



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
SÃO PAULO  
Campus Birigui

# Disciplina: Linguagem de Programação Web II



## PHP & MySQL

Prof. Murilo Varges da Silva

**email:** [murilo.varges@gmail.com](mailto:murilo.varges@gmail.com) / [murilo.varges@ifsp.edu.br](mailto:murilo.varges@ifsp.edu.br)



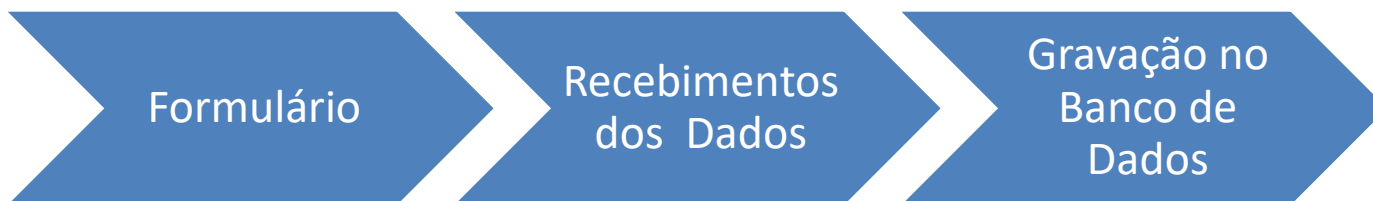
# PhpMyAdmin

O PhpMyAdmin é a ferramenta que utilizaremos para facilitar a criação e manutenção do nosso banco de dados. Para testar se ele está funcionando corretamente, faça o seguinte:

- ✓ Abra o navegador de internet e digite o seguinte endereço: <http://localhost/phpmyadmin/>

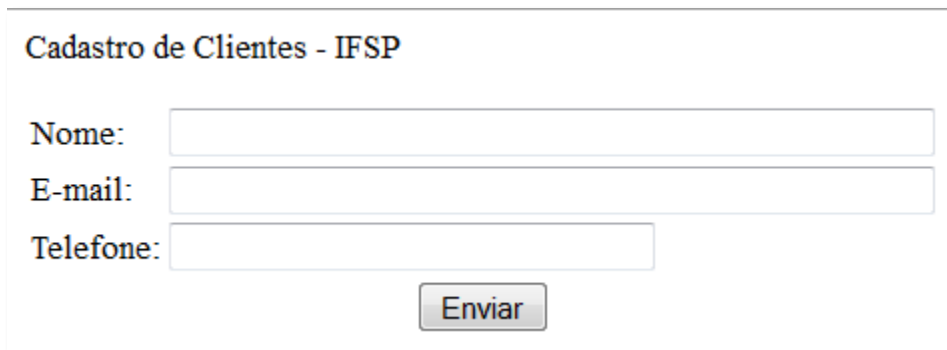


## Fluxo php + MySql



# Criando Aplicação

- ✓ Criar pasta para a aplicação chamada “**projeto01**” dentro de C:\Xampp\htdocs;
- ✓ Criar um formulário chamado “**cadastro\_usuario.htm**”, este formulário html deverá conter os seguintes campos e botão enviar:
  - Nome\_usuario: tamanho 50;
  - Email\_usuario: tamanho 50;
  - Fone\_usuario: tamanho 30.



Cadastro de Clientes - IFSP

Nome:

E-mail:

Telefone:



# Criando Aplicação

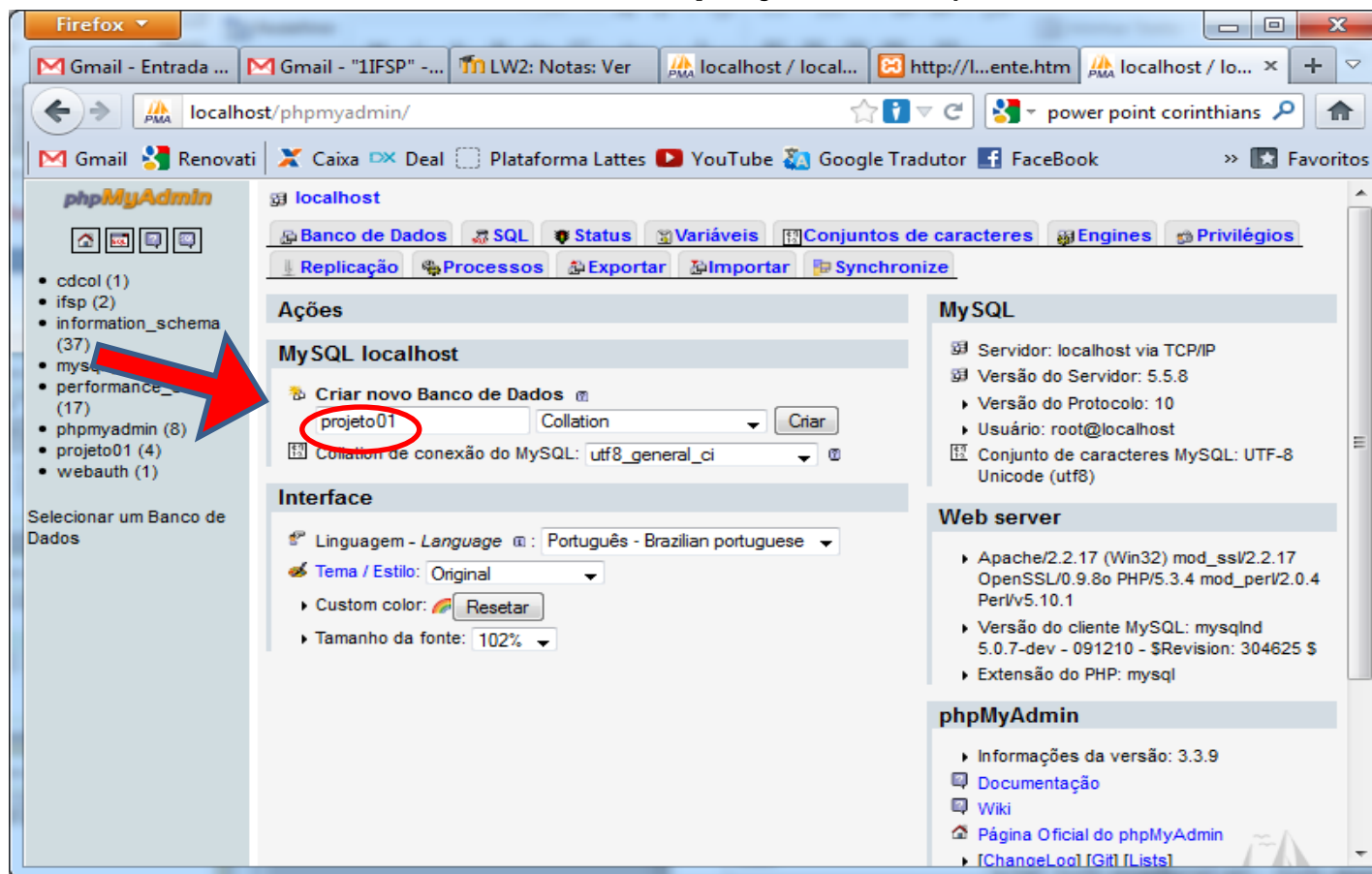
Continuação

- ✓ Criar um script php chamado “**cadastro\_usuario.php**” que receberá as informações da formulário:

```
1 <?
2 $nome_usuario = $_POST['nome_usuario'];
3 $email_usuario = $_POST['email_usuario'];
4 $fone_usuario = $_POST['fone_usuario'];
5 echo "<P>Nome do usuario: ".$nome_usuario."<BR>";
6 echo "E-mail: ".$email_usuario."<BR>";
7 echo "Telefone: ".$fone_usuario."</P>";
8 ?>
```

# Criando um Banco de dados no PhpMyAdmin

Vamos agora criar o banco de dados que armazenará os dados preenchidos no formulário, acesse o PhpMyAdmin, na tela que se apresenta, digite o nome do banco no campo “**Criar novo Banco de Dados**”, vamos colocar o nome do banco “**projeto01**”, após colocar o nome clicar no botão criar.



