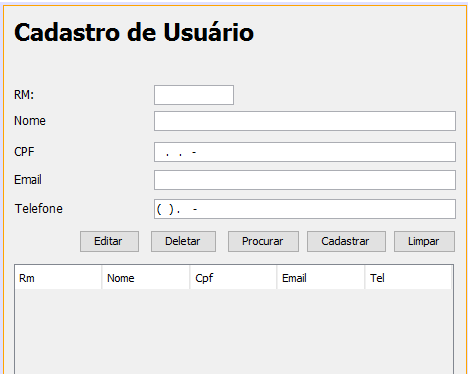
**NETBEANS – GERANDO UM CADASTRO DE CLIENTES COM INTERFACE**

 Crie o formulário abaixo e vincule a tabela e os campos a tabela do MySql o formulário deve ser criado no pacote gui com o nome de UsuarioGUI.



Crie o pacote factory e adicione a seguinte classe:

ConnectionFactory – Nome da Classe

package factory;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class ConnectionFactory {

public Connection getConnection()

{

try

{

return DriverManager.getConnection("jdbc:mysql://localhost:3306/teste","root","");

}

catch(SQLException excecao)

{

throw new RuntimeException(excecao);

}

}

}

Crie o pacote dao e adicione a seguinte classe:

UsuarioDAO – Nome da classe

package dao;

import factory.ConnectionFactory;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import modelo.Usuario;

import gui.UsuarioGUI;

/\*\*

\*

\* @author aula

\*/

public class UsuarioDAO {

private Connection connection;

Long id;

String nome;

String cpf;

String email;

String telefone;

public UsuarioDAO() {

this.connection = new ConnectionFactory().getConnection();

}

public void adiciona(Usuario usuario) {

String sql = "INSERT INTO usuarios(RM,nome,CPF,email,tel) VALUES(?,?,?,?,?)";

try {

PreparedStatement stmt = connection.prepareStatement(sql);

stmt.setInt(1, usuario.getRM());

stmt.setString(2, usuario.getNome());

stmt.setString(3, usuario.getCpf());

stmt.setString(4, usuario.getEmail());

stmt.setString(5, usuario.getTelefone());

stmt.execute();

stmt.close();

} catch (SQLException u) {

throw new RuntimeException(u);

}

}

public void editar(Usuario usuario) {

int rm = usuario.getRM();

String sql = "UPDATE usuarios SET nome= ?, cpf= ?, email= ?, tel= ? WHERE rm=" + rm + ";";

try {

PreparedStatement stmt = connection.prepareStatement(sql);

stmt.setString(1, usuario.getNome());

stmt.setString(2, usuario.getCpf());

stmt.setString(3, usuario.getEmail());

stmt.setString(4, usuario.getTelefone());

stmt.executeUpdate();

stmt.close();

} catch (SQLException u) {

throw new RuntimeException(u);

}

}

public void seleciona(Usuario usuario) {

int rm = usuario.getRM();

String sql = "SELECT \* FROM usuarios WHERE RM = " + rm + ";";

try {

PreparedStatement stmt = connection.prepareStatement(sql);

ResultSet rs = stmt.executeQuery(sql);

while (rs.next()) {

nome = rs.getString("nome");

cpf = rs.getString("CPF");

email = rs.getString("email");

telefone = rs.getString("tel");

}

usuario.setNome(nome);

usuario.setCpf(cpf);

usuario.setEmail(email);

usuario.setTelefone(telefone);

stmt.close();

} catch (SQLException u) {

throw new RuntimeException(u);

}

}

public void deletar(Usuario usuario) {

String res;

int rm = usuario.getRM();

String sql = "DELETE FROM usuarios WHERE RM = " + rm + ";";

try {

PreparedStatement stmt = connection.prepareStatement(sql);

stmt.execute();

stmt.close();

} catch (SQLException u) {

throw new RuntimeException(u);

}

}

}

Crie o pacote modelo e adicione a seguinte classe:

Usuario – Nome da Classe

package modelo;

public class Usuario {

int rm;

int num;

String nome;

String cpf;

String email;

String telefone;

public String getCpf() {

return cpf;

}

public void setCpf(String cpf) {

this.cpf = cpf;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public void setRM(int rm) {

this.rm = rm;

}

public String getNome() {

return nome;

}

public int getNum()

{

return num;

}

public int getRM() {

return rm;

}

public void setNome(String nome) {

this.nome = nome;

}

public String getTelefone() {

return telefone;

}

public void setTelefone(String telefone) {

this.telefone = telefone;

}

public void setNumLinhas(int num)

{

this.num = num;

}

}

Botão Cadastrar, Excluir, Alterar.

Cadastrar – Código do botão.

