CRUD – PARTE02 – NETBEANS – INCLUSÃO DE REGISTROS NA TABELA DO BANCO MYSQL

**PROFESSOR EDIBERTO MARIANO**

**- CRIANDO UM NOVO PROJETO**

Interface gráfica do usuário, Aplicativo, Word, Site

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

OBS. DESMARCA A CLASSE **Main**, POIS A PRIMEIRA CLASSE QUE PRIMEIRO É EXECUTADA SERÁ O **FORMULÁRIO**

Interface gráfica do usuário, Texto, Aplicativo

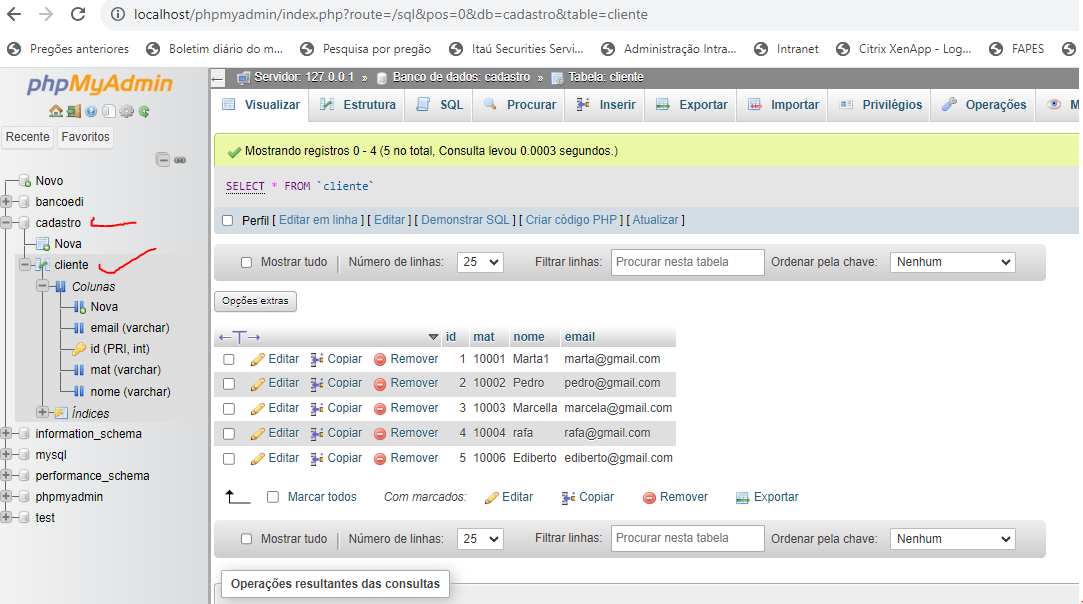
Descrição gerada automaticamente

**FAZENDO A CONEXÃO COM O BANCO DE DADOS**

**STARTAR O MYSQL CONFORME ABAIXO**

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente



**ADICIONAR AO PROJETO A BIBLIOTECA responsável pela conexão ao banco**

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

Obs. Ela se encontra na pasta **C:\Program Files\NetBeans 8.0.1\ide\modules\ext**

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

O browser do projeto fica assim

Tela de celular com publicação numa rede social

Descrição gerada automaticamente

CRIAR UMA CLASSE JAVA MAIN “**TestaConexao**” conforme abaixo

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

IMPLEMENTA A CLASSE CRIADA CONFORME ABAIXO



package cadclientes;

import com.mysql.jdbc.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class TestaConexao {

private static final String USUARIO = "root";

private static final String SENHA = "";

private static final String URL = "jdbc:mysql://localhost:3306/cadastro";

public static Connection conectar()throws SQLException, ClassNotFoundException {

DriverManager.registerDriver(new com.mysql.jdbc.Driver());

Connection conexao = (Connection) DriverManager.getConnection(URL, USUARIO, SENHA);

return conexao;

