

Sistema de Cadastro - Exemplo

PHP PDO (PHP Data Objects)





PDO em PHP

- Um módulo que está presente em praticamente todos os projetos é o **Sistema de Cadastro**.
- Seja de usuários, de produtos ou de qualquer outra coisa, quase sempre precisamos de cadastros.
- Neste material vamos desenvolver um simples sistema de cadastro, usando **PHP** e **MySQL**, utilizando a biblioteca PDO.



CRUD

- Vamos fazer um sistema de cadastro simples de usuários, com os dados:
 - nome,
 - email,
 - gênero (sexo) e
 - data de nascimento.
- Teremos uma tela de listagem de usuários, uma de cadastro, uma de edição e outra de remoção. O famoso CRUD.



Arquivos no projeto

- No final, vamos ficar com estes arquivos:
 - add.php: script que processa o cadastro de usuário
 - delete.php: script que processa a remoção de usuário
 - edit.php: script que processa a edição de usuário
 - form-add.php: formulário de cadastro de usuário
 - form-edit.php: formulário de edição de usuário
 - functions.php: arquivo de funções
 - index.php: listagem de usuários cadastrados
 - init.php: arquivo de inicialização (Bootstrapping)

Estrutura do Banco de Dados

- Nosso banco será bem simples, apenas com uma tabela de usuários.

```
CREATE TABLE users(  
  id INT UNSIGNED NOT NULL AUTO_INCREMENT, -- id  
  name VARCHAR(60) NOT NULL, -- nome  
  email VARCHAR(80) NOT NULL, -- email  
  gender ENUM('m', 'f') NOT NULL, -- gênero (masculino, feminino)  
  birthdate DATE NOT NULL, -- data de nascimento  
  PRIMARY KEY(id)  
) COLLATE=utf8_unicode_ci;
```





Arquivo de inicialização init.php

- O arquivo init.php define algumas configurações gerais do nosso projeto, como:
 - As credenciais de acesso ao banco de dados.
 - A definição da timezone
 - Habilitação de erros
- O arquivo de inicialização (*Bootstrapping*) deve ser chamado em todos os nossos scripts.
- Assim teremos um ambiente consistente e organizado, sempre com todas as mesmas configurações.

Nosso Bootstrapping

<?php

init.php

```
// constantes com as credenciais de acesso ao banco MySQL
define('DB_HOST', 'localhost');
define('DB_USER', 'root');
define('DB_PASS', '');
define('DB_NAME', 'cadastro');

// habilita todas as exibições de erros
ini_set('display_errors', true);
error_reporting(E_ALL);
date_default_timezone_set('America/Sao_Paulo');

// inclui o arquivo de funções
require_once 'functions.php';
```


Página index.php

```
1 <?php
2 require_once 'init.php';
3 // abre a conexão
4 $PDO = db_connect();
5
6 // SQL para contar o total de registros
7 // A biblioteca PDO possui o método rowCount(),
8 // mas ele pode ser impreciso.
9 // É recomendável usar a função COUNT da SQL
10 $sql_count = "SELECT COUNT(*) AS total FROM users ORDER BY name ASC";
11
12 // SQL para selecionar os registros
13 $sql = "SELECT id, name, email, gender, birthdate "
14       . "FROM users ORDER BY name ASC";
15
16 // conta o total de registros
17 $stmt_count = $PDO->prepare($sql_count);
18 $stmt_count->execute();
19 $total = $stmt_count->fetchColumn();
20
21 // seleciona os registros
22 $stmt = $PDO->prepare($sql);
23 $stmt->execute();
24 ?>
```


Página index.html

```
25      <!doctype html>
26      <html>
27      <head>
28          <meta charset="utf-8">
29          <title>Sistema de Cadastro</title>
30      </head>
31      <body>
32          <h1>Sistema de Cadastro</h1>
33          <p><a href="form-add.php">Adicionar Usuário</a></p>
34          <h2>Lista de Usuários</h2>
35          <p>Total de usuários: <?php echo $total ?></p>
36          <?php if ($total > 0): ?>
37              <table width="50%" border="1">
38                  <thead>
39                      <tr>
40                          <th>Nome</th>
41                          <th>Email</th>
42                          <th>Gênero</th>
43                          <th>Data de Nascimento</th>
44                          <th>Idade</th>
45                          <th>Ações</th>
46                      </tr>
47                  </thead>
48                  <tbody>
```

Página index.php

```
49 <?php while ($user = $stmt->fetch(PDO::FETCH_ASSOC)): ?>
50 <tr>
51 <td><?php echo $user['name'] ?></td>
52 <td><?php echo $user['email'] ?></td>
53 <td><?php echo ($user['gender'] == 'm') ? 'Masculino' : 'Feminino' ?></td>
54 <td><?php echo dateConvert($user['birthdate']) ?></td>
55 <td><?php echo calculateAge($user['birthdate']) ?> anos</td>
56 <td>
57 <a href="form-edit.php?id=<?php echo $user['id'] ?>">Editar</a>
58 <a href="delete.php?id=<?php echo $user['id'] ?>"
59     onclick="return confirm('Tem certeza de que deseja remover?');">
60     Remover
61 </a>
62 </td>
63 </tr>
64 <?php endwhile; ?>
65 </tbody>
66 </table>
67 <?php else: ?>
68 <p>Nenhum usuário registrado</p>
69 <?php endif; ?>
70 </body>
</html>
```

Baixe o arquivo txt como indicado pelo professor.



