

Conexão Java/MySQL – parte 2

(preencher a tabela e as caixas de texto, navegação registros)

1 - O método **PreencherTabela** servirá para carregar com dados da tabela a Jtable:

a) Importar estas bibliotecas:

```
import javax.swing.table.DefaultTableModel;  
import java.sql.*;
```

```
1  
2 package controle;  
3  
4 import conexao.Conexao; // importar do package a classe  
5 import javax.swing.JOptionPane;  
6 import javax.swing.table.DefaultTableModel; // para reconhecimento da Jtable  
7 import java.sql.*; // para reconhecimento dos comandos SQL  
8  
9 public class frmTela extends javax.swing.JFrame {  
10
```

Figura 1 - Importação das bibliotecas

b) incluir no construtor:

```
con_cliente.executaSQL("select * from tbclientes order by cod");  
preencherTabela();
```

```
13 public frmTela() {  
14     initComponents();  
15     con_cliente = new Conexao(); // inicialização do objeto como instância  
16     con_cliente.conecta(); // chama o método que conecta  
17     con_cliente.executaSQL("select * from tbclientes order by cod");  
18     preencherTabela();  
19 }
```

Figura 2 - Implementação do construtor da classe

c) digitar o método, antes do método main:

```
138 public void preencherTabela() {  
139     tblClientes.getColumnModel().getColumn(0).setPreferredWidth(4);  
140     tblClientes.getColumnModel().getColumn(1).setPreferredWidth(150);  
141     tblClientes.getColumnModel().getColumn(2).setPreferredWidth(11);  
142     tblClientes.getColumnModel().getColumn(3).setPreferredWidth(14);  
143     tblClientes.getColumnModel().getColumn(4).setPreferredWidth(100);  
144  
145     DefaultTableModel modelo = (DefaultTableModel) tblClientes.getModel();  
146     modelo.setNumRows(0);  
147  
148     try {  
149         con_cliente.resultset.beforeFirst();  
150         while(con_cliente.resultset.next()){  
151             modelo.addRow(new Object[] {  
152                 con_cliente.resultset.getString("cod"), con_cliente.resultset.getSt:  
153             });  
154         }  
155     } catch (SQLException erro) {  
156         JOptionPane.showMessageDialog(null, "\n Erro ao listar dados da tabela!! :\n");  
157     }  
158 }  
159  
160 public static void main(String args[]) {
```

Figura 3 - Método preencherTabela()

Linha 152 completa:

```
con_cliente.resultset.getString("cod"),con_cliente.resultset.getString("nome"),con_cliente.resultset.get  
String("dt_nasc"),con_cliente.resultset.getString("telefone"), con_cliente.resultset.getString("email")
```

Obs: o que está dentro dos parênteses e aspas são os campos da tabela no MySQL, portanto devem ser escritos conforme foram criados no bd.

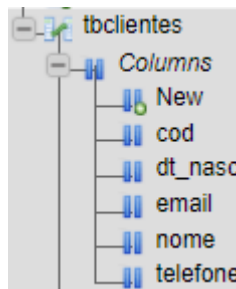


Figura 4 - Campos da tabela tblclientes do MySQL

Linha 156 completa:

```
JOptionPane.showMessageDialog(null,"\\n Erro ao listar dados da tabela!! :\\n "+erro,"Mensagem do  
Programa",JOptionPane.INFORMATION_MESSAGE);
```

Salve e execute:

A imagem mostra uma janela de software intitulada 'Cadastro de Clientes'. No topo, há uma barra de título com o ícone de uma pasta e o texto 'Cadastro de Clientes'. Abaixo, há uma interface com vários campos de entrada rotulados: 'Código:', 'Nome:', 'Data Nascimento:', 'Telefone:' e 'Email:'. Cada campo tem um campo de texto correspondente. Abaixo dos campos, há uma tabela com cinco colunas: 'Código', 'Nome', 'Data Nascime...', 'Telefone' e 'Email'. A tabela contém seis linhas de dados.

Código	Nome	Data Nascime...	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

Figura 5 - Execução do Projeto

2 - O método `posicionarRegistro()` servirá para posicionar no 1º registro da tabela, ou seja, na linha do cliente Antonio, já o método **`mostrar_Dados()`** irá carregar com dados da tabela as caixas de texto:

a) incluir no construtor:

```
posicionarRegistro();
```

```
tblClientes.setAutoCreateRowSorter(true); // ativa a classificação ordenada da tabela
```

```

13 public frmTela() {
14     initComponents();
15     con_cliente = new Conexao(); // inicialização do objeto como instância
16     con_cliente.conecta(); // chama o método que conecta
17     con_cliente.executaSQL("select * from tbclientes order by cod");
18     preencherTabela();
19     posicionarRegistro();
20     tblClientes.setAutoCreateRowSorter(true); // ativa a classificação ordenada da tabela
21 }

