ALEXIS JEFRY FRANCISCO VIDES CARRANZA

SECCION F

CARNÉ 9490194100

1.Login para control de seguridad(La contraseña es “admin”)

1.1código para validación de contraseña evento presskey y de el cambio a la ventana de menú

**Este código está en el jframe AJFVClogin dentro del pacage AJFVCjframes**

private void contraKeyPressed(java.awt.event.KeyEvent evt) {

String con = contra.getText();

if (evt.getKeyCode() == KeyEvent.VK\_ENTER) {

if ("admin".equals(con)) {

main.setVisible(true);

this.setVisible(false);

JOptionPane.showMessageDialog(null, "Bienvenido");

} else {

JOptionPane.showMessageDialog(null, "Contraseña incorrecta");

contra.setText(null);

}

}

}

2.Ventana de menú(se colocaron etiquetas con eventos para hacer las acciones y se adjunta un Joptionpane para mostrar un diálogo de bienvenida)

2.2Se colocó un evento de mouse para que cada etiqueta funcionara como botón

Este código se encuentra en **jframe AJFVCmainwindow dentro del pacage AJFVCjframes**

private void ingresarMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

AJFVCing\_mod\_elim ingresos = new AJFVCing\_mod\_elim();

ingresos.setVisible(true);

this.setVisible(false);

}

private void reporteMouseClicked(java.awt.event.MouseEvent evt) {

AJFVCreportes reporte = new AJFVCreportes();

reporte.setVisible(true);

this.setVisible(false);

// TODO add your handling code here:

}

private void salirMouseClicked(java.awt.event.MouseEvent evt) {

System.exit(0);

// TODO add your handling code here:

}

3 Al seleccionar ingresos se abre una ventana de ingreso de datos

3.1Funcíon de conexión que se encuentra en el paquete AJFVCclases que se utiliza para hacer la conexión con la base de datos

public class AJFVCconexion {

public static final String URL = "jdbc:mysql://localhost/escuela"+"?useTimezone=true&serverTimezone=UTC";

public static final String USERNAME = "root";

public static final String PASSWORD = "ky9029";

PreparedStatement ps;

ResultSet rs;

public static Connection getConnection() {

Connection Conexion = null;

try {

//no es necesario por lo que se deja en comentado

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

Conexion = (Connection) DriverManager.getConnection(URL, USERNAME, PASSWORD);

if (Conexion != null){

System.out.println( "Conexion Exitosa");

}

} catch (ClassNotFoundException | SQLException e) {

System.out.println(e);

}

return Conexion;

}

}

3.2 Después de ingresar los datos se hace click en el botón de ingresar lo cual coloca los datos en esta tabla y muestra un mensaje de confirmación

3.3El código del botón ingresar

**jframe AJFVCingresos dentro del pacage AJFVCjframes**

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

AJFVCingresardatos.ingresarpago(txtclave.getText(), txtnombre.getText(), txttel.getText(),

txtdirec.getText(), txtvariable.getText(), txtemail.getText(),genero.getSelectedItem().toString(),

cuotas.getSelectedItem().toString(), mora.getSelectedItem().toString());

MostrarDatos();

}

3.3.1 código de la función ingresar datos que está en el pacckege de clases

public class AJFVCingresardatos {

public static void ingresarpago(String clave, String nombre, String telefono, String direccion, String fecha, String email,

String genero, String pagado, String mora) {

PreparedStatement ps;

ResultSet rs;

Connection con = null;

double total = 0;

int pagos = Integer.parseInt(pagado);

if(mora.equals("SI")){

total = (pagos \* 3500)+(pagos\*35);

}else{

total = (pagos \* 3500);

}

int faltas =10 - pagos;

try {

con = AJFVCconexion.getConnection();

int rep = AJFVCrepetido.buscar(clave);

if (rep == 0) {

ps = con.prepareStatement("INSERT INTO alumno (clave,nombre,teléfono, dirección, email, "

