**Salvando dados de um formulário em um banco de dados utilizando PHP**

**Otávio A. A. Silva¹**

**¹ Faculdade de Engenharia da Computação e Telecomunicações - Instituto de Tecnologia - Universidade Federal do Pará**

**Cidade Universitária “Prof. José da Silveira Netto” – Campus II (Profissional) Belém/PA, Brasil, 66075-110**

tavioalves@gmail.com

***Abstract.*** *This meta-article aims to show the need to save the data to fill a form in a database using PHP, and the communication between the server and the client will use Ajax.*

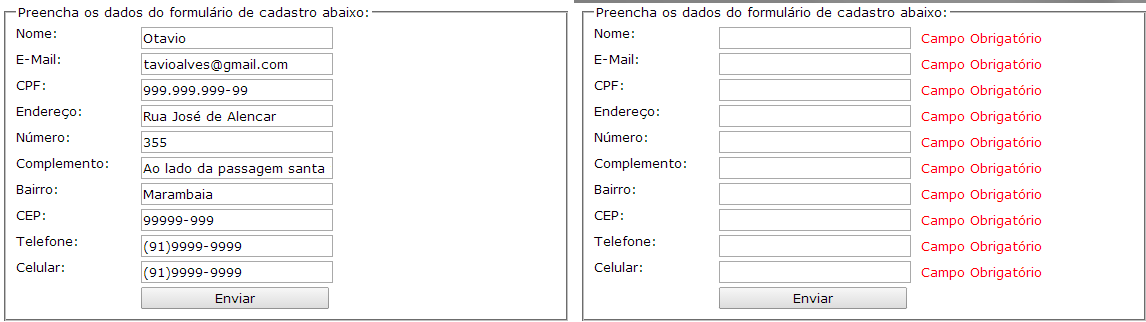
***Resumo.*** *Este meta-artigo tem como objetivo mostrar o os passos necessário para se salvar os dados de preenchimento de um formulário em um banco de dados utilizando PHP, sendo que a comunicação entre o servidor e o cliente utilizará o Ajax.*

**1. Introdução**

A utilização de formulários na Internet atualmente é sem dúvida alguma muito importante, já que é a partir dela que pode-se coletar informações sobre os usuários que utilizam alguma aplicação, como o cadastro de usuários, pesquisas de campo ou a confirmação de pagamento de uma loja online e etc. Por tanto, o desenvolvimento de aplicações que possam lidar com esses dados e serem manipuladas com segurança, ou seja guardas, é vital para eficácia da aplicação. Sendo criando o banco de dados, estabelecendo a comunicação entre o banco e o cliente, e etc. Veremos esses passos nas próximas seções desse artigo.

**2. Criação do banco de dados e tabela**

Sabendo que já existe um formulário previamente criado, com mascara e validações já presentes, como mostra na figura 1. Necessitamos de um local no qual vai guardar todas as informações vinda dessa aplicação. Sendo assim, será necessário a criação de um banco de dados, sendo que bancos de dados podem ser definidos como: coleções organizadas de dados que se relacionam de forma a criar algum sentido(Informação) e dar mais eficiência durante uma pesquisa ou estudo.[1]

Fig. 1 – Formulário

Para isso vamos precisar acessar em nosso servidor, o programa *phpMyAdmin*, onde ele é um aplicativo web desenvolvido em PHP para administração do MySQL pela Internet. A partir deste sistema é possível criar e remover bases de dados, criar, remover e alterar tabelas, inserir, remover e editar campos, executar códigos SQL e manipular campos chaves.[2]

Acessamos o *phpMyAdmin* através do console do XAMPP, caso seja necessário a inserção do usuário e senha, por padrão, o usuário com privilégios para criação do banco é o “*root*”  e a senha é vazia, ou seja é “ “.

Após o login, necessiamos criar o banco de dados no qual irá conter a tabela que terá por função receber os dados enviados pelo usuário. Sendo assim, criaremos o banco de dados com o nome “bancodedados”. Podemos ver isso claramente no destaque na figura 2.

