



# PDO

## *PHP Data Object – Parte 01*

**Prof. Fábio Procópio**  
**Prof. João Nascimento**



3

# O que é PDO?

- **PHP Data Object** é uma classe desenvolvida especificamente para **trabalhar** com procedimentos relacionados a **banco de dados**;
- Define uma **interface de comunicação** consistente para **acessar BD** usando PHP;
- Para acessar um determinado servidor de BD usando PDO, é necessário usar um **driver**;
- PDO fornece uma camada de abstração para acesso aos dados, ou seja, são usadas as mesmas funções para **acessar os dados em qualquer SGBD (isso é lindo!)**
  - A aplicação não sabe qual SGBD (MySQL, SQL Server, Oracle) está sendo usado
- PDO garante que, em uma **eventual mudança** de SGBD, **não surgirão grandes problemas** para a aplicação (**isso é mais lindo!!!**)



4

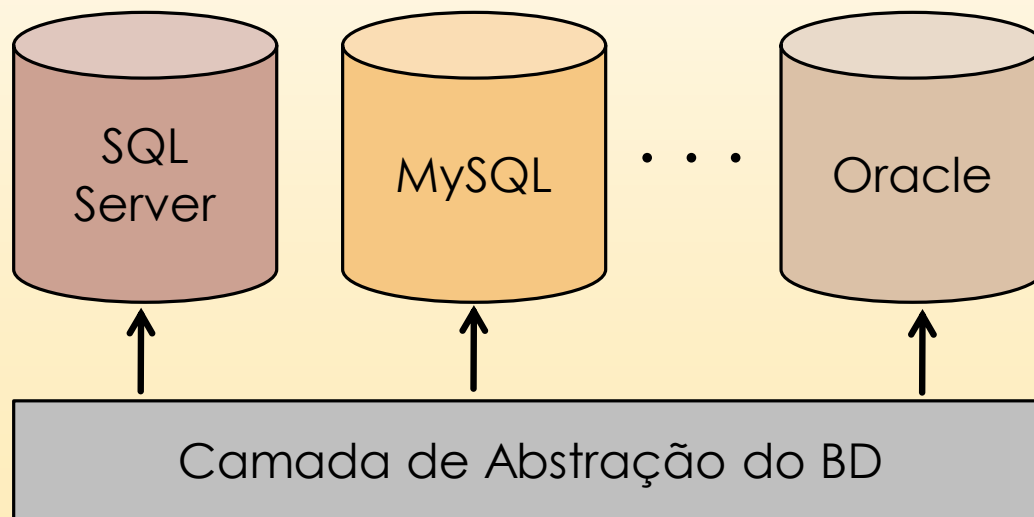
# Tratando exceções em PDO

```
<?php
    try{
        #Tentativa de execução de comandos
    }
    catch(PDOException $e){
        #Tratamento do erro...
    }
?>
```



# Aplicação

- Os principais SGBDs são compatíveis com o padrão SQL, no entanto, as interfaces utilizadas pelos desenvolvedores podem variar;
- Para resolver problemas como esses, programadores vêm desenvolvendo camadas de abstração para acesso a dados.





6

# Abrindo uma conexão

- Cria-se uma instância da classe PDO;
- Especifica-se o driver a ser usado que, no nosso caso, é o MySQL;
- Sintaxe:

```
<?php
    $conn = new PDO('mysql:host=$serv;dbname=$nomeBD',
                    $usuario, $senha,
                    array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8"));
?>
```

- Nossas conexões serão com o banco de dados **dbBiblioteca**. Caso não o tenha em seu servidor, clique [aqui](#) para baixar o script.



# A classe de conexão

```
<?php
class ConexaoBD {
    public static $instancia;

    public static function conecta() {
        $serv = "localhost";
        $bd    = "dbBiblioteca";
        $usr   = "root";
        $pwd   = "";
        if (!isset(self::$instancia)) {
            self::$instancia = new PDO('mysql:host=' . $serv . ';dbname=' . $bd, $usr, $pwd,
                                       array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8"));
            self::$instancia->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
            self::$instancia->setAttribute(PDO::ATTR_ORACLE_NULLS, PDO::NULL_EMPTY_STRING);
            echo "<b>Sucesso na conexão com $bd!</b><br><br>";
        }
        return self::$instancia;
    }
}
```



