccionamos la base de datos base1:

```
$conexion=mysqli_connect("localhost","root","","base1") or
```

```
die("Problemas con la conexión");
```

El comando SQL que nos permite recuperar datos de tablas se llama SELECT. Indicamos los campos a rescatar de la tabla y luego de la palabra clave from indicamos el nombre de la tabla:

```
$registros=mysqli_query($conexion,"select codigo,nombre, mail, codigocurso  
from alumnos") or  
die("Problemas en el select:".mysqli_error($conexion));
```

En caso de haber codificado incorrectamente, el comando SQL select la función mysqli_query retorna false, por lo que se ejecuta el comando siguiente al operador or, es decir la función die.

Si el comando SQL es correcto, en la variable \$registros se almacena una referencia a los datos rescatados de la tabla alumnos. Ahora debemos ir mostrando registro a registro los datos extraídos:

```
while ($reg=mysqli_fetch_array($registros))
```

Para rescatar registro a registro los datos obtenidos por el select debemos llamar a la función **mysqli_fetch_array**. Esta función retorna un vector asociativo con los datos del registro rescatado, o false en caso de no haber más registros. Es decir que si retorna un registro se almacena en el vector \$reg y la condición del while se valida como verdadero y pasa a ejecutarse el bloque del while:

```
{  
    echo "Codigo:".$reg['codigo']."<br>";  
    echo "Nombre:".$reg['nombre']."<br>";  
    echo "Mail:".$reg['mail']."<br>";  
    echo "Curso:";  
    switch ($reg['codigocurso']) {  
        case 1:echo "PHP";  
            break;  
        case 2:echo "ASP";  
            break;  
        case 3:echo "JSP";  
            break;  
    }  
    echo "<br>";  
    echo "<hr>";  
}
```

El bloque del while muestra el contenido del registro rescatado por la función **mysqli_fetch_array**. Como vemos, para rescatar cada campo accedemos mediante el vector asociativo \$reg indicando como subíndice un campo indicado en el select:\$reg['codigo']

Cada vez que llamamos a la función **mysqli_fetch_array** nos retorna el siguiente registro.

Cuando debemos mostrar el curso mediante la instrucción switch, analizamos si tiene un 1,2 ó 3 y procedemos a mostrar el nombre del curso.

Para separar cada alumno en la página HTML disponemos el elemento "<hr>"

Problema propuesto.

Confeccionar un programa que recupere los datos de la tabla cursos de la base de datos base1. Mostrar el código de curso y su nombre.

```
<html>
<head>
<title>Problema</title>
</head>
<body>

<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select codigo,nombrecurso from cursos") or
    die("Problemas en el select:".mysqli_error($conexion));

while ($reg=mysqli_fetch_array($registros))
{
    echo "Codigo:". $reg['codigo']. "<br>";
    echo "Nombre del curso:". $reg['nombrecurso']. "<br>";
    echo "<hr>";
}

mysqli_close($conexion);
?>

</body>
</html>
```

22 - Consulta (selección de registros de una tabla)

El proceso de consulta de datos de una tabla es similar al del listado, la diferencia es que se muestra sólo aquel o aquellos que cumplen la condición por la que buscamos.

Haremos un programa que nos permita consultar los datos de un alumno ingresando su mail para su búsqueda. Tengamos en cuenta que no puede haber dos alumnos con el mismo mail, por lo que la consulta nos puede arrojar uno o ningún registro.

Debemos codificar un formulario para el ingreso del mail a consultar:

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<form action="pagina2.php" method="post">
Ingrese el mail del alumno a consultar:
<input type="text" name="mail">
<br>
<input type="submit" value="buscar">
</form>
</body>
```

```
</html>
```

Por otro lado tenemos el archivo "pagina2.php" que se encarga de buscar el mail ingresado en el formulario:

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select codigo,nombre, codigocurso
                                from alumnos where mail='$_REQUEST[mail]'" ) or
    die("Problemas en el select:".mysqli_error($conexion));

if ($reg=mysqli_fetch_array($registros))
{
    echo "Nombre:".$reg['nombre']."<br>";
    echo "Curso:";
    switch ($reg['codigocurso']) {
        case 1:echo "PHP";
            break;
        case 2:echo "ASP";
            break;
        case 3:echo "JSP";
            break;
    }
}
else
{
    echo "No existe un alumno con ese mail.";
}
mysqli_close($conexion);
?>
</body>
</html>
```

Lo más importante está en el comando select:

```
$registros=mysqli_query($conexion,"select codigo,nombre, codigocurso
                                from alumnos where mail='$_REQUEST[mail]'" ) or
    die("Problemas en el select:".mysqli_error($conexion));
```

Acá es donde con la clausula where seleccionamos sólo el registro que cumple con la condición que el mail sea igual al que ingresamos.

Como sólo puede haber un registro que cumpla la condición, llamamos a la función **mysqli_fetch_array** en **un if** y no una estructura repetitiva como el listado:

```
if ($reg=mysqli_fetch_array($registros))
```

En caso de retornar un vector asociativo la condición del if se verifica como verdadera y pasa a mostrar los datos, en caso de retornar false se ejecuta el else.

Problema propuesto.

