Componente de conexión

En esta clase se controla la salida de las ventanas del sistema y la ejecución en segundo plano de algunas, no recibe parámetros

```
package bi;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Arrays;

import javax.swing.JOptionPane;

public class Conexion {

  VentanaMain ventana = new VentanaMain();
  static JOptionPane jop = new JOptionPane();

  public static Connection getConnection() {

        String user = "Angel";
        String password = "muu295t";
        String ruta = "jdbc:mysql://localhost/proyecto";
        Boolean x = true;

        try {
                Class.forName("com.mysql.jdbc.Driver"); // se carga el driver

                return DriverManager.getConnection(ruta, user, password);

        } catch (ClassNotFoundException e) {
                System.out.println("Error al cargar el driver");
                jop.showMessageDialog(null,
                                "Error al cargar el driver, el JDBC no esta ",
                                "Error critico", JOptionPane.ERROR_MESSAGE, null);

                x = false;
        } catch (SQLException e) {
            System.out.println("Error en al conexion con XAMPP");
            jop.showMessageDialog(null, "Error en la conexion con XAMPP ",
                    "Error critico", JOptionPane.ERROR_MESSAGE, null);

            x = false;
        }

        if (x == true) {

        }

        return null;

    }

    BufferedReader tcl = new BufferedReader(new InputStreamReader(System.in));

    int matricula;
    String nombre, grupo;
    MainPrincipal main = new MainPrincipal();
    public boolean seEncontro;

    public void altas() throws IOException {
        Connection con = Conexion.getConnection();

        try {
            PreparedStatement pstmt = con
                    .prepareStatement("insert into
alumnos(nombre,matricula,grupo)values(?,?,?)");

            System.out.println("nombre del alumno:");
            nombre = tcl.readLine();
            System.out.println("matricula del alumno:");
            matricula = Integer.parseInt(tcl.readLine());
            System.out.println("grupo del alumno:");
            grupo = tcl.readLine();

            pstmt.setString(1, nombre);
            pstmt.setInt(2, matricula);
            pstmt.setString(3, grupo);
            pstmt.executeUpdate();

        } catch (SQLException e) {
            System.out.println("ocurrio algun error");
            jop.showMessageDialog(null, "Ocurrio algun error", "Error general",
                    JOptionPane.ERROR_MESSAGE, null);

        }

    }

    public void mostrar() {

    }

    public void borrar() {
        Connection con = Conexion.getConnection();
        try {
            PreparedStatement pstmt = con
                    .prepareStatement("delete from alumnos where matricula=?");

            pstmt.setString(1, nombre);
            pstmt.setInt(2, matricula);
            pstmt.setString(3, grupo);
            pstmt.executeUpdate();
            pstmt.executeUpdate();

            pstmt.executeUpdate();

        } catch (SQLException e) {
            System.out.println("ocurrio un error");
            jop.showMessageDialog(null, "Ocurrio algun error", "Error general",
                    JOptionPane.ERROR_MESSAGE, null);
        }
    }

    public void modificar() {
        Connection con = Conexion.getConnection();
        try {
            PreparedStatement pstmt = con
                    .prepareStatement("update alumnos set nombre, matricula=?
where grupo");

            pstmt.setString(1, nombre);
            pstmt.setInt(2, matricula);
            pstmt.setString(3, grupo);
            pstmt.executeUpdate();

        } catch (SQLException e) {
            System.out.println("ocurrio un error");
```

```
                jop.showMessageDialog(null, "Ocurrio algun error", "Error general",
                        JOptionPane.ERROR_MESSAGE, null);
            }
    }

    public void consultas() throws IOException {
            String aux;
            Connection con = Conexion.getConnection();
            try {
                Statement st = con.createStatement();
                System.out.println("digite la matricula del alumno a consultar");
                aux = tcl.readLine();
                matricula = Integer.parseInt(aux);
                ResultSet tabla = st
                        .executeQuery("select * from alumnos where matricula="
                                + matricula);
                if (tabla.next()) {
                    System.out.println(" matricula: " + tabla.getInt("matricula")
                            + "\n");
                    System.out.println("nombre: " + tabla.getString("nombre"));
                    System.out.println("grupo: " + tabla.getFloat("grupo") + "\n");

                }
            } catch (SQLException e) {
                System.out.println("Error al consultar");
                jop.showMessageDialog(null, "Error al consultar", "Error general",
                        JOptionPane.ERROR_MESSAGE, null);
            }
    }

    VentanaMain hora = new VentanaMain();

    public void acceso(String codigoDeBarras) {

            String aux;
            Connection con = Conexion.getConnection();
            try {
                Statement st = con.createStatement();
                System.out.println("si");
                System.out.println(codigoDeBarras);

                ResultSet tabla = st
                        .executeQuery("select * from admins where
alumno_no_control =("
                                + codigoDeBarras + ")");
                if (tabla.next()) {
```

```
                main.OcultarVentanaPrincipal();

            hora.initialize();

                String etiqueta = tabla.getString("alumno_nombre");

                hora.label.setText(etiqueta);

            }

        } catch (SQLException s) {
            System.out.println("Error al acceder");
            jop.showMessageDialog(null,
                    "Error al acceder, la matricula no es la correcta",
                    "Error general", JOptionPane.ERROR_MESSAGE, null);

        }

    }

    static ResultSet registro = null;

    public static void generarArrays() throws SQLException {

    }

}
```

Main Principal
Es en donde se instancia el primer método para cargar la pantalla de inicio

```
package bi;

import java.awt.EventQueue;

public class MainPrincipal {

    public static boolean pasar = false;

    static VentanaAgregarAlumno frame1 = new VentanaAgregarAlumno();

    static VentanaAgregarGrupo frame2 = new VentanaAgregarGrupo();

    static VentanaAgregarMaestro frame3 = new VentanaAgregarMaestro();
```

```
}

static VentanaBaseDeDatos frame4 = new VentanaBaseDeDatos();

static VentanaConsultar frame5 = new VentanaConsultar();

static VentanaGrupos frame6 = new VentanaGrupos();

static VentanaPrincipal frame7 = new VentanaPrincipal();

static VentanaMain frame8 = new VentanaMain();

static MeinRealReal lareal = new MeinRealReal();

static VentanaGenerarGrafica frame9 = new VentanaGenerarGrafica();

public static void MonstrarVentanaAgregarAlumno() {

        frame1.setVisible(true);
}

public void OcultarVentanaAgregarAlumno() {

        frame1.setVisible(false);
}

public static void MostrarVentanaAgregarGrupo() {

        frame2.setVisible(true);
}

public static void OcultarVentanaAgregarGrupo() {

        frame2.setVisible(false);
}

public static void MostrarVentanaAgregarMaestro() {

        frame3.setVisible(true);
}

public static void OcultarVentanaAgregarMaestro() {

        frame3.setVisible(false);
}

public static void MostrarVentanaBaseDeDatos() {
        frame4.setVisible(true);
}

public static void OcultarVentanaBaseDeDatos() {

        frame4.setVisible(false);
}

public static void MostrarVentanaConsultar() {

        frame5.setVisible(true);
}

public static void OcultarVentanaConsultar() {

        frame5.setVisible(false);
}

public static void MostrarVentanaGrupos() {

        frame6.setVisible(true);
}

public static void OcultarVentanaGrupos() {

        frame6.setVisible(false);
}

public static void MostrarVentanaPrincipal() {

        frame7.setVisible(true);
        frame7.setEnabled(true);

}

public static void OcultarVentanaPrincipal() {

        frame7.setVisible(false);
        frame7.enable(false);
        frame7.hide();

}

public static void MostrarVentanaMain() {
        frame8.setVisible(true);// ///////// ---- aqui se nesesitan instanciar
                                                          // todas las cosas para el frame
    }

    public static void OcultarVentanaMain() {

        frame8.panel.setVisible(false);

        frame8.setVisible(false);

        frame8.panel.enable(false);
        frame8.panel.hide();
        System.exit(0);

    }
    public static void MostrarVentanaGenerarGrafica(){
        frame9.setVisible(true);
    }
    public static void ocultarVentanaGenerarGrafica(){
        frame9.setVisible(false);
    }
}
```

Componente Agregar Alumno:

Este es el componente encargado de agregar alumnos a la ventana principal que por alguna razón no esté en la base de datos principal o sea un visitante. Esta ventana no da de alta alumnos en la base de datos, es solo para agregar una persona ajena al sistema.

