

Aula Introdutória sobre PDO em PHP

Claudio Azevedo Passos e João Luiz Lagôas

O que é PDO?

- PDO significa **PHP Data Objects**.
É uma camada de acesso a banco de dados do PHP que permite conectar e manipular diferentes bancos utilizando uma interface padronizada.

Com PDO, você pode trabalhar com:

- MySQL
- PostgreSQL
- SQLite
- Oracle
- SQL Server
- entre outros

Vantagens do PDO

1. Segurança

- PDO permite o uso de **Prepared Statements**, protegendo contra SQL Injection.

2. Portabilidade

- O mesmo código pode funcionar em vários bancos com pequenas alterações.

3. Organização

- Código mais limpo e profissional.

4. Tratamento de Erros

- Possui sistema robusto de exceções.

Estrutura Básica do PDO

```
<?php
```

```
$pdo = new PDO(  
    "mysql:host=localhost;dbname=escola",  
    "root",  
    ""  
);
```

Entendendo a Conexão

- DSN (Data Source Name)

"mysql:host=localhost;dbname=escola"

Define:

- tipo do banco
- servidor
- nome do banco

Exemplo Completo de Conexão

```
<?php
```


```
try {
```

```
    $pdo = new PDO(
        "mysql:host=localhost;dbname=escola;charset=utf8",
        "root",
        ""
    );
```

```
    echo "Conexão realizada com sucesso!";
```

```
} catch (PDOException $e) {
```

```
    echo "Erro: " . $e->getMessage();
}
```



Estrutura TRY/CATCH
serve para tratar erros

Executando um INSERT

- INSERT INTO alunos(nome, turma)
- VALUES('João', 'INFO1');

INSERT com PDO

```
<?php
```

```
$sql = "INSERT INTO alunos(nome, turma)  
VALUES('João', 'INFO1)";
```

```
$pdo->exec($sql);
```

```
echo "Registro inserido!";
```

Problema de Segurança

- Imagine:

```
$nome = $_POST['nome'];
```

Se o usuário digitar comandos SQL maliciosos, o sistema pode ser atacado.

Isso é chamado de:

SQL Injection

Prepared Statements

- São comandos preparados que separam:
 - código SQL
 - dados do usuário

INSERT Seguro com Prepared Statement

```
<?php
```

```
$sql = "INSERT INTO alunos(nome, turma)
```

```
    VALUES(:nome, :turma)";
```

```
$stmt = $pdo->prepare($sql);
```

```
$stmt->bindValue(':nome', 'Maria');
```

```
$stmt->bindValue(':turma', 'INFO2');
```

```
$stmt->execute();
```

```
echo "Aluno cadastrado!";
```

Consultando Dados

- SELECT com PDO

```
<?php
$sql = "SELECT * FROM alunos";
$stmt = $pdo->query($sql);
$dados = $stmt->fetchAll();
echo "<pre>";
print_r($dados);
echo "</pre>";
```

fetch() e fetchAll()

fetch()

Retorna apenas 1 registro.

fetchAll()

Retorna todos os registros.

Exibindo Dados com foreach

```
<?php
```

```
$sql = "SELECT * FROM alunos";
```

```
$stmt = $pdo->query($sql);
```

```
$dados = $stmt->fetchAll();
```

```
foreach ($dados as $aluno) {
```

```
    echo $aluno['nome'] . "<br>";
```

```
}
```

SELECT com WHERE

```
<?php
```

```
$sql = "SELECT * FROM alunos
```

```
    WHERE turma = :turma";
```

```
$stmt = $pdo->prepare($sql);
```

```
$stmt->bindValue(':turma', 'INFO1');
```

```
$stmt->execute();
```

```
$resultado = $stmt->fetchAll();
```

Update

```
<?php
```

```
$sql = "UPDATE alunos  
      SET turma = :turma  
      WHERE id = :id";  
$stmt = $pdo->prepare($sql);  
$stmt->bindValue(':turma', 'INFO3');  
$stmt->bindValue(':id', 1);  
$stmt->execute();  
echo "Atualizado!";
```

Delete

```
<?php
```

```
$sql = "DELETE FROM alunos  
      WHERE id = :id";
```

```
$stmt = $pdo->prepare($sql);
```

```
$stmt->bindValue(':id', 1);
```

```
$stmt->execute();
```

```
echo "Registro removido!";
```

Principais Métodos PDO

Método	Função
<code>prepare()</code>	Prepara SQL
<code>execute()</code>	Executa
<code>query()</code>	Executa SELECT simples
<code>exec()</code>	Executa INSERT/UPDATE/DELETE simples
<code>fetch()</code>	Retorna um registro
<code>fetchAll()</code>	Retorna vários registros

Exemplo Completo

```
<?php
try {
    $pdo = new PDO(
        "mysql:host=localhost;dbname=escola;charset=utf8",
        "root",
        "" );
    $pdo->setAttribute(
        PDO::ATTR_ERRMODE,
        PDO::ERRMODE_EXCEPTION );
    $sql = "INSERT INTO alunos(nome, turma)
        VALUES(:nome, :turma)";
    $stmt = $pdo->prepare($sql);
    $stmt->bindValue(':nome', 'Carlos');
    $stmt->bindValue(':turma', 'INFO1');
    $stmt->execute();
    echo "Aluno cadastrado!";
} catch (PDOException $e) {
    echo "Erro: " . $e->getMessage();
}
```

Exercício Prático 1

Crie um sistema com:

formulário HTML

campos:

nome

turma

Ao clicar em “Cadastrar”:

salvar no banco usando PDO

Exercício Prático 2

Crie uma página que:

liste todos os alunos

utilize:

SELECT

fetchAll()

foreach

Exercício Prático 3

Crie uma busca por turma:

- usuário digita a turma
- sistema exibe apenas os alunos daquela turma
- Utilize:

prepare()

bindValue()

execute()