

**Universidad Autónoma de Baja California**

**Grupo: 541**

**Docente:**

**Mayra Janeth Duran Rodriguez**



**Alumno:**

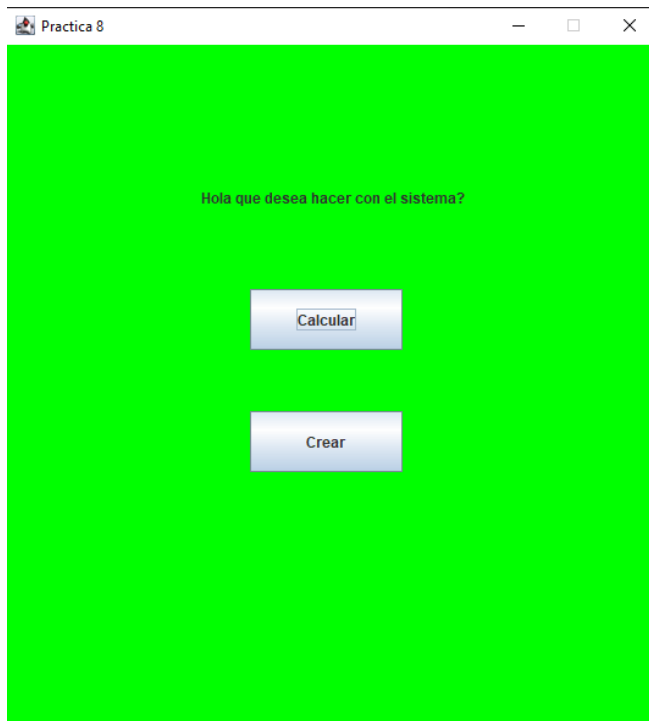
**Landa Luna Edgar Miguel 1263337**

**Programación orientada a objetos**

**Practica #8**

**Fecha de entrega: 03-diciembre-2020**

## 1. Menú



## 2. Ventana Crear

The screenshot shows a window titled 'Practica 8' with a cyan background. The form contains the following fields and buttons:

- Nombre:** Edgar
- Identificacion:** 1
- Tipo Empleado:** Empleado asalariado
- Sueldo semanal:** 1200
- Sueldo por hora:**
- Sueldo por comision:**
- Regresar** button
- Crear** button

### 3. Ventana crear 2

Practica 8

Regresar

Nombre:  
Juan

Identificacion:  
2

Tipo Empleado:  
Empleado por hora

Sueldo semanal:

Sueldo por hora:  
200

Sueldo por comision:

Crear

### 4. Ventana crear 3

Practica 8

Regresar

Nombre:  
Luis

Identificacion:  
3

Tipo Empleado:  
Empleado por comi...

Sueldo semanal:

Sueldo por hora:

Sueldo por comision:  
150

Crear

## 5. Ventana crear 4

Practica 8

Regresar

Nombre:  
Andres

Identificacion:  
4

Tipo Empleado:  
Emplador A/C

Sueldo semanal:  
1300

Sueldo por hora:

Sueldo por comision:  
90

Crear

## 6. Ventana calcular 1

Practica 8

Regresar

Nombre:  
Edgar

Identificacion:

Buscar

Tipo Empleado:  
Empleado asalariado

Sueldo semanal:  
1200.0

Horas trabajadas:

Sueldo por hora:

Sueldo total:

Ventas totales

Sueldo por comision:

Sueldo total:

Calcular Salario

## 7. Ventana calcular 2

Practica 8

Nombre: Juan

Identificacion:

Buscar

Tipo Empleado: Empleado por hora

Sueldo semanal:

Horas trabajadas: 10

Sueldo por hora: 200.0

Sueldo total: 2000.0

Ventas totales:

Sueldo por comision:

Sueldo total:

Calcular Salario

## 8. Ventana calcular 3

Practica 8

Nombre: Luis

Identificacion:

Buscar

Tipo Empleado: Empleado por comi...

