

## Conexão Java/MySql – parte 1 (design formulário e classe conexão)

1 - Criar um projeto no **Netbeans Apache**.

2 – Copiar para dentro da pasta deste projeto o driver de conexão MySql.

3 - Importar o driver mysql (Slide 13 – arquivo: Aula 01 - Conexao BD - Netbeans Apache.pdf)

4 – No “**PHPMyAdmin**” - MySql - importar no o banco de dados “bd\_cliente”

5 - No “**NetBeans**” (dentro do projeto) **criar** um “**java package**” (pacote) com o nome "conexao" dentro da pasta "Source Packages" (Pacotes de Código-fonte):

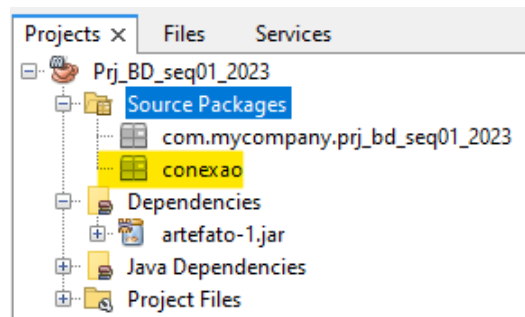


Figura 1 – Organização dos pacotes (packages)

6 - Dentro do *package* "conexao" **criar** uma **classe java** com o nome “Conexao” e digitar o código a seguir:

7.1 - Início **código da classe de conexão**:

```
1  /*
2  * Esta classe será responsável pela conexão com o MySql e terá 3 métodos:
3  * -- método 1: abertura da conexão
4  * -- método 2: fechamento da conexão
5  * -- método 3: execução de comandos SQL
6  */
7
8  package conexao;
9
10 import javax.swing.JOptionPane;
11 import java.sql.*; // para execução de comandos SQL no ambiente Java
12
13 public class Conexao {
14     final private String driver = "com.mysql.cj.jdbc.Driver"; // definição do driver MySql para acesso aos dados
15     final private String url = "jdbc:mysql://localhost/clientes"; // acesso ao bd "clientes" no servidor (myAdmin)
16     final private String usuario = "root"; // usuário do MySql
17     final private String senha = ""; // senha do MySql
18     private Connection conexao; // variável que armazenará a conexão aberta
19     public Statement statement; // variável para execução dos comandos SQL dentro do ambiente Java
20     public ResultSet resultSet; // variável que armazenará o resultado da execução de um comando SQL
21
22
23     public boolean conecta() {
24         boolean result = true;
25         try {
26             Class.forName(driver);
27             conexao = DriverManager.getConnection(url,usuario,senha);
28             JOptionPane.showMessageDialog(null, "Conexão estabelecida", "Mensagem do Programa", JOptionPane.INFORMATION_MESSAGE);
29         } catch (ClassNotFoundException Driver){
30             JOptionPane.showMessageDialog(null,"Driver não localizado"+Driver,"Mesangem do Programa", JOptionPane.INFORMATION_MESSAGE);
31             result = false;
32         } catch (SQLException Fonte){
33             JOptionPane.showMessageDialog(null,"Fonte de dados não localizada!"+"Fonte","Mensagem do Programa",JOptionPane.INFORMATION_MESSAGE);
34             result = false;
35         }
36         return result;
37     }
38 }
```

Figura 2- Início da classe Conexao

## 7.2 - Continuação e término do código da classe de conexão:

```
38
39
40 public void desconecta() {
41     try {
42         conexao.close();
43         JOptionPane.showMessageDialog(null, "Conexão com o banco fechada", "Mensagem do Programa", JOptionPane.INFORMATION_MESSAGE);
44     } catch (SQLException fecha) {
45         JOptionPane.showMessageDialog(null, "Erro ao fechar o banco", "Mensagem do Programa", JOptionPane.INFORMATION_MESSAGE);
46     }
47 }
48
49 public void executaSQL(String sql) {
50     try {
51         statement = conexao.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);
52         resultset = statement.executeQuery(sql);
53     } catch (SQLException excecao) {
54         JOptionPane.showMessageDialog(null, "Erro no comando SQL! \n Erro: " + excecao, "Mensagem do Programa", JOptionPane.INFORMATION_MESSAGE);
55     }
56 }
57 }
```

