

## PDO – PHP Data Objects

Em razão da crescente adoção do PHP, surgiu a necessidade de unificar o acesso as diferentes extensões de bancos de dados presentes no PHP. O objetivo da PDO (PHP Data Objects) é promover uma API limpa e consistente, unificando a maioria das características presentes nas extensões de acesso a banco de dados.

- Unifica o acesso às diferentes extensões de bancos de dados presentes em PHP;
- Provê uma API limpa e consistente, unificando a maioria das características presentes nas extensões de acesso a banco de dados (unifica a chamada de métodos, graças à orientação a objetos presentes no PHP5);
- Sequência de passos:
  1. A aplicação estabelece uma conexão com o SGBD;
  2. A aplicação seleciona uma base de dados a ser utilizada durante a conexão;
  3. A aplicação executa uma consulta específica (seleção, inserção, remoção...);
  4. A aplicação fecha a conexão corrente com o SGBD;

PDO – Conexão com bancos de dados:

Banco	String de Conexão
SQLite	<code>new PDO('sqlite: teste.db');</code>
FireBird	<code>new PDO("firebird:dbname=C:\\base.GDB", "SYSDBA", "masterkey");</code>
MySQL	<code>new PDO('mysql:host=localhost;dbname=base', 'login', 'senha');</code>
Postgres	<code>new PDO('pgsql:dbname=example; user=user; password=senha;host=localhost');</code>

Para podermos utilizar o PDO, deve-se ser habilitado no php.ini as bibliotecas (drivers) de acesso para o banco de dados de nossa aplicação, especificamente para o mysql:

### Extension=php\_pdo\_mysql.dll

Para realizar esta configuração no usbwebserver, entre na pasta settings e abra com o bloco de notas o arquivo php.ini. Localize o texto **php\_pdo\_mysql.dll**. Haverá um caractere de ; (ponto e vírgula) na frente desta linha deixando-a como comentário, retire este ; e salve o arquivo.

Feche a abra o usbwebserver para ele carregar as novas configurações do seu PHP.

```

php - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
extension=php_mysql.dll
extension=php_mysql.dll
;extension=php_oci8.dll      ; Use with Oracle 10gR2 Instant Client
;extension=php_oci8_11g.dll ; Use with Oracle 11gR2 Instant Client
extension=php_openssl.dll
;extension=php_pdo_firebird.dll
;extension=php_pdo_mssql.dll
extension=php_pdo_mysql.dll
;extension=php_pdo_oci.dll
;extension=php_pdo_odbc.dll
extension=php_pdo_pgsql.dll
extension=php_pdo_sqlite.dll
extension=php_pgsql.dll
;extension=php_pspell.dll
;extension=php_shmop.dll

```

## PDO – Exemplos

- No phpmyadmin, configure o banco de dados da atividade:
  - Database: livros
  - Tabela: famosos
    - Campos: codigo (int, auto increment, chave primária) e nome (varchar[100]);

Iremos trabalhar o acesso ao CRUD que vem do inglês, das palavras (Create, Read, Update, Delete) que quer dizer basicamente as 4 (quatro) principais operações com um banco de dados (inserir, ler, atualizar, excluir). Quando pensamos em qualquer website dinâmico ou sistemas em gerais, a primeira coisa que vem na nossa cabeça é o CRUD, sem ele nada funciona, pois é com ele que realizamos as interações com o banco de dados.

### Listagem 1 - Inserindo dados via PDO (inserir.php)

```
<?php

$conn = new PDO('mysql:host=localhost;port=3307;dbname=livros', 'root', 'usbw');

$conn->exec("INSERT INTO famosos (nome) VALUES ('Crônicas de Nárnia')");

$conn = null;

?>
```

### Listagem 2 - Listagem via PDO (listagem.php)

```
<?php
$conn = new PDO('mysql:host=localhost;port=3307;dbname=livros', 'root', 'usbw');

$result = $conn->query("SELECT codigo, nome from famosos");

if($result)
    foreach($result as $row)
        echo $row['codigo'] . '-' . $row['nome'] . '<br>';

$conn = null;

?>
```

### Listagem 3 - Excluindo dados via PDO (excluir.php)

```
<?php

$conn = new PDO('mysql:host=localhost;port=3307;dbname=livros', 'root', 'usbw');

$conn->exec("DELETE FROM famosos WHERE codigo = 1");

$conn = null;

?>
```

A possibilidade de reutilizar partes de código já definidas é o que nos dá maior agilidade no dia-a-dia, além de eliminar a necessidade de eventuais duplicações ou reescritas de código.

## **Exercício**

- Etapa 1 – Preparação do Banco de Dados no phpmyadmin
  - Database: sistema
  - Tabela: cadastro
    - Campos: codigo (int, auto increment, chave primária), nome (varchar[100]), telefone (varchar[100]) e email (varchar[100]);
- Etapa 2 – Preparação do Domínio
  - Crie uma pasta em htdocs chamada sistema;
- Etapa 3 – Criação de um formulário HTML
  - Crie um formulário HTML que permita ao usuário digitar seu nome, telefone e e-mail;
    - Método de envio: POST;
    - Os dados deverão ser submetidos para o script “cadastro.php”;
- Salve o documento HTML na pasta sistema com o nome index.html;
- Etapa 4 – Criação do script cadastro.php na pasta sistema
  - O script deverá inicialmente receber os dados submetidos na etapa anterior;
  - Em seguida, via PDO, o script deverá inserir os dados no banco de dados;
  - O script deverá redirecionar o usuário para o script lista\_usuarios.php;
- Etapa 5 – Criação do script lista\_usuarios.php na pasta sistema
  - O script deverá exibir todos os cadastros efetuados até o momento, além de disponibilizar um link que permita ao usuário efetuar um novo cadastro;