



Tema 4: Técnicas de acceso a datos en PHP

- Utilización de bases de datos relacionales.
- Establecimiento de conexiones.
- Ejecución de sentencias SQL.
- Recuperación y edición de información.
- Utilización de conjuntos de resultados.
- Visualización de la información en páginas Web.
- Mecanismos de edición de la información en un cliente Web.
- Actualizaciones y eliminaciones.
- Transacciones.
- Utilización de otros orígenes de datos.



Tema 4: Técnicas de acceso a datos en PHP



Trabajar con bases de datos en PHP:

SGBD

+

Estructuras de datos

+

Datos

MySQL
MariaDB

Driver nativo de MySQL **mysqlnd**

Extensión **MySQLi**
(para MySQL)

Extensión **PDO** de PHP
(Para cualquier motor)

Código **PHP**

HTML



Tema 4: Técnicas de acceso a datos en PHP



Motores de almacenamiento:

MyISAM Rápido - no controla la integridad referencial

InnoDB Controla la integridad referencial

XtraDB Sustituye a InnoDB en MariaDB



Tema 4: Técnicas de acceso a datos en PHP



Instalación y configuración de MySQL o MariaDB:

mysql-server	mariadb-server	paquete del servicio
mysql-client	paquete del cliente	
/etc/mysql/my.cnf	fichero de configuración	my.ini (windows)
mysqld	mysql	proceso - servicio
TCP 3306	puerto TCP por defecto	



Tema 4: Técnicas de acceso a datos en PHP



Herramientas administrativas:

En linea de comandos:

mysql		-u nombre-usuario
mysqladmin		-pcontraseña
mysqlshow		-h equipo-servidor

phpMyAdmin

Administración web (Servicio web para administrar DB)

My SQL Workbench

DbVisualizer

Herramienta que nos permite administrar el servidor y sus DB



Tema 4: Técnicas de acceso a datos en PHP



Operaciones sobre la base de datos:

Establecer conexión

Ejecutar sentencias SQL:

SELECT	consulta
INSERT	actualización - añadir
UPDATE	actualización - cambiar
DELETE	actualización - borrar

Obtener registros afectados o devueltos por la sentencia SQL

Emplear transacciones

Ejecutar procedimientos almacenados

Gestionar los errores que se produzcan durante la conexión o en las operaciones sobre la base de datos



Tema 4: Técnicas de acceso a datos en PHP



Extensión MySQLi:

MySQL Improved Extension

SGBD: MySQL

Interfaz procedimental y orientada a objetos

Utilizando llamadas a funciones

(Programación estructurada)

Utilizando constructores y métodos POO

(Programación orientada a objetos)

Ambos métodos devuelven un objeto de la clase `mysqli`

`php.ini`: configuramos la extensión `mysqli` / `PDO`

mysqli



Tema 4: Técnicas de acceso a datos en PHP



Extensión PDO: Objetos de datos de PHP

SGBD: Cualquiera Interfaz procedimental

Utilizando constructores y métodos POO (Programación orientada a objetos)

Ambos métodos devuelven un objeto de la clase PDO

php.ini: configuramos la extensión PDO / mysqli

PDO



Tema 4: Técnicas de acceso a datos en PHP



Establecimiento de conexión / Cierre de conexión

MySQLi

`$miDB = new mysqli();` Instanciamos un objeto mysqli

`$miDB->connect('equipo', 'usuario', 'password', 'basededatos');`

Establecimiento de conexión

`$error = $miDB->connect_errno;` Para controlar los posibles errores de conexión

...

Uso de la conexión

...

`$miDB->close();` Cierre de la conexión

mysqli



Tema 4: Técnicas de acceso a datos en PHP



Ejecución de consultas

MySQLi

...

