

INTEGRAÇÃO DO PHP COM BANCO DE DADOS

DIOGO CEZAR TEIXEIRA BATISTA
<http://inf.cp.utfpr.edu.br/diogo>
diogo@diogocezar.com

QUAIS BANCOS?

- o PHP suporta diversos **SGBD's** incluindo um conjunto de funções NATIVAS para executar operações como:
 - consultas;
 - inclusões;
 - alterações;
 - exclusões;
 - etc...
- estudaremos dois bancos que estão envolvidos em nosso âmbito de estudos:
 - MySQL;
 - PostGreSQL;

BOAS PRÁTICAS

- um bom analista tem em mente a utilização de padrões de projetos, ou AO MENOS a utilização de seus conceitos principais;
- devemos adquirir o bom hábito de separar nossa aplicação em três partes básicas:
 - apresentação (Layout)
 - regra de negócios (Como a coisa funciona)
 - base de dados (Onde isso vai ser gravado)

BOAS PRÁTICAS

- pelo fato do PHP permitir diferentes tipos de programação (Estruturada, Orientada a Objetos ou Uma mistura), é comum ver codificações sem qualquer tipo de organização;
- vamos em um primeiro momento estudar como funcionam os comandos básicos para o acesso a dados, mas com o COMPROMISSO de ao implementá-los, fazer isso com o paradigma da orientação a objetos.

BOAS PRÁTICAS

- é comum separarmos a parte de conexão com o banco de dados em um arquivo de configuração:
 - Fácil configuração ao trocar de computador;
 - Reaproveitamento em diversos arquivos;
 - Organização global do projeto;

MODELO DE EXEMPLO

- vamos utilizar o seguinte modelo conceitual em ambos SGBD's:

Agenda

idregistro

nome

endereço

email

telefone

MODELO DE EXEMPLO

- Modelo Lógico - MySQL

Agenda

idregistro: INTEGER

nome: VARCHAR(50)

endereço: VARCHAR(50)

email: VARCHAR(50)

telefone: VARCHAR(50)

```
CREATE TABLE Agenda (  
    idregistro          INTEGER NOT NULL auto_increment,  
    nome                VARCHAR(50) NOT NULL,  
    endereço            VARCHAR(50) NULL,  
    email               VARCHAR(50) NULL,  
    telefone            VARCHAR(50) NOT NULL,  
    PRIMARY KEY (idregistro)  
);
```

MODELO DE EXEMPLO

- Modelo Lógico - PostgreSQL

Agenda

| |
|-----------------------|
| idregistro: SERIAL |
| nome: VARCHAR(50) |
| endereco: VARCHAR(50) |
| email: VARCHAR(50) |
| telefone: VARCHAR(50) |

```
CREATE TABLE Agenda (  
    idregistro          SERIAL NOT NULL,  
    nome                VARCHAR(50) NOT NULL,  
    endereco            VARCHAR(50) NULL,  
    email               VARCHAR(50) NULL,  
    telefone            VARCHAR(50) NOT NULL,  
    PRIMARY KEY (idregistro)  
);
```


MYSQL - CONECTANDO-SE AO BANCO

- o primeiro comando deve estabelecer uma conexão com o banco de dados:

```
$recurso = mysql_connect ($server, $username , $password)
```

- \$server: é a localização do servidor de banco de dados, se o **servidor for local podemos utilizar 'localhost' ou '127.0.0.1'**;
- \$username: é o nome de usuário para acesso ao banco de dados, apesar de não aconselhável por motivos de **segurança, para o estudo vamos utilizar 'root'**;
- \$password: é a senha do usuário selecionado;
- Retorna um identificador, o recurso do bd.

MYSQL - SELECIONANDO UM BD

- depois de conectado devemos então selecionar um banco de dados:

```
mysql_select_db($dbname, $recurso)
```

- \$dbname: é o nome do banco de dados ao qual queremos conectar
- \$recurso: é o recurso criado pelo comando mysql_connect.

MYSQL - EXECUTANDO UM COMANDO SQL

- para executar um comando sql, utilizamos o comando:

```
$resultado = mysql_query($sql)
```

- valor retornado:
 - para comandos SELECT, SHOW, DESCRIBE ou EXPLAIN, mysql_query() retorna uma coleção de dados em caso de sucesso, ou FALSE em caso de falha.
 - para outros tipos de consultas SQL, UPDATE, DELETE, DROP, etc, mysql_query() retorna TRUE em caso de sucesso ou FALSE em caso de erro.