Confeccionar un programa que permita ingresar el nombre de un alumno en un formulario, luego mostrar los datos del mismo (tener en cuenta que puede haber más de un alumno con el mismo nombre)

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<form action="pagina22_1_resposta.php" method="post">
Ingrese nombre del/los alumno/s:
<input type="text" name="nombre">
<br>
<input type="submit" value="buscar">
</form>
</body>
</html>
```

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select codigo,nombre, codigocurso
                                from alumnos where nombre='$_REQUEST[nombre]'") or
    die("Problemas en el select:".mysqli_error($conexion));

$cant=0;
while ($reg=mysqli_fetch_array($registros))
{
    echo "Nombre: ".$reg['nombre']."<br>";
    echo "Curso:";
    switch ($reg['codigocurso']) {
        case 1:echo "PHP";
            break;
        case 2:echo "ASP";
            break;
        case 3:echo "JSP";
            break;
    }
    $cant++;
    echo "<hr>";
}
```



```

}

if ($cant==0)
{
    echo "No existen alumnos con dicho nombre";
}

mysqli_close($conexion);
?>
</body>
</html>

```

23 - DELETE (Baja de un registro en una tabla)

El objetivo de este punto es el borrado de un registro de una tabla. Para ello, implementaremos un algoritmo que solicite ingresar el mail de un alumno y posteriormente efectúe su borrado.

Para eliminar filas en una tabla debemos utilizar el comando SQL delete.

La primera página es idéntica a la consulta, ya que debemos implementar un formulario que solicite la carga del mail del alumno:

```

<html>
<head>
<title>Problema</title>
</head>
<body>
<form action="pagina2.php" method="post">
Ingrese el mail del alumno a borrar:
<input type="text" name="mail">
<br>
<input type="submit" value="buscar">
</form>
</body>
</html>

```

Por otro lado tenemos el archivo "pagina2.php" que se encarga de buscar el mail ingresado en el formulario y en caso que exista se procede a borrarlo:

```

<html>
<head>
<title>Problema</title>
</head>
<body>

```

```

<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion, "select codigo from alumnos
                                where mail='$_REQUEST[mail]') or
    die("Problemas en el select:".mysqli_error($conexion));
if ($reg=mysqli_fetch_array($registros))
{
    mysqli_query($conexion,"delete from alumnos where mail='$_REQUEST[mail]')
or
    die("Problemas en el select:".mysqli_error($conexion));
    echo "Se efectuó el borrado del alumno con dicho mail.";
}
else
{
    echo "No existe un alumno con ese mail.";
}
mysqli_close($conexion);
?>
</body>
</html>

```

En esta segunda página efectuamos dos llamadas a la función `mysqli_query`, una para consultar si existe el mail ingresado y otra para efectuar el borrado del registro respectivo. Si no existe el mail ingresado mostramos un mensaje de tal situación.

Problema propuesto.

Confeccionar un programa que permita ingresar el nombre de un curso por teclado y posteriormente efectúe el borrado de dicho registro en la tabla `cursos`. Mostrar un mensaje si no existe el curso.

```

<html>
<head>
<title>Problema</title>
</head>
<body>
<form action="pagina23_1_resposta.php" method="post">
Ingrese nombre del curso a borrar:
<input type="text" name="nombrecurso">
<br>
<input type="submit" value="Borrar">
</form>
</body>
</html>

```

```

<html>
<head>
<title>Problema</title>
</head>

```

```

<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select codigo from cursos
                                where nombrecurso='$_REQUEST[nombrecurso]'" ) or
    die("Problemas en el select:".mysqli_error($conexion));
if ($reg=mysqli_fetch_array($registros))
{
    mysqli_query($conexion,"delete from cursos where nombrecurso='$_REQUEST[nombrecurso]'" )
or
    die("Problemas en el select:".mysqli_error($conexion));
    echo "Se efectuó el borrado del curso ingresado.";
}
else
{
    echo "No existe un curso con dicho nombre.";
}
mysqli_close($conexion);
?>
</body>
</html>

```

24 - DELETE (Baja de todos los registros de una tabla)

Vimos en el concepto anterior que podemos borrar una o más filas de una tabla indicando en el where del comando SQL delete la condición que debe cumplir la fila para ser borrada.

Para borrar todos los registros de una tabla debemos llamar al comando delete de SQL sin disponer la cláusula where.

Es importante tener en cuenta que la ausencia de la cláusula where en el comando delete eliminará todas las filas de la tabla.

```

<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

mysqli_query($conexion,"delete from alumnos") or
    die("Problemas en el select:".mysqli_error($conexion));
echo "Se efectuó el borrado de todos los alumnos.";
mysqli_close($conexion);
?>
</body>
</html>

```

Problema propuesto.

Efectuar el borrado de todos los registros de la tabla cursos.