Sueldo semanal:

Horas trabajadas:

Sueldo por hora:

Sueldo total:

Ventas totales: 7

Sueldo por comision: 150.0

Sueldo total: 1050.0

Calcular Salario

## 9. Ventana calcular 4

 Practica 8—□×

Regresar

Nombre:

Andres

Identificacion:

Tipo Empleado:

Emplador A/C

Sueldo semanal:

1300.0

Horas trabajadas:

Sueldo por hora:

Sueldo total:

Ventas totales

10

Sueldo por comision:

90.0

Sueldo total:

2200.0

Buscar

Calcular Salario

## Codigo

### 1. Menú principal

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

public class Practica8 {
    JFrame frame = new JFrame("Practica 8");//frame principal
    ///llama a los paneles
    PanelMenu panelMenu = new PanelMenu();
    PanelCrear panelCrear = new PanelCrear();
    PanelCalcular panelCalcular = new PanelCalcular();
    ///instancia de empleados
    public static Empleados[] empleados = new Empleados[10];

    public static void main(String[] args) {
        Practica8 menu = new Practica8();
    }

    public Practica8() {
        ///propiedades del frame
        frame.setBounds(450, 100, 550, 600);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.add(panelMenu);
        frame.setVisible(true);
        frame.setResizable(false);
        acciones();
    }

    public void acciones() {
        ///acciones de botones que simplemente permiten el cambio de paneles de uno a
        otro
        panelMenu.btnCrear.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                panelMenu.setVisible(false);
                panelCrear.setVisible(true);
                frame.add(panelCrear);
            }
        });

        panelMenu.btnCalcular.addActionListener(new ActionListener(){
            @Override
```

```

        public void actionPerformed(ActionEvent e) {
            panelMenu.setVisible(false);
            panelCalculuar.setVisible(true);
            frame.add(panelCalculuar);
        }
    });

    panelCrear.btnBack.addActionListener(new ActionListener(){
        @Override
        public void actionPerformed(ActionEvent e) {
            panelMenu.setVisible(true);
            panelCrear.setVisible(false);
            frame.add(panelMenu);
        }
    });

    panelCalculuar.btnBack.addActionListener(new ActionListener(){
        @Override
        public void actionPerformed(ActionEvent e) {
            panelMenu.setVisible(true);
            panelCalculuar.setVisible(false);
            frame.add(panelMenu);
        }
    });
}
}

```

## 2. PanelMenu

```

import java.awt.Color;
import javax.swing.*.*;

public class PanelMenu extends JPanel{

    public JButton btnCrear;
    public JButton btnCalculuar;
    JLabel lblHola;

    public PanelMenu(){//creacion del pequeno menu
        setLayout(null);
        setBackground(Color.GREEN);
        lblHola = new JLabel("Hola que desea hacer con el sistema?");
        lblHola.setBounds(160, 100, 240, 50);

        btnCalculuar = new JButton("Calcular");
    }
}

```



```

        btnCalcular.setBounds(200,200,125,50);

        btnCrear = new JButton("Crear");
        btnCrear.setBounds(200, 300, 125, 50);

        add(lblHola);
        add(btnCalcular);
        add(btnCrear);
    }
}

```

### 3. PanelCrear

```

import static javax.swing.JOptionPane.ERROR_MESSAGE;
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.*;

//este codigo no lo explico por que creo que
//esta muy dinamico y entendible
public class PanelCrear extends JPanel {
    //Creacion de objeto
    JButton btnCrear = new JButton("Crear");
    JButton btnBack = new JButton("Regresar");

    JTextField txtName = new JTextField();
    JTextField txtId = new JTextField();
    JTextField txtSueldo = new JTextField();
    JTextField txtHoras = new JTextField();
    JTextField txtComision = new JTextField();

    JComboBox<String> comboTypeE = new JComboBox<>();

