




UNIVERSIDAD DE SEVILLA  
Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

# PHP y Acceso a Datos

*Departamento de Lenguajes y Sistemas Informáticos  
Grupo de Ingeniería del Software  
Febrero de 2012*



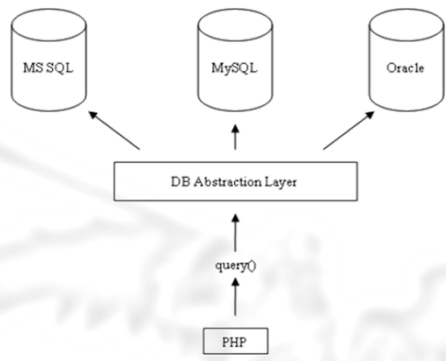
ETSII  
Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos



UNIVERSIDAD DE SEVILLA  
Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

## Introducción

- SQL es el lenguaje común para trabajar con todas las BD.
- Las interfaces que los programadores usan para trabajar con BD varían mucho. ➡ Dependencia de las aplicaciones respecto de la BD
- Para resolver estos problemas se necesitan capas de abstracción respecto de la BD (una interfaz común).



```
graph TD
    PHP[PHP] -- query() --> DAL[DB Abstraction Layer]
    DAL --> MSSQL[(MS SQL)]
    DAL --> MySQL[(MySQL)]
    DAL --> Oracle[(Oracle)]
```

1. Introducción

2. Objetos de acceso a datos (PDO)

- El objeto PDO
- El objeto PDOStatement
- Tratamiento de errores

3. Buenas prácticas usando PHP

Febrero 2012

Introducción a la Ingeniería del Software y a los Sistemas de Información

1

**UNIVERSIDAD DE SEVILLA**  
Escuela Técnica Superior de Ingeniería Informática  
Departamento de Lenguajes y Sistemas Informáticos

1. Introducción  
2. Objetos de acceso a datos (PDO)  
• El objeto PDO  
• El objeto PDOStatement  
• Tratamiento de errores  
3. Buenas prácticas usando PHP

## Objetos de Acceso a Datos

- **Conceptos básicos**
  - PDO = Portable Data Objects
  - Es una librería de objetos “ligeros” PHP que permiten acceder a bases de datos desde el código PHP.
  - PDO provee una capa de abstracción para el acceso a datos, lo que permite que nuestro código sea independiente de la BD que usemos.
  - El objeto más importante de PDO es precisamente PDO que representa una conexión con la BD.
  - PDO forma parte de la distribución PHP desde la versión v5.1

```

graph TD
    PDO[PDO] --- PDODriver[PDODriver]
    PDO --- PDOException[PDOException]
    PDOStatement[PDOStatement] --- PDO
    PDOStatement --- Parametros[Parámetros]
  
```

Febrero 2012
Introducción a la Ingeniería del Software y a los Sistemas de Información
2

**UNIVERSIDAD DE SEVILLA**  
Escuela Técnica Superior de Ingeniería Informática  
Departamento de Lenguajes y Sistemas Informáticos

1. Introducción  
2. Objetos de acceso a datos (PDO)  
• El objeto PDO  
• El objeto PDOStatement  
• Tratamiento de errores  
3. Buenas prácticas usando PHP

## Objetos de Acceso a Datos

- **Actores**
  - Proveedor de datos (PDO Specific Driver)
  - Conexión (Objeto PDO)
  - Base de Datos (MySQL, MS SQL Server, Oracle)

```

graph TD
    PDO[PDO] --- PDOSpecificDriver[PDO Specific Driver]
    PDO --- PDOSpecificDriver2[PDO Specific Driver 2]
    PDOSpecificDriver --- BD[(BD)]
    PDOSpecificDriver2 --- BD2[(BD2)]
  
```

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "
<html>
<head>
<meta http-equiv="Content-Type" content="te
<title>Untitled Document</title>
</head>
<body>
<?php
foreach(PDO::getAvailableDrivers() as $driver)
{
    echo $driver."<br />";
}
?>
</body>
</html>
  
```

MS SQL  
MySQL  
Oracle

Febrero 2012
Introducción a la Ingeniería del Software y a los Sistemas de Información
3

**UNIVERSIDAD DE SEVILLA**  
Escuela Técnica Superior de Ingeniería Informática  
Departamento de Lenguajes y Sistemas Informáticos

1. Introducción  
2. Objetos de acceso a datos (PDO)  
• El objeto PDO  
• El objeto PDOStatement  
• Tratamiento de errores  
3. Buenas prácticas usando PHP

## El Objeto PDO

- Definición del Objeto PDO**
  - Este objeto representa una conexión con una base de datos, y nos permite ejecutar consultas y obtener sus resultados.

