

## Driver Mysql para Banco de Dados

MySQL.:

Driver.: "com.mysql.jdbc.Driver"

URL.: "jdbc:mysql://localhost/nomedobanco"

Usuário.: root

senha.: \*\*\*\*\*

## Conexão com um Banco de Dados

Colocar o driver do banco de dados no diretório.:/jdk/jre/lib/ext/aqui.

Incluir o conector mysql na pasta de bibliotecas do projeto.

"Projeto"/"Bibliotecas"

/"adicionar JAR/Pasta"/selecionar o cennectorMysql

```
import java.sql.*;
import javax.swing.*;
public class conexao
{
```

```
    final private String driver = "com.mysql.jdbc.Driver";
    final private String url = "jdbc:mysql://localhost/nome_do_banco";
    final private String usuario = "root";
    final private String senha = "123456";
```

//VARIÁVEIS GLOBAIS PARA CONEXÃO COM O BANCO DE DADOS MYSQL.

//final = constante, nao podem ser alteradas

//private = privado, so podem ser acessadas por métodos da mesma classe

```
    private Connection con; //conexão com o banco de dados
    public Statement stm; //responsável por abrir caminho até o banco de dados
    public ResultSet rs; //Será armazenado as consultas sql com o banco de dados
```

```
//
```

---

//EFETUANDO CONEXÃO COM O BANCO DE DADOS MYSQL

```
    public boolean conecta()
    {
        boolean result = true; // variável local para armazenar o resultado

        try
        {
            Class.forName(driver); //Tenta encontrar o driver
            con = DriverManager.getConnection(url, usuario, senha); //Faz a
conexão
            JOptionPane.showMessageDialog(null, "Conexão efetuada com sucesso!");
        }
        catch (ClassNotFoundException driverErro) //Exception caso o driver não
seja encontrado
        {
            JOptionPane.showMessageDialog(null, "Driver não localizado.:
"+driverErro);
            result = false;
        }
    }
}
```

```

    }
    catch(SQLException fonte) // Exception caso a conexão falhe.
    {

        JOptionPane.showMessageDialog(null,"Erro na conexão com a fonte de
dados.: "+fonte);
        result = false;
    }

    return result; //retorna o resultado da tentativa de conexão
}

//
//DESCONECTAR O BANCO DE DADOS AO FECHAR O FORMULÁRIO

public void desconecta()
{

    boolean result;

    try
    {
        con.close(); //Fecha a conexão
        JOptionPane.showMessageDialog(null,"Banco de dados fechado!");
    }
    catch (SQLException fecharErro)
    {
        JOptionPane.showMessageDialog(null,"Não foi possível fechar o Banco
Dados"+fecharErro);
        result = false;
    }

}

//
//EXECUTAR UM COMANDO SQL

public void executaSQL(String sql)
{

    try
    {
        stm =
con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_READ_ONLY);
//Criar o Statement, ou seja, o caminho até o banco
        rs = stm.executeQuery(sql);//o rs armazena o comando sql executado pelo
stm
    }
    catch (SQLException sqlErro)
    {

        JOptionPane.showMessageDialog(null,"Não foi possível executar o comando sql
"+sqlErro+", comando sql passado.: "+sql);
    }

}

}

} // fim classe Conexao

```

## Dicas JavaSE

**Classe.: java.util.Date.:** Manipular a data.

**Métodos.:**

`getDate()` = dia do mês  
`getDay()` = dia da semana  
`getMonth()` = mês  
`getYear()` = ano

**Classe.: java.text.SimpleDateFormat.:** Formato da hora.

**Métodos.:**

`format()` = formata a hora - HH:mm:ss

**Classe.: javax.swing.JOptionPane.:** Tela gráfica

**Métodos.:**

`JOptionPane.showConfirmDialog(null,"texto","titulo",JOptionPane.YES_NO_OPTION);`