POSTGRESQL - CONECTANDO-SE AO BANCO

- o primeiro comando deve estabelecer uma conexão com o banco de dados:

```
pg_connect ( string $connection_string )
```

- o PostGreSQL utiliza uma string de conexão:

```
"host=localhost port=5432 dbname=agenda user=root password=***"
```

- note que na string de conexão do PostGreSQL já escolhemos com qual banco de dados vamos trabalhar;

POSTGRESQL - EXECUTANDO UM COMANDO SQL

- para executar um comando sql, utilizamos o comando:

```
$resultado = pg_query($sql)
```

- Valor retornado:
 - Para comandos SELECT, SHOW, DESCRIBE ou EXPLAIN, pg_query() retorna uma coleção de dados em caso de sucesso, ou FALSE em caso de falha.
 - Para outros tipos de consultas SQL, UPDATE, DELETE, DROP, etc, pg_query() retorna TRUE em caso de sucesso ou FALSE em caso de erro.

PERCORRENDO OS DADOS

- existem diversas maneiras de se percorrer o recurso retornado (em caso de coleção);
- é possível utilizar comando para transformar a coleção de dados em arrays com indexações numéricas, literais, ou até mesmo transformar um registro em um objeto;
- vamos utilizar os comandos mysql_fetch_array e pg_fetch_array que retornam um array indexado pelo nome do campo da tabela;

EXEMPLOS - CONSULTA

MySQL

```
...  
$resultado = mysql_query("SELECT * FROM Agenda");  
while($dados = mysql_fetch_array($resultado)) {  
    echo $dados['nome'];  
    echo "<br />";  
}
```

PostGreSQL

```
...  
$resultado = pg_query("SELECT * FROM Agenda");  
while($dados = pg_fetch_array($resultado)) {  
    echo $dados['nome'];  
    echo "<br />";  
}
```

EXEMPLOS - INSERÇÃO

MySQL

```
$nome = "Diogo";  
$endereco = "Av. 15 Novembro";  
$email = "diogo@globo.com";  
$telefone = "3564-6897";  
mysql_query("INSERT INTO Agenda (nome, endereco, email,  
telefone) VALUES ('$nome', '$endereco', '$email',  
'$telefone')");
```

PostgreSQL

```
$nome = "Diogo";  
$endereco = "Av. 15 Novembro";  
$email = "diogo@globo.com";  
$telefone = "3564-6897";  
pg_query("INSERT INTO Agenda (nome, endereco, email,  
telefone) VALUES ('$nome', '$endereco', '$email',  
'$telefone')");
```

CAPTURANDO O ERRO

- muitas vezes o banco de dados retorna um erro, se isso não for tratado não temos como saber o que aconteceu, para isso utiliza-se a seguinte sintaxe:

```
$resultado = mysql_query($sql) or die(mysql_error());
```

```
$resultado = pg_query($sql) or die(pg_last_error());
```

- or die(\$msg) garante que se ao falhar o comando de consulta, a execução do programa será finalizada com a \$msg;
- mysql_error() e pg_last_error() retornam o último erro ocorrido na conexão;

ATIVIDADE

- crie uma estrutura de controle de banco de dados utilizando os conceitos de programação orientada a objetos. Utilize as seguintes classes:
 - Connection: Classe responsável por fazer a conexão com o banco de dados;
 - DataBase: Classe responsável por executar consultas no banco de dados, utilize os métodos:
 - exec(\$sql) : executa um sql e retorna seu resultado se houver;
 - insert(\$tabela, array \$campos, array \$valores): insere um registro em \$tabela, utilizando os campos e valores passados como array;
 - update(\$tabela, array \$campos, array \$valores): atualiza um registro em \$tabela, utilizando os campos e valores passados como array;
 - delete(\$tabela, \$condicao): exclui um registro de \$tabela contemplando \$condicao;

ATIVIDADE

- utilizando a estrutura anterior, implemente um mini sistema de cadastro e consulta de pessoas na agenda passada como modelo.