+ "fecha\_de\_nacimiento, genero,cuotas\_pagadas , cuotas\_pendientes , total , mora) VALUES(?,?,?,?,?,?,?,?,?,?,?);");

ps.setString(1, clave);

ps.setString(2, nombre);

ps.setString(3, telefono);

ps.setString(4, direccion);

ps.setString(5, email);

ps.setString(6, fecha);

ps.setString(7, genero);

ps.setString(8, pagado);

ps.setString(9, String.valueOf(faltas));

ps.setString(10, String.valueOf(total));

ps.setString(11, mora);

int Resultado = ps.executeUpdate();

con.close();

if (Resultado > 0) {

JOptionPane.showMessageDialog(null, "Registro Guardado con Exito");

} else {

JOptionPane.showMessageDialog(null, "Registro no se guardo");

}

} else {

JOptionPane.showMessageDialog(null, "Clave de alumno ya existe");

} con.close();

} catch (HeadlessException | SQLException e) {

}

}

}

3.3.2Cófigo de la función mostrar datos que se encuentra en el jframe de ingresos

public void MostrarDatos() {

Connection Conexion\_MostrarDatos = null;

String[] Titulos = {"Clave", "Nombre", "Teléfono", "Direccion", "Email", "Fecha de Nacimiento", "genero",

"Pagos \_hechos","pagos\_pendientes","total","mora"};

String[] Registros = new String[11];

DefaultTableModel Model = new DefaultTableModel(null, Titulos);

try {

Conexion\_MostrarDatos = AJFVCconexion.getConnection();

ps = Conexion\_MostrarDatos.prepareStatement("SELECT \* FROM alumno");

rs = ps.executeQuery();

while (rs.next()) {

Registros[0] = rs.getString("clave");

Registros[1] = rs.getString("nombre");

Registros[2] = rs.getString("teléfono");

Registros[3] = rs.getString("dirección");

Registros[4] = rs.getString("email");

Registros[5] = rs.getString("fecha\_de\_nacimiento");

Registros[6] = rs.getString("genero");

Registros[7] = rs.getString("cuotas\_pagadas");

Registros[8] = rs.getString("cuotas\_pendientes");

Registros[9] = rs.getString("total");

Registros[10] = rs.getString("mora");

Model.addRow(Registros);

}

tabla.setModel(Model);

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, "Error al mostrar Datos" + e.getMessage());

}

}

Código del botón limpiar

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

// TODO add your handling code here:

Limpiar\_TextField();

}

Código de la función Limpiar\_TextField();

private void Limpiar\_TextField() {

txtclave.setText(null);

txtdirec.setText(null);

txtemail.setText(null);

txtnombre.setText(null);

txttel.setText(null);

txtvariable.setText(null);

}

Código del botón borrar

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

AJFVCupdate.borrar(txtclave.getText());

Código de la función borrar

public static void borrar(String clave) {

PreparedStatement ps;

ResultSet rs;

Connection con = null;

try {

con = AJFVCconexion.getConnection();

ps = con.prepareStatement("DELETE FROM alumno WHERE clave =?;");

ps.setString(1, clave);

int Resultado = ps.executeUpdate();

if (Resultado > 0) {

JOptionPane.showMessageDialog(null, "Registro borrado con Exito");

} else {

JOptionPane.showMessageDialog(null, "Registro no se guardo");

}

} catch (HeadlessException | SQLException e) {

}

}

}

Código del botón buscar

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

Connection Conexion\_Buscar = null;

String[] Titulos = {"Clave", "Nombre", "Teléfono", "Direccion", "Email", "Fecha de Nacimiento", "genero",

"Pagos \_hechos","pagos\_pendientes","total","mora"};

String[] Registros = new String[11];

DefaultTableModel Model = new DefaultTableModel(null, Titulos);

try {

Conexion\_Buscar = AJFVCconexion.getConnection();

ps = Conexion\_Buscar.prepareStatement("SELECT \* FROM alumno where clave =?");

ps.setString(1, txtclave.getText());

rs = ps.executeQuery();

if (rs.next()) {

txtclave.setText(rs.getString("clave"));

txtnombre.setText(rs.getString("nombre"));

txttel.setText(rs.getString("teléfono"));

txtdirec.setText(rs.getString("dirección"));

txtemail.setText(rs.getString("email"));

txtvariable.setText(rs.getString("fecha\_de\_nacimiento"));

ps = Conexion\_Buscar.prepareStatement("SELECT \* FROM alumno where clave =?");

ps.setString(1, txtclave.getText());

rs = ps.executeQuery();

while (rs.next()) {

Registros[0] = rs.getString("clave");