**Classe.: java.io.File.:** Arquivos

**Abrir uma caixa de diálogo para escolha de uma foto.:**

```
JFileChooser escolhe_foto = new JFileChooser();
escolhe_foto.setCurrentDirectory(new File("/estoque/Pictures/"));
escolhe_foto.setDialogTitle("Escolha a foto do Cliente");
escolhe_foto.showOpenDialog(this);
String foto = ""+escolhe_foto.getSelectedFile().getName();
tf_foto.setText(foto);
label_foto.setIcon(new
ImageIcon("/estoque/Pictures/"+tf_foto.getText()));
```

obs.: Este código tem que estar dentro de um bloco try-catch

---

## Ações

**Chamar um formulário abrindo apenas uma vez.:**

```
public void mostraDados()
{
    if(objeto == null)
    {
        objeto = new ObjetoClasse();
        objeto.setVisible(true);
    }
    else
    {
        objeto.setVisible(true);
        objeto.setState(JFrame.NORMAL);
    }
}
```

**Fechar um formulário mas não fechar sistema.:**

```
dispose();
```

**Fechar o sistema.:**

```
System.exit(0);
```

**Chamar um formulário.:**

```
new NomeClasse().show();
```

**Colocar texto no label ou campo de texto.:**

```
nome_label.setText("Coloque o texto aqui");
```

**Fazer consulta no Banco.:**

```
resultset = statement.executeQuery("select * from tabela");
```

**Fazer atualização no Banco.:**

```
statement.executeUpdate("insert into tabela....");
```

**Colocar o cursor no campo de texto.:**

```
tf_nome.requestFocus();
```

**Proibir edição de um campos de texto.:**

```
tf_nome.setEditable(false);
```

**Tamanho da palavra(Quantos caracteres).:**

```
int tamanho = tf_nome.getText().length();
```

**Sobreescrever uma String com a quantidade de caracteres desejados.:** EX.: Belo → Be

```
String word = "Belo Horizonte";
```

```
String word_changed = word.substring(0,2);
```

**Adicionar item a um Combo Box.:**

```
cb_nome.addItem("MG");
```

**Remover item de um Combo Box.:**

```
cb_nome.removeItem("MG");
```

**Remover todos os itens de um Combo Box.:**

```
cb_nome.removeAllItems();
```

**Para analisar se uma parte contém na frase.:**

```
frase.matches(".*"+ pesquisado +".*");
```

**Remove espaços em branco no início ou no final de uma String.:**

```
String.trim();
```

**Formatar data MySql(YYYY-MM-DD) para (DD/MM/YYYY).:**

```
String[] datanasc =
```

```
con_cliente.rs.getString("cli_datanasc").split("-");
```

```
int ano = Integer.parseInt(datanasc[0]) - 1900;
```

---

```
int mes = Integer.parseInt(datanasc[1])-1;
int dia = Integer.parseInt(datanasc[2]);

DateFormat format = new SimpleDateFormat("dd/MM/yyyy");
java.sql.Date newDate = new java.sql.Date(ano,mes,dia);
String datanasci = format.format(newDate);
tf_datanasc.setText(datanasci);
```

**Formatar data MySql(DD/MM/YYYY) para (YYYY-MM-DD) .:**

```
String[] datanasc = tf_datanasc.getText().split("/");
int ano = Integer.parseInt(datanasc[2])-1900;
int mes = Integer.parseInt(datanasc[1])-1;
int dia = Integer.parseInt(datanasc[0]);
DateFormat format = new SimpleDateFormat("yyyy-MM-dd");
java.sql.Date newDate = new java.sql.Date(ano,mes,dia);
String datanasci = format.format(newDate);
```

**Ignorar uma exceção - Cláusula 'throws':**

```
public static void main(String args[]) throws IOException
```

**Sempre executar uma ação em um try-catch - Cláusula 'finally':**

```
finally
{
    JOptionPane.showMessageDialog(null,"Fim da aplicação");
}
```

**Criar uma exceção – Cláusula 'throw new Exception()**

```
throw new Exception("erroOcorrido");
```

**Obter detalhes de um erro em uma exceção.:**

```
erroExcessão.printStackTrace();
```

---