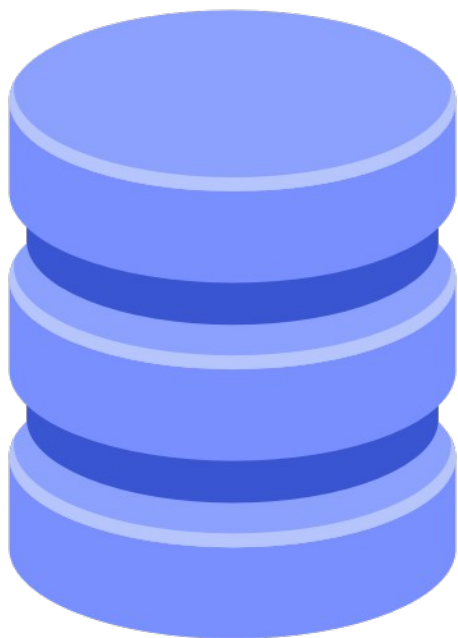


Linguagem PHP

Prof. Daniel Di Domenico



Banco de dados

PHP com banco de dados

- **Objetivo da aula:**
 - Conectar uma aplicação PHP ao banco de dados
 - Utilizar a classe de conexão PDO
 - Gravar e buscar os dados de uma base



PHP com banco de dados

- Utilizaremos a classe **PDO** para conectar ao MySQL (PDO tem suporte a outros bancos)
 - https://www.php.net/manual/pt_BR/class.pdo.php
- Exemplo para criar uma conexão:

```
$opcoes = array(<...OPÇÕES>);  
  
$conn =  
    new PDO("mysql:host=XXXX;dbname=XXX",  
            "user", "pass", $opcoes);
```

PHP com banco de dados

- Exemplo de **opções** da conexão PDO:

```
$opcoes = array(  
    //Define o charset da conexão  
    PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8",  
    //Define o tipo do erro como exceção  
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,  
    //Define o retorno das consultas como  
    //array associativo (campo => valor)  
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC  
);
```

PHP com banco de dados

- Acesso ao banco de dados com PDO:
 - **1-** Criar uma classe que retorna a conexão com o banco (objeto PDO)
 - **2-** Incluir o arquivo da conexão onde for necessário acesso ao banco de dados

```
//Arquivo Connection.php
class Connection {

    private static $conn = null;

    public static
        function getConnection() {

            if(self::$conn == null) {
                try {
                    //Cria a conexão...
                } catch(PDOException $e) {
                    echo "Erro";
                }
            }
            return self::$conn;
        }
}
```

require_once: adiciona um arquivo .php na página

```
//Arquivo que utilizará a conexão

//Adiciona o arquivo Connection.php
require_once("Connection.php");

//Chama o método que cria a conexão
$conn = Connection::getConnection();
```

PHP com banco de dados

- Executando uma consulta na base de dados com PDO:

```
//Obtém a conexão já implementada
$conn = Connection::getConnection();

$sql = "SELECT * FROM exemplo";

//Prepara e executa o comando SQL
$stmt = $conn->prepare($sql);
$stmt->execute();

//Armazena os resultados ($result é uma matriz)
$result = $stmt->fetchAll();
```

PHP com banco de dados

- Retornando os dados da consulta com PDO:

```
//... continuação do slide anterior  
  
//Acesso aos registros da matriz $result  
foreach ($result as $reg) {  
    echo $reg['coluna1'];  
    echo ' - ';  
    echo $reg['coluna2'];  
    echo '<br>';  
}
```

\$result é um *array* indexado
que possui um *array*
associativo dentro dele

\$reg é um *array* associativo.
As **chaves** são os **nomes das**
colunas na base de dados

PHP com banco de dados

- Executando INSERT/UPDATE/DELETE com PDO:

```
//Obtém a conexão já implementada
$conn = Connection::getConnection();

$sql = "INSERT/UPDATE/DELETE... <?, ?, ?...>";

$stmt = $conn->prepare($sql);
$stmt->execute(...PARAMETROS);
```

Array de parâmetros:
deve ser passado um valor
para cada ? do SQL

Parâmetros da
instrução SQL

Comando úteis PHP

- **self**: refere-se à classe (operador `::`)
 - Utilizado em métodos e com variáveis **static**
 - **\$this** refere-se ao objeto (operador `->`)

Exercícios

- **1-** Faça uma aplicação para cadastrar produtos (id, descrição e unidade de medida) em uma base de dados. A aplicação deve ter 4 páginas, sendo:
 - 1.1 – Conexão com a base de dados
 - 1.2 – Listar os produtos cadastrados em uma tabela HTML
 - 1.3 – Excluir os produtos (chamada a partir da listagem)
 - 1.4 – Cadastrar os produtos

O *script* para criar a tabela de produtos é:

```
CREATE TABLE produtos (  
  id INTEGER NOT NULL AUTO_INCREMENT,  
  descricao VARCHAR(50) NOT NULL,  
  un_medida VARCHAR(50) NOT NULL,  
  CONSTRAINT pk_produtos PRIMARY KEY (id)  
);
```