

Anexo Proyecto personal

El siguiente software debe ser implementado con las siguientes tecnologías.

Software para el registro de asistencia de trabajadores. (Registro de trabajadores, administración de trabajadores (editar, eliminar y agregar) y administración de asistencia).

Obligatorio:

- MySql.
- Java (Orientado a Objetos), Swing, AWT.
- Arquitectura Modelo Vista Controlador.
- Presentación del proyecto Sábado 2 de Diciembre. (Proyecto Completo).

Código vistas

```
package asistencia;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class ViewMenu {

    public ViewMenu() {
        iniciar();
    }

    public void iniciar(){
        JFrame marco = new JFrame("ASISTENCIA");
        marco.setSize(800, 650);
        JPanel panel = new JPanel();
        ventana_menu(panel);
        marco.add(panel);
        marco.setVisible(true);
        marco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        marco.setLocationRelativeTo(null);
    }
}
```

```

}

public void ventana_menu(JPanel panel){

    panel.setLayout(null);

    JLabel campo1 = new JLabel("ASISTENCIA");
    campo1.setFont(new java.awt.Font("Tahoma", 0, 24));
    campo1.setBounds(10, 20, 150, 50);

    JButton boton1 = new JButton("Nuevo Empleado");
    JButton boton2 = new JButton("Asistencia");
    JButton boton3 = new JButton("Listar empleados");
    JButton boton4 = new JButton("Listar Asistencia");
    //JButton boton5 = new JButton("Conectar");
    JButton boton6 = new JButton("Eliminar empleado");
    JButton boton7 = new JButton("Editar empleado");

    boton1.setBounds(100, 100, 150, 50);
    boton2.setBounds(500, 100, 150, 50);
    boton3.setBounds(100, 300, 150, 50);
    boton4.setBounds(500, 300, 150, 50);
    //boton5.setBounds(280, 500, 150, 50);
    boton6.setBounds(100, 200, 150, 50);
    boton7.setBounds(100, 400, 150, 50);

    boton1.addActionListener(new ActionListener(){
        @Override
        public void actionPerformed(ActionEvent e) {
            ViewNuevoEmpleado ventana_nuevo = new ViewNuevoEmpleado();
        }
    });

    boton2.addActionListener(new ActionListener(){
        @Override
        public void actionPerformed(ActionEvent e) {
            ViewRegistroAsistencia ventana_asistencia = new
ViewRegistroAsistencia();
        }
    });

    boton3.addActionListener(new ActionListener(){
        @Override

```

```

        public void actionPerformed(ActionEvent e) {
            ViewMostrar mostrar = new ViewMostrar();
            mostrar.setVisible(true);
        }
    });

    boton6.addActionListener(new ActionListener(){
        @Override
        public void actionPerformed(ActionEvent e) {
            ViewEliminaEmpleado ventana = new ViewEliminaEmpleado();
        }
    });

    boton7.addActionListener(new ActionListener(){
        @Override
        public void actionPerformed(ActionEvent e) {
            ViewEditar ventana_editar = new ViewEditar();
            ventana_editar.setVisible(true);
        }
    });

    panel.add(campo1);
    panel.add(boton1);
    panel.add(boton2);
    panel.add(boton6);
    panel.add(boton3);
    panel.add(boton4);
    //panel.add(boton5);
    panel.add(boton7);
}

}

package asistencia;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JFrame;

```

```

import javax.swing.JPanel;
import javax.swing.*;

/**
 *
 * @author Sergio
 */
public class ViewNuevoEmpleado {

    private String rut_ingresado, digito_input;

    public ViewNuevoEmpleado() {
        inicio();
    }

    public void inicio(){

        JFrame marco = new JFrame("Nuevo Empleado");
        marco.setSize(500, 500);
        JPanel panel = new JPanel();
        ventana_empleado(panel);
        marco.add(panel);
        marco.setVisible(true);
        marco.setLocationRelativeTo(null);
    }

    public void ventana_empleado(JPanel panel){

        panel.setLayout(null);

        JLabel texto1 = new JLabel("Rut");
        JLabel texto2 = new JLabel("Nombres");
        JLabel texto3 = new JLabel("Apellido paternno");
        JLabel texto4 = new JLabel("Apellido materno");
        JTextField campo_rut = new JTextField();
        JTextField digito = new JTextField();
        JTextField campo_nombre = new JTextField();
        JTextField campo_apellido_p = new JTextField();
        JTextField campo_apellido_m = new JTextField();
        JButton guardar = new JButton("Guardar");

        texto1.setBounds(15, 60, 100, 30);
        texto2.setBounds(15, 100, 100, 30);