```
package bi;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JTextArea;
import javax.swing.JToolBar;
import javax.swing.JDesktopPane;
import javax.swing.JInternalFrame;
import java.awt.Choice;
import java.awt.Label;
import javax.swing.JComboBox;
import javax.swing.JButton;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

public class VentanaAgregarAlumno extends JFrame {

    private JPanel contentPane;
    private JTextField textField;
    private JTextField textField_1;
    private JTextField textField_2;

    public VentanaAgregarAlumno() {
            setTitle("Agregar Alumnos-Sistema de control");
            setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            setBounds(100, 100, 609, 436);
            contentPane = new JPanel();
            contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
            setContentPane(contentPane);
            contentPane.setLayout(null);

            JLabel lblNewLabel = new JLabel("Nombre");
            lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 14));
            lblNewLabel.setBounds(25, 31, 95, 21);
            contentPane.add(lblNewLabel);

            textField = new JTextField();
            textField.setBounds(24, 63, 226, 21);
            contentPane.add(textField);
            textField.setColumns(10);

            textField_1 = new JTextField();
            textField_1.setColumns(10);
            textField_1.setBounds(24, 131, 226, 21);
            contentPane.add(textField_1);

            textField_2 = new JTextField();
            textField_2.setColumns(10);
            textField_2.setBounds(330, 63, 198, 21);
            contentPane.add(textField_2);

            JLabel lblMatricula = new JLabel("Matricula");
```

```
        lblMatricula.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblMatricula.setBounds(335, 31, 95, 21);
        contentPane.add(lblMatricula);

        JLabel lblGrupo = new JLabel("Grupo");
        lblGrupo.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblGrupo.setBounds(330, 95, 95, 21);
        contentPane.add(lblGrupo);

        Choice choice = new Choice();
        choice.addMouseListener(new MouseAdapter() {
                @Override
                public void mouseClicked(MouseEvent arg0) {
                        // evento textfield con triangulito
                }
        });
        choice.setBounds(330, 131, 198, 21);
        contentPane.add(choice);

        JButton btnNewButton = new JButton("Agregar");
        btnNewButton.addMouseListener(new MouseAdapter() {
                @Override
                public void mouseClicked(MouseEvent arg0) {
                        // evento boton agregar
                }
        });
        btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 14));
        btnNewButton.setBounds(222, 323, 107, 47);
        contentPane.add(btnNewButton);

        JButton btnCancelar = new JButton("Cancelar");
        btnCancelar.addMouseListener(new MouseAdapter() {
                @Override
                public void mouseClicked(MouseEvent arg0) {
                        // evento boton cancelar
                }
        });
        btnCancelar.setFont(new Font("Tahoma", Font.BOLD, 14));
        btnCancelar.setBounds(340, 323, 107, 47);
        contentPane.add(btnCancelar);

        JButton btnOk = new JButton("OK");
        btnOk.setFont(new Font("Tahoma", Font.BOLD, 14));
        btnOk.setBounds(459, 323, 107, 47);
        contentPane.add(btnOk);
}
```

```
}
```

Componente agregar grupos

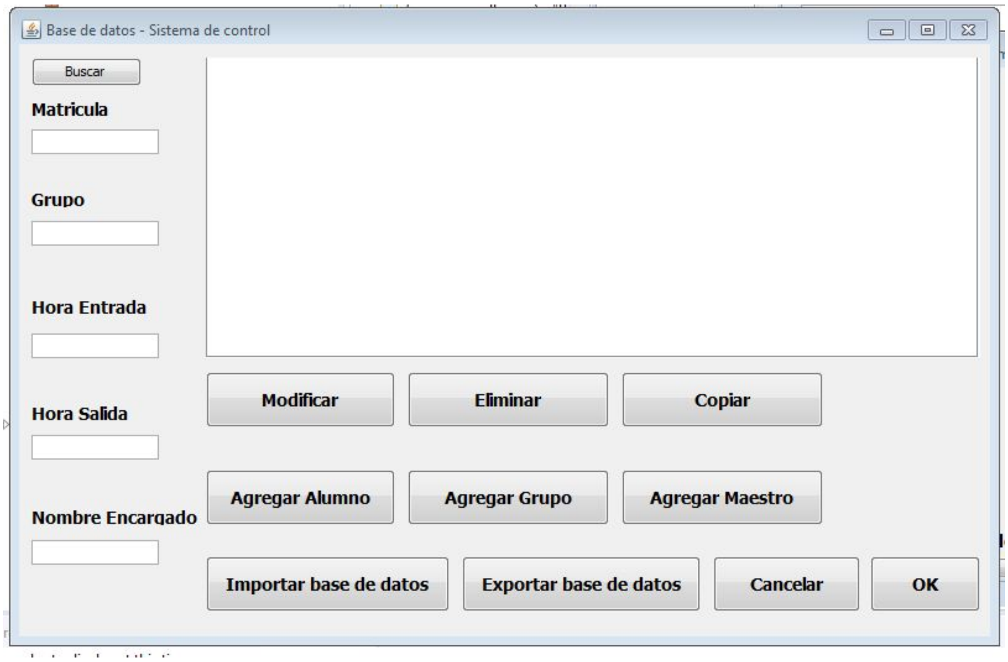
Este es el componente encargado de agregar grupos a la ventana principal que por alguna razón no esté en la base de datos principal o sean visitantes. Esta ventana no da de alta grupos en la base de datos, es solo para agregar un grupo ajeno al sistema.

```
package bi;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.JTextField;
import java.awt.Choice;
import javax.swing.JButton;
import javax.swing.JTable;
import javax.swing.JScrollPane;
import javax.swing.table.DefaultTableModel;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.MouseMotionAdapter;
import java.awt.Color;
import java.sql.SQLException;

public class VentanaAgregarGrupo extends JFrame {

   private JPanel contentPane;
   private JTextField textField;
   private JTextField textField_1;
   MainPrincipal main = new MainPrincipal();
   VentanaMain ventana = new VentanaMain();
   Conexion conectar = new Conexion();

   public VentanaAgregarGrupo() {
         setBackground(new Color(240, 248, 255));
         setAlwaysOnTop(true);
```

```
setTitle("Agregar Grupo");
setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
setBounds(100, 100, 670, 473);
contentPane = new JPanel();
contentPane.setBackground(new Color(240, 248, 255));
contentPane.addMouseMotionListener(new MouseMotionAdapter() {
        @Override
        public void mouseDragged(MouseEvent e) {

                VentanaAgregarGrupo.this.setLocation(
                                VentanaAgregarGrupo.this.getX() + e.getX(),
                                VentanaAgregarGrupo.this.getY() + e.getY());
        }
});
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);
this.setUndecorated(true);
JLabel lblNombre = new JLabel("Nombre");
lblNombre.setFont(new Font("Tahoma", Font.BOLD, 14));
lblNombre.setBounds(10, 29, 91, 22);
contentPane.add(lblNombre);

JLabel lblMatricula = new JLabel("Matricula");
lblMatricula.setFont(new Font("Tahoma", Font.BOLD, 14));
lblMatricula.setBounds(223, 29, 91, 22);
contentPane.add(lblMatricula);

JLabel lblGrupo = new JLabel("Grupo");
lblGrupo.setFont(new Font("Tahoma", Font.BOLD, 14));
lblGrupo.setBounds(410, 29, 91, 22);
contentPane.add(lblGrupo);

textField = new JTextField();
textField.setBounds(10, 62, 171, 22);
contentPane.add(textField);
textField.setColumns(10);

textField_1 = new JTextField();
textField_1.setColumns(10);
textField_1.setBounds(223, 63, 155, 22);
contentPane.add(textField_1);

final Choice choice = new Choice();
choice.setBounds(410, 64, 128, 20);
choice.addItem("grupo_3btps");
```

```
        choice.addItem("grupo_3bmt");
        contentPane.add(choice);

        JPanel panel = new JPanel();
        panel.setBackground(new Color(240, 248, 255));
        panel.setBounds(0, 110, 654, 112);
        contentPane.add(panel);
        panel.setLayout(null);

        JButton btnImportarBasededatos = new JButton("Importar BD");
        btnImportarBasededatos.setFont(new Font("Tahoma", Font.BOLD, 14));
        btnImportarBasededatos.setBounds(10, 27, 151, 50);
        panel.add(btnImportarBasededatos);

        JButton btnExportarBd = new JButton("Exportar BD");
        btnExportarBd.setFont(new Font("Tahoma", Font.BOLD, 14));
        btnExportarBd.setBounds(171, 27, 151, 50);
        panel.add(btnExportarBd);

        JButton btnCancelar = new JButton("Cancelar");
        btnCancelar.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent arg0) {
                main.OcultarVentanaAgregarGrupo();
            }
        });
        btnCancelar.setFont(new Font("Tahoma", Font.BOLD, 14));
        btnCancelar.setBounds(332, 27, 151, 50);
        panel.add(btnCancelar);

        JButton btnAgregar = new JButton("Agregar");
        btnAgregar.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {

            }
        });
        btnAgregar.setFont(new Font("Tahoma", Font.BOLD, 14));
        btnAgregar.setBounds(493, 27, 151, 50);