Uso de la conexión

...

```
$resultadoConsulta = $miDB->query('consultaSQL');
```

```
$resultadoConsulta = $miDB->query('consultaSQL', MYSQLI_USE_RESULT);
```

MYSQLI_USE_RESULT Recupera la información a medida que la usamos

MYSQLI_STORE_RESULT Recupera toda la información al ejecutar la consulta (defecto)

```
$miDB->affected_rows;
```

```
$resultadoConsulta->free();
```

 Libera el espacio ocupado tras la consulta

mysqli



Tema 4: Técnicas de acceso a datos en PHP



Transacciones

MySQLi

...

Uso de la conexión

...

`$miDB->autocommit(false);` Deshabilita el modo transaccional automático

`$resultadoConsulta1 = $miDB->query('consultaSQL1');`

`$resultadoConsulta2 = $miDB->query('consultaSQL2');`

`$miDB->commit();` Confirma los cambios y los consolida

`$miDB->rollback();` Revierte o deshace los cambios

mysqli



Tema 4: Técnicas de acceso a datos en PHP



Obtención y uso del conjunto de resultados

MySQLi

```
$resultadoConsulta = $miDB->query('consultaSQL');
```

```
$registroArray = $resultadoConsulta->fetch_array();
```

Obtiene un registro en el array (asociativo y numérico por defecto) y avanza el puntero por el conjunto de registros

```
$registroObjeto = $resultadoConsulta->fetch_object();
```

Obtiene un registro en el objeto (asociativo) y avanza el puntero por el conjunto de registros

Ambos métodos devuelven null cuando terminan de recorrer el conjunto de resultados.

mysqli



Tema 4: Técnicas de acceso a datos en PHP



Consultas preparadas (que no devuelven resultados)

MySQLi

`$consulta = $miDB->stmt_init();` Devuelve un objeto de la clase `mysqli_stmt`

`$consulta->prepare('ConsultaPreparada VALUES (?, ?, ...));`

Preparamos la consulta pendiente de dar valor a las ?

`$consulta->bind_param('cadenaformato[i,d,s,b], $var1, $var2,...);`

Sustituye las interrogaciones por el valor de las variables que pasamos en `bind_param`

`i` - entero `d` - real doble precisión `s` - cadena de texto `b` - BLOB - binario

`$consulta->execute();` Ejecuta la consulta

`$consulta->close();`

`$miDB->close();`

mysqli



Tema 4: Técnicas de acceso a datos en PHP



Consultas preparadas (que devuelven resultados) **MySQLi**

Asigna a variables los campos que se obtienen después de cada ejecución del método `fetch`

`$consulta = $miDB->stmt_init();` Devuelve un objeto de la clase **mysqli_stmt**

`$consulta->prepare('SELECT campo1, campo2,... FROM...');`

`$consulta->execute();` Ejecuta la consulta

`$consulta->bind_result($var1, $var2,...);`

`while ($consulta->fetch()) {Tratamiento de $var1, $var2,...}`

`$consulta->close();`

`$miDB->close();`

mysqli



Tema 4: Técnicas de acceso a datos en PHP



Establecimiento de conexión / Cierre de conexión

PDO

```
$miDB = new PDO('DSN', 'nombreusuario', 'password');
```

Instanciamos un objeto PDO y establecemos la conexión

Construcción de la cadena PDO: (ej. 'mysql:host=localhost; dbname=midb')

host - nombre o dirección IP del servidor

port - número de puerto en el que escucha el servidor

dbname - nombre de la base de datos

```
$miDB->getAttribute(...); Obtiene el valor de un atributo
```

```
$miDB->setAttribute(...); Establece el valor de un atributo
```

...

Uso de la conexión

...

```
unset($miDB);
```

Cierre de la conexión

PDO



Tema 4: Técnicas de acceso a datos en PHP



Ejecución de consultas

PDO

...

Establecer la conexión

...

```
$numRegistros = $miDB->exec('consultaSQLDeActualizacion');
```

```
$resultadoConsulta = $miDB->query('consultaSQLDeSeleccion');
```

PDOStatement clase a la que pertenece el objeto devuelto por el método query

```
unset($miDB);
```

 Libera el espacio ocupado por el objeto PDO

PDO



Tema 4: Técnicas de acceso a datos en PHP



Transacciones

PDO

...

