

# **Tutorial de Criação de CRUD Simples em PHP com PDO**

- 1 - Introdução
- 2 - Testando
- 3 - Conexão com o banco de dados em conn.php
- 4 - Listando os registros no index.php
- 5 - Código para adicionar registro com add.php
- 6 - Agora o código e tela de edição
- 7 - Código do arquivo delete.php

## 1 - Introdução

CRUD – Representa uma estrutura que contém todas as operações básicas para gerenciar uma tabela em um banco de dados, facilitando seu gerenciamento: C – create (inserir), R – read (select), U – update (atualizar) e D – delete (excluir).

Este é um exemplo extremamente simples. Requer apenas um conhecimento básico do PHP para sua compreensão. Não estarei detalhando o código PHP, mas sua montagem para a construção de um CRUD simples.

Nada de CSS, apenas o CRUD básico com o PDO para mostrar suas funções básicas.

A intenção é a de quando estiver encontrando dificuldade com o uso de funções poder comentar ou excluir a função e substituir pelo respectivo código do CRUD deste exemplo.

## **2 - Testando**

Para um bom entendimento sugiro testar antes

Crie o banco de dados "cadastro" em seu MySQL. Aqui estou considerando que usa "root" com senha "root". Mas se usa sem senha remova a senha no conn.php.  
Copie toda a pasta para seu diretório web.

# Codificação

## 3 - Conexão com o banco de dados conn.php

As 5 primeiras linhas contém o script delimitador do PHP na primeira linha e as 4 variáveis do servidor de banco de dados nas 4 seguintes:

```
<?php
$host = '127.0.0.1'; // URL do servidor do MySQL, que também pode ser localhost (se usando local)
$db  = 'cadastro'; // nome do banco de dados
$user = 'root'; // nome do usuário do banco
$pass = 'root'; // senha do user
```

### As próximas 5 linhas configuram o PDO

```
$opt = [
    PDO::ATTR_ERRMODE          => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES => false,
];
```

O tipo de erro, o tipo de fetch e a emulação do prepares

### As próximas e finais linhas

Testa no try a criação da instância do PDO e armazena na variável \$pdo

```
try {
    $pdo = new PDO("mysql:host=$host;dbname=$db", $user, $pass, $opt);
    // echo 'Conectado para o banco de dados<br />';
    // Fechar a conexão com o banco
    // $pdo = null;
} catch(PDOException $e){
    // Caso o try não seja executado executa a linha abaixo
    echo $e->getMessage();
}
```

## 4 - Listando os registros no index.php

Ver a listagem valendo na tela

### CRUD Simples usando apenas PHP com PDO

[Novo Cliente](#)

ID	Nome	Email	Nascimento	CPF	Ação	
19	Adrian Ballard Peters	enim.Curabitur@faucibus.com	2012-07-13	50918748283	<a href="#">Editar</a>	<a href="#">Excluir</a>
3	Aladdin Curry French	eu.augue@eutemporerat.org	2012-10-28	10376915676	<a href="#">Editar</a>	<a href="#">Excluir</a>
21	Amelia Bland Barrera	id.ante@diemissim.org	2012-06-00	12106826368	<a href="#">Editar</a>	<a href="#">Excluir</a>

### Iniciando o HTML da página e o link para chamar a página add.php

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CRUD Simples</title>
</head>
<body>

<h2 align="center">CRUD Simples usando apenas PHP com PDO</h2>

<div align="center">
  <a href="add.php">Novo Cliente</a>
</div>
<br>
```

### Veja o código seguinte

```
<?php
// Incluindo o arquivo da conexão usando o require_once
require_once 'conn.php';

try{
  // Outro trycatch, agora para executar uma query
  $stm = $pdo->query("SELECT * FROM clientes order by nome");
  // Armazenar em $executa o resultado da query
  $executa = $stm->execute();
  ?>
  <!-- Criando os títulos da tabela -->
  <table border="2" align="center">
    <tr><td><b>ID</b></td><td><b>Nome</b></td><td><b>Email</b></td><td>
    <b>Nascimento</b></td><td><b>CPF</b></td><td colspan="2" align="center">Ação</td></tr>
```

### Mais uma parte do index.php

Esta agora é para mostrar os registros no index.php