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

mysqli_query($conexion,"delete from cursos") or
    die("Problemas en el select:".mysqli_error($conexion));
echo "Se efectuó el borrado de todos los cursos.";
mysqli_close($conexion);
?>
</body>
</html>
```

25 - UPDATE (Modificación de un registro de una tabla)

De las actividades con una tabla esta es la más larga. Vamos a resolverlo implementando tres páginas, la primera un formulario de consulta del mail de un alumno, la segunda otro formulario que nos permita cargar su mail modificado y la última registrará el cambio en la tabla.

El formulario de consulta del mail del alumno es similar a problemas anteriores:

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<form action="pagina2.php" method="post">
Ingrese el mail del alumno:
<input type="text" name="mail"><br>
<input type="submit" value="buscar">
</form>
</body>
</html>
```

La segunda página es la más interesante y con conceptos nuevos:

```
<html>
<head>
<title>Problema</title>
```

```

</head>
<body>

<?php

$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select * from alumnos
                                where mail='$_REQUEST[mail]'") or
    die("Problemas en el select:".mysqli_error($conexion));
if ($reg=mysqli_fetch_array($registros))
{
?>

<form action="pagina3.php" method="post">
Ingrese nuevo mail:
<input type="text" name="mailnuevo" value="<?php echo $reg['mail'] ?>">
<br>
<input type="hidden" name="mailviejo" value="<?php echo $reg['mail'] ?>">
<input type="submit" value="Modificar">
</form>

<?php
}
else
    echo "No existe alumno con dicho mail";
?>
</body>
</html>

```

Lo primero que podemos observar es que si el if se verifica verdadero se ejecuta un bloque que contiene código HTML:

```

if ($reg=mysqli_fetch_array($registros))
{
?>
    <form action="pagina3.php" method="post">
    Ingrese nuevo mail:
    <input type="text" name="mailnuevo" value="<?php echo $reg['mail'] ?>">
    <br>
    <input type="hidden" name="mailviejo" value="<?php echo $reg['mail'] ?>">
    <input type="submit" value="Modificar">
    </form>
<?php
}

```

Es decir que podemos disponer bloques de PHP dispersos dentro de la página.

Otro concepto importante es como enviar el mail del primer formulario a la tercer página, esto se logra con los controles de tipo "hidden", este tipo de control no se visualiza en el formulario pero se envía al presionar el botón submit.

Si queremos que el control text se cargue con el mail ingresado en el formulario anterior debemos cargar la propiedad value con dicho valor:

```
<input type="text" name="mailnuevo" value="<?php echo $reg['mail'] ?>">
```

Por último la pagina3.php es la que efectúa la modificación de la tabla propiamente dicha. Con el mail ingresado en la pagina1.php, el mail modificado en la pagina2.php se efectúa el update.

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

mysqli_query($conexion, "update alumnos
                        set mail='$_REQUEST[mailnuevo]'
                        where mail='$_REQUEST[mailviejo]'" ) or
    die("Problemas en el select:".mysqli_error($conexion));
echo "El mail fue modificado con exito";
?>
</body>
</html>
```

Tengamos en cuenta que el segundo formulario nos envía dos datos: \$_REQUEST[mailnuevo] y \$_REQUEST[mailviejo].

Problema propuesto.

Efectuar la modificación del nombre del curso de la tabla "cursos". Para la búsqueda ingresar el código de curso.

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<form action="pagina25_1_procesa.php" method="post">
Ingresa el código del curso a modificar:
<input type="text" name="codigo"><br>
<input type="submit" value="buscar">
</form>
</body>
</html>
```

```
<html>
<head>
<title>Problema</title>
</head>
<body>
```



```

<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select * from cursos
                                where codigo=$_REQUEST[codigo]") or
    die("Problemas en el select:".mysqli_error($conexion));
if ($reg=mysqli_fetch_array($registros))
{
?>

<form action="pagina25_1_resultado.php" method="post">
Ingrese el nuevo nombre del curso:
<input type="text" name="nombrecurso" value="<?php echo $reg['nombrecurso'] ?>">
<input type="hidden" name="codigo" value="<?php echo $reg['codigo'] ?>">
<br>
<input type="submit" value="Modificar">
</form>

<?php
}
else
    echo "No existe un curso con dicho codigo";
?>
</body>
</html>

```

```

<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

mysqli_query($conexion,"update cursos
                        set nombrecurso='$_REQUEST[nombrecurso]'
                        where codigo=$_REQUEST[codigo]") or
    die("Problemas en el select:".mysqli_error($conexion));
    echo "Se modificó el nombre del curso";
?>
</body>
</html>

```

INSERT (y consulta de otra tabla)

En un primer paso vimos como efectuar el alta en la tabla alumnos, ahora vamos a ver como resolver el problema del alta de un alumno seleccionando el curso de la tabla "cursos".

Es decir, el formulario de carga de datos no es HTML puro ya que debemos cargar el control "select" con los datos de la tabla cursos.