}

public static void main(String[] args) throws ClassNotFoundException{

try {

Connection conexao = TestaConexao.conectar();

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

} catch (SQLException ex) {

ex.printStackTrace();

System.out.println("Conexão ao Banco falhou");

}

}

}

EXECUTA A CLASSE “**TestaConexao.java**” conforme a seguir.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**RESPOSTA**

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**CRIAR UM FORMULÁRIO**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

**O NOME DO FORMULÁRIO SERÁ Principal**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

**Label -** 

**Campos de entrada de dados** 

**PROPRIEDADES DOS ELEMENTOS DA TELA**

Tabela

Descrição gerada automaticamente

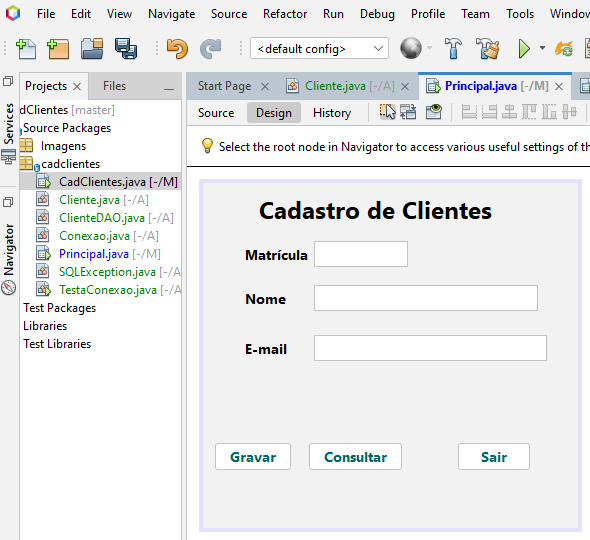
Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

**ARRUMAR O FORMULÁRIO CONFORME A SEGUIR**



ALTERAR O **TÍTULO** DA JANELA

1 – MOUSE DIREITO DENTRO DO FORMULÁRIO / PROPRIEDADE

Interface gráfica do usuário, Tabela

Descrição gerada automaticamente

2 – ESCREVE O NOVO TÍTULO DA TELA EM **title**.



VAMOS CRIAR A CLASSE CONSTRUTORA “Conexao.java” PARA REALIZAR A CONEXÃO COM O BANCO.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

IMPLEMENTAR A CLASSE **CONSTRUTORA** “**Conexao.java**”



package cadclientes;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class Conexao {

private static final String URL = "jdbc:mysql://localhost:3306/cadastro";

private static final String USER = "root";

private static final String PASS = "";

public static Connection getConnection()throws SQLException, ClassNotFoundException {

Class.forName("com.mysql.jdbc.Driver");

Connection conexao = DriverManager.getConnection(URL, USER, PASS);

return conexao;

}

public static void classConnection(Connection conn) throws Exception{

try {

Connection conexao = Conexao.getConnection();

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

} catch (SQLException e) {

throw new RuntimeException("Erro ao fechar a conexao com o banco de dados"+ e);

}

}

}

VAMOS CRIAR A CLASSE CONSTRUTORA “**Cliente.java**”



**OBS. NÃO É NECESSÁRIO CRIAR O ATRIBUTO id, MAS SE FOR CRIADO, Não dar erro**

package cadclientes;

public class Cliente {

private Integer id;

private String mat;

private String nome;

private String email;

//MÉTODO CONSTRUTOR VAZIO

public Cliente() {

}

//MÉTODO CONSTRUTOR CHEIO COM TODOS OS ATRIBUTOS

public Cliente(Integer id, String mat,String nome, String email) {

this.id = id;

this.mat = mat;

this.nome = nome;

this.email = email;

}

//MÉTODO CONSTRUTOR CHEIO COM OS ATRIBUTOS, MENOS COM A CHAVE PRIMÁRIA

public Cliente(String mat, String nome, String email) {

//this.id = id;

this.mat = mat;

this.nome = nome;

this.email = email;

