

Módulo 5 - JDBC – java.sql - Conexão com Banco de Dados

Fazer o download do drive de conexão com o BD: <http://jdbc.postgresql.org/download.html>

Clicar em Projetos -> Propriedades -> Caminho de Construção Java -> Aba Bibliotecas -> Incluir JARs externos -> e carregar o driver de conexão.

// Exemplo17 – Testando a conexão com o Driver do Banco de Dados

```
//Apenas testa o Driver jdbc
public class Ex17_Conexao{
    public static void main(String[] args) {
        try {
            Class.forName("org.postgresql.Driver");
            System.out.println("Driver carregado com sucesso!");
        }
        catch (Exception ex) {
            System.out.println("Driver nao pode ser carregado!");
        }
    }
}
```

// Exemplo18 - Conexão com o Database

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class Ex18_ConexaoDB
{
    public static void main(String[] args)
    {
        String driver = "org.postgresql.Driver";
        String user = "postgres";
        String senha = "root";
        String url = "jdbc:postgresql://localhost:5432/postgres";

        try
        {
            Class.forName(driver);
            Connection con = null;

            con = (Connection) DriverManager.getConnection(url, user, senha);

            System.out.println("Conexão realizada com sucesso.");

        }
        catch (ClassNotFoundException ex)
        {
            System.err.print(ex.getMessage());
        }
        catch (SQLException e)
        {
            System.err.print(e.getMessage());
        }
    }
}
```

// Exemplo19 - CREATE TABLE

```
import javax.swing.*;
import java.awt.*;
import java.sql.*;

public class Ex19_CriaTabela extends JFrame {
    private static final long serialVersionUID = 1L;
    private Connection con;
    private Statement st;

    public Ex19_CriaTabela()
    {
        String driver = "org.postgresql.Driver";
        String sUsuario = "postgres";
        String sSenha = "root";
        String sFonte = "jdbc:postgresql://localhost:5432/postgres";

        try {
            Class.forName(driver);
            con = DriverManager.getConnection(sFonte, sUsuario, sSenha);
            JOptionPane.showMessageDialog(this,
                "Banco conectado com sucesso!",
                "Mensagem", JOptionPane.WARNING_MESSAGE);

        } catch (SQLException eSQL) {
            // exceções de SQL
            eSQL.printStackTrace();
            JOptionPane.showMessageDialog(this,
                "Falha na conexão com o banco!\n" +
                "Mensagem: " + eSQL.getMessage(),
                "Erro", JOptionPane.ERROR_MESSAGE);
            System.exit(0);
        } catch (Exception e) {
            // demais exceções
            e.printStackTrace();
            JOptionPane.showMessageDialog(this,
                "Falha na conexão com o banco!\n" +
                "Mensagem: " + e.getMessage(),
                "Erro", JOptionPane.ERROR_MESSAGE);
            System.exit(1);
        }

        String sentencaSQL = "CREATE TABLE tabela1 (codigo integer PRIMARY KEY
, nome VARCHAR(50));" ;
        try{
            st = con.createStatement();
            st.executeUpdate(sentencaSQL);
            JOptionPane.showMessageDialog(this,
                "Tabela criada com sucesso!",
                "Mensagem", JOptionPane.WARNING_MESSAGE);
        } catch (SQLException eSQL) {
            eSQL.printStackTrace();
            JOptionPane.showMessageDialog(this,
                "Não foi possível criar a tabela!\n" +
                "Mensagem: " + eSQL.getMessage(),
                "Erro", JOptionPane.ERROR_MESSAGE);
            System.exit(2);
        }

        try {
            st.close();
            con.close();
        }
    }
}
```

```

    }catch(Exception exception){
        exception.printStackTrace();
        System.exit(3);
    }

    Container P = getContentPane();
    P.setLayout(new FlowLayout());
    JLabel mensagem = new JLabel("Você acabou de testar um exemplo usando
CREATE TABLE!");
    P.add(mensagem);
}

public static void main(String args[])
{
    Ex19_CriaTabela ex = new Ex19_CriaTabela();
    ex.setDefaultCloseOperation(EXIT_ON_CLOSE);
    ex.setTitle("USANDO CREATE TABLE");
    ex.setVisible(true);
    ex.setSize(400,300);
}
}

```

// Exemplo20 - INSERT