El código por lo tanto queda de la siguiente forma:

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<form action="pagina2.php" method="post">
Ingrese nombre:
<input type="text" name="nombre"><br>
Ingrese mail:
<input type="text" name="mail"><br>
Seleccione el curso:
<select name="codigocurso">
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select codigo,nombrecurso from cursos") or
    die("Problemas en el select:".mysqli_error($conexion));
while ($reg=mysqli_fetch_array($registros))
{
    echo "<option value=\"".$reg[codigo].\">".$reg[nombrecurso]</option>";
}
?>
</select>
<br>
<input type="submit" value="Registrar">
</form>
</body>
</html>
```

El algoritmo es similar a cuando trabajamos con una tabla, pero el control "select" lo cargamos con los datos de la tabla "cursos":

```
while ($reg=mysqli_fetch_array($registros))
{
    echo "<option value=\"".$reg[codigo].\">".$reg[nombrecurso]</option>";
}
```

Dentro del while generamos todas las opciones que contiene el "select" imprimiendo el campo nombrecurso y asociando el campo codigo a la propiedad value(que es en definitiva el código que necesitamos rescatar en la otra página)

Tengamos en cuenta que cuando queremos imprimir una comilla doble dentro de un string en PHP debemos escaparla con el caracter \

La página que efectúa el insert es exactamente la misma que vimos anteriormente:

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

mysqli_query($conexion,"insert into alumnos(nombre,mail,codigocurso) values
('$REQUEST[nombre]','$REQUEST[mail]','$REQUEST[codigocurso]')") or
    die("Problemas en el select".mysqli_error($conexion));
mysqli_close($conexion);
echo "El alumno fue dado de alta.";
?>
</body>
</html>
```

Problema propuesto.

Confeccionar el alta de la tabla alumnos empleando controles de tipo "radio" para la selección del curso.

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<form action="pagina26_1_resposta.php" method="post">
Ingrese nombre:
<input type="text" name="nombre"><br>
Ingrese mail:
<input type="text" name="mail"><br>
Seleccione el curso:<br>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select codigo,nombrecurso from cursos") or
    die("Problemas en el select:".mysql_error());
while ($reg=mysqli_fetch_array($registros))
{
    echo "<input type='radio' name='radio1'
value='\"$reg[codigo]\"'>\".$reg['nombrecurso']\".<br>";
}
?>
<input type="submit" value="Registrar">
</form>
</body>
```

```
</html>
```

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

mysqli_query($conexion,"insert into alumnos(nombre,mail,codigocurso) values
('$REQUEST[nombre]','$REQUEST[mail]','$REQUEST[radios1])") or
    die("Problemas en el select".mysqli_error($conexion));

echo "El alumno fue dado de alta.";
?>
</body>
</html>
```

27 - Listado (selección de registros de varias tablas - INNER JOIN)

Ahora veremos como imprimir todos los alumnos inscriptos a los cursos junto al nombre del curso donde está inscripto. Los datos se encuentran en las tablas "alumnos" y "cursos".

Debemos aparear el código de curso de la tabla "alumnos" con el código de la tabla "cursos".

En caso de no recordar los conceptos de relaciones de tablas en SQL pueden repasar los mismos en el tutorial de [MySQL Ya](#).

El código del programa que hace esto es el siguiente:

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");
```

```

$registros=mysqli_query($conexion,"select alu.codigo as codigo,
                                nombre,
                                mail,
                                codigocurso,
                                nombrecurso
                                from alumnos as alu
                                inner join cursos as cur on
cur.codigo=alu.codigocurso") or
    die("Problemas en el select:".mysqli_error($conexion));

while ($reg=mysqli_fetch_array($registros))
{
    echo "Codigo:".$reg['codigo']."<br>";
    echo "Nombre:".$reg['nombre']."<br>";
    echo "Mail:".$reg['mail']."<br>";
    echo "Curso:".$reg['nombrecurso']."<br>";
    echo "<hr>";
}
mysqli_close($conexion);
?>
</body>
</html>

```

Hay varias cosas nuevas cuya sintaxis necesitamos analizar, la primera es como hacer el apareo con la tabla cursos:

```
inner join cursos as cur on cur.codigo=alu.codigocurso
```

Luego de las palabras claves inner join, indicamos la tabla que necesitamos aparear, podemos crear un alias de una tabla mediante la palabra clave as. En el resto de la consulta, en vez de indicar el nombre de la tabla, hacemos referencia al alias (generalmente un nombre más corto).

Seguidamente de la palabra clave on, indicamos los campos por los que apareamos las tablas, en nuestro caso el codigo de la tabla cursos con el codigocurso de la tabla alumnos.

Otro punto a tener en cuenta es indicar en el select que campos debemos rescatar de las tablas, es decir, indicarle a que tabla pertenece en el caso que tengan el mismo nombre:

```

$registros=mysqli_query($conexion,"select alu.codigo as codigo,
                                nombre,
                                mail,
                                codigocurso,
                                nombrecurso
                                from alumnos as alu
                                inner join cursos as cur on
cur.codigo=alu.codigocurso") or
    die("Problemas en el select:".mysqli_error($conexion));

```

En este caso rescatamos el código del alumno (y no el código de curso). Debemos crear un alias si dos tablas tienen el mismo nombre de campo para evitar confusión;

como ocurre aquí con el campo codigo de las tablas alumnos y cursos. También creamos un alias para la tabla alumnos.

Cuando rescatamos los datos y los mostramos en la página, hacemos referencia al alias del campo:

```
while ($reg=mysqli_fetch_array($registros))
{
    echo "Codigo:". $reg['codigo']. "<br>";
    echo "Nombre:". $reg['nombre']. "<br>";
    echo "Mail:". $reg['mail']. "<br>";
    echo "Curso:". $reg['nombrecurso']. "<br>";
    echo "<hr>";
}
```

Problema propuesto.

Confeccionar un programa que permita ingresar el código de un alumno y nos muestre su nombre, mail y nombre del curso en el cual está inscripto. Hacer un formulario donde se ingrese el código de alumno y otra página donde se muestren los datos respectivos. Mostrar un mensaje si no existe el código de alumno ingresado.

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<form action="pagina27_1_resposta.php" method="post">
Ingrese el codigo del alumno:
<input type="text" name="codigo">
<br>
<input type="submit" value="buscar">
</form>
</body>
</html>
```

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select nombre,
                                mail,
                                nombrecurso
                                from alumnos as alu
```



```

                                inner join cursos as cur on
cur.codigo=alu.codigocurso
                                where alu.codigo=$_REQUEST[codigo]) or
                                die("Problemas en el select:".mysqli_error($conexion));

if ($reg=mysqli_fetch_array($registros))
{
    echo "Nombre:". $reg['nombre']."<br>";
    echo "Mail:". $reg['mail']."<br>";
    echo "Curso:". $reg['nombrecurso']."<br>";
}
else
{
    echo "No existe un alumno con ese código.";
}
mysqli_close($conexion);
?>
</body>
</html>

```

28 - Función count de SQL.

Hay en MySQL funciones que nos permiten contar registros, calcular sumas, promedios, obtener valores máximos y mínimos. Veamos la función count.

Cuando necesitamos saber la cantidad de registros utilizamos la función count, por ejemplo si deseamos conocer la cantidad de alumnos que hay en la tabla "alumnos" debemos implementar el siguiente comando SQL:

```
select count(*) as cantidad from alumnos
```

Implementamos una página que muestra este resultado:

```

<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select count(*) as cantidad from alumnos")
or
    die("Problemas en el select:".mysqli_error($conexion));
$reg=mysqli_fetch_array($registros);
echo "La cantidad de alumnos inscritos son :". $reg['cantidad'];
?>
</body>
</html>

```

En la sentencia select en vez de indicar los campos de la tabla, colocamos la llamada a la función count pasando como parámetro un asterisco y creando un alias para su posterior recuperación e impresión del dato:

```
$registros=mysqli_query($conexion,"select count(*) as cantidad from alumnos")
or
die("Problemas en el select:".mysqli_error($conexion));
```

El select no tiene cláusula where ya que debemos contar todos los alumnos y no los de algún curso en particular.

La llamada a la función mysqli_fetch_array se hace sin estructura condicional o repetitiva ya que sabemos que nos retornará un único registro (en realidad, un registro que tiene una sola columna llamada cantidad, en caso de estar vacía la tabla alumnos, se almacena cero en el alias cantidad):

```
$reg=mysqli_fetch_array($registros);
```

Problema propuesto.

Confeccionar un programa que muestre por pantalla los nombres de todos los cursos y al final la cantidad total de cursos.

```
<html>
<head>
<title>Problema</title>
</head>
<body>

<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select nombrecurso from cursos") or
    die("Problemas en el select:".mysqli_error($conexion));

while ($reg=mysqli_fetch_array($registros))
{
    echo "Nombre del curso:".$reg['nombrecurso']."<br>";
    echo "<hr>";
}

$registros=mysqli_query($conexion,"select count(*) as cantidad from cursos") or
    die("Problemas en el select:".mysqli_error($conexion));

$reg=mysqli_fetch_array($registros);
echo "La cantidad de cursos son :".$reg['cantidad'];

mysqli_close($conexion);
?>

</body>
</html>
```

29 - UPDATE (Modificación de un registro trabajando con dos tablas)

Ahora complicaremos un poco la modificación de un registro consultando dos tablas. Supongamos que un alumno desea cambiarse de curso, es decir, se inscribió en uno y quiere borrarse de ese e inscribirse en otro diferente. Debemos mostrar en un "select" el curso actual en el que está inscripto y los otros cursos disponibles en la tabla "cursos".

Para resolver este problema tenemos que plantear tres páginas, una donde ingresemos el mail del alumno, la segunda donde se pueda cambiar el curso y por última una que permita modificar la tabla "alumnos" con el nuevo curso seleccionado.

La primer página de ingreso del mail es:

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<form action="pagina2.php" method="post">
Ingrese el mail del alumno:
<input type="text" name="mail"><br>
<input type="submit" value="buscar">
</form>
</body>
</html>
```

La segunda página y la más importante en cuanto a novedades es la siguiente:

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select * from alumnos
                                where mail='$_REQUEST[mail]'") or
    die("Problemas en el select:".mysqli_error($conexion));
if ($regalu=mysqli_fetch_array($registros))
{
    ?>
    <form action="pagina3.php" method="post">
    <input type="hidden" name="mailviejo" value="<?php echo $regalu['mail']
?>">
    <select name="codigocurso">
    <?php
```