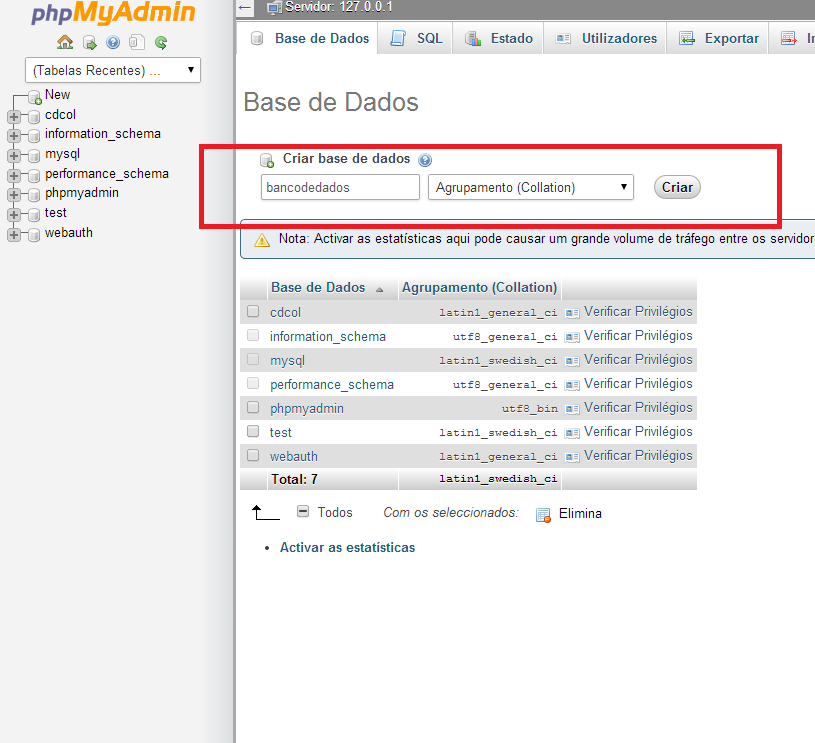


Figura 2. – Criação do banco de dados

Com a criação do banco, podemos então criar a tabela, para isso podemos utilizar manualmente a inserção do código para a criação da tabela, através do SQL ou criar coluna por coluna, colocando suas características. Em nosso caso, faremos isso de forma manual, através do código SQL, para isso utilizamos o código abaixo:

**CREATE** **TABLE** formulario(

**id** **INT** AUTO\_INCREMENT,

nome **VARCHAR**(50),

endereco **VARCHAR**(50),

numero **VARCHAR**(50),

complemento **VARCHAR**(50),

bairro **VARCHAR**(50),

cep **VARCHAR**(50),

email **VARCHAR**(50),

cpf **VARCHAR**(50),

telefone **VARCHAR**(50),

celular **VARCHAR**(50),

**PRIMARY** **KEY**(**id**)

)

Podemos ver então, que nossa tabela se chamará “formulário” e possuirá todos os campos que estão em nosso formulário mostrado na figura 1, exceto o campo “id” que tem por função mostrar a quantidade de formulários já enviados e para melhor manuseio das tabelas pelo desenvolvedor.

**3. Utilização do Ajax para comunicação cliente-servidor**

Com a criação do banco e da tabela, é necessário que exista um meio de comunicação entre o cliente e o servidor. Para tal função, utilizaremos o Ajax, basicamente o mesmo tem função de tornar páginas Web mais interativas com o usuário, utilizando-se de solicitações assíncronas de informações.[3]

Por tanto é necessário que seja inserido em nossa página o script com tal função, referenciamos o mesmo na pasta JS, através da tag:

<**script type="text/javascript" language="**javascript**" src="js/ajaxpost.js"**></**script**>

O mesmo possui as seguintes informações:

$(document).ready(**function**()

{

$("#enviar").click(**function**()