# Criando um Banco de dados no PhpMyAdmin

(Continuação)

Pronto, o seu já banco está criado. Agora vamos criar as tabelas do banco. Inicialmente nosso banco terá apenas uma única tabela, a qual chamaremos de “usuario” (sem acentos).

Esta tabela terá 4 campos, conforme os campos que criamos no formulário e mais o campo “id\_usuario” que é o número de registro único do usuário no banco. Sendo assim os campos serão os seguintes:

- id\_usuario: int(4)
- nome\_usuario: varchar(50);
- email\_usuario: varchar(50);
- telefone\_usuario: varchar(30);

Os campos “varchar(50)” indicam que serão armazenados textos com até 50 caracteres.



# Criando um Banco de dados no PhpMyAdmin

(Continuação)

Conforme a tela abaixo digite o nome da tabela e a quantidade de campos que ela terá:

Aqui está o nome do banco de dados que voce acabou de criar.

Digite aqui a quantidade de campos que terá sua tabela

Digite aqui o nome da tabela a ser criada.

Clique o botão “executar” que está à sua direita.





# Criando um Banco de dados no PhpMyAdmin

(Continuação)

Na tela seguinte crie os campos do seu banco, conforme a figura abaixo, obedecendo os nomes dos campos corretos:

id\_usuario, nome\_usuario, e\_mail\_usuario e telefone\_usuario.

phpMyAdmin

Servidor: localhost Banco de Dados: dgestruturado Tabela: usuario

Campo	Tipo	Tamanho/Definir	Collation	Atributos	Nulo	Padrão	Extra
id_usuario	INT	4			not null		auto_increment
nome_usuario	VARCHAR	50			not null		
e_mail_usuario	VARCHAR	50			not null		
telefone_usuario	VARCHAR	30			not null		

Comentários da tabela:

Storage Engine: InnoDB Collation:

Salvar Adicionar 1 campo(s) Executar

<sup>1</sup> Se um tipo de campo é "enum" ou "set", por favor entre valores usando este formato: "a","b","c"

Digite nesta coluna, o nome dos campos.

Digite nesta coluna, o tipo dos campos.

Tamanho 50, indica que só caberão 50 caracteres

Auto\_increment, indica que é uma numeração sequencial automática

Chave-primária -Indica que se trata de um identificador único.

Observe que o campo id\_usuario é **chave primária** e **auto\_increment**. Se voce esqueceu de definir estes detalhes, acesse o recurso de alteração do campo, clicando no ícone em forma de lápis que está na mesma linha do nome do campo.

Depois de definir tudo, clique no botão **Salvar**.



# Criando um Banco de dados no PhpMyAdmin

(Continuação)

Depois que você clicou no botão “Salvar”, o PhpMyAdmin, lhe apresenta a tela abaixo:

Nome do seu banco de dados

Nome da tabela que voce acabou de criar.

Aqui também estão no nome do banco e da tabela.

Aqui ele demonstra como seria os comandos em sql para criar uma tabela.

Esta é a estrutura da sua tabela.

phpMyAdmin

Servidor: localhost ▶ Banco de Dados: dgestruturado ▶ Tabela: usuario

Visualizar Estrutura SQL Procurar Inserir Exportar Importar Operações Limpar Eliminar

Tabela "dgestruturado"."usuario" foi criado.

consulta SQL:

```
CREATE TABLE `dgestruturado`.`usuario` (  
  `id_usuario` INT(4) NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
  `nome_usuario` VARCHAR(50) NOT NULL ,  
  `e_mail_usuario` VARCHAR(50) NOT NULL ,  
  `telefone_usuario` VARCHAR(30) NOT NULL  
) ENGINE = INNODB
```

Campos	Tipo	Collation	Atributos	Nulo	Padrão	Extra	Ação
<input type="checkbox"/> id_usuario	int(4)			Não		auto_increment	
<input type="checkbox"/> nome_usuario	varchar(50)	latin1_swedish_ci		Não			
<input type="checkbox"/> e_mail_usuario	varchar(50)	latin1_swedish_ci		Não			
<input type="checkbox"/> telefone_usuario	varchar(30)	latin1_swedish_ci		Não			