```
<?
$host = 'oci:dbname=localhost/XE';
$username = 'username';
$password = 'password';
try {
    $con = new PDO($host,$username,$password);
    $con->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
    echo 'Conectados a la BD';
    //...
    // Cerramos la conexión
    $con = null;
}
catch( PDOException $e ) {
    // tratamiento del error
    echo "error de conexión: ".$e->getMessage();
}
?>
```

**!!! DEBEMOS INDICAR  
EL MODO DE ERROR  
DE EXCEPCIONES PARA  
QUE LAS LANCE !!!**

**Si no capturamos la  
excepción, el servidor  
mostrará la traza  
!!! incluyendo el usuario  
y password de la BD !!!**

Febrero 2012      Introducción a la Ingeniería del Software y a los Sistemas de Información      4


**UNIVERSIDAD DE SEVILLA**  
Escuela Técnica Superior de Ingeniería Informática  
Departamento de Lenguajes y Sistemas Informáticos

1. Introducción  
2. Objetos de acceso a datos (PDO)  
• El objeto PDO  
• El objeto PDOStatement  
• Tratamiento de errores  
3. Buenas prácticas usando PHP

## El Objeto PDO

- Métodos de *PDO***
  - `PDO( cadenaConexión, [ usuario], [ pwd], [ opts])`: abre la conexión con la BD. La cadena de conexión depende del tipo de BD. Opcionalmente se puede indicar usuario, password y un conjunto de opciones de configuración.
  - `beginTransaction()`: comienza una transacción.
  - `commit()`: termina una transacción.
  - `rollback()`: aborta una transacción.
  - `exec(SQL)`: ejecuta una instrucción SQL (INSERT, UPDATE, DELETE) y devuelve el número de filas afectadas.
  - `query(SQL)`: ejecuta una consulta SQL ("*SELECT...*") contra la BD, devuelve un objeto **PDOStatement** con los resultados.
  - `lastInsertId()`: devuelve la clave primaria de la última fila insertada. (Nota: este método puede devolver resultados inconsistentes dependiendo del driver usado).

Febrero 2012      Introducción a la Ingeniería del Software y a los Sistemas de Información      5



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

- 1. Introducción
- 2. Objetos de acceso a datos (PDO)
  - El objeto PDO
  - El objeto PDOStatement
  - Tratamiento de errores
- 3. Buenas prácticas usando PHP


### El Objeto PDOStatement

- **Definición de PDOStatement**
  - Representa una consulta preparada para su ejecución en la BD y tras su ejecución, su resultado.
  - Su uso habitual es el siguiente:

```

<?
try {
    $stmt= $con->query("SELECT ..." );
    // o bien...
    $stmt = $con->prepare("SELECT ... WHERE a=:data");
    $stmt->bindParam(':data',$name);
    $stmt->execute();
} catch(PDOException $e ) {
    // tratamiento del error
    echo "error: ".$e->GetMessage();
}
// recorrido del PDOStatement
?>
```

Febrero 2012
Introducción a la Ingeniería del Software y a los Sistemas de Información
6



Escuela Técnica Superior  
de Ingeniería Informática  
Departamento de Lenguajes  
y Sistemas Informáticos

- 1. Introducción
- 2. Objetos de acceso a datos (PDO)
  - El objeto PDO
  - El objeto PDOStatement
  - Tratamiento de errores
- 3. Buenas prácticas usando PHP

### El Objeto PDOStatement

- **Métodos de *PDOStatement***
  - bindParam(parámetros): asocia un valor a los parámetros establecidos en la consulta.
  - execute(parámetros): ejecuta la consulta usando los valores introducidos en el array parámetros.
  - rowCount(): Devuelve el número de filas afectadas por la consulta (no funciona en todas las BDs).
  - columnCount(): Devuelve el número de columnas.


```

<?
$calories = 150;
$colour = 'red';

$stmt = $con->prepare('SELECT name, colour, calories
FROM fruit WHERE calories < :calories AND colour = :colour');

$stmt->execute(array(':calories' => $calories, ':colour' => $colour));
?>
```

Febrero 2012
Introducción a la Ingeniería del Software y a los Sistemas de Información
7



UNIVERSIDAD DE SEVILLA  
Escuela Técnica Superior de Ingeniería Informática  
Departamento de Lenguajes y Sistemas Informáticos

## El Objeto PDOStatement

- 1. Introducción
- 2. Objetos de acceso a datos (PDO)
  - El objeto PDO
  - El objeto PDOStatement
  - Tratamiento de errores
- 3. Buenas prácticas usando PHP


• Recorrido del *PDOStatement* (2 opciones)

```
<?
// $stmt es un PDOStatement ya ejecutado
foreach ($stmt as $fila) {
    echo $fila["campo1"];
    ...
    echo $fila["campoN"];
}
?>
```