        panel.add(btnAgregar);
  }
}
```

Componente Base de datos
Este componente es una visualización de la instancia de la base de datos actual



```
package bi;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JButton;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JTabbedPane;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.MouseMotionAdapter;

public class VentanaBaseDeDatos extends JFrame {

   private JPanel contentPane;
   private JTextField textField;
   private JTextField textField_1;
   private JTextField textField_2;
   private JTextField textField_3;
   private JTextField textField_4;
```

```
public VentanaBaseDeDatos() {
	setTitle("Base de datos - Sistema de control");
	setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
	setBounds(100, 100, 801, 517);
	contentPane = new JPanel();
	contentPane.addMouseMotionListener(new MouseMotionAdapter() {
		@Override
		public void mouseDragged(MouseEvent e) {
			contentPane.setLocation(contentPane.getX() + e.getX(),
					contentPane.getY() + e.getY());
		}
	});
	contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
	setContentPane(contentPane);
	contentPane.setLayout(null);

	JButton button = new JButton("Buscar");
	button.addMouseListener(new MouseAdapter() {
		@Override
		public void mouseClicked(MouseEvent arg0) {
			// evento boton buscar
		}
	});
	button.setBounds(10, 11, 89, 23);
	contentPane.add(button);

	JLabel label = new JLabel("Matricula");
	label.setFont(new Font("Tahoma", Font.BOLD, 14));
	label.setBounds(10, 45, 75, 14);
	contentPane.add(label);

	textField = new JTextField();
	textField.setColumns(10);
	textField.setBounds(10, 69, 103, 20);
	contentPane.add(textField);

	textField_1 = new JTextField();
	textField_1.setColumns(10);
	textField_1.setBounds(10, 143, 103, 20);
	contentPane.add(textField_1);

	JLabel label_1 = new JLabel("Grupo");
	label_1.setFont(new Font("Tahoma", Font.BOLD, 14));
	label_1.setBounds(10, 118, 75, 14);
	contentPane.add(label_1);
```

```
JLabel label_2 = new JLabel("Hora Entrada");
label_2.setFont(new Font("Tahoma", Font.BOLD, 14));
label_2.setBounds(10, 205, 108, 14);
contentPane.add(label_2);

JLabel label_3 = new JLabel("Hora Salida");
label_3.setFont(new Font("Tahoma", Font.BOLD, 14));
label_3.setBounds(10, 291, 103, 14);
contentPane.add(label_3);

textField_2 = new JTextField();
textField_2.setColumns(10);
textField_2.setBounds(10, 234, 103, 20);
contentPane.add(textField_2);

textField_3 = new JTextField();
textField_3.setColumns(10);
textField_3.setBounds(10, 316, 103, 20);
contentPane.add(textField_3);

JButton btnAgreagarAlumno = new JButton("Agregar Alumno");
btnAgreagarAlumno.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
                // evento boton agregar alumno
        }
});
btnAgreagarAlumno.setFont(new Font("Tahoma", Font.BOLD, 14));
btnAgreagarAlumno.setBounds(151, 344, 153, 45);
contentPane.add(btnAgreagarAlumno);

JButton btnEliminar = new JButton("Eliminar");
btnEliminar.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
                // evento boton eliminar
        }
});
btnEliminar.setFont(new Font("Tahoma", Font.BOLD, 14));
btnEliminar.setBounds(314, 265, 163, 45);
contentPane.add(btnEliminar);

JLabel label_4 = new JLabel("Nombre Encargado");
label_4.setFont(new Font("Tahoma", Font.BOLD, 14));
label_4.setBounds(10, 375, 147, 14);
```

```
contentPane.add(label_4);

textField_4 = new JTextField();
textField_4.setColumns(10);
textField_4.setBounds(10, 401, 103, 20);
contentPane.add(textField_4);

JButton button_3 = new JButton("Copiar");
button_3.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
                // evento boton copiar
        }
});
button_3.setFont(new Font("Tahoma", Font.BOLD, 14));
button_3.setBounds(487, 265, 163, 45);
contentPane.add(button_3);

JButton button_4 = new JButton("OK");
button_4.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
                // evento boton ok
        }
});
button_4.setFont(new Font("Tahoma", Font.BOLD, 14));
button_4.setBounds(688, 414, 89, 45);
contentPane.add(button_4);

JButton btnModificar = new JButton("Modificar");
btnModificar.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
                // evento boton modificar
        }
});
btnModificar.setFont(new Font("Tahoma", Font.BOLD, 14));
btnModificar.setBounds(151, 265, 153, 45);
contentPane.add(btnModificar);

JTabbedPane tabbedPane = new JTabbedPane(JTabbedPane.TOP);
tabbedPane.setBounds(151, 11, 626, 243);
contentPane.add(tabbedPane);

JButton btnImportarBaseDe = new JButton("Importar base de datos");
btnImportarBaseDe.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
        // evento boton importar base de datos
    }
});
btnImportarBaseDe.setFont(new Font("Tahoma", Font.BOLD, 14));
btnImportarBaseDe.setBounds(151, 414, 197, 45);
contentPane.add(btnImportarBaseDe);

JButton btnAgregarGrupo = new JButton("Agregar Grupo");
btnAgregarGrupo.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
        // evento boton agregar grupo
    }
});
btnAgregarGrupo.setFont(new Font("Tahoma", Font.BOLD, 14));
btnAgregarGrupo.setBounds(314, 344, 163, 45);
contentPane.add(btnAgregarGrupo);

JButton btnExportarBaseDe = new JButton("Exportar base de datos");
btnExportarBaseDe.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
        // evento boton exportar base de datos
    }
});
btnExportarBaseDe.setFont(new Font("Tahoma", Font.BOLD, 14));
btnExportarBaseDe.setBounds(358, 414, 193, 45);
contentPane.add(btnExportarBaseDe);

JButton btnAgregarMaestro = new JButton("Agregar Maestro");
btnAgregarMaestro.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
        // evento boton agregar maestro
    }
});
btnAgregarMaestro.setFont(new Font("Tahoma", Font.BOLD, 14));
btnAgregarMaestro.setBounds(487, 344, 163, 45);
contentPane.add(btnAgregarMaestro);

JButton btnCancelar = new JButton("Cancelar");
btnCancelar.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
```

```
                        // evento boton cancelar
                }
            });
            btnCancelar.setFont(new Font("Tahoma", Font.BOLD, 14));
            btnCancelar.setBounds(561, 414, 119, 45);
            contentPane.add(btnCancelar);
    }

}
```

Componente consultar
Este componente es el encargado de consultar en la base de datos toda la informacion relacionada con un alumno

```
package bi;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JButton;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JTabbedPane;
import javax.swing.JTextArea;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

public class VentanaConsultar extends JFrame {

    private JPanel contentPane;
    private JTextField textField;
    private JTextField textField_1;
    private JTextField textField_2;
    private JTextField textField_3;
    private JTextField textField_4;

    public VentanaConsultar() {
            setTitle("Consulta de alumnos");
            setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            setBounds(100, 100, 802, 519);
            contentPane = new JPanel();
```

```
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);

JButton btnBuscar = new JButton("Buscar");
btnBuscar.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
        // evento boton buscar
    }
});
btnBuscar.setBounds(10, 11, 89, 23);
contentPane.add(btnBuscar);

JLabel lblNombre = new JLabel("Nombre");
lblNombre.setFont(new Font("Tahoma", Font.BOLD, 14));