```

```

        texto3.setBounds(15, 140, 100, 30);
        texto4.setBounds(15, 190, 100, 30);
        campo_rut.setBounds(150, 60, 150, 30);
        digito.setBounds(320, 60, 30, 30);
        campo_nombre.setBounds(150, 100, 200, 30);
        campo_apellido_p.setBounds(150, 140, 200, 30);
        campo_apellido_m.setBounds(150, 190, 200, 30);
        guardar.setBounds(130, 280, 200, 40);

        guardar.addActionListener(new ActionListener(){
            @Override
            public void actionPerformed(ActionEvent e) {
                rut_ingresado=campo_rut.getText();
                digito_input=digito.getText();
                ControllerAsistencia controlador = new
ControllerAsistencia();
                boolean validacion = controlador.validar_rut(rut_ingresado,
digito_input);
                if (validacion){
                    ControllerAsistencia nuevo = new ControllerAsistencia();
                    nuevo.nuevo_empleado(campo_rut.getText(),
campo_nombre.getText(),
                    campo_apellido_p.getText(), campo_apellido_m.getText());
                    JOptionPane.showMessageDialog(null, "Empleado agregado
satisfactoriamente");
                }else{
                    JOptionPane.showMessageDialog(null, "Error en RUT");
                }
            }

        });
        panel.add(texto1);panel.add(texto2);panel.add(texto3);panel.add(texto4);

        panel.add(campo_rut);panel.add(digito);panel.add(campo_nombre);panel.add(cam
po_apellido_p);panel.add(campo_apellido_m);
        panel.add(guardar);
    }
}

package asistencia;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Timestamp;
import java.text.SimpleDateFormat;

```

```

import java.util.Date;
import javax.swing.*;

public class ViewRegistroAsistencia extends JFrame{

    private String rut, digito;

    public ViewRegistroAsistencia() {
        iniciar();
    }

    public void iniciar(){

        PantallaReloj reloj = new PantallaReloj(0, 0, 0, 0);

        reloj.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);//Centrado
del texto
        reloj.setVerticalAlignment(SwingConstants.TOP);
        reloj.setFont(new java.awt.Font("Arial", 1, 18));
        reloj.setBounds(270, 5, 150, 50);
        JFrame marco = new JFrame("Registro de Asistencia");
        marco.setSize(450, 450);
        JPanel panel = new JPanel();
        ventana_asistencia(panel);
        marco.getContentPane().add(reloj);
        marco.add(panel);

        marco.setVisible(true);
        marco.setLocationRelativeTo(null);

    }

    public void ventana_asistencia(JPanel panel){

        panel.setLayout(null);
        JLabel texto1 = new JLabel("Ingrese RUT del emplead@:");
        JTextField campo1 = new JTextField();
        JTextField campo2 = new JTextField();
        JButton boton1 = new JButton("INGRESAR");
        JButton boton2 = new JButton("ingreso");
        JButton boton3 = new JButton("Salida");
        boton2.setEnabled(false);
        boton3.setEnabled(false);
        //JLabel fecha = new JLabel();
        //SimpleDateFormat sdf = new SimpleDateFormat();
    }
}

```

```

//fecha.setText(sdf.format(new Date(System.currentTimeMillis())));

Date fecha_i = new Date();
Timestamp ingreso = new Timestamp(fecha_i.getTime());
//JLabel fecha2 = new JLabel();

//fecha.setBounds(300, 5, 150, 30);
texto1.setBounds(20, 70, 180, 30);
campo1.setBounds(200, 70, 100, 30);
campo2.setBounds(320, 70, 30, 30);
boton1.setBounds(90, 150, 250, 30);
boton2.setBounds(70, 270, 100, 30);
boton3.setBounds(250, 270, 100, 30);
//fecha2.setBounds(300, 35, 150, 30);

boton1.addActionListener(new ActionListener(){
    @Override
    public void actionPerformed(ActionEvent e) {
        ControllerAsistencia validaRut = new ControllerAsistencia();
        rut = campo1.getText();
        digito = campo2.getText();
        boolean resp1 = validaRut.validar_rut(rut, digito);
        if(resp1){
            JOptionPane.showMessageDialog(null, "bien");
        }else{
            JOptionPane.showMessageDialog(null, "que mal");
        }
        //JOptionPane.showMessageDialog(null, "Validar RUT");
        boton2.setEnabled(true);
        boton3.setEnabled(true);
    }
});

boton2.addActionListener(new ActionListener(){
    @Override
    public void actionPerformed(ActionEvent e) {
        String rut = campo1.getText();
        ControllerAsistencia controlador = new
ControllerAsistencia();
        boolean respuesta = controlador.ingreso(rut);
        if(respuesta){
            JOptionPane.showMessageDialog(null, "registro
ingresado");