```
<?
// $stmt es un PDOStatement ya ejecutado
$fila = $stmt->fetch();
while ($fila) {
    echo $fila["campo1"];
    ...
    echo $fila["campoN"];
    $fila = $stmt->fetch();
}
?>
```

**fetch():** Devuelve la siguiente fila del resultado o false si es la última fila.

Febrero 2012
Introducción a la Ingeniería del Software y a los Sistemas de Información
8



UNIVERSIDAD DE SEVILLA  
Escuela Técnica Superior de Ingeniería Informática  
Departamento de Lenguajes y Sistemas Informáticos

## Tratamiento de Errores

- 1. Introducción
- 2. Objetos de acceso a datos (PDO)
  - El objeto PDO
  - El objeto PDOStatement
  - Tratamiento de errores
- 3. Buenas prácticas usando PHP

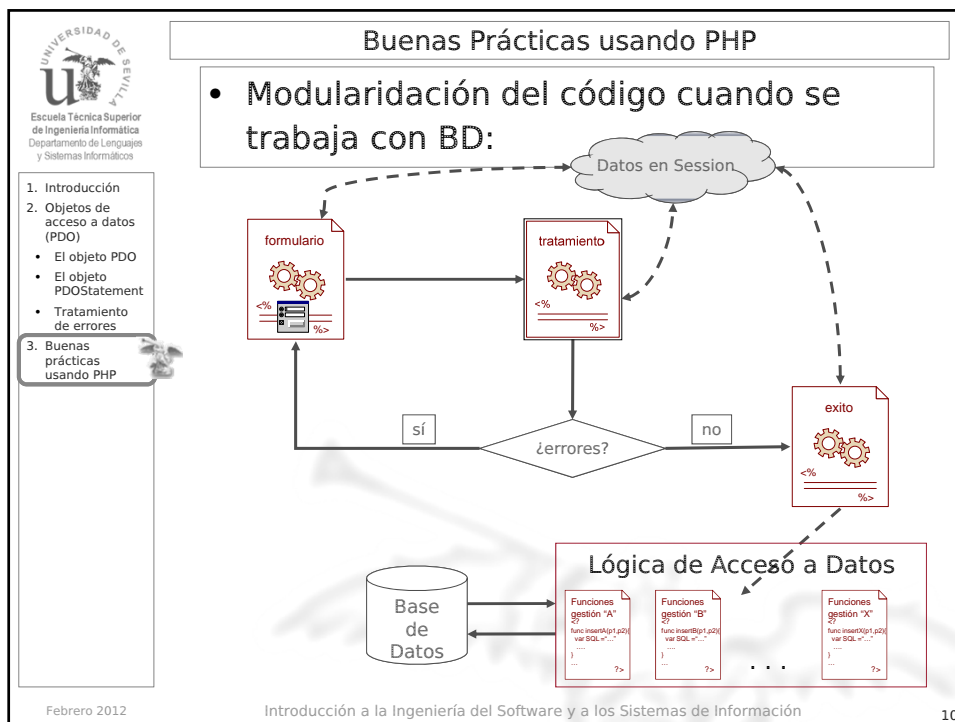
• Bloques try/catch en PHP con PDO

- Todas las operaciones con objetos PDO pueden generar excepciones que deben capturarse.
- El objeto que lanzan en las excepciones es de tipo PDOException, que es a su vez una subclase de Exception.
- Exception tiene los siguientes métodos:
  - getMessage(): devuelve el mensaje describe la excepción.
  - getFile(): fichero en que ocurrió el error.
  - getLine(): línea en que ocurrió el error.
  - getTrace() y getTraceAsString(): devuelven la traza de ejecución en el momento del error.

• Tratamiento del error en el bloque *catch*

- El código habitual en el bloque *catch* es liberar recursos y avisar al usuario del error.

Febrero 2012
Introducción a la Ingeniería del Software y a los Sistemas de Información
9



**UNIVERSIDAD DE SEVILLA**  
Escuela Técnica Superior de Ingeniería Informática  
Departamento de Lenguajes y Sistemas Informáticos

- Introducción
- Objetos de acceso a datos (PDO)
  - El objeto PDO
  - El objeto PDOStatement
  - Tratamiento de errores
- Buenas prácticas usando PHP

### Buenas Prácticas usando PHP

**exito**

- Rescatar los datos del formulario
- Anular los valores en la Session
- Hacer el tratamiento de los datos (p.e. inserción en la BDs)

```

<?php
include_once("gestionUsuarios.php");
$formulario = $_SESSION[ "formulario" ];
$_SESSION[ "formulario" ] = null;
$_SESSION[ "errores" ] = null;
?>
<html>
<head>
<title>Resultado del Registro</title>
</head>
<body>
<? if(insertarUsuario(formulario[ "nombre" ],
formulario[ "direccion" ]) {
?>
<div id="div_exito">
<h1>Hola <?= formulario[ "nombre" ] ?>,
gracias por registrarte</h1>
</div>
<? }
else { ?>
<div id="div_error_registro">
Lo sentimos ya existe un usuario con su misma
dirección.
</div>
<? } ?>