```
<?php
// Checando se a variável contém registros. Caso contenha realiza o código abaixo
if($executa){
    // Para recuperar um OBJETO utilize PDO::FETCH_OBJ
    while($reg = $stm->fetch()){
?>
        <tr><td><?=$reg->id?></td>
        <td><?=$reg->nome?></td>
        <td><?=$reg->email?></td>
        <td><?=$reg->data_nasc?></td>
        <td><?=$reg->cpf?></td>
```

### As duas linhas abaixo criar os links Editar e Excluir na index.php

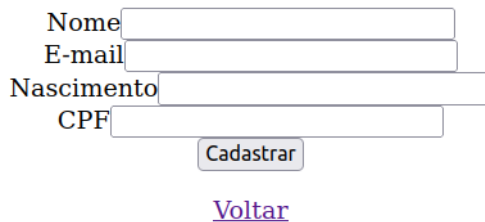
```
        <td><a href="update.php?id=<?=$reg->id?>">Editar</a></td>
        <td><a href="delete.php?id=<?=$reg->id?>">Excluir</a></td></tr>
```

Abaixo as linhas finais sem comentários pois os respectivos código já foram comentados:

```
<?php
}
print '</table>';
}else{
    echo 'Erro ao inserir os dados';
}
} catch(PDOException $e){
    echo $e->getMessage();
}
?>
</body>
</html>
```

## 5 - Código para adicionar registro com add.php

Na index.php existe o link Novo Cliente que chama add.php



Iniciando com um form simples:

```
<div align="center">
<form method="post" action="add.php">
Nome<input type="text" name="nome"><br>
E-mail<input type="text" name="email"><br>
Nascimento<input type="text" name="data_nasc"><br>
CPF<input type="text" name="cpf"><br>
<input name="enviar" type="submit" value="Cadastrar">
</form>
</div>
```

### Agora com um código iniciando

Que cria variáveis recebendo os valores das mesmas do form

```
<?php
// Requer antes incluir a conexão
require_once 'conn.php';

// Testa se o submit foi clicado e as variáveis tenham conteúdo
if(isset($_POST['enviar'])) {
    $nome = $_POST['nome'];
    $email = $_POST['email'];
    $data_nasc = $_POST['data_nasc'];
    $cpf = $_POST['cpf'];
```

**O trecho abaixo usa o prepare para deixar o código mais seguro com os respectivos bind**

```
try {
    $stmt = $pdo->prepare("INSERT INTO clientes(nome,email,data_nasc,cpf) VALUES
(?, ?, ?, ?)");
    $stmt->bindParam(1, $nome, PDO::PARAM_STR);
    $stmt->bindParam(2, $email, PDO::PARAM_STR);
    $stmt->bindParam(3, $data_nasc, PDO::PARAM_STR);
    $stmt->bindParam(4, $cpf, PDO::PARAM_STR);
    $executa = $stmt->execute();
```

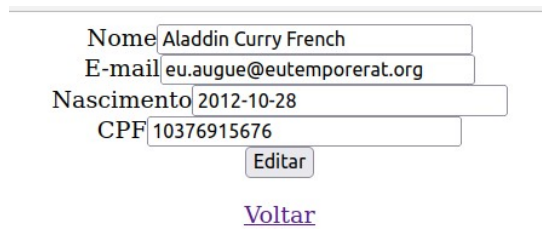
## Final do arquivo

```
if($executa){
    echo 'Dados inseridos com sucesso';
    header('location: index.php');
}
else{
    echo 'Erro ao inserir os dados';
}
}
catch(PDOException $e){
    echo $e->getMessage();
}
}
?>
<div align="center"><a href="index.php">Voltar</a></div>
```



## 6 - Agora o código e tela de edição update.php

Antes veja a tela



Nome Aladdin Curry French  
E-mail eu.augue@eutemporerat.org  
Nascimento 2012-10-28  
CPF 10376915676  
  
[Voltar](#)

Antes usamos o select com o prepare para maior segurança, para encher o form com o registros apontado.

Veja que ao início é incluído o conn.php e recebe o id do registro atual do form do index.php.