Uso de la conexión

...

`$miDB->beginTransaction();` Deshabilita el modo autocommit

`$resultadoConsulta1 = $miDB->exec('consultaSQL1');`

`$resultadoConsulta2 = $miDB->exec('consultaSQL2');`

`$miDB->commit();` Confirma los cambios y los consolida

`$miDB->rollback();` Revierte o deshace los cambios

PDO



Tema 4: Técnicas de acceso a datos en PHP



Obtención y uso del conjunto de resultados

PDO

```
$resultadoConsulta = $miDB->query('consultaSQL');
```

```
$registroArray = $resultadoConsulta->fetch();
```

Obtiene un registro en el array (asociativo y numérico por defecto) y avanza el puntero por el conjunto de registros

```
$registroObjeto = $resultadoConsulta->fetch(PDO::FETCH_OBJ);
```

Obtiene un registro en el objeto (asociativo) y avanza el puntero por el conjunto de registros

Ambos métodos devuelven null cuando terminan de recorrer el conjunto de resultados.

PDO



Tema 4: Técnicas de acceso a datos en PHP



Consultas preparadas (que no devuelven resultados) PDO

```
$consulta = $miDB->prepare('ConsultaPreparada VALUES (?, ?, ...));
```

```
$consulta = $miDB->prepare('ConsultaPreparada VALUES (:param1, :param2, ...));
```

Devuelve un objeto de la clase **PDOStatement**

Preparamos la consulta pendiente de dar valor a las ?

También podemos preparar la consulta poniendo nombre a los campos precedidos de :

```
$consulta->bindParam(1, $var1);    $consulta->bindParam(:param1, $var1);
```

```
$consulta->bindParam(2, $var2);    $consulta->bindParam(:param2, $var2);
```

Sustituye las interrogaciones por el valor de las variables que pasamos en bind_param

```
$consulta->execute();            Ejecuta la consulta
```

```
$parametros = array (":param1"=>"nombrecampo1", ":param2"=>"nombrecampo2");
```

```
$consulta->execute($parametros);    Otra opción para pasar los parámetros PDO
```



Tema 4: Técnicas de acceso a datos en PHP



Errores y manejo de excepciones

Buscamos un comportamiento sólido y coherente en nuestra web

```
try { /Código susceptible de producir un error
...
throw new Exception ('MensajeDeError'); /Utilizamos la clase Exception
...
}
catch (Exception $miException) {
    echo'Error: ' . $miException->getMessage();
    echo'Código de error: ' . $miException->getCode();
}
```

Exception



Tema 4: Técnicas de acceso a datos en PHP



Errores y manejo de excepciones en PDO

La clase PDO nos permite definir la fórmula que se utilizará cuando se produzca un error en el atributo **PDO::ATTR_ERRMODE**:

PDO::ERRMODE_SILENT

no se hace nada cuando hay error

PDO::ERRMODE_WARNING

genera un error de tipo **E_WARNING**

PDO::ERRMODE_EXCEPTION

lanza una excepción en **PDOException**

```
$miDB->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

```
try { /Bloque de código que puede tener excepciones en el objeto PDO }
```

```
catch (PDOException $miExceptionPDO) {
```

```
    echo'Error: '.$miExceptionPDO->getMessage();
```

```
    echo'Código de error: '.$miExceptionPDO->getCode();
```

```
}
```

PDOException



Tema 4: Técnicas de acceso a datos en PHP



Funciones interesantes

`XMLReader` `XMLWriter` `SimpleXMLElement` // Clases para
trabajar con ficheros XML

Manipulación de XML

`fputcsv()` `fgetcsv()` `str_getcsv()` //Para trabajar con ficheros csv

`fopen()` `fget()` `fread()` `fputs()` `fwrite()` //Para trabajar con
ficheros

Filesystem Functions

¿Preguntas?



Mapa conceptual