    JLabel lblName = new JLabel("Nombre:");
    JLabel lblId = new JLabel("Identificacion:");
    JLabel lblTypeE = new JLabel("Tipo Empleado:");
    JLabel lblSueldo = new JLabel("Sueldo semanal:");
    JLabel lblHoras = new JLabel("Sueldo por hora:");
    JLabel lblComision = new JLabel("Sueldo por comision:");

    int i = 0;

    public PanelCrear() {
        setLayout(null);
    }
}

```

```
this.setBackground(Color.cyan);
///armando ventana
txtName.setBounds(180, 30, 145, 50);
txtId.setBounds(180, 100, 145, 50);
txtSueldo.setBounds(180, 240, 145, 50);
txtHoras.setBounds(180, 310, 145, 50);
txtComision.setBounds(180, 390, 145, 50);

txtSueldo.setEditable(true);
txtHoras.setEditable(false);
txtComision.setEditable(false);

comboTypeE.setBounds(180, 170, 145, 50);

comboTypeE.addItem("Empleado asalariado");
comboTypeE.addItem("Empleado por hora");
comboTypeE.addItem("Empleado por comision");
comboTypeE.addItem("Emplador A/C");

lblName.setBounds(180, 10, 145, 10);
lblId.setBounds(180, 85, 145, 10);
lblTypeE.setBounds(180, 152, 145, 12);
lblSueldo.setBounds(180, 225, 145, 10);
lblHoras.setBounds(180, 295, 145, 12);
lblComision.setBounds(180, 370, 145, 12);

btnBack.setBounds(10, 30, 145, 50);
btnCrear.setBounds(180, 475, 145, 50);

accionesCrear();

add(lblName);
add(lblId);
add(lblTypeE);
add(lblSueldo);
add(lblHoras);
add(lblComision);

add(txtName);
add(txtId);
add(txtSueldo);
add(txtHoras);
add(txtComision);

add(comboTypeE);
```

```

        add(btnBack);
        add(btnCrear);
    }
    //metodos de los botones
    public void accionesCrear() {
        btnCrear.addActionListener(new ActionListener() {
            ///esto permite la creacion del tipo de empleado
            ///dependiendo del combobox
            @Override
            public void actionPerformed(ActionEvent e) {
                switch (comboTypeE.getSelectedItem().toString()) {
                    case "Empleado asalariado":
                        try {
                            Practica8.empleados[i] = new EmpleadosA(txtName.
getText(),
                                Integer.parseInt(txtId.getText()), 1, Fl
oat.parseFloat(txtSueldo.getText()));
                            i++;
                        } catch (Exception er) {
                            JOptionPane.showMessageDialog(txtId, "Datos no e
stan completos", "Error", ERROR_MESSAGE,
                                null);
                        }
                        break;
                    case "Empleado por hora":
                        try {
                            Practica8.empleados[i] = new EmpleadosH(txtName.
getText(),
                                Integer.parseInt(txtId.getText()), 2, Fl
oat.parseFloat(txtHoras.getText()));
                            i++;
                        } catch (Exception er) {
                            JOptionPane.showMessageDialog(txtId, "Datos no e
stan completos", "Error", ERROR_MESSAGE,
                                null);
                        }
                        break;
                    case "Empleado por comision":
                        try {
                            Practica8.empleados[i] = new EmpleadosC(txtName.
getText(),
                                Integer.parseInt(txtId.getText()), 3, Fl
oat.parseFloat(txtComision.getText()));
                            i++;

```