Visualização para impressão Propri estrutura da tabela

Adicionar 1 campo(s) No final da tabela No início da tabela Depois id\_usuario Executar

Índices: 0					Uso de espaço		Estatísticas do registros	
Nome chave	Tipo	Cardinalidade	Ação	Campo	Tipo	Uso	Comandos	Valor
PRIMARY	PRIMARY	0		id_usuario	Dados	Bytes		
Criar um índice em 1 colunas Executar					Índice	Bytes		
					Total	Bytes		



# Criando um Banco de dados no PhpMyAdmin

(Continuação)

Depois que você clicou no botão “Salvar”, o PhpMyAdmin, lhe apresenta a tela abaixo:

Nome do seu banco de dados

Nome da tabela que voce acabou de criar.

Aqui também estão no nome do banco e da tabela.

Aqui ele demonstra como seria os comandos em sql para criar uma tabela.

Esta é a estrutura da sua tabela.

phpMyAdmin

Servidor: localhost > Banco de Dados: dgestruturado > Tabela: usuario

Visualizar Estrutura SQL Procurar Inserir Exportar Importar Operações Limpar Eliminar

Tabela "dgestruturado"."usuario" foi criado.

consulta SQL:

```
CREATE TABLE `dgestruturado`.`usuario` (  
  `id_usuario` INT(4) NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
  `nome_usuario` VARCHAR(50) NOT NULL ,  
  `e_mail_usuario` VARCHAR(50) NOT NULL ,  
  `telefone_usuario` VARCHAR(30) NOT NULL  
) ENGINE = INNODB
```

Campos	Tipo	Collation	Atributos	Nulo	Padrão	Extra	Ação
<input type="checkbox"/> id_usuario	int(4)			Não		auto_increment	
<input type="checkbox"/> nome_usuario	varchar(50)	latin1_swedish_ci		Não			
<input type="checkbox"/> e_mail_usuario	varchar(50)	latin1_swedish_ci		Não			
<input type="checkbox"/> telefone_usuario	varchar(30)	latin1_swedish_ci		Não			

Visualização para impressão Propri estrutura da tabela

Adicionar 1 campo(s) No final da tabela No início da tabela Depois id\_usuario Executar

Índices: 0					Uso de espaço		Estatísticas do registros	
Nome chave	Tipo	Cardinalidade	Ação	Campo	Tipo	Uso	Comandos	Valor
PRIMARY	PRIMARY	0		id_usuario	Dados	Bytes		
Criar um índice em 1 colunas Executar					Índice	Bytes		
					Total	Bytes		



# Criando um Banco de dados no PhpMyAdmin

(Continuação)

Agora vamos verificar se voce fez tudo corretamente.

- Clique no nome da tabela, que está aparecendo no seu lado esquerdo.
- Clique, em seguida, na aba "Exportar".
- E então clique no botão "executar" que aparece no canto inferior direito.

Na tela seguinte, será mostrado o código em SQL de criação da tabela **"usuario"**, conforme este a seguir:

```
CREATE TABLE `usuario` (  
  `id_usuario` int(4) NOT NULL auto_increment,  
  `nome_usuario` varchar(50) NOT NULL,  
  `e_mail_usuario` varchar(50) NOT NULL,  
  `telefone_usuario` varchar(30) NOT NULL,  
  PRIMARY KEY (`id_usuario`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

Código SQL  
gerado pelo  
phpMyAdmin.  
Após criarmos a  
tabela **usuario**

Se o seu código SQL estiver igualzinho a este então voce está de parabéns nesta tarefa, podendo seguir em frente tranquilamente. Se não está igual, faça as devidas correções, para evitar problemas posteriores.

# Conexão com Banco de Dados

Para conectar-se ao banco é necessário informar ao MySQL quem é o usuário que está se conectando, qual a senha e qual banco pretende conectar-se, levando em consideração que no MySQL pode-se ter vários bancos de dados. Na verdade o MySQL é um gerenciador de banco de dados.

O ideal é que se crie no MySQL um usuário para cada banco de dados criado, mas no nosso caso iremos utilizar o usuário **root**, que é o usuário principal, e não utilizaremos senha por enquanto. Sendo assim, nosso código PHP ficará da seguinte forma.

Salvar arquivo como “**conexao.php**” na pasta “**projeto01**”