```

```

        }else{
            JOptionPane.showMessageDialog(null, "Ingreso no
registrado");
        }
    }

});

boton3.addActionListener(new ActionListener(){
    @Override
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Registrar SALIDA");
    }

});

panel.add(texto1);
panel.add(campo1);
panel.add(campo2);
panel.add(boton1);
//panel.add(fecha);
//panel.add(fecha2);
panel.add(boton2);
panel.add(boton3);

}

}
package asistencia;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author Sergio
 */
public class ViewEditor extends javax.swing.JFrame {

    DefaultTableModel tablaempleados = new DefaultTableModel();

```



```

        String id_input;
        private ControllerAsistencia empleado;

    public ViewEditor() {
        tablaempleados = new DefaultTableModel(null, getColumnas());
        setFilas();
        //mostrar();
        initComponents();
    }

    private String[] getColumnas(){
        String columna[] = new String[]{"Id", "Rut", "Nombres", "Apellido
paterno", "Apellido materno"};
        return columna;
    }

    private void setFilas(){
        ControllerAsistencia controlador = new ControllerAsistencia();
        ArrayList<ControllerAsistencia> empleados = new
ArrayList<ControllerAsistencia>();
        empleados = controlador.listado_empleados();
        //ArrayList<String> fila = new ArrayList<String>();
        Object[] fila = new Object[5];
        for(ControllerAsistencia empleado: empleados){
            fila[0]=Integer.toString(empleado.getId());
            fila[1]=empleado.getRut_input();
            fila[2]=empleado.getNombre_input();
            fila[3]=empleado.getApellido_p_input();
            fila[4]=empleado.getApellido_m_input();
            tablaempleados.addRow(fila);
        }
    }

    /**
     * This method is called from within the constructor to initialize the
form.
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents

```

```
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jLabel1 = new javax.swing.JLabel();
    campo_id = new javax.swing.JTextField();
    btn_buscar = new javax.swing.JButton();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    campo_rut = new javax.swing.JTextField();
    campo_nombres = new javax.swing.JTextField();
    campo_apellidop = new javax.swing.JTextField();
    campo_apellidom = new javax.swing.JTextField();
    btn_editar = new javax.swing.JButton();
    btn_eliminar = new javax.swing.JButton();
    jLabel6 = new javax.swing.JLabel();
    btnactualizar = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

    jTable1.setModel(tablaempleados);
    jScrollPane1.setViewportViewView(jTable1);

    jLabel1.setText("ID del empleado");

    campo_id.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            campo_idActionPerformed(evt);
        }
    });

    btn_buscar.setBackground(new java.awt.Color(149, 194, 255));
    btn_buscar.setText("Buscar");
    btn_buscar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btn_buscarActionPerformed(evt);
        }
    });

    jLabel2.setText("RUT");
```

```

jLabel3.setText("NOMBRES");

jLabel4.setText("APELLIDO PATERNO");

jLabel5.setText("APELLIDO MATERNO");

campo_rut.setEditable(false);

btn_editar.setBackground(new java.awt.Color(149, 220, 231));
btn_editar.setText("Editar");
btn_editar.setEnabled(false);
btn_editar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btn_editarActionPerformed(evt);
    }
});

btn_eliminar.setText("Eliminar");
btn_eliminar.setEnabled(false);
btn_eliminar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btn_eliminarActionPerformed(evt);
    }
});

jLabel6.setFont(new java.awt.Font("Calisto MT", 0, 18)); // NOI18N
jLabel6.setText("Edición de registros");

btnactualizar.setText("Actualiza Listado");
btnactualizar.setEnabled(false);
btnactualizar.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnactualizarActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()

```

```

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addComponent(jScrollPane1)
    .addGroup(jPanel1Layout.createSequentialGroup())

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(jPanel1Layout.createSequentialGroup())

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

    .addComponent(jLabel5,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 110, Short.MAX_VALUE)
    .addComponent(jLabel11,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE, 98,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel4,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    .addComponent(jLabel12,
javax.swing.GroupLayout.PREFERRED_SIZE, 52,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel13,
javax.swing.GroupLayout.PREFERRED_SIZE, 64,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(jPanel1Layout.createSequentialGroup())
    .addComponent(campo_id,
javax.swing.GroupLayout.PREFERRED_SIZE, 78,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(64, 64, 64)
    .addComponent(btn_buscar,
javax.swing.GroupLayout.PREFERRED_SIZE, 145,
javax.swing.GroupLayout.PREFERRED_SIZE))