```

Figura 6 - Implementação do construtor da classe

b) digitar os métodos abaixo, antes do método main:

```

161 public void posicionarRegistro() {
162     try {
163         con_cliente.resultset.first(); // posiciona no 1º registro da tabela
164         mostrar_Dados(); // chama o método que irá buscar o dado da tabela
165     } catch (SQLException erro) {
166         JOptionPane.showMessageDialog(null, "Não foi possível posicionar no primeiro registro: "+erro, "Mensagem do Programa", JOptionPane.INFORMATION_MESSAGE);
167     }
168 }
169
170 public void mostrar_Dados() {
171     try {
172         txtCod.setText(con_cliente.resultset.getString("cod")); // Associar a caixa de texto ao campo cod
173         txtNome.setText(con_cliente.resultset.getString("nome")); // Associar a caixa de texto ao campo nome
174         txtNasc.setText(con_cliente.resultset.getString("dt_nasc"));
175         txtFone.setText(con_cliente.resultset.getString("telefone"));
176         txtEmail.setText(con_cliente.resultset.getString("email"));
177     } catch (SQLException erro) {
178         JOptionPane.showMessageDialog(null, "Não localizou dados: "+erro, "Mensagem do Programa", JOptionPane.INFORMATION_MESSAGE);
179     }
180 }

```

Figura 7 - Métodos para apresentar dados nas caixas de texto

Linha 166 completa:

JOptionPane.showMessageDialog(null, "Não foi possível posicionar no primeiro registro: "+erro, "Mensagem do Programa", JOptionPane.INFORMATION_MESSAGE);

Linha 178 completa:

JOptionPane.showMessageDialog(null, "Não localizou dados: "+erro, "Mensagem do Programa", JOptionPane.INFORMATION_MESSAGE);

3 - O evento da Jtable `MouseClicked()` servirá para carregar as caixas de texto com dados do registro que foi selecionado na tabela com o clique do mouse, também usaremos o evento **KeyPressed** que fará o mesmo efeito só que ao toque das teclas para cima ou para baixo:

- a) para ativar este evento, selecione a grid, janela propriedades, guia evento, ao encontrar este evento, clique sobre ele e aparecerá o nome da JTable, clique novamente:

```

143 private void tblClientesMouseClicked(java.awt.event.MouseEvent evt) {
144     int linha_selecionada = tblClientes.getSelectedRow();
145     txtCod.setText(tblClientes.getValueAt(linha_selecionada,0).toString());
146     txtNome.setText(tblClientes.getValueAt(linha_selecionada, 1).toString());
147     txtNasc.setText(tblClientes.getValueAt(linha_selecionada, 2).toString());
148     txtFone.setText(tblClientes.getValueAt(linha_selecionada, 3).toString());
149     txtEmail.setText(tblClientes.getValueAt(linha_selecionada, 4).toString());
150 }

```

Figura 8 - Evento para o clique do mouse sobre o registro da Jtable

```

157 private void tblClientesKeyPressed(java.awt.event.KeyEvent evt) {
158     // evento que sincroniza a grid com as setas do teclado
159     int linha_selecionada = tblClientes.getSelectedRow();
160     txtCod.setText(tblClientes.getValueAt(linha_selecionada,0).toString());
161     txtNome.setText(tblClientes.getValueAt(linha_selecionada, 1).toString());
162     txtNasc.setText(tblClientes.getValueAt(linha_selecionada, 2).toString());
163     txtFone.setText(tblClientes.getValueAt(linha_selecionada, 3).toString());
164     txtEmail.setText(tblClientes.getValueAt(linha_selecionada, 4).toString());
165 }

```

Figura 9 - Evento para as teclas de posicionamento

4 – Botões de ação para Navegação de registros:

a) desenhar os botões no formulário

The screenshot shows a Java Swing window with a light gray background. At the top, there is a label "Email:" followed by a text field containing "antonio@antonio.com". Below the text field, there are four navigation buttons arranged horizontally: a first button (left arrow), a previous button (double left arrow), a next button (double right arrow), and a last button (right arrow). At the bottom of the window, there is a table with two columns: "Código" and "Nome". The table contains one row with the values "1" and "Antonio F. Cruz".

Figura 10 - Botões de navegação de registros

b) inserir o código fonte de cada um deles:

```

private void btnPrimeiroRegistroActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        con_cliente.resultset.first();
        mostrar_Dados();
    } catch (SQLException erro) {
        JOptionPane.showMessageDialog(null, "Não foi possível acessar o primeiro registro: "+erro.getMessage());
    }
}

private void btnUltimoRegistroActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        con_cliente.resultset.last();
        mostrar_Dados();
    } catch (SQLException erro) {
        JOptionPane.showMessageDialog(null, "Não foi possível posicionar no último registro: "+erro.getMessage());
    }
}

```

```

private void btnVoltarUmRegistroActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        con_cliente.resultset.previous();
        mostrar_Dados();
    } catch (SQLException erro) {
        JOptionPane.showMessageDialog(null, "Não foi possível posicionar no registro anterior: ").
    }
}

private void btnAvancarUmRegistroActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        con_cliente.resultset.next();
        mostrar_Dados();
    } catch (SQLException erro) {
        JOptionPane.showMessageDialog(null, "Não foi possível posicionar no proximo registro: "+
    }
}

```

5 – Até agora temos este resultado:

- Tabela e caixas de texto alimentadas com dados,
- Sincronia das caixas de texto ao clique do mouse ou ao toque das teclas “seta acima” ou “seta abaixo,”
- Funcionamento dos botões de navegação.

Código	Nome	Data Nascime...	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

Figura 11 - Formulário na 2ª execução