```

        } catch (Exception er) {
            JOptionPane.showMessageDialog(txtId, "Datos no e
stan completos", "Error", ERROR_MESSAGE,
                null);
        }
        break;
        case "Emplador A/C":
            try {
                Practica8.empleados[i] = new EmpleadosAC(txtName
.getText(),
                    Integer.parseInt(txtId.getText()), 4, Fl
oat.parseFloat(txtComision.getText()),
                    Float.parseFloat(txtSueldo.getText()));
                i++;
            } catch (Exception er) {
                JOptionPane.showMessageDialog(txtId, "Datos no e
stan completos", "Error", ERROR_MESSAGE,
                    null);
            }
            break;
        default:
            break;
    }

    limpiar();
}
});
///Dependiendo del combobox
///deja editar en unos u otros
comboTypeE.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        switch (comboTypeE.getSelectedItem().toString()) {
            case "Empleado asalariado":
                txtSueldo.setEditable(true);
                txtHoras.setEditable(false);
                txtComision.setEditable(false);
                break;
            case "Empleado por hora":
                txtSueldo.setEditable(false);
                txtHoras.setEditable(true);
                txtComision.setEditable(false);
                break;
            case "Empleado por comision":
                txtSueldo.setEditable(false);

```

```

        txtHoras.setEditable(false);
        txtComision.setEditable(true);
        break;
    case "Emplador A/C":
        txtSueldo.setEditable(true);
        txtHoras.setEditable(false);
        txtComision.setEditable(true);
        break;
    default:
        break;
    }
}

});
}
//limpiar los textfield
public void limpiar() {
    txtName.setText("");
    txtId.setText("");
    txtSueldo.setText("");
    txtHoras.setText("");
    txtComision.setText("");
    comboTypeE.setSelectedIndex(0);
    txtSueldo.setEditable(true);
    txtHoras.setEditable(false);
    txtComision.setEditable(false);
}
}

```

#### 4. PanelCalcular

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class PanelCalcular extends JPanel {
    int posicion;
    //creacion de objetos
    JButton btnCalcular = new JButton("Calcular Salario");
    JButton btnBuscar = new JButton("Buscar");
    JButton btnBack = new JButton("Regresar");

    JTextField txtName = new JTextField();
    JTextField txtId = new JTextField();
    JTextField txtSueldo = new JTextField();
}

```

```

JTextField txtHoras = new JTextField();
JTextField txtComision = new JTextField();

JTextField txtCalHoras = new JTextField();
JTextField txtCalComision = new JTextField();
JTextField txtHorasT = new JTextField();
JTextField txtComisionT = new JTextField();
JComboBox<String> comboTypeE = new JComboBox<>();

JLabel lblName = new JLabel("Nombre:");
JLabel lblId = new JLabel("Identificacion:");
JLabel lblTypeE = new JLabel("Tipo Empleado:");
JLabel lblSueldo = new JLabel("Sueldo semanal:");
JLabel lblHoras = new JLabel("Sueldo por hora:");
JLabel lblComision = new JLabel("Sueldo por comision:");
JLabel lblCalHoras = new JLabel("Sueldo total:");
JLabel lblCalComision = new JLabel("Sueldo total:");
JLabel lblHorasT = new JLabel("Horas trabajadas:");
JLabel lblComisionT = new JLabel("Ventas totales");
PanelCrear panelCrear = new PanelCrear();

public PanelCalcular() {
    setLayout(null);
    this.setBackground(Color.ORANGE);
    txtName.setBounds(180, 30, 145, 50);
    txtId.setBounds(180, 100, 145, 50);
    txtSueldo.setBounds(180, 240, 145, 50);
    txtHoras.setBounds(180, 310, 145, 50);
    txtComision.setBounds(180, 390, 145, 50);
    txtCalHoras.setBounds(350, 310, 145, 50);
    txtCalComision.setBounds(350, 390, 145, 50);
    txtHorasT.setBounds(10, 310, 145, 50);
    txtComisionT.setBounds(10, 390, 145, 50);

    txtCalComision.setEditable(false);
    txtCalHoras.setEditable(false);
    txtName.setEditable(false);
    txtSueldo.setEditable(false);
    txtHoras.setEditable(false);
    txtComision.setEditable(false);

    comboTypeE.setBounds(180, 170, 145, 50);

    comboTypeE.addItem("Empleado asalariado");
    comboTypeE.addItem("Empleado por hora");
}