```
<?php
require_once 'conn.php';

$Id=$_GET['id'];

$sth = $pdo->prepare("SELECT id, nome,email,data_nasc,cpf from clientes WHERE id = :id");
// No select e no delete basta um bindValue
$sth->bindValue(':id', $Id, PDO::PARAM_STR);
$sth->execute();
```

### Próximo trecho de código

Abaixo o fetch é convertido para objetos e armazenado em \$reg  
E são criadas variáveis recebendo cada campo da tabela

```
$reg = $sth->fetch(PDO::FETCH_OBJ);
$nome = $reg->nome;
$email = $reg->email;
$data_nasc = $reg->data_nasc;
$cpf = $reg->cpf;
?>
```

Logo abaixo o form e em cada campo recebendo o valor dos campos do registro recebido

```
<div align="center">
<form method="post" action="">
Nome<input type="text" name="nome" value="<?=$nome?>"><br>
E-mail<input type="text" name="email" value="<?=$email?>"><br>
Nascimento<input type="text" name="data_nasc" value="<?=$data_nasc?>"><br>
CPF<input type="text" name="cpf" value="<?=$cpf?>"><br>
<input name="id" type="hidden" value="<?=$id?>">
<input name="enviar" type="submit" value="Editar">
</form>
</div>
```

### Código final do arquivo:

```
<?php

if(isset($_POST['enviar'])){
    $id = $_POST['id'];
    $nome = $_POST['nome'];
    $email = $_POST['email'];
    $data_nasc = $_POST['data_nasc'];
    $cpf = $_POST['cpf'];

    $sql = "UPDATE clientes SET nome = :nome, email=:email, data_nasc=:data_nasc, cpf=:cpf
WHERE id = :id";
    $sth = $pdo->prepare($sql);
    $sth->bindParam(':id', $_POST['id'], PDO::PARAM_INT);
    $sth->bindParam(':nome', $_POST['nome'], PDO::PARAM_INT);
    $sth->bindParam(':email', $_POST['email'], PDO::PARAM_INT);
    $sth->bindParam(':data_nasc', $_POST['data_nasc'], PDO::PARAM_INT);
    $sth->bindParam(':cpf', $_POST['cpf'], PDO::PARAM_INT);

    if($sth->execute()){
        print "<script>alert('Registro alterado com sucesso!');location='index.php';</script>";
    }else{
        print "Erro ao editar o registro!<br><br>";
    }
}
?>

<div align="center"><a href="index.php">Voltar</a></div>
```

Similar a outros trechos.

## 7 - Código do arquivo delete.php

Como o código é similar a vários trechos já abordados não comentarei. Mas pode deixar um comentário caso fique alguma dúvida.

```
<?php
require_once 'conn.php';

$id=$_GET['id'];

$stmt = $pdo->prepare("SELECT id,nome,email,data_nasc,cpf from clientes WHERE id = :id");
$stmt->bindValue(':id', $id, PDO::PARAM_STR);
$stmt->execute();

$reg = $stmt->fetch(PDO::FETCH_OBJ);
$nome = $reg->nome;
$email = $reg->email;
$data_nasc = $reg->data_nasc;
$cpf = $reg->cpf;

?>
<h3 align="center">Tem certeza de que deseja excluir o registro <?=$id?>?</h3>
<div align="center">
Nome: <?=$nome?><br>
E-mail: <?=$email?><br>
Nascimento: <?=$data_nasc?><br>
CPF: <?=$cpf?><br>

<form method="post" action="">
<input name="id" type="hidden" value="<?=$id?>">
<input name="enviar" type="submit" value="Excluir!">
</form>

</div>

<?php
if(isset($_POST['enviar'])){
$id = $_POST['id'];
$sql = "DELETE FROM clientes WHERE id = :id";
$stmt = $pdo->prepare($sql);
$stmt->bindParam(':id', $id, PDO::PARAM_INT);
if( $stmt->execute()){
    print "<script>alert('Registro excluído com sucesso!');location='index.php';</script>";
} else{
    print "Erro ao excluir o registro!<br><br>";
}
}
?>
<div align="center"><a href="index.php">Voltar</a></div>
```