}

//MÉTODOS GETTERS E SETTERS

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getMat() {

return mat;

}

public void setMat(String mat) {

this.mat = mat;

}

public String getNome() {

return nome;

}

public void setNome(String nome) {

this.nome = nome;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

//ESSE MÉTODO IRÁ PRINTAR NA TELA UM OBJETO DA TABELA CLIENTE

@Override

public String toString() {

return "id" + id + ", Matricula = " + mat + ", nome = " + nome + ", email = " + email;

}

}

VAMOS CRIAR A CLASSE CONSTRUTORA “**ClienteDAO.java**” PARA PERMITIR A INSERÇÃO DOS REGISTROS NA TABELA



package cadclientes;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet ;

import java.sql.SQLException;

import javax.swing.JOptionPane;

public class ClienteDAO {

public void adicionar(Cliente obj ) throws Exception{

String sql = "INSERT INTO cliente(mat, nome, email)VALUES(?,?,?)";

Connection conn = null;

PreparedStatement pstm = null;

try{

conn = (Connection) Conexao.getConnection();

pstm = (PreparedStatement) conn.prepareStatement(sql);

pstm.setString(1, obj.getMat());

pstm.setString(2, obj.getNome());

pstm.setString(3, obj.getEmail());

pstm.execute();

}catch(SQLException e) {

JOptionPane.showConfirmDialog(null, "Erro ao salvar dados "+ e.toString());

//System.out.println("Erro ao salvar dados "+ e.getMessage().toLowerCase());

}finally {

try {

if(pstm!=null) {

JOptionPane.showMessageDialog(null, "Registro gravado com sucesso!!!");

pstm.close();

}

if(conn!=null) {

conn.close();

}

}catch (Exception e) {

System.out.println("Erro ao salvar dados "+ e.getMessage().toLowerCase());

}

}

}

//@Override

public Cliente consultar(String chave) throws ClassNotFoundException {

Cliente cliente = null;

try {

Connection conn = Conexao.getConnection();

PreparedStatement ps = conn.prepareStatement("SELECT \* FROM Cliente WHERE mat = ?");

ps.setString(1, chave);

ResultSet r1 = ps.executeQuery();

if (r1.next()) {

cliente = new Cliente(r1.getInt("id"), r1.getString("mat"), r1.getString("nome"), r1.getString("email"));

}

ps.close();

conn.close();

} catch (SQLException e) {

e.printStackTrace(); // Trate o erro apropriadamente

}

return cliente;

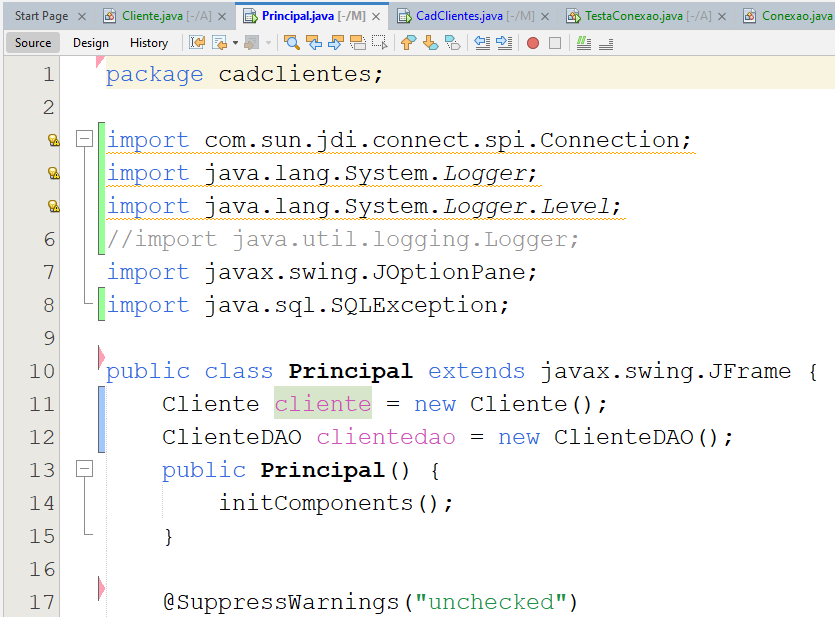
}

}

VAMOS IMPLEMENTAR A CLASSE “**Principal.java**”, para que a regra do negócio seja realizada