```

```
comboTypeE.addItem("Empleado por comision");
comboTypeE.addItem("Emplador A/C");
comboTypeE.setEditable(false);

lblName.setBounds(180, 10, 145, 10);
lblId.setBounds(180, 85, 145, 10);
lblTypeE.setBounds(180, 152, 145, 12);
lblSueldo.setBounds(180, 225, 145, 10);
lblHoras.setBounds(180, 295, 145, 12);
lblComision.setBounds(180, 370, 145, 12);
lblCalHoras.setBounds(350, 295, 145, 12);
lblCalComision.setBounds(350, 370, 145, 12);
lblComisionT.setBounds(10, 370, 145, 10);
lblHorasT.setBounds(10, 295, 145, 12);

btnBack.setBounds(10, 30, 145, 50);
btnBuscar.setBounds(350, 100, 145, 50);
btnCalcular.setBounds(180, 475, 145, 59);

accionesCrear();

add(lblName);
add(lblId);
add(lblTypeE);
add(lblSueldo);
add(lblHoras);
add(lblComision);
add(lblCalHoras);
add(lblCalComision);
add(lblHorasT);
add(lblComisionT);

add(txtName);
add(txtId);
add(txtSueldo);
add(txtHoras);
add(txtComision);
add(txtHorasT);
add(txtComisionT);
add(txtCalHoras);
add(txtCalComision);
add(comboTypeE);

add(btnBuscar);
add(btnCalcular);
```

```

        add(btnBack);
    }

    private void accionesCrear() {

        btnBuscar.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    for (int i = 0; i < Practica8.empleados.length; i++) {
                        ///permite leer la informacion de los empleados
                        ///y reflejarlos
                        if (Practica8.empleados[i].getId().toString().equals
(txtId.getText())) {

                            switch (Practica8.empleados[i].typeE) {
                                case 1:
                                    limpiar();
                                    txtName.setText(Practica8.empleados[i].g
etName());

                                    txtSueldo.setText(
                                        ((EmpleadosA) Practica8.empleado
s[i]).getSueldoSemanal().toString());
                                    comboTypeE.setSelectedIndex(0);
                                    posicion = i;

                                    break;
                                case 2:
                                    limpiar();
                                    txtName.setText(Practica8.empleados[i].g
etName());

                                    txtHoras.setText(((EmpleadosH) Practica8
.empleados[i]).getSueldoHoras().toString());
                                    comboTypeE.setSelectedIndex(1);
                                    posicion = i;

                                    break;
                                case 3:
                                    limpiar();
                                    txtName.setText(Practica8.empleados[i].g
etName());

                                    txtComision.setText(((EmpleadosC) Practi
ca8.empleados[i]).getComision().toString());
                                    comboTypeE.setSelectedIndex(2);
                                    posicion = i;

```



```

                break;
            case 4:
                limpiar();
                txtName.setText(((EmpleadosAC) Practica8
.empleados[i]).getName());
                txtComision.setText(((EmpleadosAC) Pract
ica8.empleados[i]).getSueldoC().toString());
                txtSueldo.setText(((EmpleadosAC) Practic
a8.empleados[i]).getSueldoA().toString());
                comboTypeE.setSelectedIndex(3);
                posicion = i;
                break;
            default:
                break;
        }
    }
}

} catch (Exception er) {
}
}
});
btnCalcular.addActionListener(new ActionListener() {
    ///depnde del tipo de empleado escogera su calculo
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            System.out.println(posicion + " " + Practica8.empleados[
posicion].typeE);
            if (Practica8.empleados[posicion].typeE == 1) {
                ((EmpleadosA) Practica8.empleados[posicion]).calSala
rio(Float.parseFloat(txtSueldo.getText()));
            } else if (Practica8.empleados[posicion].typeE == 2) {
                txtCalHoras.setText(((EmpleadosH) Practica8.empleado
s[posicion])
                .calSalario(Float.parseFloat(txtHorasT.getTe
xt())).toString());
            } else if (Practica8.empleados[posicion].typeE == 3) {
                txtCalComision.setText(((EmpleadosC) Practica8.emple
ados[posicion])
                .calSalario(Float.parseFloat(txtComisionT.ge
tText())).toString());
            }
        } catch (Exception er) {
            JOptionPane.showMessageDialog(null, er.getMessage());
        }
    }
});