Registros[1] = rs.getString("nombre");

Registros[2] = rs.getString("teléfono");

Registros[3] = rs.getString("dirección");

Registros[4] = rs.getString("email");

Registros[5] = rs.getString("fecha\_de\_nacimiento");

Registros[6] = rs.getString("genero");

Registros[7] = rs.getString("cuotas\_pagadas");

Registros[8] = rs.getString("cuotas\_pendientes");

Registros[9] = rs.getString("total");

Registros[10] = rs.getString("mora");

Model.addRow(Registros);

}

tabla.setModel(Model);

} else {

JOptionPane.showMessageDialog(null, "No Existe la Clave solicitada");

}

Conexion\_Buscar.close();

} catch (HeadlessException | SQLException e) {

System.err.println(e);

}

// TODO add your handling code here:

}

Código del botón actualizar

private void limpiar1ActionPerformed(java.awt.event.ActionEvent evt) {

AJFVCupdate.actualizar(txtclave.getText(), txtnombre.getText(), txttel.getText(),

txtdirec.getText(), txtvariable.getText(), txtemail.getText(),genero.getSelectedItem().toString(),

cuotas.getSelectedItem().toString(), mora.getSelectedItem().toString());

MostrarDatos();

}

Código de la función actualizar

public class AJFVCupdate {

public static void actualizar(String clave, String nombre, String telefono, String direccion, String fecha, String email,

String genero, String pagado, String mora) {

PreparedStatement ps;

ResultSet rs;

Connection con = null;

double total = 0;

int pagos = Integer.parseInt(pagado);

if(mora.equals("SI")){

total = (pagos \* 3500)+(pagos\*35);

}else{

total = (pagos \* 3500);

}

int faltas =10 - pagos;

try {

con = AJFVCconexion.getConnection();

int rep = AJFVCrepetido.buscar(clave);

if (rep == 0) {

ps = con.prepareStatement("UPDATE alumno SET nombre =?,teléfono =?, dirección =?, email =?, "

+ "fecha\_de\_nacimiento =?, genero =?,cuotas\_pagadas =?, cuotas\_pendientes =?, total =?, mora =? WHERE clave=?;");

ps.setString(11, clave);

ps.setString(1, nombre);

ps.setString(2, telefono);

ps.setString(3, direccion);

ps.setString(4, email);

ps.setString(5, fecha);

ps.setString(6, genero);

ps.setString(7, pagado);

ps.setString(8, String.valueOf(faltas));

ps.setString(9, String.valueOf(total));

ps.setString(10, mora);

int Resultado = ps.executeUpdate();

con.close();

if (Resultado > 0) {

JOptionPane.showMessageDialog(null, "Registro Guardado con Exito");

} else {

JOptionPane.showMessageDialog(null, "Registro no se guardo");

}

} else {

JOptionPane.showMessageDialog(null, "Clave de alumno ya existe");

} con.close();

} catch (HeadlessException | SQLException e) {

}

}

Código usado para insertar imágenes

class FondoPanel extends JPanel {

public Image imagen;

@Override

public void paint(Graphics g) {

try {

imagen = new ImageIcon(getClass().getResource("/AJFVCimágenesvarias/fondo.jpg")).getImage();

} catch (NullPointerException e) {

}

g.drawImage(imagen, 0, 0, getWidth(), getHeight(), this);

setOpaque(false);

super.paint(g);

}

}

}

Código que verifica que no haya repetidos

public class AJFVCrepetido {

//esta clase se encarga de que no existan claves repetidas

public static int buscar(String clave){

Connection con = null;

con = AJFVCconexion.getConnection();

PreparedStatement ps;

ResultSet rs ;

try{

ps = con.prepareStatement("Select count(clave) from alumno where clave =?;");

ps.setString(1, clave);

rs=ps.executeQuery();

if(rs.next()){

return rs.getInt(1);

}

return 1;