```

import javax.swing.*;
import java.awt.Container;
import java.awt.FlowLayout;
import java.sql.*;

public class Ex20_InserirDados extends JFrame {
    private static final long serialVersionUID = 1L;
    private Connection con;
    private Statement st;

    public Ex20_InserirDados() throws SQLException
    {

        String driver = "org.postgresql.Driver";
        String sUsuario = "postgres";
        String sSenha = "root";
        String sFonte = "jdbc:postgresql://localhost:5432/postgres";

        try {
            Class.forName(driver);
            con = DriverManager.getConnection(sFonte, sUsuario, sSenha);
            JOptionPane.showMessageDialog(this,
                "Banco conectado com sucesso!",
                "Mensagem", JOptionPane.WARNING_MESSAGE);

        }catch (SQLException eSQL) {
            // exceções de SQL
            eSQL.printStackTrace();
            JOptionPane.showMessageDialog(this,
                "Falha na conexão com o banco!\n" +
                "Mensagem: " + eSQL.getMessage(),
                "Erro", JOptionPane.ERROR_MESSAGE);
        }catch (Exception e) {
            // demais exceções
            e.printStackTrace();
            JOptionPane.showMessageDialog(this,
                "Falha na conexão com o banco!\n" +
                "Mensagem: " + e.getMessage(),
                "Erro", JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

```

        System.exit(0);
    }

    try{
        //cria statement para consultar banco de dados
        st = con.createStatement();

        //Cadastro de valores pre-definidos
        st.executeUpdate ("INSERT INTO tabelal values(1,'Joao');");
    }catch(SQLException eSQL){
        JOptionPane.showMessageDialog(this,
            "Erro na expressão do INSERT!\n" +
            "Mensagem: " + eSQL.getMessage(),
            "Erro", JOptionPane.ERROR_MESSAGE);
        System.exit(1);
    }

    try {
        st.close();
        con.close();

    }catch(Exception exception){
        exception.printStackTrace();
        System.exit(2);
    }

    Container P = getContentPane();
    P.setLayout(new FlowLayout());
    JLabel mensagem = new JLabel("Você acabou de testar um exemplo usando
INSERT!");
    P.add(mensagem);

}

public static void main(String args[]) throws SQLException
{
    Ex20_InserirDados ex = new Ex20_InserirDados();
    ex.setDefaultCloseOperation(EXIT_ON_CLOSE);
    ex.setVisible(true);
    ex.setTitle("USANDO INSERT");
    ex.setSize(400,200);
}
}

```

// Exemplo21 - SELECT

```

import javax.swing.*;
import java.awt.Container;
import java.awt.FlowLayout;
import java.sql.*;

public class Ex21_SelecionarDados extends JFrame {
    private static final long serialVersionUID = 1L;
    private Connection con;
    private Statement st;

    public Ex21_SelecionarDados() throws SQLException
    {

        String driver = "org.postgresql.Driver";
        String sUsuario = "postgres";
        String sSenha = "root";
        String sFonte = "jdbc:postgresql://localhost:5432/postgres";
    }
}

```

```

try {
    Class.forName(driver);
    con = DriverManager.getConnection(sFonte, sUsuario, sSenha);
    JOptionPane.showMessageDialog(this,
        "Banco conectado com sucesso!",
        "Mensagem", JOptionPane.WARNING_MESSAGE);

} catch (SQLException eSQL) {
    // exceções de SQL
    eSQL.printStackTrace();
    JOptionPane.showMessageDialog(this,
        "Falha na conexão com o banco!\n" +
        "Mensagem: " + eSQL.getMessage(),
        "Erro", JOptionPane.ERROR_MESSAGE);
} catch (Exception e) {
    // demais exceções
    e.printStackTrace();
    JOptionPane.showMessageDialog(this,
        "Falha na conexão com o banco!\n" +
        "Mensagem: " + e.getMessage(),
        "Erro", JOptionPane.ERROR_MESSAGE);
    System.exit(0);
}

try{
    st = con.createStatement();
    ResultSet resultSet = st.executeQuery("SELECT * FROM tabela1;");

    //processa resultados da consulta
    ResultSetMetaData metaData = resultSet.getMetaData();
    int numberOfColumns = metaData.getColumnCount();
    System.out.println("Relação de pessoas cadastradas");

    for (int i=1; i<=numberOfColumns; i++){
        //exibe o nome dos campos da tabela
        System.out.printf("%-8s\t",metaData.getColumnName(i));
    }
    while (resultSet.next()){
        System.out.println();
        String linha = "";
        for (int x=1; x<=numberOfColumns; x++){
            //exibe o conteúdo dos campos
            System.out.printf("%-8s\t",resultSet.getObject(x));
            linha = linha + " " + resultSet.getObject(x);
        }
    }

} catch (SQLException eSQL){
    JOptionPane.showMessageDialog(this,
        "Erro na expressão do SELECT!\n" +
        "Mensagem: " + eSQL.getMessage(),
        "Erro", JOptionPane.ERROR_MESSAGE);
    System.exit(1);
}

try {
    st.close();
    con.close();

} catch (Exception exception){
    exception.printStackTrace();
    System.exit(2);
}

Container P = getContentPane();

