



MINISTÉRIO DAS TELECOMUNICAÇÕES, TECNOLOGIAS
DE INFORMAÇÃO E COMUNICAÇÃO SOCIAL
MINISTÉRIO DA EDUCAÇÃO

PROGRAMABILIDADE: LIGAÇÃO COM OUTRAS LINGUAGENS



INSTITUTO DE TELECOMUNICAÇÕES



ÍNDICE

1. Sistema de informação
2. Porquê Integrar SQL com outras Linguagens;
3. Ligação por Meios de APís;
4. PDO;
5. Exemplo de consultas simples;
6. Exercícios;
7. Criar Tabela de Login;



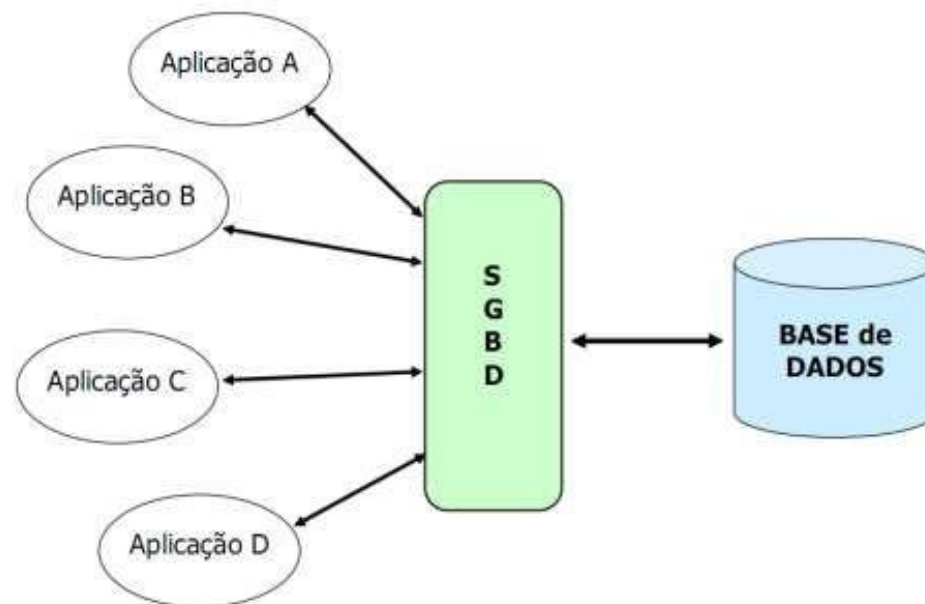
ÍNDICE

1. Sistema de informação
2. Porquê Integrar SQL com outras Linguagens;
3. Ligação por Meios de APís;
4. PDO;
5. Exemplo de consultas simples;
6. Exercícios;
7. Criar Tabela de Login;

LIGAÇÃO COM OUTRAS LINGUAGENS DE PROGRAMAÇÃO

Nos sistemas computacionais a base de dados é o “coração” da aplicação e o seu bom funcionamento é imprescindível para o sucesso de todo o sistema.

As aplicações são escritas em linguagens hospedeiras (hosts) como por exemplo: JAVA, C#, linguagens web, etc. E estas comunicam-se com a base de dados, havendo a necessidade de integrar a linguagem SQL com outras linguagens de programação.





Porquê integrar SQL com outras linguagens?

- ❑ O SQL é uma linguagem não procedimental, onde se indica apenas o que se pretende fazer e

não a forma específica de fazer. Perdendo-se assim algumas funcionalidades como:

- Processamento mais complexo de dados;
- Interfaces especializadas com utilizador;
- Necessidade de aceder a mais do que uma BD;
- Capacidade de tomar decisões.