```

```

        .addComponent(campo_rut)
        .addComponent(campo_nombres)
        .addComponent(campo_apellidop)
        .addComponent(campo_apellidom))
        .addGap(112, 112, 112)

.addGroup(jPanel11Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(btn_editar,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        .addComponent(btn_eliminar,
javax.swing.GroupLayout.DEFAULT_SIZE, 148, Short.MAX_VALUE)
        .addComponent(btnactualizar,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)))
        .addComponent(jLabel6,
javax.swing.GroupLayout.PREFERRED_SIZE, 213,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(0, 57, Short.MAX_VALUE)))
        .addContainerGap())
);
jPanel11Layout.setVerticalGroup(

jPanel11Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel11Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel6,
javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 167,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel11Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel11Layout.createSequentialGroup()
                        .addGap(29, 29, 29)

.addGroup(jPanel11Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

```

```

        .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 24,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(campo_id,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btn_buscar)
        .addComponent(btn_actualizar))
        .addGap(26, 26, 26)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel2)
        .addComponent(campo_rut,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btn_editar))
        .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel3)
        .addComponent(campo_nombres,
javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(campo_apellidoop,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel4)))
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(131, 131, 131)
            .addComponent(btn_eliminar)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

```

```

        .addComponent(campo_apellido,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel15))
        .addContainerGap(61, Short.MAX_VALUE))
    );

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addContainerGap())
        );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addContainerGap())
        );

    pack();
    setLocationRelativeTo(null);
} // </editor-fold> // GEN-END: initComponents

private void campo_idActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_campo_idActionPerformed
    //String id_input = campo_id.getText();
    // TODO add your handling code here:
} // GEN-LAST:event_campo_idActionPerformed

private void btn_buscarActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_btn_buscarActionPerformed
    // TODO add your handling code here:
    // JOptionPane.showMessageDialog(null, "buscar");
    //int id = 0;

```

```

        int id_input = Integer.parseInt(campo_id.getText());
        System.out.println(id_input);
        ControllerAsistencia enviar_dato = new
ControllerAsistencia();
        //ArrayList<ControllerAsistencia> empleado = new
ArrayList<ControllerAsistencia>();
        ArrayList<ControllerAsistencia> resultado =
enviar_dato.view_editar_mostrar_empleado(id_input);
        for(ControllerAsistencia dato: resultado){
            campo_rut.setText(dato.getRut_input());
            campo_nombres.setText(dato.getNombre_input());
            campo_apellidop.setText(dato.getApellido_p_input());
            campo_apellidom.setText(dato.getApellido_m_input());
        }
        btn_editar.setEnabled(true);
        btn_eliminar.setEnabled(true);
    }//GEN-LAST:event_btn_buscarActionPerformed

    private void btn_editarActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_btn_editarActionPerformed
        int id_input = Integer.parseInt(campo_id.getText());
        ControllerAsistencia enviardato = new ControllerAsistencia();
        boolean respuesta = enviardato.view_editar_editar_empleado(id_input,
campo_nombres.getText(), campo_apellidop.getText(),
campo_apellidom.getText());
        if(respuesta){
            JOptionPane.showMessageDialog(null, "Dato actualizado");
            btnactualizar.setEnabled(true);
            campo_id.setText("");
            campo_rut.setText("");
            campo_nombres.setText("");
            campo_apellidop.setText("");
            campo_apellidom.setText("");
            btn_eliminar.setEnabled(false);
            btn_editar.setEnabled(false);
        }else{
            JOptionPane.showMessageDialog(null, "Dato sin procesar");
            btnactualizar.setEnabled(true);
            campo_id.setText("");
            campo_rut.setText("");
            campo_nombres.setText("");
            campo_apellidop.setText("");
            campo_apellidom.setText("");
            btn_eliminar.setEnabled(false);
            btn_editar.setEnabled(false);
        }
    }