```
1 <?php
2 $hostname = "localhost";
3 $database = "projeto01";
4 $username = "root";
5 $password = "";
6 $con = mysql_connect($hostname, $username, $password)
7 or die(mysql_error()."Erro ao tentar conectar-se ao banco");
8 mysql_select_db($database, $con);
9 ?>
```



# Conexão com Banco de Dados

(Continuação)

Para testar se a conexão com o Banco de Dados MySQL está funcionando basta abrir o navegador de internet e digitar o seguinte caminho:

**<http://localhost/projeto01/conexao.php>**

- ✓ Se não aparecer nenhuma mensagem de erro está tudo correto.
- x Se não funcionar, vai aparecer uma mensagem de erro no navegador, tente descobrir onde está o erro.





# Criando Aplicação

Continuação

Agora que já temos o html da aplicação, o banco de dados criado e a conexão com o banco de dados funcionando, vamos retornar ao nosso arquivo “**cadastro\_usuario.php**” que já havia sido criado conforme consta no slide 5.

Neste arquivo inclua o seguinte comando na primeira linha de código, logo abaixo da instrução <?

```
include ('conexao.php');
```

O comando “include” serve para incluir um arquivo dentro do outro. No caso estamos incluindo o arquivo “**conexao.php**” dentro do arquivo “**cadastro\_usuario.php**”,



# Criando Aplicação

Continuação

Agora faça o teste para verificar se está tudo correto até aqui, abra o navegador de internet e digite o endereço:

[http://localhost/projeto01/cadastro\\_usuario.htm](http://localhost/projeto01/cadastro_usuario.htm)

- ✓ Preencha o formulário e clique no botão “**Enviar**”, se não der erro prossiga.
- x Se não funcionar, vai aparecer uma mensagem de erro no navegador, tente descobrir onde está o erro.





# Criando Aplicação

Continuação

A utilização do comando “include” é uma forma de não ter que ficar repetindo um mesmo código PHP várias vezes. Por exemplo, quando inserimos o arquivo “conexao.php” dentro do arquivo “cadastro\_usuario.php” seria a mesma coisa que copiar e colar o código do arquivo “conexao.php” dentro do arquivo “cadastro\_usuario.php”. Se você simplesmente digitasse o código de conexão dentro do arquivo “cadastro\_usuario.php” também funcionaria, porém, se você tiver que utilizar os comandos de conexão em um outro arquivo, teria que digitá-los novamente.



# Iniciando os primeiros comandos SQL no PHP

## Inserindo dados

Lembre-se que já criamos um banco de dados chamado “**projeto01**” e uma tabela chamada “**usuario**”, que possui os seguintes campos: `id_usuario`, `nome_usuario`, `email_usuario` e `fone_usuario`, logo para se fazer a inserção de dados em uma tabela do banco de dados precisamos saber em quais campos desejamos inserir os dados.