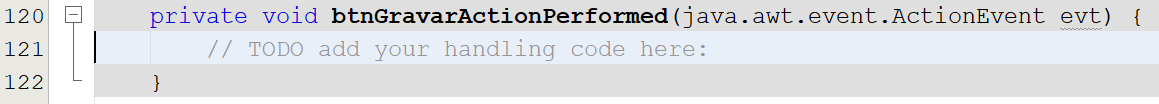
1 – VAMOS INSTANCIAR AS CLASSES Cliente e **ClienteDAO**, conforme segue



2 – VAMOS IMPLEMENTAR O EVENTO DO BOTÃO Gravar “**btnGravar”**, conforme segue.

2.1 – Valida os campos, ou seja, não aceita campos sem preenchimento.

2.2 – Grava os dados do formulário na tabela.



**FICANDO ASSIM**

**private void btnGravarActionPerformed(java.awt.event.ActionEvent evt) {**

**try {**

**Salvar\_Dados();**

**} catch(Exception ex) {**

**// Logger.getLogger(CadClientes.class.getName()).log(Level.SEVERE, null, ex);**

**}**

**}**

**private boolean validaCamposObrigatorios() {**

**return (txtMat.getText().equals("")**

**|| (txtNome.getText().equals("")**

**|| txtEmail.getText().equals(""));**

**}**

**private void Salvar\_Dados() throws Exception{**

**if(validaCamposObrigatorios()) {**

**JOptionPane.showMessageDialog(null, "Preencha todos os campos antes de gravar!!");**

**} else {**

**try {**

**cliente.setMat(txtMat.getText());**

**cliente.setNome(txtNome.getText());**

**cliente.setEmail(txtEmail.getText());**

**clientedao.adicionar(cliente);**

**}catch(SQLException e){**

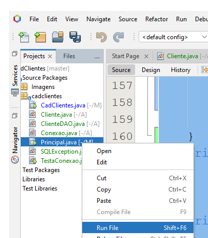
**JOptionPane.showMessageDialog(null, "Erro aos salvar os dados" + e.toString());**

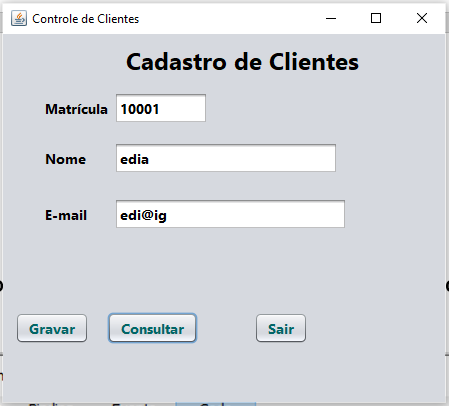
**}**

**}**

**}**

EXECUTA O FORMULÁRIO (Principal.Java)