}catch(SQLException e){

Logger.getLogger(AJFVCrepetido.class.getName()).log(Level.SEVERE, null, e);

return 1;

}

}

FUNCION Que busca los alumnos que pagaron

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

Connection con = null;

String[] Titulos = {"Clave", "Nombre", "Pagos \_hechos",};

String[] Registros = new String[3];

DefaultTableModel Model = new DefaultTableModel(null, Titulos);

try {

con = AJFVCconexion.getConnection();

ps = con.prepareStatement("SELECT clave, nombre, cuotas\_pagadas FROM alumno; ");

rs = ps.executeQuery();

while (rs.next()) {

Registros[0] = rs.getString("clave");

Registros[1] = rs.getString("nombre");

Registros[2] = rs.getString("cuotas\_pagadas");

Model.addRow(Registros);

}

tabla.setModel(Model);

con.close();

} catch (HeadlessException | SQLException e) {

System.err.println(e);

}

// TODO add your handling code here:

}

FUNCIONES Y BOTONES QUE REALIZAN LOS REPORTES

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

Connection con = null;

String[] Titulos = {"Clave", "Nombre", "Pagos \_hechos"};

String[] Registros = new String[3];

DefaultTableModel Model = new DefaultTableModel(null, Titulos);

try {

con = AJFVCconexion.getConnection();

ps = con.prepareStatement("SELECT clave, nombre, cuotas\_pagadas FROM alumno where clave =?; ");

ps.setString(1, clave.getText());

rs = ps.executeQuery();

while (rs.next()) {

Registros[0] = rs.getString("clave");

Registros[1] = rs.getString("nombre");

Registros[2] = rs.getString("cuotas\_pagadas");

Model.addRow(Registros);

}

tabla.setModel(Model);

con.close();

} catch (HeadlessException | SQLException e) {

System.err.println(e);

}

// TODO add your handling code here:

}

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

Connection con = null;

String[] Titulos = {"Clave", "Nombre", "Pagos \_faltantes",};

String[] Registros = new String[3];

DefaultTableModel Model = new DefaultTableModel(null, Titulos);

try {

con = AJFVCconexion.getConnection();

ps = con.prepareStatement("SELECT clave, nombre, cuotas\_pendientes FROM alumno where clave =?; ");

ps.setString(1, clave.getText());

rs = ps.executeQuery();

while (rs.next()) {

Registros[0] = rs.getString("clave");

Registros[1] = rs.getString("nombre");

Registros[2] = rs.getString("cuotas\_pendientes");

Model.addRow(Registros);

}

tabla.setModel(Model);

con.close();

} catch (HeadlessException | SQLException e) {

System.err.println(e);

}

// TODO add your handling code here:

}

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

Connection con = null;

String[] Titulos = {"Clave", "Nombre", "Pagos \_hechos"};

String[] Registros = new String[3];

DefaultTableModel Model = new DefaultTableModel(null, Titulos);

try {

con = AJFVCconexion.getConnection();

ps = con.prepareStatement("SELECT nombre, cuotas\_pagadas FROM alumno where clave =?; ");

ps.setString(1, clave.getText());

rs = ps.executeQuery();

while (rs.next()) {

Registros[1] = rs.getString("nombre");

Registros[2] = rs.getString("cuotas\_pagadas");

}

JOptionPane.showMessageDialog(null, "El Alumno " + " " + Registros[1] + " " + "ha realizado : \n" + Registros[2] + " " + "Cuotas.");

con.close();

} catch (HeadlessException | SQLException e) {

System.err.println(e);

}

// TODO add your handling code here:

}

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

Connection con = null;

String[] Titulos = {"Clave", "Nombre", "Pagos \_Pendientes",};

String[] Registros = new String[3];

DefaultTableModel Model = new DefaultTableModel(null, Titulos);

try {

con = AJFVCconexion.getConnection();

ps = con.prepareStatement("SELECT clave, nombre, cuotas\_pendientes FROM alumno; ");

rs = ps.executeQuery();

while (rs.next()) {

Registros[0] = rs.getString("clave");

Registros[1] = rs.getString("nombre");

Registros[2] = rs.getString("cuotas\_pendientes");