lblNombre.setBounds(10, 57, 89, 14);
contentPane.add(lblNombre);

JLabel lblMatricula = new JLabel("Matricula");
lblMatricula.setFont(new Font("Tahoma", Font.BOLD, 14));
lblMatricula.setBounds(10, 129, 75, 14);
contentPane.add(lblMatricula);

textField = new JTextField();
textField.setBounds(10, 83, 103, 20);
contentPane.add(textField);
textField.setColumns(10);

textField_1 = new JTextField();
textField_1.setColumns(10);
textField_1.setBounds(10, 153, 103, 20);
contentPane.add(textField_1);

textField_2 = new JTextField();
textField_2.setColumns(10);
textField_2.setBounds(10, 227, 103, 20);
contentPane.add(textField_2);

JLabel lblGrupo = new JLabel("Grupo");
lblGrupo.setFont(new Font("Tahoma", Font.BOLD, 14));
lblGrupo.setBounds(10, 202, 75, 14);
contentPane.add(lblGrupo);

JLabel lblHoraEntrada = new JLabel("Hora Entrada");
lblHoraEntrada.setFont(new Font("Tahoma", Font.BOLD, 14));
lblHoraEntrada.setBounds(10, 289, 108, 14);
contentPane.add(lblHoraEntrada);

JLabel lblHoraSalida = new JLabel("Hora Salida");
lblHoraSalida.setFont(new Font("Tahoma", Font.BOLD, 14));
lblHoraSalida.setBounds(10, 375, 103, 14);
contentPane.add(lblHoraSalida);

textField_3 = new JTextField();
textField_3.setColumns(10);
textField_3.setBounds(10, 318, 103, 20);
contentPane.add(textField_3);

textField_4 = new JTextField();
textField_4.setColumns(10);
textField_4.setBounds(10, 400, 103, 20);
contentPane.add(textField_4);

JTabbedPane tabbedPane = new JTabbedPane(JTabbedPane.TOP);
tabbedPane.setBounds(130, 0, 656, 420);
contentPane.add(tabbedPane);

JButton btnNewButton = new JButton("OK");
btnNewButton.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
                // evento boton ok
        }
});
btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 14));
btnNewButton.setBounds(687, 425, 89, 45);
contentPane.add(btnNewButton);

JButton btnCancelar = new JButton("Cancelar");
btnCancelar.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
                // evento boton no se acuerda
        }
});
btnCancelar.setFont(new Font("Tahoma", Font.BOLD, 14));
btnCancelar.setBounds(554, 425, 107, 45);
contentPane.add(btnCancelar);

JButton btnCopiar = new JButton("Copiar");
btnCopiar.addMouseListener(new MouseAdapter() {
```

```
            @Override
            public void mouseClicked(MouseEvent arg0) {
                // evento boton copiar
            }
        });
        btnCopiar.setFont(new Font("Tahoma", Font.BOLD, 14));
        btnCopiar.setBounds(455, 425, 89, 45);
        contentPane.add(btnCopiar);
    }
}
```

Ventana grupos;
Es el encargado de mostrar o agregar grupos a la ventana principal

```
package bi;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JButton;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JTabbedPane;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

public class VentanaGrupos extends JFrame {

    private JPanel contentPane;
    private JTextField textField_1;
    private JTextField textField_2;
    private JTextField textField_3;
    private JTextField textField_4;
    private JTextField textField;

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    VentanaGrupos frame = new VentanaGrupos();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    public VentanaGrupos() {
        setTitle("Grupo-Sistema de control");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 803, 513);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JButton button = new JButton("Buscar");
        button.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent arg0) {
                // evento boton buscar
            }
        });
        button.setBounds(10, 11, 89, 23);
        contentPane.add(button);

        JLabel label_1 = new JLabel("Matricula");
        label_1.setFont(new Font("Tahoma", Font.BOLD, 14));
        label_1.setBounds(10, 45, 75, 14);
        contentPane.add(label_1);

        textField_1 = new JTextField();
        textField_1.setColumns(10);
        textField_1.setBounds(10, 69, 103, 20);
        contentPane.add(textField_1);

        textField_2 = new JTextField();
        textField_2.setColumns(10);
        textField_2.setBounds(10, 143, 103, 20);
        contentPane.add(textField_2);

        JLabel label_2 = new JLabel("Grupo");
        label_2.setFont(new Font("Tahoma", Font.BOLD, 14));
        label_2.setBounds(10, 118, 75, 14);
```

```
contentPane.add(label_2);

JLabel label_3 = new JLabel("Hora Entrada");
label_3.setFont(new Font("Tahoma", Font.BOLD, 14));
label_3.setBounds(10, 205, 108, 14);
contentPane.add(label_3);

JLabel label_4 = new JLabel("Hora Salida");
label_4.setFont(new Font("Tahoma", Font.BOLD, 14));
label_4.setBounds(10, 291, 103, 14);
contentPane.add(label_4);

textField_3 = new JTextField();
textField_3.setColumns(10);
textField_3.setBounds(10, 234, 103, 20);
contentPane.add(textField_3);

textField_4 = new JTextField();
textField_4.setColumns(10);
textField_4.setBounds(10, 316, 103, 20);
contentPane.add(textField_4);

JButton btnExportarPdf = new JButton("Exportar PDF");
btnExportarPdf.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
                // evento boton exportar PDF
        }
});
btnExportarPdf.setFont(new Font("Tahoma", Font.BOLD, 14));
btnExportarPdf.setBounds(428, 425, 142, 45);
contentPane.add(btnExportarPdf);

JButton btnExpandir = new JButton("Expandir");
btnExpandir.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
                // evento boton expandir
        }
});
btnExpandir.setFont(new Font("Tahoma", Font.BOLD, 14));
btnExpandir.setBounds(317, 425, 101, 45);
contentPane.add(btnExpandir);

JLabel lblNombreEncargado = new JLabel("Nombre Encargado");
lblNombreEncargado.setFont(new Font("Tahoma", Font.BOLD, 14));
```

```
        lblNombreEncargado.setBounds(10, 375, 147, 14);
        contentPane.add(lblNombreEncargado);

        textField = new JTextField();
        textField.setColumns(10);
        textField.setBounds(10, 401, 103, 20);
        contentPane.add(textField);

        JButton button_3 = new JButton("Copiar");
        button_3.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent arg0) {
                // evento boton copiar
            }
        });
        button_3.setFont(new Font("Tahoma", Font.BOLD, 14));
        button_3.setBounds(580, 425, 89, 45);
        contentPane.add(button_3);

        JButton btnOk = new JButton("OK");
        btnOk.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseExited(MouseEvent arg0) {
                // evento boton OK
            }
        });
        btnOk.setFont(new Font("Tahoma", Font.BOLD, 14));
        btnOk.setBounds(688, 425, 89, 45);
        contentPane.add(btnOk);

        JButton button_5 = new JButton("");
        button_5.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                // evento boton sin nombre
            }
        });
        button_5.setFont(new Font("Tahoma", Font.BOLD, 14));
        button_5.setBounds(218, 425, 89, 45);
        contentPane.add(button_5);

        JTabbedPane tabbedPane = new JTabbedPane(JTabbedPane.TOP);
        tabbedPane.setBounds(151, 11, 626, 403);
        contentPane.add(tabbedPane);
    }
}
```

Ventana principal y componente básico de seguridad

Es el componente de inicio de sesion encargado de buscar el nombre de usuario y la contraseña en la base de datos para abrir una nueva sesión en el sistema. Si no lo encuentra no se puede abrir la ventana de trabajo. El inicio de sesión se hace por parámetros.

