

Ejercicio 1:

Para comenzar la comunicación con un servidor de base de datos MySQL, es necesario abrir una conexión a ese servidor. Para inicializar esta conexión, PHP ofrece la función **mysqli_connect**: Esta función nos permite crear una conexión con una base de datos concreta.

```
$nombreConexion = mysqli_connect($hostname , $nombreUsuario , $contraseña);
```

Todos sus parámetros son opcionales, pero hay tres de ellos que generalmente son necesarios:

\$hostname: Es un nombre de servidor (dato que debemos consultar en nuestra cuenta de hosting).

\$nombreUsuario: Es el nombre de usuario de base de datos, nombre que habremos especificado al crear la base de datos (si no lo hemos hecho, puede ser root).

\$contraseña: Es la contraseña de acceso para el usuario de base de datos, contraseña que habremos especificado al crear la base de datos.

Una vez abierta la conexión, se debe seleccionar una base de datos para su uso, mediante la función

mysqli_select_db: Esta función nos permite seleccionar una base de datos concreta dentro de todas las existentes en el servidor al que nos conectemos.

```
mysqli_select_db ($nombreConexion, $nombreBaseDatos);
```

Esta función debe pasar como parámetro

\$nombreBaseDatos: Nombre de la base de datos.

\$nombreConexion: Conexion devuelta por **mysqli_connect**.

La función **mysqli_query()** se utiliza para

Ejecutar una consulta a la base de datos que especifiquemos

```
$resultado = mysqli_query($nombreConexion, $sql_query);
```

y requiere como parámetros

\$nombreConexion: Conexion devuelta por **mysqli_connect**.

\$sql_query: La string de la consulta.

La cláusula *or die()* se utiliza para

Mostrar un mensaje en caso de no poder concretar la conexión y finalizar la ejecución del script. *die* es equivalente a utilizar *exit*.

Por ejemplo, en la siguiente sentencia:

\$link = mysqli_connect("localhost", "root") or die ("Problemas de conexión a la base de datos");

Si el primer argumento (*mysqli_connect("localhost", "root")*) retorna *false*, se ejecuta la función *die*, mostrando el mensaje correspondiente y finalizando la ejecución del script.

Si el primer argumento retorna *true*, se asume que toda la sentencia va a ser *true*, entonces el *or* no sigue su curso y no ejecuta *die*.

y la función *mysqli_error()* se puede usar para

Devolver el último mensaje de error para la llamada más reciente a una función de *MySQLi* que puede haberse ejecutado correctamente o haber fallado.

Si la función *mysqli_query()* es exitosa, el conjunto resultante retornado se almacena en una variable, por ejemplo ***\$vResultado***, y a continuación se puede ejecutar el siguiente código (explicarlo):

```
<?php
while ($fila = mysqli_fetch_array($vResultado))
{
?>
<tr>
    <td><?php echo ($fila[0]); ?></td>
    <td><?php echo ($fila[1]); ?></td>
    <td><?php echo ($fila[2]); ?></td>
</tr>
<tr>
    <td colspan="5">
<?php
}
mysqli_free_result($vResultado);

mysqli_close($link);
?>
```

- El resultado se almacena en ***\$vResultado***.
- Luego se ejecuta ***\$fila = mysqli_fetch_array(\$vResultado)*** con lo cual ***\$fila*** toma un valor que puede ser un array resultado o *NULL*. Por ende, entra al bucle mientras no se hayan extraído todas las filas del resultado. Como ***mysqli_fetch_array*** carece del segundo parámetro opcional (El cual indica el tipo de array con el que debe retornar el resultado) se retorna de manera que pueda referenciarse mediante índices o mediante el nombre de la columna.

- Por cada fila, muestra el valor para la columna de índice 0, la columna de índice 1 y la columna de índice 2.
- **<td colspan="5">** genera un espacio entre cada **\$fila**.
- La función **mysqli_free_result** libera la memoria en la que se almaceno la información de los registros.
- La función **mysqli_close** cierra la conexión con la base de datos.