```



```

    }

    // TODO add your handling code here:
} //GEN-LAST:event_btn_editarActionPerformed

private void btn_eliminarActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_btn_eliminarActionPerformed
    int id_input2 = Integer.parseInt(campo_id.getText());
    ControllerAsistencia controlador = new ControllerAsistencia();
    boolean respuesta =
controlador.view_editar_eliminar_empleado(id_input2);
    if(respuesta){
        JOptionPane.showMessageDialog(null, "Registro eliminado");
        btnactualizar.setEnabled(true);
        campo_id.setText("");
        campo_rut.setText("");
        campo_nombres.setText("");
        campo_apellidop.setText("");
        campo_apellidom.setText("");
        btn_eliminar.setEnabled(false);
        btn_editar.setEnabled(false);
    }else{
        JOptionPane.showMessageDialog(null, "Solicitud sin procesar");
        btnactualizar.setEnabled(true);
        campo_id.setText("");
        campo_rut.setText("");
        campo_nombres.setText("");
        campo_apellidop.setText("");
        campo_apellidom.setText("");
        btn_eliminar.setEnabled(false);
        btn_editar.setEnabled(false);
    }

} //GEN-LAST:event_btn_eliminarActionPerformed

private void btnactualizarActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_btnactualizarActionPerformed
    try {
        int filas=tablaempleados.getRowCount();
        for (int i = 0;filas>i; i++) {
            tablaempleados.removeRow(0);
        }
        btn_editar.setEnabled(false);
        btn_eliminar.setEnabled(false);
        btnactualizar.setEnabled(false);
    } catch (Exception e) {

```

```

        JOptionPane.showMessageDialog(null, "Error al limpiar la
tabla.");
    }
    setFilas();
} //GEN-LAST:event_btnactualizarActionPerformed

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus Look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(ViewEditor.class.getName()).log(java.util
.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(ViewEditor.class.getName()).log(java.util
.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(ViewEditor.class.getName()).log(java.util
.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(ViewEditor.class.getName()).log(java.util
.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

```

```

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new ViewEditor().setVisible(true);
            }
        });

    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton btn_buscar;
    private javax.swing.JButton btn_editar;
    private javax.swing.JButton btn_eliminar;
    private javax.swing.JButton btn_actualizar;
    private javax.swing.JTextField campo_apellido_m;
    private javax.swing.JTextField campo_apellido_p;
    private javax.swing.JTextField campo_id;
    private javax.swing.JTextField campo_nombres;
    private javax.swing.JTextField campo_rut;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable jTable1;
    // End of variables declaration//GEN-END:variables

}

```

Código controladores

```

package asistencia;

import java.sql.Timestamp;
import java.util.ArrayList;
import java.util.Date;
import javax.swing.JOptionPane;

```

```

/**
 *
 * @author Sergio
 */
public class ControllerAsistencia {

    private String rut_input, codigo_input, nombre_input, apellido_p_input,
apellido_m_input, i;
    private int digito, id;

    public ControllerAsistencia() {

    }

    public ControllerAsistencia(int id, String rut_input, String
nombre_input, String apellido_p_input, String apellido_m_input) {
        this.id = id;
        this.rut_input = rut_input;
        this.nombre_input = nombre_input;
        this.apellido_p_input = apellido_p_input;
        this.apellido_m_input = apellido_m_input;

    }

    public ControllerAsistencia(String rut_input, String nombre_input,
String apellido_p_input, String apellido_m_input) {
        this.rut_input = rut_input;
        this.nombre_input = nombre_input;
        this.apellido_p_input = apellido_p_input;
        this.apellido_m_input = apellido_m_input;

    }

    public ControllerAsistencia(String nombre_input, String
apellido_p_input, String apellido_m_input) {
        this.nombre_input = nombre_input;
        this.apellido_p_input = apellido_p_input;
        this.apellido_m_input = apellido_m_input;

    }

    public int getId() {
        return id;
    }
}

```

```
public void setId(int id) {
    this.id = id;
}

public String getRut_input() {
    return rut_input;
}

public void setRut_input(String rut_input) {
    this.rut_input = rut_input;
}

public String getNombre_input() {
    return nombre_input;
}

public void setNombre_input(String nombre_input) {
    this.nombre_input = nombre_input;
}

public String getApellido_p_input() {
    return apellido_p_input;
}

public void setApellido_p_input(String apellido_p_input) {
    this.apellido_p_input = apellido_p_input;
}

public String getApellido_m_input() {
    return apellido_m_input;
}

public void setApellido_m_input(String apellido_m_input) {
    this.apellido_m_input = apellido_m_input;
}

public boolean validar_rut(String rut_input, String digito_input){

    int rut_int = 0, suma=0, div=0, res=0, fin=0, var;
    StringBuilder rut = new StringBuilder(rut_input);
    rut.reverse();
    try{
        rut_int=Integer.parseInt(rut_input);
```

```

        for(int i=0;i<rut.length();i++){
            int serie[] = {2,3,4,5,6,7,2,3,4,5,6,7};
            char temp = rut.charAt(i);
            int tempn = Character.getNumericValue(temp);
            suma = suma + (tempn * serie[i]);
        }
        div = suma/11;
        res = suma - ( 11 * div);
        fin = 11 - res;
        if(fin == 11){
            i=Integer.toString(0);
        }else if(fin == 10){
            i="K";
        }else{
            i=Integer.toString(fin);
        }
        if(i.equalsIgnoreCase(digito_input)){
            return true;
        }else{
            return false;
        }
    }catch(Exception p){
        return false;
    }
}

public boolean nuevo_empleado(String rut, String nombre, String
apellido_p, String apellido_m){
    try{
        boolean respuesta;
        ModelAsistencia nuevo = new ModelAsistencia();
        return respuesta = nuevo.guarda_empleado(rut, nombre,
apellido_p, apellido_m);
    }catch(Exception e){
        JOptionPane.showMessageDialog(null, "Datos sin guardar");
        return false;
    }
}

public boolean buscar_empleado(String rut_input){
    try{
        ModelAsistencia modelo = new ModelAsistencia();
        boolean encontrado = modelo.buscar_empleado(rut_input);
        System.out.println(encontrado);
        if(encontrado){

