



INSTITUTO FEDERAL

Mato Grosso do Sul

Desenvolvimento web baseado em Frameworks:
Banco de dados

Curso técnico integrado em informática para internet

Banco de dados nativo



- O PHP é capaz de se comunicar com diversos SGBD's.
 - Sybase;
 - MySQL;
 - SQL Server;
 - PostgreSQL;
 - Oracle;
 - SQLite.
- Para qualquer SGBD suportado, será possível criar uma conexão a partir de um driver e sua chamada respectiva pelo PHP.

Banco de dados nativo



- Para o exemplo de hoje, será usada a tabela de usuário.
- Aplicação de login.



Banco de dados nativo



- DDL para criação da tabela usuário:

```
CREATE TABLE `bd_1029`.`usuario` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(45) NOT NULL,  
  `senha` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`id`));
```

Banco de dados nativo



- Para testar a conexão com driver nativo:

```
<?php
$conn = new mysqli('localhost', seu_usuario, 'sua_senha');
$db = mysqli_select_db($conn, 'nome_bd');
mysqli_set_charset($conn, 'utf8');

if($conn->connect_error){
    die('Não foi possível conectar: ' . $conn->connect_error);
}
else{
    //código se conectado
}

?>
```

Banco de dados nativo



- Uma vez conectado, as operações de manipulação ou busca de registros podem ser enviadas.
 - SELECT;
 - INSERT;
 - DELETE;
 - UPDATE;
- Outros comandos relacionados a manutenção da estrutura do banco de dados podem ser enviadas, mas não serão abordadas neste momento.

Problema



- E se fizermos todo o código pensando em MySQL e eventualmente precisarmos mudar de banco de dados?

Problema



- E se fizermos todo o código pensando em MySQL e eventualmente precisarmos mudar de banco de dados?
 - Para todo código em PHP relacionado à manipulação ou conexão ao SGBD seria necessário refatorar o código.
- Em PHP, há uma classe especializada em tornar o código relacionado a banco de dados transparente!
 - **PDO (PHP Data Objects).**
- Nos exemplos de aula usaremos PDO.
- É necessário habilitar os drivers no arquivo php.ini

PDO - Conexão



```
<?php
try{
    $conn = new PDO('mysql:host=localhost; dbname=nome_bd, 'seu_usuario', 'senha');
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
}
catch(PDOException $e){
    die("A conexão falhou: ". $e->getMessage() );
}

//código se conectado
```

PDO - Listagem



```
<?php
```

```
//código se conectado
```

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

```
$res = $conn->query($sql);
```

```
foreach($res as $linha){
```

```
    echo($linha["id"]."<br/>");
```

```
    echo($linha["nome"]."<br/>");
```

```
    echo($linha["senha"]."<br/>");
```



PDO - Inserção



```
<?php
```

```
//código se conectado
```

```
$nome = "teste";
```

```
$senha = "testee";
```

```
$sql = "INSERT INTO usuario (nome,senha) VALUES (?,?)";
```

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

```
$stmt->bindParam(1, $nome);
```

```
$stmt->bindParam(2, $senha);
```

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

```
//$stmt->fetchAll() caso fosse realizar uma consulta de filtragem
```

```
//$stmt->rowCount() caso fosse contar quantas linhas resultam da pesquisa
```

Login de usuário



- Premissas:
 - Não há duplicações de nome de usuário.
 - Para todo usuário há uma senha válida.
- Algoritmo de autenticação de credencial:
 1. Receber informações de nome de usuário e senha da aplicação.
 2. Para todo registro no banco de dados, verificar o nome **E** senha são iguais aos armazenados no banco de dados.
 - Se sim, então encontrou uma credencial válida.
 - Armazene a credencial em uma lista.
 3. Se na lista de credenciais houve um registro então o usuário está autenticado.
 - Redirecione para a área restrita.
 4. Caso contrário emita uma mensagem de erro.

Exercícios



1. Adicione na página home.php a funcionalidade de autenticação.
 - Caso a credencial alegada seja **inválida**, redirecione-o de volta para login.
 - Caso a credencial alegada seja válida, crie um cookie de usuário contendo o id do usuário no banco (**não é a melhor prática no mundo real!**).
2. Modifique a página listar.php para realizar a listagem dos dados a partir do banco de dados.
3. Modifique a página listar.php para realizar a inserção dos dados no banco de dados.

PHP - Próxima aula



- **PHP - Banco de dados com PDO.**
 - **Atualização e remoção de registros.**
- **Hospedagem do esqueleto da aplicação.**

Referências



- [1] - DALL'OGGIO, Pablo. **PHP Programando com orientação a Objetos**. Novatec Editora, 2018.
- [2] - [Documentação oficial do PHP](#)
- [3] - [PDO: Documentação oficial do PHP](#)

Dúvidas, críticas ou sugestões

maximilian.melo@ifms.edu.br

max.mjm.melo@gmail.com