```
package bi;
import java.awt.BorderLayout;
import java.awt.EventQueue;
import java.awt.Frame;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import java.awt.Window.Type;
import java.awt.Color;
import javax.swing.UIManager;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.JPasswordField;
import java.awt.Toolkit;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.SwingConstants;
import javax.swing.ImageIcon;
import javax.swing.JLayeredPane;
import java.awt.event.MouseMotionAdapter;

// this windows shows login and acces, (codebarreader and cecyte logo)

public class VentanaPrincipal extends JFrame {
     private JTextField textField;
    private JPasswordField passwordField;
    private JLayeredPane contentPane;
    public boolean SeEncontro = false;
    Conexion conectar = new Conexion();
```

```
MainPrincipal maestro = new MainPrincipal();
       String matricula;
String codigoDeBarras;



public VentanaPrincipal() {
        setResizable(false);
        setIconImage(Toolkit.getDefaultToolkit().getImage("J:\\logo gatito,perrito.jpg"));
        setTitle("Login-Sistema de control");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(500, 100, 406, 520);
        contentPane = new JLayeredPane();
        contentPane.setBackground(Color.black);
        this.setOpacity(1);
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
        this.setUndecorated(true);


        JLabel lblMatricula = new JLabel("Usuario");
        lblMatricula.setForeground(new Color(255, 255, 255));
        lblMatricula.setFont(new Font("Trebuchet MS", Font.BOLD, 15));
        lblMatricula.setBounds(64, 190, 152, 25);
        contentPane.add(lblMatricula);

        textField = new JTextField();
        String codigodeTextfield;
        textField.setBounds(64, 212, 282, 20);
        contentPane.add(textField);
        textField.setColumns(10);

        JLabel lblContrasea = new JLabel("Contrase\u00F1a");
        lblContrasea.setForeground(new Color(255, 255, 255));
        lblContrasea.setFont(new Font("Trebuchet MS", Font.BOLD, 15));
        lblContrasea.setBounds(64, 258, 88, 25);
        contentPane.add(lblContrasea);

        final String varUsuariocorrecto="";
        final JLabel lblUsuarioCorrecto = new JLabel(varUsuariocorrecto );
        lblUsuarioCorrecto.setEnabled(false);
        lblUsuarioCorrecto.setForeground(new Color(25, 25, 112));
        lblUsuarioCorrecto.setBounds(64, 325, 260, 25);
        contentPane.add(lblUsuarioCorrecto);
```

```
JButton btnOk = new JButton("OK");
btnOk.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
        }
});
btnOk.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {

                String codigo = textField.getText();


                conectar.acceso(codigo);




        }


        }
);
btnOk.setForeground(new Color(25, 25, 112));
btnOk.setFont(new Font("Tahoma", Font.BOLD, 15));
btnOk.setBounds(64, 341, 69, 52);
contentPane.add(btnOk);

JButton btnCancelar = new JButton("Cancelar");
btnCancelar.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {

        }
});
btnCancelar.setForeground(new Color(25, 25, 112));
btnCancelar.setFont(new Font("Tahoma", Font.BOLD, 15));
btnCancelar.setBounds(143, 341, 112, 52);
contentPane.add(btnCancelar);

passwordField = new JPasswordField();
passwordField.setBounds(64, 281, 282, 20);
contentPane.add(passwordField);

JButton btnNewButton = new JButton("Salir");
btnNewButton.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
            System.exit(EXIT_ON_CLOSE);
        }
    });
    btnNewButton.setForeground(new Color(139, 0, 0));
    btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 15));
    btnNewButton.setBounds(265, 341, 81, 52);
    contentPane.add(btnNewButton);

    JLabel lblNewLabel = new JLabel("");
    lblNewLabel.setIcon(new
ImageIcon(VentanaPrincipal.class.getResource("/Imagenes/dos.png")));

    contentPane.addMouseMotionListener(new MouseMotionAdapter() {
        @Override
        public void mouseDragged(MouseEvent e) {


VentanaPrincipal.this.setLocation(VentanaPrincipal.this.getX()+e.getX(),VentanaPrincipal.thi
s.getY()+e.getY());
        }
    });

    JLabel lblNewLabel_1 = new JLabel("");
    lblNewLabel_1.setIcon(new
ImageIcon(VentanaPrincipal.class.getResource("/Imagenes/TRES.png")));

    lblNewLabel_1.setBounds(46, 23, 396, 183);
    contentPane.add(lblNewLabel_1);
    lblNewLabel.setVerticalAlignment(SwingConstants.TOP);
    lblNewLabel.setHorizontalAlignment(SwingConstants.LEFT);
    lblNewLabel.setBounds(0, 0, 406, 535);;
    contentPane.add(lblNewLabel);
    //lblNewLabel.setIcon(new
ImageIcon(VentanaPrincipal.class.getResource("/bin/dos.png")));

  }
}
```

Componente principal:
Es la ventana principal del sistema con el que el usuario ya identificado interactúa directamente. Es el lugar en donde residen todos los demás componentes.

```
package bi;
```

```
import java.awt.Color;
import java.awt.Dialog.ModalExclusionType;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.print.PrinterException;
import java.io.BufferedReader;
import java.io.FileOutputStream;
import java.io.InputStreamReader;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.DateFormat;
import java.text.MessageFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;

import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JSeparator;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.LayoutStyle.ComponentPlacement;
import javax.swing.ListSelectionModel;
import javax.swing.ScrollPaneConstants;
import javax.swing.UIManager;
import javax.swing.UnsupportedLookAndFeelException;
import javax.swing.border.TitledBorder;
import javax.swing.table.DefaultTableModel;
import org.apache.poi.hssf.usermodel.HSSFCell;
import org.apache.poi.hssf.usermodel.HSSFRichTextString;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;

public class VentanaMain extends JPanel {

    /**
     * Este codigo esta optimizado para que aparezca correctamente en Windows
     * Builder, si solo aparece una ventana gris dar muchos clics en "repase" o
     * si no quitar el comentario en el constructor. si se quita este comentario
     * y asi se guarda se ejecutara con errores.
     */

    // public VentanaMain() {
    // initialize();
    // }

    Float matriculalocal;
    String nombrelocal;
    int contadorGeneralDeAlumnos = 0;
    MainPrincipal main = new MainPrincipal();
    MeinRealReal real = new MeinRealReal();
    boolean usuarioSalio = false;
    String horadesalida = "";
    Object elRegistro = new Object();
    JLabel label = new JLabel("-------------------");
    JPanel panel = new JPanel();
    JTable mitable = new JTable();
    String horadeentrada = "";
    Autoguardado autosave = new Autoguardado();
    int Meces[] = new int[12];
    static JOptionPane jop = new JOptionPane();
    int contadorparafila1Excel = 0;

    Date date = new Date();
    String gafeteABuscar;

    BufferedReader tlc = new BufferedReader(new InputStreamReader(System.in));

    Object[] fila = { "Numero", "Apellido paterno", "Apellido materno",
                    "Nombre", "Carrera", "Matricula", "Trabajo o practica", "Notas",
                    "Hr Entrada", "Hr Salida" };

    Object[] titulos =
    { fila[0], "Apellido paterno", "Apellido paterno", "Nombre", "Carrera",
                    "Turno", "Matricula", "Trabajo o practica", "Notas", "Hr Entrada" };

    DateFormat hourdateFormat = new SimpleDateFormat("HH:mm:ss dd/MM/yyyy");
    DefaultTableModel model = new DefaultTableModel(titulos, 0);

    JTextField lectora;

    public void initialize() {

            JFrame frmSistemaDeControl;
            frmSistemaDeControl = new JFrame();
            frmSistemaDeControl

.setModalExclusionType(ModalExclusionType.APPLICATION_EXCLUDE);
            frmSistemaDeControl.setAutoRequestFocus(false);
            frmSistemaDeControl.setVisible(true);
            frmSistemaDeControl.getContentPane().setBackground(
                            new Color(119, 136, 153));

            frmSistemaDeControl.getContentPane().setForeground(new Color(0, 0, 0));

            JMenuBar menuBar = new JMenuBar(); // esta es la barra de menus
            menuBar.setBackground(new Color(192, 192, 192));
            frmSistemaDeControl.setJMenuBar(menuBar);

            JMenu mnArchivo = new JMenu("Archivo ");
            menuBar.add(mnArchivo);

            JMenu mnGrupos = new JMenu("Grupos");
            mnGrupos.setEnabled(false);
            menuBar.add(mnGrupos);

            JMenu mnHerramientas = new JMenu("Herramientas");
            menuBar.add(mnHerramientas);

            JMenu mnAyuda = new JMenu("Ayuda");
            menuBar.add(mnAyuda);

            JMenu mnVer = new JMenu("Ver");
            menuBar.add(mnVer);

            JMenu mnBarraDeHerramientas = new JMenu("Barra de Herramientas");
            mnVer.add(mnBarraDeHerramientas);
```

```
JLabel lblPutas = new JLabel("");
mnArchivo.add(lblPutas);

JLabel lblAbrir = new JLabel("Abrir");
lblAbrir.setEnabled(false);
lblAbrir.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
        }
});
mnArchivo.add(lblAbrir);

JLabel lblCerrar = new JLabel("Cerrar");
lblCerrar.setEnabled(false);
lblCerrar.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
        }
});
mnArchivo.add(lblCerrar);

JLabel lblGuardar = new JLabel("Guardar");
lblGuardar.setEnabled(false);
lblGuardar.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
        }
});
mnArchivo.add(lblGuardar);

JLabel lblCerrarSesion = new JLabel("Cerrar sesi\u00F3n");
lblCerrarSesion.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {

                real.ejecutar2();