Model.addRow(Registros);

}

tabla.setModel(Model);

con.close();

} catch (HeadlessException | SQLException e) {

System.err.println(e);

}

// TODO add your handling code here:

}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

Connection con = null;

String[] Registros = new String[3];

try {

con = AJFVCconexion.getConnection();

ps = con.prepareStatement("SELECT nombre, cuotas\_pendientes FROM alumno where clave =?; ");

ps.setString(1, clave.getText());

rs = ps.executeQuery();

while (rs.next()) {

Registros[1] = rs.getString("nombre");

Registros[2] = rs.getString("cuotas\_pendientes");

}

JOptionPane.showMessageDialog(null, "El Alumno" + " " + Registros[1] + " " + " le falta cancelar : \n" + Registros[2] + " " + "Cuotas.");

con.close();

} catch (HeadlessException | SQLException e) {

System.err.println(e);

} // TODO add your handling code here:

}

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {

Connection con = null;

String[] Titulos = {"Clave", "Nombre", "Saldo\_actual",};

String[] Registros = new String[3];

DefaultTableModel Model = new DefaultTableModel(null, Titulos);

try {

con = AJFVCconexion.getConnection();

ps = con.prepareStatement("SELECT clave, nombre, total FROM alumno where clave =?;");

ps.setString(1, clave.getText());

rs = ps.executeQuery();

while (rs.next()) {

Registros[0] = rs.getString("clave");

Registros[1] = rs.getString("nombre");

Registros[2] = "Q" + rs.getString("total");

Model.addRow(Registros);

}

tabla.setModel(Model);

con.close();

} catch (HeadlessException | SQLException e) {

System.err.println(e);

}

// TODO add your handling code here:

}

private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {

Connection con = null;

String[] Titulos = {"Clave", "Nombre", "Mora","Total csin mora"};

String[] Registros = new String[4];

DefaultTableModel Model = new DefaultTableModel(null, Titulos);

try {

con = AJFVCconexion.getConnection();

ps = con.prepareStatement("SELECT clave, nombre, mora, total FROM alumno where mora ='No';");

rs = ps.executeQuery();

while (rs.next()) {

Registros[0] = rs.getString("clave");

Registros[1] = rs.getString("nombre");

Registros[2] = rs.getString("mora");

Registros[3] = rs.getString("total");

Model.addRow(Registros);

}

tabla.setModel(Model);

con.close();

} catch (HeadlessException | SQLException e) {

System.err.println(e);

}

// TODO add your handling code here:

}

private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {

Connection con = null;

String[] Titulos = {"Clave", "Nombre", "Mora","Total con mora"};

String[] Registros = new String[4];

DefaultTableModel Model = new DefaultTableModel(null, Titulos);

try {

con = AJFVCconexion.getConnection();

ps = con.prepareStatement("SELECT clave, nombre, mora, total FROM alumno where mora ='SI';");

rs = ps.executeQuery();

while (rs.next()) {

Registros[0] = rs.getString("clave");

Registros[1] = rs.getString("nombre");

Registros[2] = rs.getString("mora");

Registros[3] = rs.getString("total");

Model.addRow(Registros);

}

tabla.setModel(Model);

con.close();

} catch (HeadlessException | SQLException e) {

System.err.println(e);

}

// TODO add your handling code here:

}

Querys usados para la manipulación de base de datos

create table if not exists alumno(clave int(4) NOT NULL, nombre varchar(50), teléfono INT(8),dirección varchar(30), email varchar(30), fecha\_de\_nacimiento VARCHAR(30), genero varchar(10), cuotas\_pagadas int(2), cuotas\_pendientes int(2), total varchar(10), mora varchar(2));

SELECT \* FROM alumno where clave =?;

INSERT INTO alumno (clave,nombre,teléfono, dirección, email, fecha\_de\_nacimiento, genero,cuotas\_pagadas , cuotas\_pendientes , total , mora) VALUES(?,?,?,?,?,?,?,?,?,?,?);

Select count(clave) from alumno where clave =?;

DELETE FROM alumno WHERE clave =?;

SELECT clave, nombre, cuotas\_pagadas FROM alumno;