Resultado

Sistema de Cadastro

[Adicionar Usuário](#)

Lista de Usuários

Total de usuários: 1

Nome	Email	Gênero	Data de Nascimento	Idade	Ações
Rodrigo	rodrigo@gmail.com	Masculino	10/10/1977	39 anos	Editar Remover

Adicionar usuário: form-add.php

```
1  <?php
2  require 'init.php';
3  ?>
4  <!doctype html>
5  <html>
6      <head>
7          <meta charset="utf-8">
8          <title>Cadastro de Usuário</title>
9      </head>
10     <body>
11         <h1>Sistema de Cadastro</h1>
12         <h2>Cadastro de Usuário</h2>
13         <form action="add.php" method="post">
14             <label for="name">Nome: </label>
15             <br>
16             <input type="text" name="name" id="name">
17             <br><br>
18             <label for="email">Email: </label>
19             <br>
20             <input type="text" name="email" id="email">
21             <br><br>
```


Adicionar usuário: form- add.php

```
22      Gênero:  
23      <br>  
24      <input type="radio" name="gender" id="gener_m" value="m">  
25      <label for="gener_m">Masculino </label>  
26      <input type="radio" name="gender" id="gener_f" value="f">  
27      <label for="gener_f">Feminino </label>  
28      <br><br>  
29      <label for="birthdate">Data de Nascimento: </label>  
30      <br>  
31      <input type="text" name="birthdate" id="birthdate" placeholder="dd/mm/aa">  
32      <br><br>  
33      <input type="submit" value="Cadastrar">  
34  </form>  
35  </body>  
36  </html>
```

Resultado

Sistema de Cadastro

Cadastro de Usuário

Nome:

Email:

Gênero:

☐ Masculino ☐ Feminino

Data de Nascimento:

Cadastrar

Arquivo que trata o recebimento dos dados. add.php

```
1 <?php
2 require_once 'init.php';
3
4 // pega os dados do formulário
5 $name = isset($_POST['name']) ? $_POST['name'] : null;
6 $email = isset($_POST['email']) ? $_POST['email'] : null;
7 $gender = isset($_POST['gender']) ? $_POST['gender'] : null;
8 $birthdate = isset($_POST['birthdate']) ? $_POST['birthdate'] : null;
9
10 // validação (bem simples, só pra evitar dados vazios)
11 if (empty($name) || empty($email) || empty($gender) || empty($birthdate))
12 {
13     echo "Volte e preencha todos os campos";
14     exit;
15 }
16 // a data vem no formato dd/mm/YYYY
17 // então precisamos converter para YYYY-mm-dd
18 $isoDate = dateConvert($birthdate);
```

Arquivo que trata o recebimento dos dados. add.php

80%

20%

20%

```
19 // insere no banco
20 $PDO = db_connect();
21 $sql = "INSERT INTO users(name, email, gender, birthdate) VALUES(:name, :email, :gender, :birthdate)";
22 $stmt = $PDO->prepare($sql);
23 $stmt->bindParam(':name', $name);
24 $stmt->bindParam(':email', $email);
25 $stmt->bindParam(':gender', $gender);
26 $stmt->bindParam(':birthdate', $isoDate);
27
28 if ($stmt->execute())
29 {
30     header('Location: index.php');
31 }
32 else
33 {
34     echo "Erro ao cadastrar";
35     print_r($stmt->errorInfo());
36 }
```


Página para Edição dos Dados form-edit.php

```
1 <?php
2 require 'init.php';
3 // pega o ID da URL
4 $id = isset($_GET['id']) ? (int) $_GET['id'] : null;
5 // valida o ID
6 if (empty($id)) {
7     echo "ID para alteração não definido";
8     exit;
9 }
10 // busca os dados do usuário a ser editado
11 $PDO = db_connect();
12 $sql = "SELECT name, email, gender, birthdate FROM users WHERE id = :id";
13 $stmt = $PDO->prepare($sql);
14 $stmt->bindParam(':id', $id, PDO::PARAM_INT);
15 $stmt->execute();
16 $user = $stmt->fetch(PDO::FETCH_ASSOC);
17 // se o método fetch() não retornar um array, significa que o ID não corresponde
18 // a um usuário válido
19 if (!is_array($user)) {
20     echo "Nenhum usuário encontrado";
21     exit;
22 }
23 ?>
24 <!doctype html>
25 <html>
26 <head>
27     <meta charset="utf-8">
28     <title>Edição de Usuário</title>
29 </head>
```