```

```

        System.out.println("bandera");
        return true;
    }
} catch (Exception e) {
    return false;
}
return false;
}

public ArrayList<ControllerAsistencia> mostrar_empleado(String rut){
    ArrayList<ControllerAsistencia> empleado = new
ArrayList<ControllerAsistencia>();
    ModelAsistencia modelo_empleado = new ModelAsistencia();
    empleado =
modelo_empleado.mostrar_empleados(rut); //.mostrar_empleado(rut);
    return empleado;
}

public ArrayList<ControllerAsistencia> view_editar_mostrar_empleado(int
id){
    ArrayList<ControllerAsistencia> empleado = new
ArrayList<ControllerAsistencia>();
    ModelAsistencia modelo_empleado = new ModelAsistencia();
    empleado =
modelo_empleado.view_editar_mostrar_empleados(id); //.mostrar_empleado(rut);
    return empleado;
}

public boolean view_editar_eliminar_empleado(int id){
    try{
        ModelAsistencia modelo = new ModelAsistencia();
        boolean respuesta = modelo.view_editar_elimina_empleado(id);
        return respuesta;
    } catch (Exception e) {
        return false;
    }
}

public boolean editar_empleado(String rut, String nombres, String
apellido_p, String apellido_m){
    try{
        ModelAsistencia modelo = new ModelAsistencia();
        boolean resultado = modelo.actualizar_empleado(rut, nombres,
apellido_p, apellido_m);
    }
}

```

```

        return resultado;
    }catch(Exception e){
        return false;
    }
}

    public boolean view_editar_editar_empleado(int id, String nombres,
String apellido_p, String apellido_m){
        try{
            ModelAsistencia modelo = new ModelAsistencia();
            boolean resultado = modelo.view_editar_actualizar_empleado(id,
nombres, apellido_p, apellido_m);
            return resultado;
        }catch(Exception e){
            return false;
        }
    }

    public ArrayList<ControllerAsistencia> listado_empleados(){
        ArrayList<ControllerAsistencia> empleados = new
ArrayList<ControllerAsistencia>();
        ModelAsistencia modelo = new ModelAsistencia();
        empleados = modelo.listaEmpleados();
        return empleados;
    }

    public boolean ingreso(String rut){
        try{
            //Date d = new Date();
            //java.sql.Date date2 = new java.sql.Date(d.getTime());
            //System.out.println(date2);
            Date fecha_i = new Date();
            Timestamp ingreso = new Timestamp(fecha_i.getTime());
            ModelAsistencia intento = new ModelAsistencia();
            boolean resultado = intento.registrar_ingreso(rut, ingreso);
            return resultado;
        }catch(Exception e){
            return false;
        }
    }
}
}