```
INSERT INTO usuario (nome_usuario, email_usuario, fone_usuario)
VALUES ('Murilo Vargês', 'murilo.vargês@gmail.com', '3643-1160')
```

# Iniciando os primeiros comandos SQL no PHP

(Continuação)

## Explicando o comando INSERT

- Observe que os comando SQL coloquei em letras maiúscula, isto não é necessário, é apenas por questão de melhor visualizar os comandos SQL. Recomendo que façam sempre assim, por simples questão de organização e melhor visualização do seu código. Lembre-se que, quando melhor estiver organizado o seu código, mais fácil será identificar erros e problemas no seu programa.
- Logo após o “INSERT INTO” vem o nome da tabela que receberá os dados. No nosso caso é a tabela “usuario”. E logo após o nome da tabela vem os nomes dos campos que receberão dados. Se eu fosse inserir apenas o nome da pessoa e não quisesse inserir o telefone, eu não colocaria o nome do campo “telefone\_usuario”.

# Iniciando os primeiros comandos SQL no PHP

(Continuação)

## Explicando o comando INSERT

Após o nome dos campos vem o comando VALUES, para informarmos os valores que serão inseridos nos campos. Observe que como são dois campos que receberão letras (Strings), é necessário portanto, que os valores estejam entre aspas. Observe também que não foi necessário colocar o campo “id\_usuario”, este é um campo de identificação única da tabela “usuario” ele é um campo que recebe numeração automaticamente. Reveja o script de criação da tabela que está slide 12, observe que na linha do “id\_usuario” há um parâmetro chamado “**auto\_increment**”, este parâmetro indica que este campo receberá um valor numérico e sequencial automaticamente. Este campo é chamado de **CHAVE PRIMÁRIA** ou **PRIMARY KEY**.

# Iniciando os primeiros comandos SQL no PHP

(Continuação)

## Explicando o comando INSERT

### Detalhe importante:

A Chave primária neste caso é um campo com numeração automática, que tem uma particularidade: você verá que ao inserir o primeiro registro na tabela “usuario”, este campo (id\_usuario) receberá o valor “1”, e ao inserir mais um registro, este campo receberá automaticamente o valor “2”, no entanto se logo depois você excluir o registro 2 do banco de dados, e em seguida inserir um novo registro, este novo registro receberá o número “3”, mesmo não existindo mais o registro número 2. Isso acontece por questão de segurança do próprio banco de dados, de modo a evitar de todas as formas, a existência de dois campos com um mesmo número de CHAVE PRIMÁRIA.

# Iniciando os primeiros comandos SQL no PHP

(Continuação)

## Utilizando o comando INSERT

Agora que já entendemos como o comando INSERT funciona abra o PhpMyAdmin <http://localhost/phpmyadmin>, selecione o banco de dados que criamos “**projeto01**”, basta clicar no nome do banco que aparece à direita da página e depois clique aba em “**SQL**”, que fica na parte superior da página.

Feito isso vamos digitar o comando INSERT INTO, que já vimos no slide 18 que segue abaixo:

```
INSERT INTO usuario (nome_usuario, email_usuario, fone_usuario)  
VALUES ('Murilo Vargês', 'murilo.vargês@gmail.com', '3643-1160')
```

Depois de digitar clique no botão executar e observe se o comando foi executado com sucesso, para conferir se os dados foram inseridos clique no botão “**Visualizar**”.



# Iniciando os primeiros comandos SQL no PHP

(Continuação)

Aqui começa a ficar um pouco mais complicado, mas ainda não é tão difícil assim. Não é nada que você não consiga fazer. Tendo calma, muita atenção e paciência consegue-se tudo aqui.

Seu arquivo “cadastro\_usuario.php” já deverá estar então com o comando “include 'conexao.php';”, que faz a inclusão do arquivo de conexão. A partir daí iremos inserir as seguintes linhas de comando no final do arquivo “cadastro\_usuario.php”:

```
$sql="INSERT INTO usuario(nome_usuario, email_usuario, fone_usuario)
      VALUES ('".$nome_usuario."', '".$email_usuario."', '".$fone_usuario."')";
$result = mysql_query($sql);
if ($result)
    echo "Dados cadastrados com sucesso!";
else
    | echo "Erro ao tentar cadastrar dados no banco!";
```



# Iniciando os primeiros comandos SQL no PHP

(Continuação)

## Explicando o código:

Estamos utilizando aí, duas variáveis: **\$sql** e **\$result**. Lembre-se que, nomes de variáveis sempre iniciam com o sinal “\$”. Pode-se atribuir qualquer nome às variáveis, mas, por questão de padronização e por uma melhor organização, sempre utilizamos os nomes \$sql e \$result para utilização de comandos de banco de dados.

Observe que a variável **\$sql** recebeu como valor um comando SQL de inserção de dados, que, se observar bem, vai ver que é o mesmo comando que utilizamos lá no PhpMyAdmin.

Compare este comando SQL com os comandos SQLs que aparecem lá no PhpMyAdmin e verá que são praticamente iguais.





# Iniciando os primeiros comandos SQL no PHP

(Continuação)

## Explicando o código:

Observe também que as variáveis que aparecem no comando `INSERT INTO`, logo depois da palavra “VALUES”, estão entre aspas simples. No comando SQL, os valores a serem inseridos no banco tem que estar entre aspas simples.

A variável `$result` pega o conteúdo da variável `$sql` e envia para o banco de dados utilizando o comando `mysql_query()`.

Existem várias outras formas de se fazer inserção de dados no banco utilizando PHP, esta é apenas uma delas e acredito que é a mais simples.



# Consultando Dados

## LISTANDO OS DADOS QUE FORAM INSERIDOS NO BANCO

Primeiramente vamos ver como fazer para listar todos os registros cadastrados no banco, depois veremos como listar apenas um registro específico ou como criar o formulário para alteração dos dados.

Como eu já disse anteriormente, é recomendável que tenhamos uma página para cada função do nosso sistema. Então, se já temos a página que inclui os dados no banco de dados, vamos criar agora a página que vai listar estes dados em ordem alfabética.



# Consultando Dados

(Continuação)

Vamos criar um arquivo  
chamado  
“listar\_usuarios.php”

```
1 <?php
2 include ('conexao.php');
3 $sql="SELECT * FROM usuario";
4 $result = mysql_query($sql);
5 $row = mysql_fetch_array($result);
6 ?>
7 <html>
8   <body>
9     <h2 align="center">Consulta de usuários</h2>
10    <table align="center" border="1" width="500">
11      <tr>
12        <td>Código</td>
13        <td>Nome</td>
14        <td>E-mail</td>
15        <td>Telefone</td>
16      </tr>
17      <?php
18        do {
19          echo "<tr>";
20          echo "<td>".$row['id_usuario']. "</td>";
21          echo "<td>".$row['nome_usuario']. "</a></td>";
22          echo "<td>".$row['email_usuario']. "</td>";
23          echo "<td>".$row['telefone_usuario']. "</td>";
24          echo "</tr>";
25        }while ($row = mysql_fetch_array($result));
26      ?>
27    </table>
28    <a href="index.php">voltar</a>
29  </body>
30 </html>
```



# Página Principal

Vamos criar um arquivo chamado “index.php”

```
1 <html>
2     <body>
3         <p><h3 align="Center">Página Inicial</h3></p>
4         <hr align="tr">
5         <u><b>Opções</b></u>
6         <p><a href="cadastro_usuario.htm">Cadastrar</a></BR>
7         <p><a href="listar_usuario.php">Listar</a></p>
8     </body>
9 </html> |
```

Pronto, feito isso enfeite a página como quiser, o importante são os links que criamos (< a href > ).

Agora para acessar seu sistema basta acessar o seguinte link>

**<http://localhost/projeto01>**



# Página Principal

(Continuação)

Ao abrir o link O PHP irá abrir automaticamente a página ***“index.php”***, todo programa em PHP que você desenvolver, ou todo site que você criar, sempre terá que ter um arquivo chamado ***“index.php”*** ou ***“index.html”***. Este arquivo é sempre o primeiro arquivo a ser aberto por um site ou sistema de internet.

Ao abrir esta página no seu browser, verá que aparecerá uma tela com os dois links que criamos: um para acessar o formulário de cadastro e outro para acessar a lista de usuários cadastrados.



# Alterando Dados

Abra o arquivo **“cadastro\_usuario.htm”** e salve-o com outro nome (clique em salvar como... e digite **“altera\_usuario.php”**).

Pronto! O arquivo que você tem aberto na sua tela agora é o **“altera\_usuario.php”**. Certo?

Neste arquivo faremos algumas modificações. No slide seguinte está o código, com as modificações a serem feitas.



# Alterando Dados

(Continuação)