Página para Edição dos Dados

form-edit.php

```
30 <body>
31 <h1>Sistema de Cadastro</h1>
32 <h2>Edição de Usuário</h2>
33 <form action="edit.php" method="post">
34 <label for="name">Nome: </label>
35 <br>
36 <input type="text" name="name" id="name" value="<?php echo $user['name'] ?>">
37 <br><br>
38 <label for="email">Email: </label>
39 <br>
40 <input type="text" name="email" id="email" value="<?php echo $user['email'] ?>">
41 <br><br>
42 Gênero:
43 <br>
44 <input type="radio" name="gender" id="gener_m" value="m" <?php if ($user['gender'] == 'm'): ?>
45 <br> <br> checked="checked" <?php endif; ?>>
46 <label for="gener_m">Masculino </label>
47 <input type="radio" name="gender" id="gener_f" value="f" <?php if ($user['gender'] == 'f'): ?>
48 <br> <br> checked="checked" <?php endif; ?>>
49 <label for="gener_f">Feminino </label>
50 <br><br>
51 <label for="birthdate">Data de Nascimento: </label>
52 <br>
53 <input type="text" name="birthdate" id="birthdate" placeholder="dd/mm/YYYY"
54 <br> <br> value="<?php echo dateConvert($user['birthdate']) ?>">
55 <br><br>
56 <input type="hidden" name="id" value="<?php echo $id ?>">
57 <input type="submit" value="Alterar">
58 </form>
59 </body>
60 </html>
```

Resultado

80%

20%

20%

Nome:

Rodrigo

Email:

rodrigo@gmail.com

Gênero:

☒ Masculino ☐ Feminino

Data de Nascimento:

Arquivo que trata o recebimento dos dados.

edit.php

```
1 <?php
2 require_once 'init.php';
3 // resgata os valores do formulário
4 $name = isset($_POST['name']) ? $_POST['name'] : null;
5 $email = isset($_POST['email']) ? $_POST['email'] : null;
6 $gender = isset($_POST['gender']) ? $_POST['gender'] : null;
7 $birthdate = isset($_POST['birthdate']) ? $_POST['birthdate'] : null;
8 $id = isset($_POST['id']) ? $_POST['id'] : null;
9 // validação (bem simples, mais uma vez)
10 if (empty($name) || empty($email) || empty($gender) || empty($birthdate))
11 {
12     echo "Volte e preencha todos os campos";
13     exit;
14 }
15 // a data vem no formato dd/mm/YYYY
16 // então precisamos converter para YYYY-mm-dd
17 $isoDate = dateConvert($birthdate);
18 // atualiza o banco
19 $PDO = db_connect();
```


Arquivo que trata o recebimento dos dados. edit.php

```
20 $sql = "UPDATE users SET name = :name, email = :email,"
21      . " gender = :gender, birthdate = :birthdate WHERE id = :id";
22 $stmt = $PDO->prepare($sql);
23 $stmt->bindParam(':name', $name);
24 $stmt->bindParam(':email', $email);
25 $stmt->bindParam(':gender', $gender);
26 $stmt->bindParam(':birthdate', $isoDate);
27 $stmt->bindParam(':id', $id, PDO::PARAM_INT);
28 if ($stmt->execute())
29 {
30     header('Location: index.php');
31 }
32 else
33 {
34     echo "Erro ao alterar";
35     print_r($stmt->errorInfo());
36 }
```

Arquivo para remoção delete.php

```
1 <?php
2 require_once 'init.php';
3 // pega o ID da URL
4 $id = isset($_GET['id']) ? $_GET['id'] : null;
5 // valida o ID
6 if (empty($id))
7 {
8     echo "ID não informado";
9     exit;
10 }
11 // remove do banco
12 $PDO = db_connect();
13 $sql = "DELETE FROM users WHERE id = :id";
14 $stmt = $PDO->prepare($sql);
15 $stmt->bindParam(':id', $id, PDO::PARAM_INT);
16 if ($stmt->execute())
17 {
18     header('Location: index.php');
19 }
20 else
21 {
22     echo "Erro ao remover";
23     print_r($stmt->errorInfo());
24 }
```



80% 20%
Pronto

- **Referências Bibliográficas**
 - BERALDO, Roberto; **Sistema de Cadastro com PHP e MySQL Usando PDO**, disponível em <http://blog.ultimatephp.com.br/sistema-de-cadastro-php-mysql-pdo/> acessado em <12/09/2017>.