```


Código modelo

```
package asistencia;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.Timestamp;
import java.util.ArrayList;

public class ModelAsistencia {

    private Connection conectar;
    private static final String usuario = "root";
    private static final String password = "";
    private static final String enlace =
"jdbc:mysql://localhost:3306/ASISTENCIA";
    private Statement smt = null;

    public ModelAsistencia(){
        try{
            this.conectar = DriverManager.getConnection(enlace, usuario,
password);
            this.smt = (Statement) this.conectar.createStatement();
            //System.out.println("CONECTADO");
            //this.smt.close();
        }catch(SQLException e){
            System.out.println(e);
        }
    }

    public boolean guarda_empleado(String rut, String nombre, String
apellido_p, String apellido_m){
        try{
            String intento = "INSERT INTO
empleados(`rut_empleado`,`nombres`,`apellido_paterno`,`apellido_materno`)
VALUES('"+rut+"','"+nombre+"','"+apellido_p+"','"+apellido_m+"')";
            this.smt.executeUpdate(intento);
            this.smt.close();
            return true;
        }
    }
}
```

```

        }catch(SQLException e){
            System.err.println(e);
            return false;
        }
    }

    public boolean elimina_empleado(String rut){
        try{
            String intento = "DELETE FROM empleados WHERE
rut_empleado="+rut;
            this.smt.executeUpdate(intento);
            this.smt.close();
            System.out.println("eliminado");
            return true;
        }catch(Exception e){
            System.out.println("no eliminado");
            return false;
        }
    }

    public boolean view_editar_elimina_empleado(int id){
        try{
            String intento = "DELETE FROM empleados WHERE id_empleado="+id;
            this.smt.executeUpdate(intento);
            this.smt.close();
            System.out.println("eliminado");
            return true;
        }catch(Exception e){
            System.out.println("no eliminado");
            return false;
        }
    }

    public ArrayList<ControllerAsistencia> mostrar_empleados(String rut){
        ArrayList<ControllerAsistencia> datos_empleado = new
ArrayList<ControllerAsistencia>();
        try{
            String intento = "SELECT *FROM empleados WHERE
rut_empleado="+rut;
            ResultSet resultado = this.smt.executeQuery(intento);
            if(resultado.first()){
                String nombre = resultado.getString("nombres");
                String apellido_p = resultado.getString("apellido_paterno");
                String apellido_m = resultado.getString("apellido_materno");
            }
        }
    }

```

```

        ControllerAsistencia empleado = new
ControllerAsistencia(nombre, apellido_p, apellido_m);
        datos_empleado.add(empleado);
        System.out.println("enviado");
        this.smt.close();
    }
} catch(SQLException e){
    System.out.println("error");
    return null;
}
return datos_empleado;

}

public ArrayList<ControllerAsistencia> view_editar_mostrar_empleados(int
id){
    ArrayList<ControllerAsistencia> datos_empleado = new
ArrayList<ControllerAsistencia>();
    try{
        String intento = "SELECT *FROM empleados WHERE id_empleado="+id;
        ResultSet resultado = this.smt.executeQuery(intento);
        System.out.println("consulta");
        if(resultado.first()){
            String rut = resultado.getString("rut_empleado");
            String nombre = resultado.getString("nombres");
            String apellido_p = resultado.getString("apellido_paterno");
            String apellido_m = resultado.getString("apellido_materno");
            ControllerAsistencia empleado = new
ControllerAsistencia(rut, nombre, apellido_p, apellido_m);
            datos_empleado.add(empleado);
            System.out.println("enviado");
            this.smt.close();
        }
    } catch(SQLException e){
        System.out.println("error");
        System.err.println(e);
        return null;
    }
    return datos_empleado;
}

public boolean buscar_empleado(String rut){
    try{
        String buscar = "SELECT *FROM empleados WHERE
rut_empleado="+rut;

```

```

        ResultSet resultado = this.smt.executeQuery(buscar);
        System.out.println(resultado);
        if(resultado.first()){
            String rutBD = resultado.getString("rut_empleado");
            System.out.println(rutBD);
            //boolean resp = true;
            return true;
        }
    }catch(SQLException e){
        return false;
    }
    return false;
}

public ControllerAsistencia mostrar_empleado(String rut){
    try{
        String consulta = "SELECT *FROM empleados WHERE
rut_empleado="+rut;
        ResultSet resultado = this.smt.executeQuery(consulta);
        if(resultado.first()){
            String nombre = resultado.getString("nombres");
            String apellido_p = resultado.getString("apellido_paterno");
            String apellido_m = resultado.getString("apellido_materno");
            ControllerAsistencia empleado = new
ControllerAsistencia(nombre, apellido_p, apellido_m);
            this.smt.close();
            return empleado;
        }
    }catch(SQLException e){
        return null;
    }
    return null;
}

public boolean actualizar_empleado(String rut, String nombre, String
apellido_p, String apellido_m){
    try{
        String intento = "UPDATE empleados SET nombres='"+nombre+"',
apellido_paterno='"+apellido_p+"', apellido_materno='"+apellido_m+"' WHERE
rut_empleado='"+rut+"'";
        this.smt.executeUpdate(intento);
        this.smt.close();
        return true;
    }catch(SQLException e){
        System.err.println();
    }
}