                main.OcultarVentanaMain();
                main.MostrarVentanaPrincipal();
                panel.enable(false);
                panel.show(false);
                panel.hide();

                real.ejecutar();

        }
});
```

```
    mnArchivo.add(lblCerrarSesion);

    JLabel lblConsultar = new JLabel("Consultar");
    lblConsultar.setEnabled(false);
    lblConsultar.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
        }
    });
    mnHerramientas.add(lblConsultar);

    JLabel lblExportarAPdf = new JLabel("Exportar a PDF");
    lblExportarAPdf.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {

            try {
                UIManager

.setLookAndFeel("com.sun.java.swing.plaf.nimbus.NimbusLookAndFeel");
            } catch (ClassNotFoundException e) {
                e.printStackTrace();
            } catch (InstantiationException e) {
                e.printStackTrace();
            } catch (IllegalAccessException e) {
                e.printStackTrace();
            } catch (UnsupportedLookAndFeelException e) {
                e.printStackTrace();
            }

            try {

                // tabla1.print();//envia los datos de la tabla a la
                // impresora

                MessageFormat headerFormat = new
MessageFormat("Grupo ");
                MessageFormat footerFormat = new MessageFormat(
                        "- Página {0} -");
                mitable.print(null, headerFormat, footerFormat);
            } catch (PrinterException ex) {
                JOptionPane
                        .showMessageDialog(null,
                                "No se ha podido imprimir
correctamente, intentalo más tarde.");
            }
    }
});

JLabel lblGenerarGraficaMensual = new JLabel("Generar grafica mensual");
lblGenerarGraficaMensual.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {

        main.MostrarVentanaGenerarGrafica();
    }
});

JLabel lblEliminarFila = new JLabel("Eliminar fila");
lblEliminarFila.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
        eliminarUno();
    }
});
mnHerramientas.add(lblEliminarFila);
mnHerramientas.add(lblGenerarGraficaMensual);

JLabel lblAbrirAutoguardados = new JLabel("Abrir autoguardados");
mnHerramientas.add(lblAbrirAutoguardados);
mnHerramientas.add(lblExportarAPdf);

JSeparator separator = new JSeparator();
mnHerramientas.add(separator);

JLabel lblDispositivos = new JLabel("Dispositivos ...");
lblDispositivos.setEnabled(false);
lblDispositivos.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
    }
});
mnHerramientas.add(lblDispositivos);

JSeparator separator_1 = new JSeparator();
mnHerramientas.add(separator_1);

JLabel lblpersonalizar = new JLabel("Personalizar");
lblpersonalizar.setEnabled(false);
lblpersonalizar.addMouseListener(new MouseAdapter() {
    @Override
```

```
        public void mouseClicked(MouseEvent arg0) {
        }
});
mnHerramientas.add(lblpersonalizar);

JLabel lblBarraDeEstado = new JLabel("Barra de estado");
lblBarraDeEstado.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
        }
});
mnBarraDeHerramientas.add(lblBarraDeEstado);

JLabel lblBarraDeBotones = new JLabel("Barra de botones");
lblBarraDeBotones.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
        }
});
mnBarraDeHerramientas.add(lblBarraDeBotones);

JLabel lblLectorDeCodigo = new JLabel("Lector de Codigo ...");
lblLectorDeCodigo.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
        }
});
mnVer.add(lblLectorDeCodigo);

JLabel lblPantallaCompleta = new JLabel("Pantalla completa");
lblPantallaCompleta.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
        }
});
mnVer.add(lblPantallaCompleta);

JLabel lblManualDeUsuario = new JLabel("Manual de usuario tecnico");
lblManualDeUsuario.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
        }
});
mnAyuda.add(lblManualDeUsuario);

JLabel lblManulDeUsuario = new JLabel("Manual de usuario");
```

```
lblManulDeUsuario.addMouseListener(new MouseAdapter() {
	@Override
	public void mouseClicked(MouseEvent arg0) {
	}
});
mnAyuda.add(lblManulDeUsuario);

JLabel lblAgregarGrupo = new JLabel("Agregar grupo");
lblAgregarGrupo.addMouseListener(new MouseAdapter() {
	@Override
	public void mouseClicked(MouseEvent arg0) {
		main.MostrarVentanaAgregarGrupo();
	}
});
mnGrupos.add(lblAgregarGrupo);

JLabel lblBorrarGrupo = new JLabel("Borrar grupo");
lblBorrarGrupo.addMouseListener(new MouseAdapter() {
	@Override
	public void mouseClicked(MouseEvent arg0) {
	}
});
mnGrupos.add(lblBorrarGrupo);

JLabel lblGuardarGrupo = new JLabel("Guardar grupo");
lblGuardarGrupo.addMouseListener(new MouseAdapter() {
	@Override
	public void mouseClicked(MouseEvent arg0) {
	}
});
mnGrupos.add(lblGuardarGrupo);

JLabel lblSeleccionarTodo = new JLabel("Seleccionar todo");
lblSeleccionarTodo.addMouseListener(new MouseAdapter() {
	@Override
	public void mouseClicked(MouseEvent arg0) {
	}
});
mnGrupos.add(lblSeleccionarTodo);

JLabel lblDeseleccionar = new JLabel("Deseleccionar");
lblDeseleccionar.addMouseListener(new MouseAdapter() {
	@Override
	public void mouseClicked(MouseEvent arg0) {
	}
});
```

```
        mnGrupos.add(lblDeseleccionar);

        JLabel lblTodosLosGrupos = new JLabel("Todos los grupos ...");
        lblTodosLosGrupos.addMouseListener(new MouseAdapter() {
                @Override
                public void mouseClicked(MouseEvent arg0) {
                }
        });
        mnGrupos.add(lblTodosLosGrupos);

        Class[] columnTypes = new Class[] { Integer.class, Object.class,
                        Boolean.class, String.class, String.class, Long.class,
                        String.class, String.class, Double.class, Double.class };

        boolean[] columnEditables = new boolean[] { false, true, true, true,
                        true, true, true, true, true, true };
        ;

        JPanel panel_de_paneles = new JPanel();
        panel_de_paneles.setBackground(Color.LIGHT_GRAY);

        panel.setBorder(new TitledBorder(null, "", TitledBorder.LEADING,
                        TitledBorder.TOP, null, null));

        JMenuBar menuBar_1 = new JMenuBar(); //

        JLabel lblNombreDelMaestro = new JLabel("          Nombre del maestro: ");
        menuBar_1.add(lblNombreDelMaestro);

        menuBar_1.add(label);
        GroupLayout gl_panel = new GroupLayout(panel);
        gl_panel.setHorizontalGroup(gl_panel.createParallelGroup(
                        Alignment.LEADING).addComponent(menuBar_1,
                        GroupLayout.DEFAULT_SIZE, 1189, Short.MAX_VALUE));
        gl_panel.setVerticalGroup(gl_panel.createParallelGroup(
                        Alignment.LEADING).addGroup(
                        gl_panel.createSequentialGroup()
                                        .addComponent(menuBar_1,
GroupLayout.DEFAULT_SIZE,
                                                        GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                                        .addGap(2)));
        panel.setLayout(gl_panel);

        JPanel panel_2 = new JPanel();
```

```
		panel_2.setOpaque(false);

		JButton btnNewButton_1 = new JButton("Eliminar Grupo");
		btnNewButton_1
				.setIcon(new ImageIcon(
	"C:\\Users\\Yodemi\\Desktop\\nuevos_iconos\\varios\\delete_square16_h.ico"));

		JButton btnNewButton_4 = new JButton("Generar Reporte");

		JButton btnNewButton_3 = new JButton("Abrir Sesion");
		btnNewButton_3.setEnabled(false);

		JButton btnNewButton_2 = new JButton("Guardar Sesion");
		btnNewButton_2.setEnabled(false);

		JButton btnNewButton = new JButton("Agregar Grupo");
		btnNewButton.setEnabled(false);
		btnNewButton.addMouseListener(new MouseAdapter() {
			@Override
			public void mouseClicked(MouseEvent arg0) {

			}
		});
		lectora = new JTextField();
		lectora.addKeyListener(new KeyAdapter() {
			@Override
			public void keyPressed(KeyEvent e) {

				int key = e.getKeyCode();

				if (key == KeyEvent.VK_ENTER) {
					gafeteABuscar = lectora.getText();
					agregarUno(gafeteABuscar);

				}
			}
		});
		lectora.setColumns(10);

		JLabel lblIngresoManual = new JLabel("Gafete");

		GroupLayout gl_panel_2 = new GroupLayout(panel_2);
		gl_panel_2.setHorizontalGroup(gl_panel_2.createParallelGroup(
				Alignment.LEADING).addGroup(
				gl_panel_2
```

```
                                        .createSequentialGroup()
                                        .addContainerGap()
                                        .addComponent(btnNewButton_1,
                                                   GroupLayout.PREFERRED_SIZE, 129,
                                                   GroupLayout.PREFERRED_SIZE)
                                        .addGap(18)
                                        .addComponent(btnNewButton_4)
                                        .addPreferredGap(ComponentPlacement.RELATED)
                                        .addComponent(btnNewButton_3)

.addPreferredGap(ComponentPlacement.UNRELATED)
                                        .addComponent(btnNewButton_2)

.addPreferredGap(ComponentPlacement.UNRELATED)
                                        .addComponent(btnNewButton,
GroupLayout.PREFERRED_SIZE,
                                                   137, GroupLayout.PREFERRED_SIZE)
                                        .addPreferredGap(ComponentPlacement.RELATED)
                                        .addComponent(lectora,
GroupLayout.PREFERRED_SIZE, 159,
                                                   GroupLayout.PREFERRED_SIZE)

.addPreferredGap(ComponentPlacement.UNRELATED)
