



Departamento de Lenguajes y
Sistemas Informáticos



Universitat d'Alacant
Universidad de Alicante

mysql

Programación en Internet
Curso 2008-2009

Programación en Internet – Curso 2008-2009

Índice

- ¿Qué es?
- Interfaz procedimental
- Interfaz orientado a objetos
- Sentencias preparadas

¿Qué es?

- mysqli es una extensión de PHP que permite acceder a ciertas funciones disponibles a partir de MySQL 4.1 que no se pueden emplear con la extensión tradicional
- Proporciona:
 - Mayor velocidad y seguridad
 - Interfaz procedimental u orientado a objetos
 - Soporte de transacciones
 - Nuevo protocolo binario de MySQL 4.1 que permite ciertas funciones como la ejecución de sentencias preparadas

Interfaz procedimental

- Principales funciones:
 - `mysqli_connect(servidor, usuario, contraseña, baseDatos)`: devuelve un identificador de recurso o `false` si la conexión falla
 - `mysqli_connect_error()`: devuelve una descripción del último error de conexión
 - `mysqli_query(identificador, sentencia)`: ejecuta una sentencia SQL
 - `mysqli_error(identificador)`: devuelve una descripción del último error
 - `mysqli_fetch_assoc(resultado)`: devuelve la siguiente fila en el resultado o `false` en caso contrario
 - `mysqli_free_result(resultado)`: libera la memoria empleada por un resultado
 - `mysqli_close(identificador)`: cierra la conexión con una base de datos

Interfaz procedimental

```
<?php
// Conecta con el servidor de MySQL
$link = @mysqli_connect(
    'localhost',    // El servidor
    'wwwwdata',    // El usuario
    '',             // La contraseña
    'biblioteca'); // La base de datos

if(!$link) {
    echo '<p>Error al conectar con la base de datos: '
    . mysqli_connect_error();
    echo '</p>';
    exit;
}
```

Interfaz procedimental

```
// Ejecuta una sentencia SQL
$sentencia = 'SELECT * FROM libros';
if(!($resultado = @mysqli_query($link, $sentencia)))
{
    echo "<p>Error al ejecutar la sentencia
    <b>$sentencia</b>: " . mysqli_error($link);
    echo '</p>';
    exit;
}
```

Interfaz procedimental

```
echo '<table><tr>';
echo '<th>IdLibro</th><th>Título</th><th>Resumen</th>';
echo
'<th>Autor</th><th>Categoría</th><th>Editorial</th><th>Anyo
</th></tr>';
// Recorre el resultado y lo muestra en forma de tabla HTML
while($fila = mysqli_fetch_assoc($resultado)) {
    echo '<tr>';
    echo '<td>' . $fila['IdLibro'] . '</td>';
    echo '<td>' . $fila['Titulo'] . '</td>';
    echo '<td>' . $fila['Resumen'] . '</td>';
    echo '<td>' . $fila['Autor'] . '</td>';
    echo '<td>' . $fila['Categoria'] . '</td>';
    echo '<td>' . $fila['Editorial'] . '</td>';
    echo '<td>' . $fila['Anyo'] . '</td>';
    echo '</tr>';
}
echo '</table>';
```

Interfaz procedimental

```
// Libera la memoria ocupada por el resultado
mysqli_free_result($resultado);
// Cierra la conexión
mysqli_close($link);
?>
```

Interfaz orientado a objetos

- Mismas funciones que el interfaz procedimental, pero son métodos de objetos
- Clases:
 - `mysqli`: representa una conexión entre PHP y una base de datos en MySQL
 - `mysqli_stmt`: representa una sentencia preparada
 - `mysqli_result`: representa el resultado de una consulta

Interfaz orientado a objetos

- `mysqli->métodos`:
 - `connect(servidor, usuario, contraseña, baseDatos)`: conecta con una base de datos
 - `query(sentencia)`: ejecuta una sentencia SQL, devuelve un objeto de tipo `mysqli_result`
 - `prepare(sentencia)`: crea una sentencia preparada, devuelve un objeto de tipo `mysqli_stmt`
 - `close()`: cierra la conexión con una base de datos
- `mysqli->propiedades`:
 - `affected_rows`: devuelve el número de filas afectadas por la última operación
 - `error`: devuelve una descripción del último error
 - `field_count`: devuelve el número de columnas de la última consulta