```

$registros=mysqli_query($conexion,"select * from cursos") or
die("Problemas en el select:".mysqli_error($conexion));
while ($reg=mysqli_fetch_array($registros))
{
    if ($regalu['codigocurso']==$reg['codigo'])
        echo "<option value=\"\$reg[codigo]\\"
selected>$reg[nombrecurso]</option>";
    else
        echo "<option value=\"\$reg[codigo]\\">$reg[nombrecurso]</option>";
    }
?>
</select>
<br>
<input type="submit" value="Modificar">
</form>
<?php
}
else
    echo "No existe alumno con dicho mail";
?>
</body>
</html>

```

La primera consulta de la tabla alumnos es para verificar si existe un alumno con el mail ingresado por teclado:

```

$registros=mysqli_query($conexion,"select * from alumnos
                                where mail='$_REQUEST[mail]'" ) or
die("Problemas en el select:".mysqli_error($conexion));
if ($regalu=mysqli_fetch_array($registros))
{

```

En caso de existir un alumno con dicho mail, el if se verifica verdadero y pasamos a poblar el control select con los distintos cursos que contiene la tabla "cursos":

```

$registros=mysqli_query($conexion,"select * from cursos") or
die("Problemas en el select:".mysqli_error($conexion));
while ($reg=mysqli_fetch_array($registros))
{
    if ($regalu['codigocurso']==$reg['codigo'])
        echo "<option value=\"\$reg[codigo]\\"
selected>$reg[nombrecurso]</option>";
    else
        echo "<option value=\"\$reg[codigo]\\">$reg[nombrecurso]</option>";
    }
?>
</select>

```

Para que aparezca seleccionado el curso actual debemos inicializar lo marca option con el texto selected. Es decir que el if dentro del while se verifica verdadero una solo vez.

Por último la tercer página procede a registrar el cambio en la tabla "alumnos":

```

<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"update alumnos
                                set codigocurso=$_REQUEST[codigocurso]
                                where mail='$_REQUEST[mailviejo]'" ) or
    die("Problemas en el select:".mysqli_error($conexion));
    echo "El curso fue modificado con éxito";
?>
</body>
</html>

```

Problema propuesto.

Confeccionar la modificación del mail, nombre y curso de la tabla "alumnos". Ingresar por teclado el código de alumno para su búsqueda

```

<html>
<head>
<title>Problema</title>
</head>
<body>
<form action="pagina29_1_procesa.php" method="post">
Ingrese el codigo del alumno:
<input type="text" name="codigo"><br>
<input type="submit" value="buscar">
</form>
</body>
</html>

```

```

<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select * from alumnos
                                where codigo=$_REQUEST[codigo]" ) or
    die("Problemas en el select:".mysqli_error($conexion));

```

```

if ($regalu=mysqli_fetch_array($registros))
{
?>
    <form action="pagina29_1_resultado.php" method="post">
    Nombre: <input type="text" name="nombre" value="<?php echo $regalu['nombre'] ?>"><br>
    Mail: <input type="text" name="mail" value="<?php echo $regalu['mail'] ?>"><br>
    <input type="hidden" name="codigo" value="<?php echo $regalu['codigo'] ?>">
    <select name="codigocurso">
    <?php
    $registros=mysqli_query($conexion,"select * from cursos") or
        die("Problemas en el select:".mysql_error());
    while ($reg=mysqli_fetch_array($registros))
    {
        if ($regalu['codigocurso']==$reg['codigo'])
            echo "<option value=\"\$reg[codigo]\" selected>$reg[nombrecurso]</option>";
        else
            echo "<option value=\"\$reg[codigo]\">$reg[nombrecurso]</option>";
    }
?>
    </select>
    <br>
    <input type="submit" value="Modificar">
    </form>
<?php
}
else
    echo "No existe alumno con dicho mail";
?>
</body>
</html>

```

```

<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"update alumnos
                                set codigocurso=$_REQUEST[codigocurso],
                                nombre='$_REQUEST[nombre]',
                                mail='$_REQUEST[mail]'
                                where codigo=$_REQUEST[codigo]") or
    die("Problemas en el select:".mysql_error($conexion));
echo "El curso fue modificado con éxito";
?>
</body>
</html>

```


30 - Cláusula Group By de SQL.

Entre las muchas posibilidades que nos brinda SQL, una es agrupar registros y obtener información resumida de tablas.

En nuestro problema, un listado interesante sería mostrar la cantidad de alumnos inscriptos por curso. Para resolver de una manera sencilla esta situación, SQL nos permite agrupar los registros de la tabla "alumnos" por la columna "codigocurso" y contar la cantidad de registros que hay por cada código de curso igual.

El programa que nos permite resolver este problema es el siguiente:

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select count(alu.codigo) as cantidad,
                                nombrecurso
                                from alumnos as alu
                                inner join cursos as cur on
                                cur.codigo=alu.codigocurso
                                group by alu.codigocurso") or
    die("Problemas en el select:".mysqli_error($conexion));
while ($reg=mysqli_fetch_array($registros))
{
    echo "Nombre del curso:".$reg['nombrecurso']."<br>";
    echo "Cantidad de inscriptos:".$reg['cantidad']."<br>";
    echo "<hr>";
}
mysqli_close($conexion);
?>
</body>
</html>
```

Hay varias partes importantes en este código; primero, en el select indicamos que cuente la cantidad de registros de la tabla "alumnos":

```
select count(alu.codigo) as cantidad,
        nombrecurso
from alumnos as alu
```

Pero, como veremos más adelante, en 'cantidad' no se almacena la cantidad total de registros de la tabla "alumnos" debido a que más adelante empleamos la cláusula group by.