8

# Teste de conexão

```
<?php
include_once("ConexaoBD.php");
try{
    $conn = ConexaoBD::conecta();
}
catch(PDOException $e) {
    echo "ERRO: ".$e->getMessage();
}
?>
```

**TestaConexao.php**



# Método `fetch()`

- Recebe um parâmetro informando o tipo de busca que será retornada:
  - **PDO::FETCH\_ASSOC** → Retorna um *array* indexado pelo nome da coluna
  - **PDO::FETCH\_NUM** → Retorna um *array* indexado pelo número da coluna
  - **PDO::FETCH\_BOTH** → Retorna um *array* indexado pelo nome da coluna e pelo número correspondente
  - **PDO::FETCH\_OBJECT** → Retorna um objeto anônimo (`stdClass`), de modo que cada coluna é acessada como uma propriedade;





# Classe Curso – 1 de 8

Curso
- idcurso - nocurso
+ __construct(allAttributes) + lista() + consulta(codigo) + insere(allAttributes) + altera(allAttributes) + exclui(codigo)



# Classe Curso – 2 de 8

Curso.php

```
<?php
include_once('ConexaoBD.php');
class Curso {
    private $idcurso;
    private $nocurso;

    function getIdcurso() {
        return $this->idcurso;
    }
    function getNocurso() {
        return $this->nocurso;
    }
    function setIdcurso($idcurso) {
        $this->idcurso = $idcurso;
    }
    function setNocurso($nocurso) {
        $this->nocurso = $nocurso;
    }
}
?>
```



# Classe Curso – 3 de 8

Curso.php

```
class Curso{  
    # métodos getters e setters...  
    function __construct($nocurso = null) {  
        $this->nocurso = $nocurso;  
    }  
  
}
```



# Classe Curso – 4 de 8

Curso.php

```
class Curso{
    # métodos anteriores. . .
    public function lista(){
        try {
            $sql = "SELECT IdCurso, NoCurso FROM TbCurso ORDER BY NoCurso";
            $conn = ConexaoBD::conecta();
            $sql = $conn->query($sql);

            $lst_cursos = array();
            while($row = $sql->fetch(PDO::FETCH_OBJ)) {
                $curso = new Curso();
                $curso->setIdcurso($row->IdCurso);
                $curso->setNocurso($row->NoCurso);
                $lst_cursos[] = $curso;
            }
            return $lst_cursos;
        } catch (Exception $e) {
            echo "ERRO: ".$e->getMessage(). "<br><br>";
        } //try-catch
    } //método lista
} //classe
```



# Testando o método lista() – 1 de 3

TestaCurso.php

```
include_once("Curso.php");

$c = new Curso();
echo "<b>Método lista():</b><br>";
$lista_curso = $c->lista();
foreach($lista_curso as $lst_curso) {
    echo $lst_curso->getIdcurso().'<br>';
    echo $lst_curso->getNocurso().'<br>';
}
```



15

## Testando o método lista() – 2 de 3

- Neste exemplo, vamos montar um elemento Select, do HTML, usando o método lista:
- MontaSelect.php**

```
<?php
    include_once("Curso.php");

    $c = new Curso();
    $lista_curso = $c->lista(); //Método já faz uma conexão com o BD
    echo "Curso: ";
    echo "<select>";
        echo "<option value=0>(Selecione)</option>";
        foreach ($lista_curso as $lst_curso) {
            echo "<option value=\".$lst_curso->getIdcurso().\">".
                $lst_curso->getNocurso()."</option>";
        }
    echo "</select>";
?>
```



16

## Testando o método lista() – 3 de 3

- Neste exemplo, vamos montar um elemento Table, do HTML, usando o método lista:

**MontaTable.php**

```
<?php
include_once("Curso.php");
$c = new Curso();

$lista_curso = $c->lista(); //Método já faz uma conexão com o BD
echo "<table border=1>";
    echo "<tr>";
        echo "<th>Nome</th>";
    echo "</tr>";
    foreach ($lista_curso as $lst_curso) {
        echo "<tr>";
            echo "<td><a href='PreencheForm.php?idcurso=" .
                $lst_curso->getIdcurso()."'>".
                $lst_curso->getNocurso()."</a></td>";
        echo "</tr>";
    }
echo "</table>";

?>
```