                                        .addComponent(lblIngresoManual).addGap(231)));
      gl_panel_2
                  .setVerticalGroup(gl_panel_2
                                        .createParallelGroup(Alignment.LEADING)
                                        .addGroup(
                                                   gl_panel_2
                                                              .createSequentialGroup()
                                                              .addGap(5)
                                                              .addGroup(

gl_panel_2

      .createParallelGroup(

                  Alignment.BASELINE)

      .addComponent(

                  btnNewButton_1)

      .addComponent(

                  btnNewButton_4)
```

```
        .addComponent(

                btnNewButton_3)

        .addComponent(

                btnNewButton_2)

        .addComponent(

                btnNewButton)

        .addComponent(

                lectora,

                GroupLayout.PREFERRED_SIZE,

                GroupLayout.DEFAULT_SIZE,

                GroupLayout.PREFERRED_SIZE)

        .addComponent(

                lblIngresoManual))
                                                    .addContainerGap()));
        panel_2.setLayout(gl_panel_2);

        JPanel panel_1 = new JPanel();
        panel_1.setOpaque(false);
        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setViewportBorder(UIManager.getBorder("EditorPane.border"));
        scrollPane

.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS)
;

        mitable.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
        mitable.setBorder(new TitledBorder(UIManager
                .getBorder("TitledBorder.border"), "", TitledBorder.CENTER,
                TitledBorder.TOP, null, null));

        mitable.setModel(new DefaultTableModel());
        scrollPane.setViewportView(mitable);
        GroupLayout gl_panel_1 = new GroupLayout(panel_1);
        gl_panel_1.setHorizontalGroup(gl_panel_1.createParallelGroup(
                Alignment.LEADING).addGroup(
                gl_panel_1
                        .createSequentialGroup()
                        .addComponent(scrollPane,
GroupLayout.DEFAULT_SIZE,
                                1165, Short.MAX_VALUE).addGap(9)));
        gl_panel_1.setVerticalGroup(gl_panel_1.createParallelGroup(
                Alignment.LEADING).addComponent(scrollPane,
                GroupLayout.DEFAULT_SIZE, 589, Short.MAX_VALUE));
        panel_1.setLayout(gl_panel_1);
        GroupLayout groupLayout = new GroupLayout(
                frmSistemaDeControl.getContentPane());
        groupLayout.setHorizontalGroup(groupLayout.createParallelGroup(
                Alignment.LEADING).addComponent(panel_de_paneles,
                GroupLayout.DEFAULT_SIZE, 1172, Short.MAX_VALUE));
        groupLayout.setVerticalGroup(groupLayout.createParallelGroup(
                Alignment.LEADING).addGroup(
                groupLayout
                        .createSequentialGroup()
                        .addComponent(panel_de_paneles,
                                GroupLayout.DEFAULT_SIZE, 660,
Short.MAX_VALUE)
                        .addContainerGap()));
        GroupLayout gl_panel_de_paneles = new GroupLayout(panel_de_paneles);
        gl_panel_de_paneles.setHorizontalGroup(gl_panel_de_paneles
                .createParallelGroup(Alignment.LEADING)
                .addGroup(
                        gl_panel_de_paneles
                                .createSequentialGroup()
                                .addComponent(panel,
GroupLayout.DEFAULT_SIZE,
                                        1155,
Short.MAX_VALUE).addGap(17))
                .addGroup(
                        gl_panel_de_paneles
                                .createSequentialGroup()
                                .addGap(10)
                                .addComponent(panel_1,

GroupLayout.DEFAULT_SIZE, 1145,

Short.MAX_VALUE).addGap(17))
                .addGroup(
                        gl_panel_de_paneles
                                .createSequentialGroup()
```

```
                                                                .addGap(18)
                                                                .addComponent(panel_2,

GroupLayout.PREFERRED_SIZE, 1107,

GroupLayout.PREFERRED_SIZE)
                                                                .addContainerGap(47,
Short.MAX_VALUE)));
          gl_panel_de_paneles.setVerticalGroup(gl_panel_de_paneles
                              .createParallelGroup(Alignment.LEADING).addGroup(
                                              gl_panel_de_paneles
                                                                .createSequentialGroup()
                                                                .addGap(20)
                                                                .addComponent(panel,
GroupLayout.DEFAULT_SIZE,
                                                                                  23, Short.MAX_VALUE)

.addPreferredGap(ComponentPlacement.RELATED)
                                                                .addComponent(panel_2,

GroupLayout.PREFERRED_SIZE,

GroupLayout.DEFAULT_SIZE,

GroupLayout.PREFERRED_SIZE)

.addPreferredGap(ComponentPlacement.RELATED)
                                                                .addComponent(panel_1,

GroupLayout.DEFAULT_SIZE, 568,

Short.MAX_VALUE).addGap(29)));
          panel_de_paneles.setLayout(gl_panel_de_paneles);
          frmSistemaDeControl.getContentPane().setLayout(groupLayout);
          btnNewButton_2.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent arg0) {
                    }
          });
          btnNewButton_3.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent arg0) {
                    }
          });

          btnNewButton_4.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent arg0) {
```

```
            int seleccion = jop.showOptionDialog(null,
                    "Selecione opcion a exportar", "Generar reporte",
                    JOptionPane.YES_NO_CANCEL_OPTION,
                    JOptionPane.QUESTION_MESSAGE, null, new
Object[] {
                            "PDF", "Excel" }, null);

            if (seleccion != -1) {
                System.out.println(" seleccionada la opcion "
                        + (seleccion + 1));

                if (seleccion == 1) {
                    GenerarReporteExcel();
                } else {

                    GenerarREportePorPDF();
                }
            }

        }
    });
    btnNewButton_1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {

            VaciarElModelo();
        }
    });

    JLabel lblNewLabel = new JLabel("New label");
    lblNewLabel.setBounds(646, 50, 46, 14);
    frmSistemaDeControl.getContentPane().add(lblNewLabel);

    frmSistemaDeControl.setBackground(new Color(192, 192, 192));
    frmSistemaDeControl.setForeground(new Color(0, 0, 0));
    frmSistemaDeControl
            .setIconImage(Toolkit
                    .getDefaultToolkit()
                    .getImage(

"C:\\Users\\Angel\\Pictures\\302-iconos\\nuevos_iconos\\varios\\user_properties16_h.ico"));
    frmSistemaDeControl.setTitle("Sistema de control");
    frmSistemaDeControl.setBounds(100, 100, 1188, 759);
    frmSistemaDeControl.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    lectora.requestFocus();
```

```
}

public void AgregarModelo() {
    this.mitable.setModel(model);
}

public void VaciarElModelo() {
    DefaultTableModel modeloVacio;
    modeloVacio = new DefaultTableModel(titulos, 0);

    this.mitable.setModel(modeloVacio);

    while (model.getRowCount() > 0) {

        model.removeRow(model.getRowCount() - 1);
    }

    contadorGeneralDeAlumnos = 0;

}

public void agregarUno(String gafeteABuscar2) {
    Connection conn;
    java.sql.Statement sent;
    conn = Conexion.getConnection();
    Calendar hora = new GregorianCalendar();

    try {
        sent = conn.createStatement();
        ResultSet rs = sent
                .executeQuery("select * from alumnos where row_id = " + "("
                        + gafeteABuscar2 + ")");

        horadeentrada = hora.get(Calendar.HOUR_OF_DAY) + ":"
                + hora.get(Calendar.MINUTE) + ":"
                + hora.get(Calendar.SECOND);

        while (rs.next()) {
            fila[0] = contadorGeneralDeAlumnos;
            fila[1] = rs.getString("alumno_apell_pat");
            fila[2] = rs.getString("alumno_apell_mat");
            fila[3] = rs.getString("alumno_nombre");
            fila[4] = rs.getString("esp_nombre");
            fila[5] = rs.getString("gpo_turno");
            fila[6] = rs.getString("alumno_no_control");
            fila[7] = "BIBLIOTECA";
            fila[8] = " ";
            fila[9] = horadeentrada;
        }
            // aqui se agrega fila por fila, modificando
        model.addRow(fila);
            horadesalida = "";
            horadeentrada = "";
            contadorGeneralDeAlumnos = contadorGeneralDeAlumnos + 1;


     mitable.setModel(model);

        lectora.setText("");

    } catch (Exception e) {
        e.printStackTrace();
    }
    // autosaveConexion();
    try {

        lectora.setText("Agregando...");
        lectora.setEnabled(true);
        Thread.sleep(500);
        lectora.setText("");
        lectora.setEnabled(true);

        lectora.requestFocus();

    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

}

public void eliminarUno() {

    String rowAEliminar;

    try {