Como necesitamos rescatar el nombre del curso hacemos el apareo con la tabla "cursos":

```
inner join cursos as cur on cur.codigo=alu.codigocurso
```

Por último en la sentencia select de SQL disponemos la cláusula group by:

```
group by alu.codigocurso"
```

Con esta cláusula se hace un corte de control por cada grupo de registros que tienen el mismo código de curso almacenado.

Luego mostramos el nombre de curso con la cantidad de inscriptos para dicho curso:

```
while ($reg=mysqli_fetch_array($registros))
{
    echo "Nombre del curso:".$reg['nombrecurso']."<br>";
    echo "Cantidad de inscriptos:".$reg['cantidad']."<br>";
    echo "<hr>";
}
```

Problema propuesto.

Confeccionar un programa que muestre el nombre del curso, la cantidad de inscriptos y todos los inscriptos a dicho curso. Repetir esto con todos los cursos.

Es decir, en la página debe aparecer algo parecido a:

Nombre del curso:PHP

Cantidad de inscriptos:3

Nombres: Martinez Luis - Rodriguez Pablo - Gonzalez Ana

Nombre del curso:JSP

Cantidad de inscriptos:2

Nombres: Hernandez Hector - Roca Marta

31 - Parámetros en un hipervínculo.

Hasta ahora hemos visto como enviar datos de una página a otra mediante formularios. Otra forma muy utilizada que complementa la anterior es como parámetro en un hipervínculo.

Confeccionaremos una página que muestre tres hipervínculos, cada uno tiene por objetivo mostrar en una página las tablas de multiplicar del 2, del 3 o la del 5.

La primer página es un archivo HTML puro, ya que sólo disponemos las marcas de hipervínculos:

```
<html>
<head>
<title>Problema</title>
</head>
<body>
```

```
<A href="pagina2.php?tabla=2">Tabla del 2</A> <br>
<A href="pagina2.php?tabla=3">Tabla del 3</A> <br>
<A href="pagina2.php?tabla=5">Tabla del 5</A>
</body>
</html>
```

La sintaxis para pasar parámetros en un hipervínculo es:

```
<A href="pagina2.php?tabla=2">Tabla del 2</A> <br>
```

Es decir, luego del caracter "?" indicamos el nombre del parámetro y seguidamente el valor del mismo.

La página que rescata el valor pasado como parámetro es la siguiente:

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
echo "Listado de la tabla del $_REQUEST[tabela] <br>";
for($f=1;$f<=10;$f++)
{
$valores=$f*$_REQUEST['tabela'];
echo $valores."-";
}
?>
</body>
</html>
```

Es decir que con el mismo vector asociativo \$_REQUEST recuperamos los datos enviados por parámetro en la llamada a la página.

Problema propuesto.

Confeccionar un programa que muestre una página con todos los nombres de la tabla "cursos". Los nombres deben aparecer como hipervínculos a otra página que deberá mostrar todos los alumnos inscriptos a dicho curso. Como parámetro en el hipervínculo, pasar el código de curso.

```
<html>
<head>
<title>Problema</title>
</head>
<body>

<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select codigo,nombrecurso from cursos") or
    die("Problemas en el select:".mysqli_error($conexion));
```

```

while ($reg=mysqli_fetch_array($registros))
{
    echo "Codigo: ".$reg['codigo']."<br>";
    echo "Nombre del curso:<a
href=\"pagina31_1_resposta.php?codigo=$reg[codigo]\">".$reg['nombrecurso']."</a><br>";
    echo "<hr>";
}

mysqli_close($conexion);
?>

</body>
</html>

```

```

<html>
<head>
<title>Problema</title>
</head>
<body>

<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select nombre from alumnos where
codigocurso=$_REQUEST[codigo]") or
    die("Problemas en el select:".mysqli_error($conexion));

echo "Alumnos inscritos a dicho curso<br>";
while ($reg=mysqli_fetch_array($registros))
{
    echo $reg['nombre']." - ";
}

mysqli_close($conexion);
?>

</body>
</html>

```

32 - Paginación de registros.

En situaciones en las cuales una consulta retorna muy muchos datos, en vez de enviarlos todos al navegador, se puede enviar un conjunto limitado de registros. Luego, mediante hipervínculos, ver el resto de datos. Por ejemplo, cuando hacemos búsquedas con el servidor google, generalmente no nos retorna todas las direcciones donde se encuentran los resultados buscados, nos retorna páginas con 10 enlaces por página (pensemos el tiempo de transferencia si nos retornara 1.000.000 de enlaces).

Bueno, ahora resolvamos con el lenguaje PHP este problema de paginación, mostraremos dos alumnos por página:

```
<?php
if (isset($_REQUEST['pos']))
    $inicio=$_REQUEST['pos'];
else
    $inicio=0;
?>
<html>
<head>
<title>Problema</title>
</head>
<body>