Figura 3 - Continuação e término da classe *Conexao*

## 8 - Criar o java package (pacote) "**controle**" dentro da pasta "Pacotes de Código-fonte"

## 9 - Dentro do package "controle" criar o JFrame (design da tela):

Código	Nome	Data Nascimento	Telefone	Email

Figura 4 - Design inicial do formulário

## 10 – Código para *criação* da "*JTable*":

```
1
2 package controle;
3
4 import java.awt.*;
5 import java.text.*;
6 import javax.swing.*;
7 import javax.swing.text.MaskFormatter;
8
9 import javax.swing.table.DefaultTableModel; // para reconhecimento de Jtable
10
11
12 public class FrmTelaCad extends JFrame {
13
14     JLabel rCodigo, rNome, rEmail, rTel, rData;
15     JTextField tcodigo, tnome, temail;
16     JFormattedTextField tel, data;
17     MaskFormatter mTel, mData;
18
19     JTable tblClientes; // datagrid
20     JScrollPane scp_tabela; // container para o datagrid
21
22     public FrmTelaCad() {
23         setTitle("Conexão Java com MySql");
24         setResizable(false);
25     }
26 }
```

Este código a seguir **deverá ser digitado** antes da definição do tamanho e visibilidade do formulário, **no final do “construtor”**:

```
86 // configuração da Jtable
87 tblClientes = new javax.swing.JTable();
88 scp_tabela = new javax.swing.JScrollPane();
89
90 tblClientes.setBounds(50,200,550,200);
91 scp_tabela.setBounds(50,200,550,200);
92
93 tela.add(tblClientes);
94 tela.add(scp_tabela);
95
96 tblClientes.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
97
98 tblClientes.setFont(new java.awt.Font("Arial", 1, 12));
99
100 tblClientes.setModel(new javax.swing.table.DefaultTableModel(
101     new Object [][] {
102         {null, null, null, null, null},
103         {null, null, null, null, null},
104         {null, null, null, null, null},
105         {null, null, null, null, null}
106     },
107     new String [] { "Código", "Nome", "Data Nascimento", "Telefone", "Email" }
108 );
109 boolean[] canEdit = new boolean [] {
110     false, false, false, false, false };
111
112 public boolean isCellEditable(int rowIndex, int columnIndex) {
113     return canEdit [columnIndex];
114 };
115 scp_tabela.setViewportViewView(tblClientes);
116
117     tblClientes.setAutoCreateRowSorter(true); // ativa a classificação ordenada da tabela
118
119 // fim da config da jtable
120
121 setSize(800, 600);
122 setVisible(true);
123 setLocationRelativeTo(null);
```

Figuras 5 e 6 – Código para criação da JTable

**11** - No **código fonte do formulário**, iniciar inserindo os imports:

```
import conexao.Conexao;
import javax.swing.JOptionPane;
```

**12** - Incluir após a abertura da classe (*public class FrmTelaCad extends JFrame {}*):

```
Conexao con_cliente;
```

**13** - Incluir após a abertura do construtor (*public FrmTelaCad(){}*)

```
con_cliente = new conexao(); // inicialização do objeto
con_cliente.conecta(); // chama o método que conecta
```

```

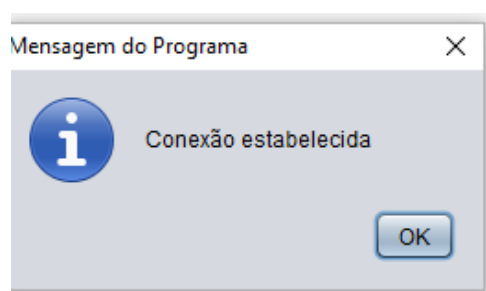
2  package controle;
3
4  import java.awt.*;
5  import java.text.*;
6  import javax.swing.*;
7  import javax.swing.text.MaskFormatter;
8
9  import conexao.Conexao;
10 import javax.swing.JOptionPane;
11
12 import javax.swing.table.DefaultTableModel; // para reconhecimento de Jt:
13
14
15 public class FrmTelaCad extends JFrame {
16
17     Conexao con_cliente;
18
19     JLabel rCodigo, rNome, rEmail, rTel, rData;
20     JTextField tcodigo, tnome, temail;
21     JFormattedTextField tel, data;
22     MaskFormatter mTel, mData;
23
24     JTable tblClientes; // datagrid
25     JScrollPane scp_tabela; // container para o datagrid
26
27     public FrmTelaCad() {
28
29         con_cliente = new Conexao(); // inicialização do objeto
30         con_cliente.conecta(); // chama o método que conecta
31
32         setTitle("Conexão Java com MySql");
33         setResizable(false);

```

*Figura 7 – Chamada a classe Conexao*

**14** - Salvar, testar.

**15** - Na execução, primeiro deverá aparecer a caixa de diálogo: Conexão Estabelecida.



*Figura 8 - Caixa de diálogo: Conexão estabelecida*

**16** – E logo após apresentará o formulário:

Conexão Java com MySql

Código::

Nome::

Data::

Telefone::

Email::

Código	Nome	Data Nascimento	Telefone	Email

*Figura 9 - Formulário na 1ª execução*

**Obs:** Centralize o formulário na tela e coloque um título na “barra de títulos”.