```

```

        P.setLayout(new FlowLayout());
        JLabel mensagem1 = new JLabel("Você acabou de testar um exemplo usando
SELECT!");
        JLabel mensagem2 = new JLabel("O resultado da consulta esta no
Console!");
        P.add(mensagem1);
        P.add(mensagem2);
    }

    public static void main(String args[]) throws SQLException
    {
        Ex21_SelecionarDados ex = new Ex21_SelecionarDados ();
        ex.setDefaultCloseOperation(EXIT_ON_CLOSE);
        ex.setVisible(true);
        ex.setTitle("USANDO SELECT");
        ex.setSize(400,200);
    }
}

```

// Exemplo22 - DELETE

```

import javax.swing.*;
import java.awt.Container;
import java.awt.FlowLayout;
import java.sql.*;

public class Ex22_ApagarDados extends JFrame {
    private static final long serialVersionUID = 1L;
    private Connection con;
    private Statement st;

    public Ex22_ApagarDados() throws SQLException
    {

        String driver = "org.postgresql.Driver";
        String sUsuario = "postgres";
        String sSenha = "root";
        String sFonte = "jdbc:postgresql://localhost:5432/postgres";

        try {
            Class.forName(driver);
            con = DriverManager.getConnection(sFonte, sUsuario, sSenha);
            JOptionPane.showMessageDialog(this,
                "Banco conectado com sucesso!",
                "Mensagem", JOptionPane.WARNING_MESSAGE);

        } catch (SQLException eSQL) {
            // exceções de SQL
            eSQL.printStackTrace();
            JOptionPane.showMessageDialog(this,
                "Falha na conexão com o banco!\n" +
                "Mensagem: " + eSQL.getMessage(),
                "Erro", JOptionPane.ERROR_MESSAGE);
            System.exit(0);
        } catch (Exception e) {
            // demais exceções
            e.printStackTrace();
            JOptionPane.showMessageDialog(this,
                "Falha na conexão com o banco!\n" +
                "Mensagem: " + e.getMessage(),
                "Erro", JOptionPane.ERROR_MESSAGE);
            System.exit(0);
        }
    }
}

```

```

    try{
        st = con.createStatement();
        String ExecutaDelete= "DELETE FROM tabelal WHERE codigo=1";
        st.execute (ExecutaDelete);
    }catch(SQLException eSQL){
        JOptionPane.showMessageDialog(this,
            "Erro na expressão do DELETE!\n" +
            "Mensagem: " + eSQL.getMessage(),
            "Erro", JOptionPane.ERROR_MESSAGE);
        System.exit(2);
    }

    try {
        st.close();
        con.close();

    }catch(Exception exception){
        exception.printStackTrace();
        System.exit(3);
    }
    Container P = getContentPane();
    P.setLayout(new FlowLayout());
    JLabel mensagem = new JLabel("Você acabou de testar um exemplo usando
DELETE!");
    P.add(mensagem);

}

public static void main(String args[]) throws SQLException
{
    Ex22_ApagarDados ex = new Ex22_ApagarDados ();
    ex.setDefaultCloseOperation(EXIT_ON_CLOSE);
    ex.setVisible(true);
    ex.setTitle("USANDO DELETE");
    ex.setSize(400,200);
}
}

```

EXERCÍCIOS PARA ENTREGA

EE_08) Implemente um programa em Java que resulte na seguinte janela e suas funcionalidades para criar uma tabela de Banco de Dados:

Wizard para criação de tabelas de BD

Fonte de Dados: jdbc:postgresql://localhost:5432/postgres Usuário

Não Conectado

Nome da Tabela

Nome Tipo