- ❑ Duas abordagens comuns de integração:

- API (Application Programming Interface) - bibliotecas específicas que interagem com a base de dados (exemplo: JDBC, ODBC, ADO.NET, OLE DB);
- SQL embutido (Embedded SQL) - pedaços de código SQL embutido na linguagem “tradicional” de programação. – Não muito popular



Ligação por meio de API'S

❑ Características:

- Usa objectos e funções especiais da linguagem host.
- Fornece uma interface padronizada com procedimentos e objetos pré-definidos.
- As instruções e apresentação de dados é feita de forma “amigável” através da linguagem host.
- As APIs mais comuns são: PDO, MYSQL, MYSQLI, JDBC, ODBC, OLE DB e DO.NET.
- Iremos optar por usar PDO (JPHP Data Objects)
- Conjunto de classes desenvolvidas para facilitar o acesso à base de dados;
- Faz parte de JavaSE (Java Standard Edition) .

O define uma interface de conexão a banco de dados leve e consistente para PHP. Há a possibilidade de utilização de diversos drivers de conexão que implementam a interface do PDO para vários tipos de bancos de dados. Como o PDO representa uma camada de abstração de acesso aos dados, as mesmas funções utilizadas para manipular dados ou recuperar informações do banco serão as mesmas, independentemente do banco de dados que esteja sendo usado.

Existem três API's de conexão com o banco de dados em PHP, são elas:

mysql: Pacote de funções para acesso ao MySQL, foi descontinuado no PHP7.

mysqli: Extensão da API mysql com suporte a funcionalidades adicionadas a versões posteriores ao

MySQL 4.1 - <http://www.mysql.com/>

PDO - PHP Data Objects: Interface para acesso a dados do PHP.

O PDO é uma API robusta que pode ser usada, independentemente do driver que você estiver usando.

Primeiro vamos ver como se conectar usando o PDO. Nós vamos criar uma nova instância de classe e especificar o driver que vamos usar, no caso o mysql, o nome do banco de dados, nome de usuário e senha.

```
1  <?php
2
3  //Criar as constantes com as credencias de acesso ao banco de dados
4  define('HOST', 'localhost');
5  define('USER', 'root');
6  define('PASS', 'YYYYYY');
7  define('DBNAME', 'XXXXX');
8  define('PORT', '3306');
9
10 //Criar a conexão com banco de dados usando o PDO e a porta do banco de dados
11 //Utilizar o Try/Catch para verificar a conexão.
12 try {
13     $conn = new pdo('mysql:host=' . HOST . ';port=' . PORT . ';dbname=' . DBNAME, USER, PASS);
14     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
15     echo "Conexão com banco de dados realizada com sucesso.";
16 } catch (PDOException $e) {
17     echo "Erro: Conexão com banco de dados não foi realizada com sucesso. Erro gerado " . $e->getMessage();
18 }
19
```


É importante tratar os erros, para se caso aconteça algum erro na conexão, retornar ao usuário.

O erro padrão do PDO é o `PDO::ERRMODE_SILENT`, porém, no código implementamos o `PDO::ERRMODE_EXCEPTION` e abaixo segue a lista das demais opções de erros:

`PDO::ERRMODE_SILENT` //Just set error codes.

`PDO::ERRMODE_WARNING` //Raise E_WARNING.

`PDO::ERRMODE_EXCEPTION` //Throw exceptions.



Exemplo de consultas

```
21
22 $dados = $conn->query("select * from laravel.cursos;");
23
24 while ($linha = $dados->fetch(PDO::FETCH_ASSOC)) {
25     print_r($linha) ;
26 }
27
28
```

Fonte: <https://www.devmedia.com.br/php-pdo-como-se-conectar-ao-banco-de-dados/3721>



Exercício

❑ **Problema:** Usar a linguagem de programação PHP e o SGBD MySQL para criar um mecanismo de autenticação (login).

A autenticação será feita pela combinação correcta dos valores Utilizador e Senha.

O programa deverá apresentar uma mensagem de boas-vindas, caso a autenticação seja bem sucedida.

Abra o MySQL Workbench;

- Seleccione uma base de dados;
- Abra um novo script ;
- Crie a tabela Login representada na figura abaixo. Respeite as regras de integridade;
- Insira um utilizador com os respectivos dados para teste.

| | Field | Type | Null | Key |
|---|------------|-------------|------|-----|
| ▶ | utilizador | varchar(20) | NO | PRI |
| | senha | varchar(20) | NO | |
| | Nome | varchar(20) | NO | |



INSTITUTO DE TELECOMUNICAÇÕES

MUITO OBRIGADO!