        int rowEliminar = Integer.parseInt(rowAEliminar = jop
                .showInputDialog("Inserte el numero de fila a eliminar"));
        model.removeRow(rowEliminar);

        int cantidadDeRows = model.getRowCount();
```

```
                        for (int variableParaReasignarNumero = 0; variableParaReasignarNumero <
cantidadDeRows; variableParaReasignarNumero++) {
                                model.setValueAt(variableParaReasignarNumero,
                                                variableParaReasignarNumero, 0);

                        }
                        contadorGeneralDeAlumnos = contadorGeneralDeAlumnos - 1;
                } catch (Exception e) {
                        jop.showMessageDialog(null,
                                        "Error, el valor no es un numero o no existe",
                                        "Error de valores", jop.ERROR_MESSAGE);
                }

   }

// metodo para hacer un autoguardado de la base de datos local,
   //tener bien configurado el server local, hacer pruebas en caso de que
   // no se establezca conexion con el server local (Pantalla de error)

   public void autosaveConexion() {

                Connection conn;
                java.sql.Statement sent;
                conn = Conexion.getConnection();
                // String sql =
                // "INSERT INTO respaldo (Numero, Apellido Paterno, Apellido Materno, Nombre,
Carrera, Turno, Matricula, Trabajo o practica, Notas, Hora de entrada) VALUES
('?','?','?','?','?','?','?','?','?','?')";
                try {
                        // PreparedStatement pst = conn.prepareStatement(sql);
                        Statement statement = (Statement) conn.createStatement();

                        statement
                                        .execute("INSERT into respaldo "
                                                        + "(`Numero`, `Apellido_Paterno`,
`Apellido_Materno`,`Nombre`, `Carrera`, `Turno`, `Matricula`, `Trabajo_o_practica`, `Notas`,
`Hora_de_entrada`) VALUES ('"
                                                        + model.getValueAt(0, 0) + "','"
                                                        + model.getValueAt(0, 1) + "','"
                                                        + model.getValueAt(0, 2) + "','"
                                                        + model.getValueAt(0, 3) + "','"
                                                        + model.getValueAt(0, 4) + "','"
                                                        + model.getValueAt(0, 5) + "','"
                                                        + model.getValueAt(0, 6) + "','"
                                                        + model.getValueAt(0, 7) + "','"
                                                        + model.getValueAt(0, 8) + "','"
                                                + model.getValueAt(0, 9) + "");

                        // for(int x=0; x <= (model.getRowCount()); x++){
                        //
                        //

                        // }
                } catch (SQLException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }

        }

        public void GenerarREportePorPDF() {
                try {
                        UIManager

.setLookAndFeel("com.sun.java.swing.plaf.nimbus.NimbusLookAndFeel");
                } catch (ClassNotFoundException e) {
                        e.printStackTrace();
                } catch (InstantiationException e) {
                        e.printStackTrace();
                } catch (IllegalAccessException e) {
                        e.printStackTrace();
                } catch (UnsupportedLookAndFeelException e) {
                        e.printStackTrace();
                }

                try {

                        // tabla1.print();//envia los datos de la tabla a la
                        // impresora

                        MessageFormat headerFormat = new MessageFormat("Grupo ");
                        MessageFormat footerFormat = new MessageFormat("- Página {0} -");
                        mitable.print(null, headerFormat, footerFormat);
                } catch (PrinterException ex) {
                        JOptionPane
                                        .showMessageDialog(null,
                                                        "No se ha podido imprimir correctamente,
intentalo más tarde.");
                }
        }

        HSSFWorkbook libro = new HSSFWorkbook();
```

```
HSSFSheet hoja1 = libro.createSheet("Reporte");
HSSFRow filaexcel = hoja1.createRow(1);

public void crearfila(int numerodeiteracion) {
        filaexcel = hoja1.createRow(numerodeiteracion + 1);
}

public void GenerarReporteExcel() {

        HSSFRow fila0 = hoja1.createRow(0);

        HSSFCell celda0x0 = fila0.createCell(0);
        String txt0x0 = "Numero";
        HSSFRichTextString texto0x0 = new HSSFRichTextString(txt0x0);
        celda0x0.setCellValue(texto0x0);

        HSSFCell celda0x1 = fila0.createCell(1);
        String txt0x1 = "Apellido paterno";
        HSSFRichTextString texto0x1 = new HSSFRichTextString(txt0x1);
        celda0x1.setCellValue(texto0x1);

        HSSFCell celda0x2 = fila0.createCell(2);
        String txt0x2 = "Apellido materno";
        HSSFRichTextString texto0x2 = new HSSFRichTextString(txt0x2);
        celda0x2.setCellValue(texto0x2);

        HSSFCell celda0x3 = fila0.createCell(3);
        String txt0x3 = "Nombre";
        HSSFRichTextString texto0x3 = new HSSFRichTextString(txt0x3);
        celda0x3.setCellValue(texto0x3);

        HSSFCell celda0x4 = fila0.createCell(4);
        String txt0x4 = "Carrera";
        HSSFRichTextString texto0x4 = new HSSFRichTextString(txt0x4);
        celda0x4.setCellValue(texto0x4);

        HSSFCell celda0x5 = fila0.createCell(5);
        String txt0x5 = "Turno";
        HSSFRichTextString texto0x5 = new HSSFRichTextString(txt0x5);
        celda0x5.setCellValue(texto0x5);

        HSSFCell celda0x6 = fila0.createCell(6);
        String txt0x6 = "Matricula";
        HSSFRichTextString texto0x6 = new HSSFRichTextString(txt0x6);
        celda0x6.setCellValue(texto0x6);
```

```
HSSFCell celda0x7 = fila0.createCell(7);
String txt0x7 = "Trabajo o practica";
HSSFRichTextString texto0x7 = new HSSFRichTextString(txt0x7);
celda0x7.setCellValue(texto0x7);

HSSFCell celda0x8 = fila0.createCell(8);
String txt0x8 = "Notas";
HSSFRichTextString texto0x8 = new HSSFRichTextString(txt0x8);
celda0x8.setCellValue(texto0x8);

HSSFCell celda0x9 = fila0.createCell(9);
String txt0x9 = "Hr Entrada";
HSSFRichTextString texto0x9 = new HSSFRichTextString(txt0x9);
celda0x9.setCellValue(texto0x9);

int tamañodemodelo = model.getRowCount();
int contadorParaExcel = 0;

// para apellido paterno
while (contadorparafila1Excel < tamañodemodelo) {

        crearfila(contadorparafila1Excel);

        HSSFCell celda0 = filaexcel.createCell(0);
        String txt0 = contadorParaExcel + "";
        HSSFRichTextString texto0 = new HSSFRichTextString(txt0);
        celda0.setCellValue(texto0);
        contadorParaExcel++;

        HSSFCell celda1 = filaexcel.createCell(1);
        String txt = (String) model.getValueAt(contadorparafila1Excel, 1);
        HSSFRichTextString texto = new HSSFRichTextString(txt);
        celda1.setCellValue(texto);

        HSSFCell celda2 = filaexcel.createCell(2);
        String txt1 = (String) model.getValueAt(contadorparafila1Excel, 2);
        HSSFRichTextString texto1 = new HSSFRichTextString(txt1);
        celda2.setCellValue(texto1);

        HSSFCell celda3 = filaexcel.createCell(3);
        String txt3 = (String) model.getValueAt(contadorparafila1Excel, 3);
        HSSFRichTextString texto3 = new HSSFRichTextString(txt3);
        celda3.setCellValue(texto3);

        HSSFCell celda4 = filaexcel.createCell(4);
        String txt4 = (String) model.getValueAt(contadorparafila1Excel, 4);
        HSSFRichTextString texto4 = new HSSFRichTextString(txt4);
        celda4.setCellValue(texto4);

        HSSFCell celda5 = filaexcel.createCell(5);
        String txt5 = (String) model.getValueAt(contadorparafila1Excel, 5);
        HSSFRichTextString texto5 = new HSSFRichTextString(txt5);
        celda5.setCellValue(texto5);

        HSSFCell celda6 = filaexcel.createCell(6);
        String txt6 = (String) model.getValueAt(contadorparafila1Excel, 6);
        HSSFRichTextString texto6 = new HSSFRichTextString(txt6);
        celda6.setCellValue(texto6);

        HSSFCell celda7 = filaexcel.createCell(7);
        String txt7 = (String) model.getValueAt(contadorparafila1Excel, 7);
        HSSFRichTextString texto7 = new HSSFRichTextString(txt7);
        celda7.setCellValue(texto7);

        HSSFCell celda8 = filaexcel.createCell(8);
        String txt8 = (String) model.getValueAt(contadorparafila1Excel, 8);
        HSSFRichTextString texto8 = new HSSFRichTextString(txt8);
        celda8.setCellValue(texto8);

        HSSFCell celda9 = filaexcel.createCell(9);
        String txt9 = (String) model.getValueAt(contadorparafila1Excel, 9);
        HSSFRichTextString texto9 = new HSSFRichTextString(txt9);
        celda9.setCellValue(texto9);

        contadorparafila1Excel++;
    }



    try {
        FileOutputStream arch = new FileOutputStream(
                jop.showInputDialog("Inserte el nombre de el archivo")
                        + ".xls");
        libro.write(arch);
        jop.showMessageDialog(null, "Archivo creado correctamente");
    } catch (Exception e) {
        System.out.println("Error al escribir en Excel");
    }

  }
}
```