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

Usuario usuarios = new Usuario();

usuarios.setRM(Integer.parseInt(jTextField2.getText()));

usuarios.setNome(jTextField1.getText());

usuarios.setCpf(jFormattedTextField2.getText());

usuarios.setEmail(jTextField3.getText());

usuarios.setTelefone(jFormattedTextField1.getText());

if ((jTextField1.getText().isEmpty()) || (jTextField2.getText().isEmpty()) || (jFormattedTextField2.getText().isEmpty()) || (jTextField3.getText().isEmpty()) || (jFormattedTextField1.getText().isEmpty())) {

JOptionPane.showMessageDialog(null, "Os campos não podem retornar vazios");

} else {

UsuarioDAO dao = new UsuarioDAO();

dao.adiciona(usuarios);

JOptionPane.showMessageDialog(null, "Usuário " + jTextField1.getText() + " Inserido com sucesso! ");

}

jTextField2.setText("");

jTextField1.setText("");

jFormattedTextField2.setText("");

jTextField3.setText("");

jFormattedTextField1.setText("");

String sql = "SELECT \* FROM usuarios;";

try {

tabela(sql);

} catch (SQLException ex) {

Logger.getLogger(UsuarioGUI.class.getName()).log(Level.SEVERE, null, ex);

}

}

Limpar – Código do botão.

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

jTextField2.setText("");

jTextField1.setText("");

jFormattedTextField2.setText("");

jTextField3.setText("");

jFormattedTextField1.setText("");

}

Procurar – Código do botão.

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

Usuario usuarios = new Usuario();

usuarios.setRM(Integer.parseInt(jTextField2.getText()));

if((jTextField2.getText().isEmpty()))

{

JOptionPane.showMessageDialog(null, "Insira um RM");

}

else

{

UsuarioDAO dao = new UsuarioDAO();

dao.seleciona(usuarios);

jTextField1.setText(usuarios.getNome());

jFormattedTextField2.setText(usuarios.getCpf());

jTextField3.setText(usuarios.getEmail());

jFormattedTextField1.setText(usuarios.getTelefone());

String sql = "SELECT \* FROM usuarios WHERE rm="+usuarios.getRM()+";";

try {

tabela(sql);

} catch (SQLException ex) {

Logger.getLogger(UsuarioGUI.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

Deletar– Código do botão.

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

Usuario usuarios = new Usuario();

usuarios.setRM(Integer.parseInt(jTextField2.getText()));

if((jTextField2.getText().isEmpty()))

{

JOptionPane.showMessageDialog(null, "Insira um RM");

}

else

{

UsuarioDAO dao = new UsuarioDAO();

dao.deletar(usuarios);

jTextField2.setText("");

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

}

String sql = "SELECT \* FROM usuarios;";

try {

tabela(sql);

} catch (SQLException ex) {

Logger.getLogger(UsuarioGUI.class.getName()).log(Level.SEVERE, null, ex);

}

}

Alterar – Código do botão.

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

Usuario usuarios = new Usuario();

usuarios.setRM(Integer.parseInt(jTextField2.getText()));

usuarios.setNome(jTextField1.getText());

usuarios.setCpf(jFormattedTextField2.getText());

usuarios.setEmail(jTextField3.getText());

usuarios.setTelefone(jFormattedTextField1.getText());

if ((jTextField1.getText().isEmpty()) || (jTextField2.getText().isEmpty()) || (jFormattedTextField2.getText().isEmpty()) || (jTextField3.getText().isEmpty()) || (jFormattedTextField1.getText().isEmpty())) {

JOptionPane.showMessageDialog(null, "Os campos não podem retornar vazios");

} else {

UsuarioDAO dao = new UsuarioDAO();

dao.editar(usuarios);

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

}

jTextField2.setText("");

jTextField1.setText("");

jFormattedTextField2.setText("");

jTextField3.setText("");

jFormattedTextField1.setText("");

String sql = "SELECT \* FROM usuarios;";

try {

tabela(sql);

} catch (SQLException ex) {

Logger.getLogger(UsuarioGUI.class.getName()).log(Level.SEVERE, null, ex);

}

}

Metódo - Adicionar após o código do botão editar.

private void formWindowOpened(java.awt.event.WindowEvent evt) {

String sql = "SELECT \* FROM usuarios;";

try {

tabela(sql);

} catch (SQLException ex) {

Logger.getLogger(UsuarioGUI.class.getName()).log(Level.SEVERE, null, ex);

}

}

Metódo - Adicionar após o código do botão método formWindowOpened.

public void tabela(String sql) throws SQLException

{

DefaultTableModel tabela = new DefaultTableModel();

Connection connection = new ConnectionFactory().getConnection();

tabela.addColumn("RM");

tabela.addColumn("Nome");

tabela.addColumn("CPF");

tabela.addColumn("Email");

tabela.addColumn("Telefone");

PreparedStatement stmt = connection.prepareStatement(sql);

ResultSet rs = stmt.executeQuery(sql);

while (rs.next()) {

int rm = rs.getInt("RM");

String nome = rs.getString("nome");

String cpf = rs.getString("CPF");

String email = rs.getString("email");

String telefone = rs.getString("tel");

String dados[] = {""+rm,nome,cpf,email,telefone};

tabela.addRow(dados);

}

stmt.close();

jTable1.setModel(tabela);

}