```

```

        } else if (Practica8.empleados[posicion].typeE == 4) {
            txtCalComision.setText(((EmpleadosAC) Practica8.empl
eados[posicion])
                                .calSalario(Float.parseFloat(txtComisionT.ge
tText()))).toString());
        }
    } catch (Exception er) {
    }

}

});
}

public void limpiar() {
    txtName.setText("");
    txtId.setText("");
    txtSueldo.setText("");
    txtHoras.setText("");
    txtComision.setText("");
    txtCalComision.setText("");
    txtCalHoras.setText("");
    txtHorasT.setText("");
    txtComisionT.setText("");
    comboTypeE.setSelectedIndex(0);
    txtSueldo.setEditable(true);
    txtHoras.setEditable(false);
    txtComision.setEditable(false);
}
}

```

## 5. interfazE

```

public interface InterfazE {
    Float calSalario(float hola);
}

```

## 6. Empleados

```
public abstract class Empleados implements InterfazE {
    String name;
    public Integer id;
    int typeE;

    public Empleados(String name, Integer id, int typeE) {
        this.name=name;
        this.id=id;
        this.typeE=typeE;
    }

    public Float calSalario(float f){
        return null;
    }

    public String getName() {
        return this.name;
    }

    public Integer getId() {
        return this.id;
    }
}
```

## 7. EmpleadosH

```
public class EmpleadosH extends Empleados {
    Float sueldoHoras;
    public EmpleadosH(String name, Integer id, int typeE, Float sueldoHoras)
    {
        super(name, id, typeE);
        this.sueldoHoras=sueldoHoras;
    }

    public Float getSueldoHoras() {
        return this.sueldoHoras;
    }
    @Override
    public Float calSalario(float horas) {
        return sueldoHoras * horas;
    }
}
```

## 8. EmpleadosA

```
public class EmpleadosA extends Empleados {  
    public Float sueldoSemenal;  
  
    public EmpleadosA(String name, Integer id, int typeE, Float sueldoSemenal)  
1) {  
        super(name,id,typeE);  
        this.sueldoSemenal=sueldoSemenal;  
    }  
  
    public Float getSueldoSemenal() {  
        return this.sueldoSemenal;  
    }  
  
    public Float calSalario(float sueldoSemenal) {  
        return this.sueldoSemenal;  
    }  
}
```

## 9. EmpleadosAC

```
public class EmpleadosAC extends Empleados{  
  
    Float sueldoA;  
    Float sueldoC;  
  
    public EmpleadosAC(String name, Integer id, int typeE, Float sueldoC, Fl  
oat sueldoA) {  
        super(name, id, typeE);  
        this.sueldoA = sueldoA;  
        this.sueldoC = sueldoC;  
    }  
  
    public Float calSalario(float numVentas){  
        return sueldoA + sueldoC * numVentas;  
    }  
  
    public Float getSueldoA() {  
        return this.sueldoA;  
    }  
  
    public Float getSueldoC() {  
        return this.sueldoC;  
    }  
}
```

## 10. EmpleadosC

```
public class EmpleadosC extends Empleados {  
    private Float comision;  
    public EmpleadosC(String name, Integer id, int typeE, Float comision) {  
        super(name, id, typeE);  
        this.comision = comision;  
    }  
  
    public Float getComision() {  
        return this.comision;  
    }  
  
    public Float calSalarario(float numVentas){  
        return comision * numVentas;  
    }  
}
```