

## Conexão Java/MySQL – parte 3 (botões manutenção dados e pesquisa)

1 – Botões de ação para efetuar a “Manutenção dos dados”: novo, cadastrar, alterar e excluir:

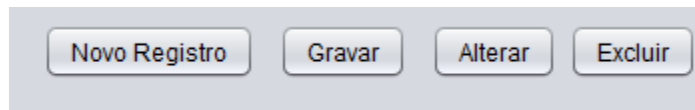


Figura 1 - Botões para Manutenção dos registros

### a) Botão Novo / Limpar:

Função: Limpar as caixas de texto para novas entradas de dados.

```
419 |         txtCod.setText(""); // limpa a caixa de texto em questão
420 |         txtNome.setText("");
421 |         txtNasc.setText("");
422 |         txtFone.setText("");
423 |         txtEmail.setText("");
424 |         txtCod.requestFocus(); // posiciona o cursor neste campo para digitação
```

Figura 2 - Código do botão Novo

### b) Botão Gravar:

Função: Incluir na tabela um novo registro a partir dos dados digitados nas caixas de texto:

```
340 |         String nome = txtNome.getText();
341 |         String data_nasc = txtNasc.getText();
342 |         String telefone = txtFone.getText();
343 |         String email = txtEmail.getText();
344 |
345 |         try {
346 |             String insert_sql="insert into tbclientes (nome,telefone, email, dt_nasc) values ('" + nome + "','" + telefone + "','" + email + "
347 |             con_cliente.statement.executeUpdate(insert_sql);
348 |             JOptionPane.showMessageDialog(null,"Gravação realizada com sucesso!!", "Mensagem do Programa",JOptionPane.INFORMATION_MESSAGE);
349 |
350 |             con_cliente.executaSQL("select * from tbclientes order by cod");
351 |             preencherTabela();
352 |
353 |         }catch(SQLException errosql){
354 |             JOptionPane.showMessageDialog(null,"\n Erro na gravação :\n "+errosql,"Mensagem do Programa",JOptionPane.INFORMATION_MESSAGE);
355 |         }
```

Figura 3 - Código do botão de Cadastro

### Linha 346 completa:

String insert\_sql="insert into tbclientes (nome,telefone, email, dt\_nasc) values ('" + nome + "','" + telefone + "','" + email + "','" + data\_nasc + "')";

- **Linhas 340 a 343:** declara as variáveis e atribui a elas o valor de cada caixa de texto respectivamente, através do método “getText()”.
- **Linha 346:** define o comando SQL de inserção, onde são informados nome da tabela, campos, e, na sequência as variáveis resultantes das caixas de texto, respectivamente.
- **Linha 347:** executa (no ambiente Java - virtualmente) o comando de inserção.
- **Linha 350:** executa (no ambiente Java - virtualmente) o comando de seleção dos dados da tabela.

- **Linha 351:** executa o método “preencherTabela”, responsável por apresentar dados atualizados na Jtable.

### c) Botão Alterar:

**Função:** Alterar na tabela os dados de um registro a partir dos dados digitados nas caixas de texto. Para encontrar o registro na tabela será usado o campo “código”, onde acontecerá um filtro e as alterações só ocorrerão quando o SQL encontrar na tabela o registro que tenha o código coincidente.

```
359 String nome = txtNome.getText();
360 String data_nasc = txtNasc.getText();
361 String telefone = txtFone.getText();
362 String email = txtEmail.getText();
363 String sql;
364 String msg="";
365
366 try {
367     if(txtCod.getText().equals("")){
368         sql="insert into tbclientes (nome,telefone, email, dt_nasc) values ('" + nome + "','" + telefone + "','" + email + "','" + da
369         msg="Gravação de um novo registro";
370     }else{
371         sql="update tbclientes set nome='" + nome + "',telefone='" + telefone + "', email='" + email + "', dt_nasc='" + data_nasc + "
372         msg="Alteração de registro";
373     }
374
375     con_cliente.statement.executeUpdate(sql);
376     JOptionPane.showMessageDialog(null,"Gravação realizada com sucesso!!","Mensagem do Programa",JOptionPane.INFORMATION_MESSAGE);
377
378     con_cliente.executaSQL("select * from tbclientes order by cod");
379     preencherTabela();
380
381 } catch (SQLException errosql) {
382     JOptionPane.showMessageDialog(null,"\n Erro na gravação :\n "+errosql,"Mensagem do Programa",JOptionPane.INFORMATION_MESSAGE);
383 }
```

Figura 4 - Código do botão da Alteração

### Linha 371 completa:

```
sql="update tbclientes set nome='" + nome + "',telefone='" + telefone + "', email='" + email + "',
dt_nasc='" + data_nasc + "' where cod = " + txtCod.getText();
```

Esta rotina é bem parecida com a anterior:

- **Linhas 359 a 362:** declara as variáveis e atribui a elas o valor de cada caixa de texto respectivamente, através do método “getText()”.