Interfaz orientado a objetos

- `mysqli_result->`métodos:
 - `fetch_assoc()`: devuelve la siguiente fila en el resultado o `false` en caso contrario
 - `free_result()`: libera la memoria empleada por un resultado
 - `close()`: cierra el resultado
- `mysqli_result->`propiedades:
 - `field_count`: devuelve el número de columnas de la última consulta
 - `num_rows`: devuelve el número de filas en el resultado

Interfaz orientado a objetos

```
<?php
// Conecta con el servidor de MySQL
$mysqli = @new mysqli(
    'localhost',      // El servidor
    'wwwwdata',      // El usuario
    '',               // La contraseña
    'biblioteca');   // La base de datos

if(mysqli_connect_errno()) {
    echo '<p>Error al conectar con la base de datos: ' .
        mysqli_connect_error();
    echo '</p>';
    exit;
}
```

Interfaz orientado a objetos

```
// Ejecuta una sentencia SQL
$sentencia = 'SELECT * FROM libros';
if(!($resultado = $mysqli->query($sentencia)) {
    echo "<p>Error al ejecutar la sentencia
<b>$sentencia</b>: " . $mysqli->error;
    echo '</p>';
    exit;
}
```

Interfaz orientado a objetos

```
echo '<table><tr>';
echo '<th>IdLibro</th><th>Título</th><th>Resumen</th>';
echo
'<th>Autor</th><th>Categoría</th><th>Editorial</th><th>Anyo
</th></tr>';
// Recorre el resultado y lo muestra en forma de tabla HTML
while($fila = $resultado->fetch_assoc()) {
    echo '<tr>';
    echo '<td>' . $fila['IdLibro'] . '</td>';
    echo '<td>' . $fila['Titulo'] . '</td>';
    echo '<td>' . $fila['Resumen'] . '</td>';
    echo '<td>' . $fila['Autor'] . '</td>';
    echo '<td>' . $fila['Categoría'] . '</td>';
    echo '<td>' . $fila['Editorial'] . '</td>';
    echo '<td>' . $fila['Anyo'] . '</td>';
    echo '</tr>';
}
echo '</table>';
```

Interfaz orientado a objetos

```
// Libera la memoria ocupada por el resultado
$resultado->close();
// Cierra la conexión
$mysqli->close();
?>
```

Sentencias preparadas

- Mecanismo que permite optimizar la ejecución de una sentencia SQL cuando se repite varias veces
 - Realizar una búsqueda pero con distintos valores

```
select nombre,edad from cliente where edad >= 18 and edad <= 25
select nombre,edad from cliente where edad >= 20 and edad <= 30
```
 - Insertar varias filas similares

```
insert into Personas values ("Juan", "López")
insert into Personas values ("Pedro", "Martínez")
```


Sentencias preparadas

- Las sentencias preparadas están optimizadas para ser ejecutadas muchas veces
- Previamente a su ejecución, su código es optimizado para tardar menos al ejecutar
 - Este proceso también consume un tiempo que puede penalizar la ejecución si solo lo utilizamos para ejecutar pocas sentencias

Sentencias preparadas

- Los valores no definidos se indican con el símbolo interrogación (?)
- Antes de ejecutar la sentencia, se asignan los valores con el método `bind_param()`

Sentencias preparadas

- `mysqli_stmt`->métodos:
 - `bind_param()`: enlaza variables como parámetros de una sentencia preparada
 - `bind_result()`: enlaza variables para almacenar el resultado de una sentencia preparada
 - `execute()`: ejecuta una sentencia preparada
 - `close()`: cierra una sentencia preparada
- `mysqli_stmt`->propiedades:
 - `field_count`: devuelve el número de columnas de la última consulta
 - `num_rows`: devuelve el número de filas en el resultado
 - `param_count`: devuelve el número de parámetros de la sentencia preparada

Sentencias preparadas

```
<?php
// Abre conexión con la base de datos
// ...
$stmt = $mysqli->prepare("INSERT INTO Productos VALUES (?,
?, ?, ?)");
// i: entero, s: otros tipos, d: double o float
$stmt->bind_param('issd', $codigo, $nombre, $descripcion,
$precio);
$codigo = 123;
$nombre = 'Mk1 20';
$descripcion = 'Televisor Mk1 20 de pantalla plana';
$precio = 1100.50;
$stmt->execute();
$stmt->close();
// Cerrar la conexión
// ...
?>
```