{

**var** nome = $("#nome").val();

**var** endereco = $("#endereco").val();

**var** numero = $("#numero").val();

**var** complemento = $("#complemento").val();

**var** bairro = $("#bairro").val();

**var** cep = $("#cep").val();

**var** email = $("#email").val();

**var** cpf = $("#cpf").val();

**var** telefone = $("#telefone").val();

**var** celular = $("#celular").val();

$.post('enviar.php', {nome: nome, endereco: endereco, numero: numero, complemento: complemento, bairro: bairro, cep: cep, email: email, cpf: cpf, telefone: telefone, celular: celular},

**function**(data)

{

$("#resposta").html(data);

}

, "html");

});

});

No momento que a página estiver pronta, ao se clicar no botão enviar, os campos das variáveis em questão serão salvas e enviadas para o “enviar.php”, através do $.post. Sendo que o arquivo enviar.php, tem como função tratar os dados do Ajax e enviar para o banco de dados.

**4. Criação do processo de envio de informações para o banco de dados**

Para que seja possível a inclusão dos dados do formulário no banco de dados, primeiro precisamos estabelecer uma conexão com o banco. Sendo assim será necessário que existam referências do nosso servidor, usuário que logara e sua senha, além das referências do banco e tabela. Sendo assim, a primeira parte do nosso código tem a seguinte cara:

<?php

//Abrindo Conexao com o banco de dados

$conexao = mysql\_pconnect("localhost","root","") or **die** (mysql\_error());

$banco = mysql\_select\_db("bancodedados");

?>

Após a referencia do nosso servidor (“localhost”), usuário (“root”) e senha(“”), podemos então selecionar o banco de dados(“bancodedados”). Por tanto agora será necessário receber os valores enviados pelo Ajax. Fazendo com que o nosso código receba mais uma parcela de informações vista abaixo.

<?php

$nome = $\_POST["nome"];

$endereco = $\_POST["endereco"];

$numero = $\_POST["numero"];

$complemento = $\_POST["complemento"];

$bairro = $\_POST["bairro"];

$cep = $\_POST["cep"];

$email = $\_POST["email"];

$cpf = $\_POST["cpf"];

$telefone = $\_POST["telefone"];

$celular = $\_POST["celular"];

//Abrindo Conexao com o banco de dados

$conexao = mysql\_pconnect("localhost","root","") or **die** (mysql\_error());

$banco = mysql\_select\_db("bancodedados");

?>

Para que os dados possam ser tratados com segurança e que possa diminuir o risco de SQL Injection, utilizamos o “mysql\_real\_escape\_string” para isso.

//Utilizando o mysql\_real\_escape\_string voce se protege o seu código contra SQL Injection.

$nome = mysql\_real\_escape\_string($nome);

$endereco = mysql\_real\_escape\_string($endereco);

$numero = mysql\_real\_escape\_string($numero);

$complemento = mysql\_real\_escape\_string($complemento);

$bairro = mysql\_real\_escape\_string($bairro);

$cep = mysql\_real\_escape\_string($cep);

$email = mysql\_real\_escape\_string($email);

$cpf = mysql\_real\_escape\_string($cpf);

$telefone = mysql\_real\_escape\_string($telefone);

$celular = mysql\_real\_escape\_string($celular);

Com as devidas ações de segurança tomadas, podemos então executar o comando SQL para inserir os dados em nossa tabela e retornar informações ao usuário caso ocorreu tudo tranquilamente ou houve um erro.

$insert = mysql\_query("insert into formulario (nome,endereco,numero,complemento,bairro,cep,email,cpf,telefone,celular) values ('{$nome}','{$endereco}','{$numero}','{$complemento}','{$bairro}','{$cep}','{$email}','{$cpf}','{$telefone}','{$celular}')");

mysql\_close($conexao);

**if**($insert) {

**print** "Formulario enviado com sucesso!";

}**else** {

**print** "Erro ao enviar formulario!";

}

Com isso nossa aplicação está feita e totalmente funcional. Acessando o localhost, recebemos o formulário, e após a inserção de dados e clicar no botão de enviar podemos ver a mensagem de sucesso de envio do formulário como pode ser visto na figura 3.