- **Linha 367,** possui um if porque se o usuário não digitar nenhum código na caixa de texto, então este botão fará uma inclusão e não alteração.

- **Linha 371:** define o comando SQL de alteração, onde é informado o nome da tabela, e, após o “set” estão sendo feitas atribuições para cada campo com sua respectiva variável, extraída das caixas de texto. É nesta hora que os atributos passam a ter o novo valor.

- **Linha 375:** executa (no ambiente Java - virtualmente) o comando de inserção e/ou alteração.

- **Linha 378:** executa (no ambiente Java - virtualmente) o comando de seleção dos dados da tabela.

- **Linha 379:** executa o método “preencherTabela”, responsável por apresentar dados na Jtable.

#### d) Botão Excluir:

**Função:** Excluir na tabela um registro a partir do código da caixa de texto. Para encontrar o registro na tabela será usado o campo “código”, onde acontecerá um filtro e a exclusão só ocorrerá quando o SQL encontrar na tabela o registro que tenha o código coincidente.

```
354 String sql="";
355
356 try {
357     int resposta = JOptionPane.showConfirmDialog(rootPane, "Deseja excluir o registro: ", "Confirmar Exclusão", JOptionPane.YES_NO_OPTION, 3);
358     if (resposta==0){
359         sql = "delete from tblclientes where cod = " + txtCod.getText();
360         int excluir = con_cliente.statement.executeUpdate(sql);
361         if (excluir==1){
362             JOptionPane.showMessageDialog(null, "Exclusão realizada com sucesso!!", "Mensagem do Programa", JOptionPane.INFORMATION_MESSAGE);
363             con_cliente.executaSQL("select * from tblclientes order by cod");
364             con_cliente.resultset.first();
365             preencherTabela();
366             posicionarRegistro();
367         }
368         else{
369             JOptionPane.showMessageDialog(null, "Operação cancelada pelo usuário!!", "Mensagem do Programa", JOptionPane.INFORMATION_MESSAGE);
370         }
371     }
372 } catch (SQLException excecacao){
373     JOptionPane.showMessageDialog(null, "Erro na exclusão: "+excecacao, "Mensagem do Programa", JOptionPane.INFORMATION_MESSAGE);
374 }
```

*Figura 5 - Código do botão da Exclusão*

Aqui nesta rotina, será feita a pergunta ao usuário se realmente “deseja excluir”, se sim, então a exclusão será oficializada.

- **Linha 358:** executa (no ambiente Java - virtualmente) o comando de deletar registros, onde o parâmetro para o filtro de registros será o conteúdo da caixa de texto do código.

A sequência é igual as rotinas anteriores.

## 2 – Pesquisa de um determinado registro:

**Função:** Filtrar os registros da JTable, mostrando somente aqueles que atendem as teclas digitadas nesta caixa de texto.

#### a) Desenhar no formulário, abaixo da tabela, estes objetos:

A imagem mostra um formulário de pesquisa. À esquerda, há o rótulo "Pesquisar pelo nome do Cliente::". À direita dele, há uma caixa de texto vazia para o usuário digitar o nome. À direita da caixa de texto, há um botão com o texto "Pesquisar".

*Figura 6 – Pesquisar Cliente*

**Exemplo 1:** Sem filtro, ou seja, sem digitação na caixa de texto (apresenta todos registros):

Código	Nome	Data Nascim...	Telefone	Email
1	Antonio F. Cruz	17/03/1939	1111-1111	antonio@antonio.com.br
2	Edneia R F	27/03/1969	2222-2222	edneia@edneia.com.br
3	Gustavo Alameda	27/05/2000	3333-3333	hustavo@gustavo.com.br
4	Ana Maria da Silva	01/01/2001	4444-4444	ana@aaa.com.br
5	Bento Gonçalves	02/02/2002	5555-5555	bento@bbb.com.br
6	Guilherme Souza e Silva	01/01/2001	6666-6666	guigui@gui.com
8	Edneia R F	27/03/1969	2222-2222	edneia@edneia.com.br

Pesquisa por nome do Cliente:

Figura 7 - Tabela sem filtro de pesquisa

**Exemplo 2:** Com filtro, ou seja, com digitação da letra “g” na caixa de texto (apresenta somente os registros cujo nome comece com a letra “g”:

Código	Nome	Data Nascim...	Telefone	Email
3	Gustavo Alameda	27/05/2000	3333-3333	hustavo@gustavo.com.br
6	Guilherme Souza e Silva	01/01/2001	6666-6666	guigui@gui.com
10	Gustavo Alameda hhhh	27/05/2000	3333-3333	hustavo@gustavo.com.br

Pesquisa por nome do Cliente:

Figura 8- Tabela com filtro através da pesquisa da letra G

b) Acione o evento “addActionListener” da caixa de texto:

```

bPesq.addActionListener( new ActionListener() {
public void actionPerformed(ActionEvent e) {
435     try{
436         String pesquisa = "select * from tbclientes where nome like '" + txt_pesquisa.getText() + "%'";
437         con_cliente.executaSQL(pesquisa);
438
439         if(con_cliente.resultset.first()){
440             preencherTabela();
441         }
442         else{
443             JOptionPane.showMessageDialog(null, "\n Não existe dados com este parâmetro!!", "Mensagem do Programa", JOI
444         }
445     }catch(SQLException errosql){
446         JOptionPane.showMessageDialog(null, "\n Os dados digitados não foram localizados!! : \n " + errosql, "Mensagem de
447     }

```

Figura 9 - Código do botão "Pesquisar"

Este evento executa o código sempre que um novo caractere for digitado. Então a cada letra teclada a pesquisa na tabela é feita através do filtro da linha 436:

**where nome like ''' + txt\_pesquisa.getText() + "%';**

O percentual significa que “qualquer complemento” será colocado a(s) letra(s) digitadas. Então digitado a letra “G”, o comando filtrou todos nomes começados com “G”.

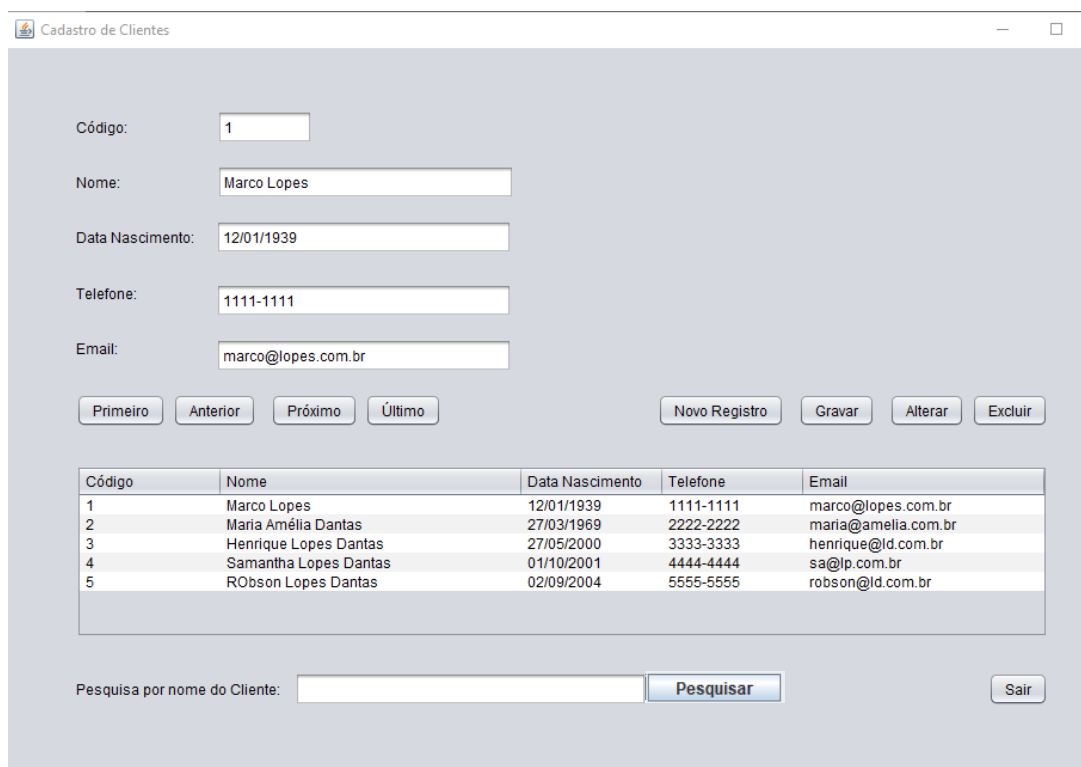
- Linha 443 completa:

```
JOptionPane.showMessageDialog(null, "\n Não existe dados com este parâmetro!!", "Mensagem do Programa", JOptionPane.INFORMATION_MESSAGE);
```

- Linha 446 completa:

```
JOptionPane.showMessageDialog(null, "\n Os dados digitados não foram localizados!! :\n "+errosql, "Mensagem do Programa", JOptionPane.INFORMATION_MESSAGE);
```

### 3 – Formulário finalizado:



Código	Nome	Data Nascimento	Telefone	Email
1	Marco Lopes	12/01/1939	1111-1111	marco@lopes.com.br
2	Maria Amélia Dantas	27/03/1969	2222-2222	maria@amelia.com.br
3	Henrique Lopes Dantas	27/05/2000	3333-3333	henrique@ld.com.br
4	Samantha Lopes Dantas	01/10/2001	4444-4444	sa@lp.com.br
5	RObson Lopes Dantas	02/09/2004	5555-5555	robson@ld.com.br

**Figura 10- Formulário Completo**

Agora cabe a você deixar o visual mais bonito, para isto, use a criatividade.