<div id="div_volver">
Pulsa <a href="formulario.php">aquí</a>
para volver al formulario.
</div>
</body>
</html>

```

**UNIVERSIDAD DE SEVILLA**  
Escuela Técnica Superior de Ingeniería Informática  
Departamento de Lenguajes y Sistemas Informáticos

1. Introducción
2. Objetos de acceso a datos (PDO)
  - El objeto PDO
  - El objeto PDOStatement
  - Tratamiento de errores
3. Buenas prácticas usando PHP

## Buenas Prácticas usando PHP

**Funciones de Gestión de "X"**  

```
<%
func insertX(p1,p2){
var SQL = "...";
...
%>
```

Implementar la lógica de acceso a datos de la aplicación para cada concepto del dominio:

- Inserción
- Modificación
- Eliminación
- Selección por diversos criterios.

```
<?
/*
+-----+
|  Gestión de Usuarios  |
+-----+
*/

// Inserción:
function insertarUsuario($usuario, $direccion){
    $resultado=true;
    $usuarios=usuariosporDireccion($direccion);
    $fila = $usuarios->fetch();
    if(!$fila){
        $con = new PDO("oci:dbname=localhost/XE",
            $username,$password);
        $con->setAttribute(PDO::ATTR_ERRMODE,
            PDO::ERRMODE_EXCEPTION);

        $stmt = $con->execute("INSERT INTO Usuarios...");
    }else
        $resultado=false;
    $fila = null;
    $usuarios=null;
    return resultado;
}

// Selección de usuarios por distintos criterios:
function usuariosporDireccion($direccion){
    $con = new PDO(...);
    $stmt = $con->prepare("SELECT * FROM Usuarios WHERE
        direccion=:direccion");
    $stmt->bindParam(':direccion',$direccion);
    $stmt->execute();
    return $stmt;
}
?>
```

Este código no pertenecería realmente a la lógica de acceso a datos, sino a la lógica de funcionamiento de la organización.

Duplicación de conexión datos / código

Febrero 2012      Introducción a la Ingeniería del Software y a los Sistemas de Información      12


**UNIVERSIDAD DE SEVILLA**  
Escuela Técnica Superior de Ingeniería Informática  
Departamento de Lenguajes y Sistemas Informáticos

1. Introducción
2. Objetos de acceso a datos (PDO)
  - El objeto PDO
  - El objeto PDOStatement
  - Tratamiento de errores
3. Buenas prácticas usando PHP

## Buenas Prácticas usando PHP

- Normalmente, los pasos a seguir para cada función de la capa de acceso a datos son:
  - Construir la sentencia SQL
    - [ Opcional] Tratamiento y formateo de los parámetros.
  - Obtener una conexión. Opciones:
    - Creación y apertura directamente en la función
    - Obtener la conexión de los contextos de Sesión o Aplicación
    - Tratar la conexión como parámetro de cada función
  - Ejecución del comando SQL
  - [ Opcional] Tratamiento de errores (podría ser delegado en el código que llama la función para realizar un tratamiento más específico)
  - Si abrimos la conexión debemos cerrarla
  - Devolver los resultados si procede

Febrero 2012      Introducción a la Ingeniería del Software y a los Sistemas de Información      13



Escuela Técnica Superior de Ingeniería Informática  
Departamento de Lenguajes y Sistemas Informáticos

1. Introducción
2. Objetos de acceso a datos (PDO)
  - El objeto PDO
  - El objeto PDOStatement
  - Tratamiento de errores
3. Buenas prácticas usando PHP

### Buenas Prácticas usando PHP

```

<?php
include_once("BDConexion.php");
include_once("gestionUsuarios.php");

$form = $_SESSION["form"];
$_SESSION["form"] = null;
$_SESSION["errors"] = null;
$conexion = conectarBD();
?>
<html>
<head>
<title>Resultado del Registro</title>
</head>
<body>
    <? if(crearUsuario(form["nomb"],form["direccion"],$conexion)
    {
        <? >
        <div id="div_exito">
        <h1>Hola <?= form["nomb"] ?>,
        gracias por registrarse</h1>
        </div>
        <? >
        <? else{ ?>
        <div id="div_error_registro">
        Lo sentimos, ya existe un usuario con su misma dirección,
        el registro como usuario fue cancelado.
        </div>
        <?>?>
        <div id="div_volver">
        Pulse <a href="formulario.asp">aquí</a>
        para volver al formulario
        </div>
        </body>
        </html>
    <? desconectarBD($conexion); ?>
    </html>
            
```

Febrero 2012

Introducción a la Ingeniería del Software y a los Sistemas de Información

14