```

```

        return false;
    }
}

public boolean view_editar_actualizar_empleado(int id, String nombre,
String apellido_p, String apellido_m){
    try{
        String intento = "UPDATE empleados SET nombres='"+nombre+"',
apellido_paterno='"+apellido_p+"', apellido_materno='"+apellido_m+"' WHERE
id_empleado="+id;
        this.smt.executeUpdate(intento);
        this.smt.close();
        return true;
    }catch(SQLException e){
        System.err.println();
        return false;
    }
}

public ArrayList<ControllerAsistencia> listaEmpleados(){
    ArrayList<ControllerAsistencia> empleados = new
ArrayList<ControllerAsistencia>();
    try{
        String consulta = "SELECT *FROM empleados";
        ResultSet listado = this.smt.executeQuery(consulta);
        while(listado.next()){
            int id = listado.getInt("Id_empleado");
            String rut = listado.getString("rut_empleado");
            String nombres = listado.getString("nombres");
            String apellidop = listado.getString("apellido_paterno");
            String apellidom = listado.getString("apellido_materno");
            ControllerAsistencia empleado = new ControllerAsistencia(id,
rut, nombres, apellidop, apellidom);
            empleados.add(empleado);
        }
    }catch(SQLException e){
        System.err.println(e);
    }
    return empleados;
}

public boolean registrar_ingreso(String rut, Timestamp fecha){
    try{
        String intento = "INSERT INTO registro(`rut`,`ingreso`)
VALUES('"+rut+"', '"+fecha+"')";

```

```
        this.smt.executeUpdate(intento);
        this.smt.close();
        return true;
    } catch (SQLException e) {
        System.err.println(e);
        return false;
    }
}
}
```

VISTAS

ASISTENCIA

ASISTENCIA

Nuevo Empleado

Asistencia

Eliminar empleado

Listar empleados

Listar Asistencia

Editar empleado

Nuevo Empleado

Rut

Nombres

Apellido paterno

Apellido materno

Guardar

Edición de registros

Id	Rut	Nombres	Apellido paterno	Apellido materno
7	5889665	Juan Andres	Perezc	Diaz
8	12589654	Manuel Ignacio	Lopez	Miranda
18	18545887	Pablo	Rojas	Diaz
19	20448789	Manuel Andres	Zambrano	Morales
20	19823647	Pedro	Vega	Flores
22	18336428	Veronica	Silva	Arancibia
23	20987321	Andrea	Aliaga	Perez

ID del empleado

Buscar

Actualiza Listado

RUT

Editar

NOMBRES

Eliminar

APELLIDO PATERNO

APELLIDO MATERNO

Registro de Asistencia

28 / 5 / 2018
15:55:20

Ingrese RUT del emplead@:

INGRESAR

ingreso

Salida

BASE DE DATOS

Servidor: 127.0.0.1 » Base de datos: asistencia

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegios Rutinas Más

Filtros

Que contengan la palabra:

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> empleados	Examinar Estructura Buscar Insertar Vaciar Eliminar	7	InnoDB	utf8_general_ci	16 KB	-
<input type="checkbox"/> registro	Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	utf8_general_ci	16 KB	-
<input type="checkbox"/> usuario	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	16 KB	-
3 tablas	Número de filas	13	InnoDB	utf8_general_ci	48 KB	0 B

Seleccionar todo Para los elementos que están marcados:

Imprimir Diccionario de datos

Crear tabla

Nombre: Número de columnas:

Continuar

Servidor: 127.0.0.1 » Base de datos: asistencia » Tabla: empleados

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Más

Continuar

	id_empleado	rut_empleado	nombres	apellido_paterno	apellido_materno
<input type="checkbox"/> Editar Copiar Borrar	7	5889665	Juan Andres	Perez	Diaz
<input type="checkbox"/> Editar Copiar Borrar	8	12589654	Manuel Ignacio	Lopez	Miranda
<input type="checkbox"/> Editar Copiar Borrar	18	18545887	Pablo	Rojas	Diaz
<input type="checkbox"/> Editar Copiar Borrar	19	20448789	Manuel Andres	Zambrano	Morales
<input type="checkbox"/> Editar Copiar Borrar	20	19823647	Pedro	Vega	Flores
<input type="checkbox"/> Editar Copiar Borrar	22	18336428	Veronica	Silva	Arancibia
<input type="checkbox"/> Editar Copiar Borrar	23	20987321	Andrea	Aliaga	Perez

Seleccionar todo Para los elementos que están marcados: Editar Copiar Borrar Exportar

Mostrar todo Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Operaciones sobre los resultados de la consulta

Imprimir Copiar al portapapeles Exportar Mostrar gráfico Crear vista