```
1 <?php
2 include('conexao.php');
3 $id_usuario=$_GET['id_usuario'];
4 $sql="SELECT * FROM usuario WHERE id_usuario='$id_usuario'";
5 $result = mysql_query($sql);
6 $row = mysql_fetch_array($result);
7 ?>
8 <html>
9 <body>
10 <p>Cadastro de Clientes</p>
11 <form name="Cadastro de usuario" method="post" action="altera_usuario_exe.php">
12 <table>
13 <tr>
14 <td>Nome:</td>
15 <td><input name="nome_usuario" type="text" size="50" maxlength="50" value="<?php echo $row['nome_usuario']?>"> </td>
16 </tr>
17 <tr>
18 <td>E-mail:</td>
19 <td><input name="email_usuario" type="text" size="50" maxlength="50" value="<?php echo $row['email_usuario']?>"> </td>
20 </tr>
21 <tr>
22 <td>Fone:</td>
23 <td><input name="telefone_usuario" type="text" size="20" maxlength="20" value="<?php echo $row['telefone_usuario']?>"> </td>
24 </tr>
25 <tr>
26 <td colspan="2"><input name="id_usuario" type="hidden" value="<?php echo $row['id_usuario']?>">
27 <td colspan="2" align="center"><input type="submit" name="Submit" value="Alterar"></td>
28 </tr>
29 </table>
30 </form>
31 </body>
32 </html>
```



# Alterando Dados

(Continuação)

Abra o arquivo “*listar\_usuarios.php*”, na linha 21, onde está o seguinte:

```
echo "<td>".$row['nome_usuario'].</td>";
```

Nesta linha iremos criar um link para, quando clicarmos no nome do usuário, o sistema deverá abrir a página de alteração de dados “*altera\_usuario.php*”.

Para fazer isso, modifique essa linha para:

```
echo "<td><a href='altera_usuario.php?id_usuario=".$row['id_usuario']."'>".$row['nome_usuario'].</a></td>";
```





# Alterando Dados

(Continuação)

Criar um arquivo chamado “*altera\_usuario\_exe.php*” o início deste arquivo vai ser idêntico ao início do arquivo “*cadastra\_usuario.php*”.

O conteúdo dele será o seguinte:

```
1 <?php
2 include ('conexao.php');
3 $id_usuario=$_POST['id_usuario'];
4 $nome_usuario=$_POST['nome_usuario'];
5 $email_usuario=$_POST['email_usuario'];
6 $telefone_usuario=$_POST['telefone_usuario'];
7 echo "<P>Nome do usuario: ".$nome_usuario."<BR></P>";
8
9 $sql="UPDATE usuario SET
10     nome_usuario= '".$nome_usuario."',
11     email_usuario= '".$email_usuario."',
12     telefone_usuario= '".$telefone_usuario.'"
13     WHERE id_usuario= '".$id_usuario.'" " ;
14 $result = mysql_query($sql);
15 if ($result)
16     echo "Dados alterados com sucesso! <br>";
17 else
18     echo "Erro ao tentar alterar dados no banco! <br>";
19 ?>
20 <a href="index.php">voltar</a>
```

Comando para alteração de dados no MySQL.

Comando WHERE é para informar que apenas um registro será alterado.



# Excluindo Dados

Abra o arquivo “*listar\_usuarios.php*”.

Depois de linha 15 ( <td>Telefone</td>) inclua a linha: <td>Excluir</td>

Depois da linha 22 (echo "<td>".\$row['telefone\_usuario']."</td>";) crie a seguinte linha:

```
echo "<td><a href='excluir_usuario.php?id_usuario=".$row['id_usuario']."'>Excluir</td>";
```



# Excluindo Dados

(Continuação)

Agora crie o arquivo “*excluir\_usuario.php*”, com o seguinte código:

```
1 <?php
2 include ('conexao.php');
3     //id_usuario vem do arquivo listar_usuario.php
4     $id_usuario=$_GET['id_usuario'];
5     $sql="DELETE FROM usuario WHERE id_usuario='".$id_usuario.'" ";
6     $result = mysql_query($sql);
7     if ($result)
8         echo "Registro excluído com sucesso! <br> ";
9     else
10         echo "Erro ao tentar excluir registro no banco! <br>";
11 ?>
12 <a href="index.php">voltar</a>
```