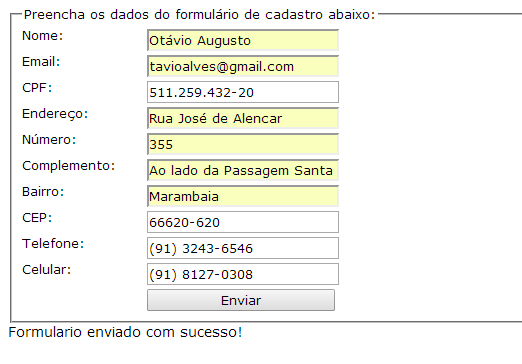


Figura 3. – Envio com Sucesso

Além disso, podemos verificar de fato que os dados estão sendo mesmo enviados ao servidor, acessando o phpMyAdmin e verificando se os a tabela “formulario”. A figura 4 mostra que os dados foram enviados com sucesso.

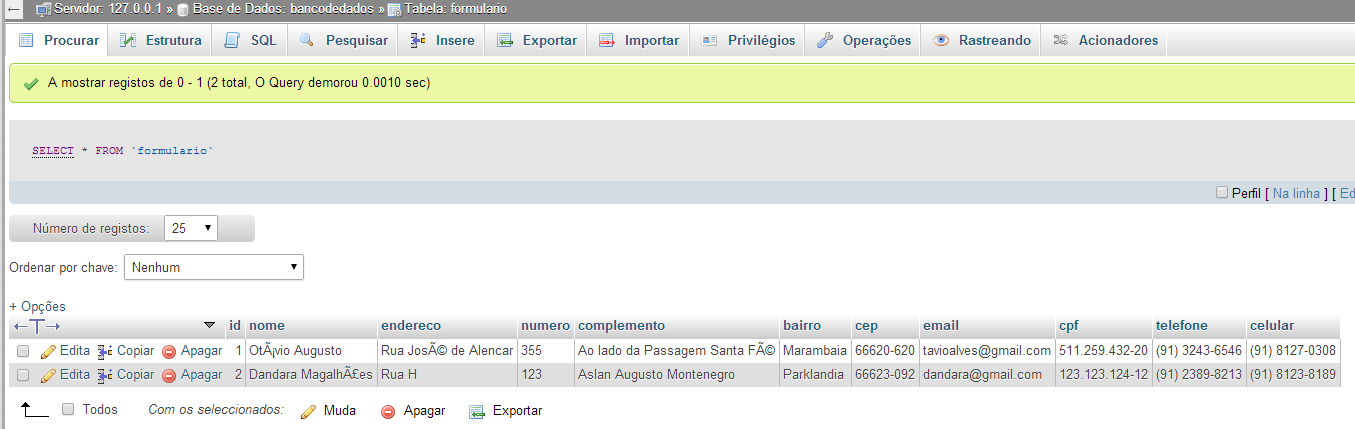


Figura 4. – Tabela formulário com dados dos usuários

**5. Conclusão**

A utilização de formulários por parte de aplicações Web nunca deixarão de existir. Já que é uma maneira bastante eficaz para se obter informações corretas de um usuário, porém é necessário que existam meios de ser tratar tal informação, a utilização dos bancos de dados e Ajax é perfeita para isso, tanto para pequenas aplicações como a proposta aqui tanto para as grandes.

**6. Referência**

Banco de Dados - http://pt.wikipedia.org/wiki/Banco\_de\_dados

phpMyAdmin - http://pt.wikipedia.org/wiki/PhpMyAdmin

Ajax - http://pt.wikipedia.org/wiki/AJAX\_(programa%C3%A7%C3%A3o)

jQuery Ajax - http://api.jquery.com/jQuery.ajax