**PROPRIEDADE DO FORMULARIO PARA QDO FOR EXECUTADO FICAR NO CENTRO DA TELA**

Tabela

Descrição gerada automaticamente

**BOTÃO CONSULTAR**

**private void btnConsultarActionPerformed(java.awt.event.ActionEvent evt) {**

**try {**

**Consultar\_Dados();**

**} catch(Exception ex) {**

**// Logger.getLogger(CadClientes.class.getName()).log(Level.SEVERE, null, ex);**

**}**

**}**

**private void Consultar\_Dados() throws Exception{**

**try {**

**// Obtém a matrícula a ser consultada do campo de texto**

**String matricula = txtMat.getText();**

**// Chama o método consultar da classe ClienteDAO para obter os dados do cliente**

**Cliente clienteConsultado = clientedao.consultar(matricula);**

**// Verifica se o cliente foi encontrado**

**if (clienteConsultado != null) {**

**// Preenche os campos de texto com os dados do cliente**

**txtNome.setText(clienteConsultado.getNome());**

**txtEmail.setText(clienteConsultado.getEmail());**

**JOptionPane.showMessageDialog(null, "Cliente encontrado!");**

**} else {**

**// Caso o cliente não seja encontrado, exiba uma mensagem**

**JOptionPane.showMessageDialog(null, "Cliente não encontrado!");**

**}**

**} catch (Exception ex) {**

**ex.printStackTrace();**

**// Trate o erro apropriadamente, exibindo uma mensagem de erro se necessário**

**JOptionPane.showMessageDialog(null, "Erro ao consultar cliente: " + ex.getMessage());**

**}**

**}**

**BOTÃO SAIR**

private void btnSairActionPerformed(java.awt.event.ActionEvent evt) {

// Fechar a tela de CadClientes (se necessário)

dispose();

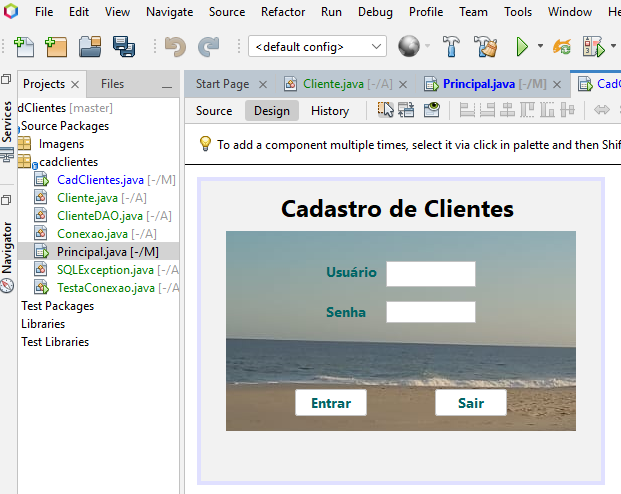
CadClientes cadclientes = new CadClientes();

cadclientes.setVisible(true);

}

**CRIAR UMA TELA DE LOGIN**

**CRIAR UM FORMULARIO**



**BOTÃO ENTRAR**

**private void txtBotaoEntrarActionPerformed(java.awt.event.ActionEvent evt) {**

**// EVENTO ONCLICK DO BOTÃO ENTRAR**

**/\*String username = txtusuario.getText().toLowerCase(); // Converter para letras minúsculas**

**String password = new String(txtsenha.getPassword());**

**if (username.equals("ediberto") && password.equals("123")) {**

**JOptionPane.showMessageDialog(null, "Usuário logado no sistema");**

**// Código para abrir a Tela Principal**

**Principal principal = new Principal();**

**principal.setVisible(true);**

**// Fechar a tela de CadClientes (se necessário)**

**dispose();**

**} else {**

**JOptionPane.showMessageDialog(null, "Usuário e/ou senha inválidos");**

**} \*/**

**//OU ASSIM**

**if(txtusuario.getText().equals("ediberto") && new String(txtsenha.getPassword()).equals("123")){**

**JOptionPane.showMessageDialog(null, "Usuário logado no sistema");**

**// Código para abrir a Tela Principal**

**Principal principal = new Principal();**

**principal.setVisible(true);**

**// Fechar a tela de CadClientes (se necessário)**

**dispose();**

**} else {**

**JOptionPane.showMessageDialog(null, "Usuário e/ou senha inválidos");**

**}**

**}**

**BOTÃO SAIR**

**private void txtBotaoSairActionPerformed(java.awt.event.ActionEvent evt) {**

**// SAIR DO SISTEMA**

**this.dispose(); // Fecha a janela atual**

**JOptionPane.showMessageDialog(null, "SISTEMA FECHADO!!!");**

**}**