<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select alu.codigo as
                                codigo,
                                nombre,
                                mail,
                                codigocurso,
                                nombrecurso
                                from alumnos as alu
                                inner join cursos as cur on
                                cur.codigo=alu.codigocurso
                                limit $inicio,2") or
    die("Problemas en el select:".mysqli_error($conexion));
$impresos=0;
while ($reg=mysqli_fetch_array($registros))
{
    $impresos++;
    echo "Codigo:".$reg['codigo']."<br>";
    echo "Nombre:".$reg['nombre']."<br>";
    echo "Mail:".$reg['mail']."<br>";
    echo "Curso:".$reg['nombrecurso']."<br>";
    echo "<hr>";
}
if ($inicio==0)
    echo "anteriores ";
else
{
    $anterior=$inicio-2;
    echo "<a href=\"pagina1.php?pos=$anterior\">Anteriores </a>";
}
if ($impresos==2)
{
    $proximo=$inicio+2;
    echo "<a href=\"pagina1.php?pos=$proximo\">Siguietes</a>";
}
else
    echo "siguietes";
mysqli_close($conexion);
```

```
?>

</body>
</html>
```

La función `isset` retorna verdadero si existe la variable que le pasamos como parámetro, en este caso le estamos pasando la componente `pos` del vector asociativo `$_REQUEST`. Cuando llamamos por primera vez a esta página, lo hacemos : `pagina1.php` sin parámetros, por lo que el `if` se verifica como falso. Es decir la variable `$inicio` se carga con el valor 0.

Otro concepto importante es la cláusula `limit` que es propia del gestor MySQL. Mediante esta cláusula limitamos la cantidad de registros que retorna el `select`. El primer valor del `limit` indica a partir de cual registro y el segundo la cantidad de registros. Es decir si un `select` sin `limit` retorna 100 registro, luego utilizando por ejemplo la sintaxis `limit 50,25` nos retornará, de esa lista de 100 registros, a partir del registro de la posición 50, 25 registros.

En nuestro problema indicamos que retorne desde valor que tenga la variable `$inicio` y como cantidad 2 (páginas con 2 registros):

```
$registros=mysqli_query($conexion,"select alu.codigo as
                                codigo,
                                nombre,
                                mail,
                                codigocurso,
                                nombrecurso
                                from alumnos as alu
                                inner join cursos as cur on
                                cur.codigo=alu.codigocurso
                                limit $inicio,2") or
die("Problemas en el select:".mysqli_error($conexion));
```

Seguidamente mostramos todos los registros retornados y además los contamos:

```
$impresos=0;
while ($reg=mysqli_fetch_array($registros))
{
    $impresos++;
    echo "Codigo:".$reg['codigo']."<br>";
    echo "Nombre:".$reg['nombre']."<br>";
    echo "Mail:".$reg['mail']."<br>";
    echo "Curso:".$reg['nombrecurso']."<br>";
    echo "<hr>";
}
```

Ahora vemos dónde dispondremos los hipervínculos, hacia adelante o atrás:

```
if ($inicio==0)
    echo "anteriores ";
else
{
    $anterior=$inicio-2;
    echo "<a href=\"pagina1.php?pos=$anterior\">Anteriores </a>";
}
```

```
}
```

Si la variable \$inicio vale 0 significa que no hay registros antes de éste, por lo que sólo mostramos un texto "anteriores". En caso que la variable \$inicio sea distinta de 0, se ejecuta el else, donde disponemos un hipervínculo con la misma página e inicializando el parámetro pos con el valor de \$inicio menos 2.

Si el contador \$impresos tiene el valor 2 significa que posiblemente hay más registros por mostrar y debemos disponer un hipervínculo con la misma página pero inicializando el parámetro pos con el valor de \$inicio más 2:

```
if ($impresos==2)
{
    $proximo=$inicio+2;
    echo "<a href=\"pagina1.php?pos=$proximo\">Siguientes</a>";
}
else
    echo "siguientes";
```

Problema propuesto.

Confeccionar un programa que muestre los registros de la tabla "cursos" con páginas de 3 registros.

```
<?php
if (isset($_REQUEST['pos']))
    $inicio=$_REQUEST['pos'];
else
    $inicio=0;
?>
<html>
<head>
<title>Problema</title>
</head>
<body>

<?php
$conexion=mysqli_connect("localhost","root","","base1") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select codigo,
                                nombrecurso from cursos
                                limit $inicio,3") or
    die("Problemas en el select:".mysqli_error($conexion));
$impresos=0;
while ($reg=mysqli_fetch_array($registros))
{
    $impresos++;
    echo "Codigo:".$reg['codigo']."<br>";
    echo "Nombre:".$reg['nombrecurso']."<br>";
    echo "<hr>";
}
mysqli_close($conexion);
if ($inicio==0)
```

```
    echo "anteriores ";
else
{
    $anterior=$inicio-3;
    echo "<a href=\"pagina32_1.php?pos=$anterior\">Anteriores </a>";
}
if ($impresos==3)
{
    $proximo=$inicio+3;
    echo "<a href=\"pagina32_1.php?pos=$proximo\">Siguietes</a>";
}
else
    echo "siguietes";
?>

</body>
</html>
```