**Universidad Autónoma de Baja California**

**Grupo: 541**

**Docente:**

**Mayra Janeth Duran Rodriguez**

****

**Alumno:**

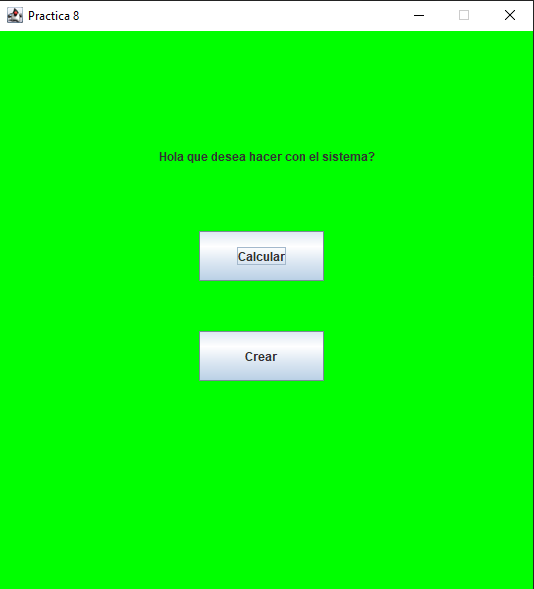
**Landa Luna Edgar Miguel 1263337**

**Programación orientada a objetos**

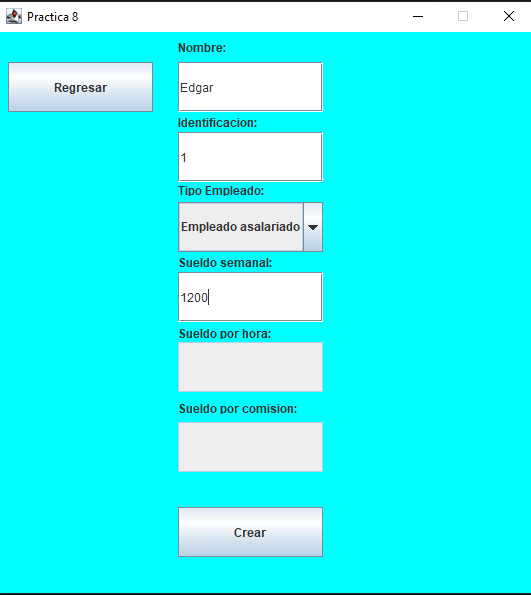
**Practica #8**

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

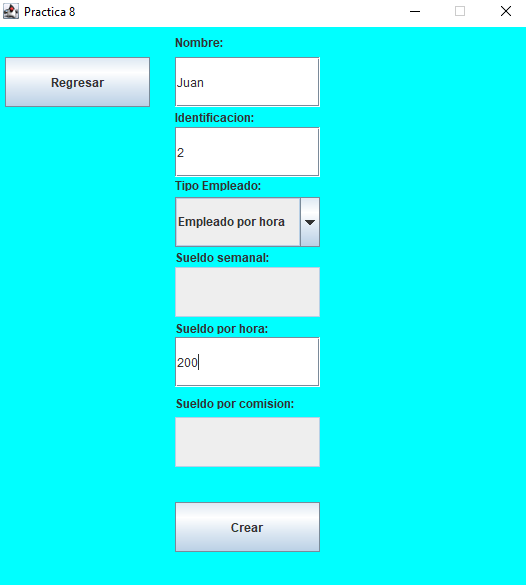
1. Menú



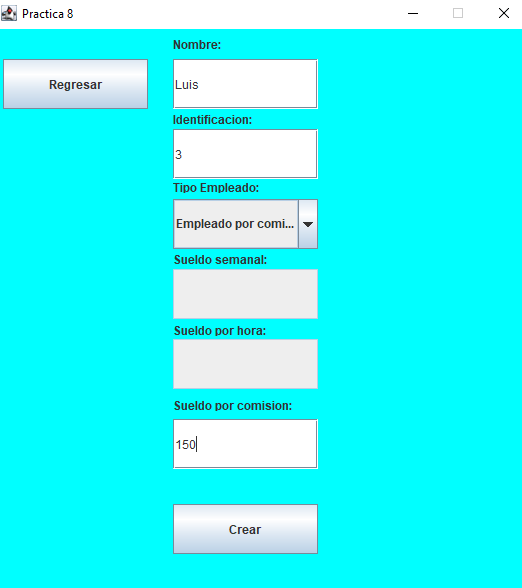
1. Ventana Crear



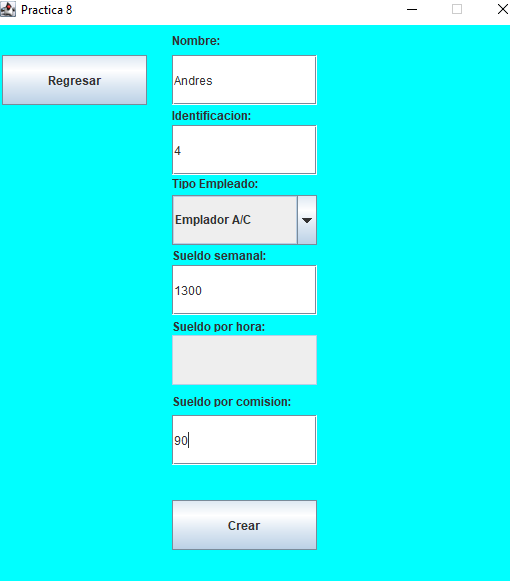
1. Ventana crear 2



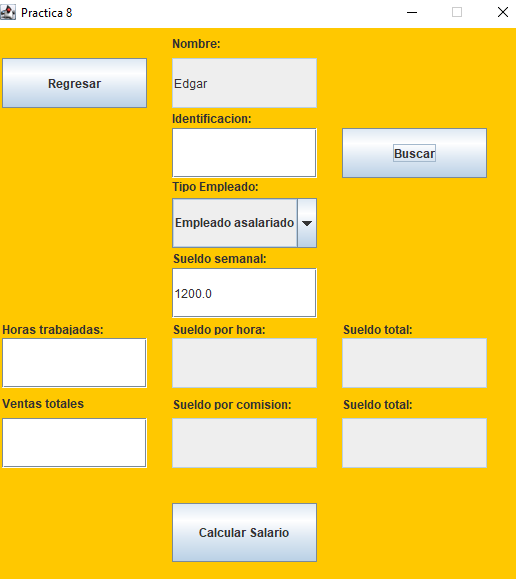
1. Ventana crear 3



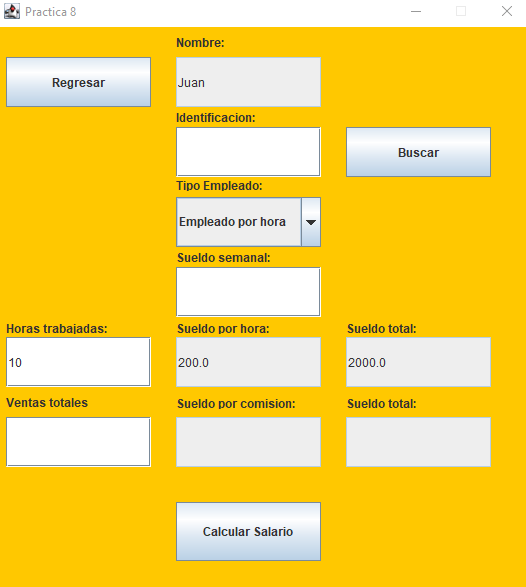
1. Ventana crear 4



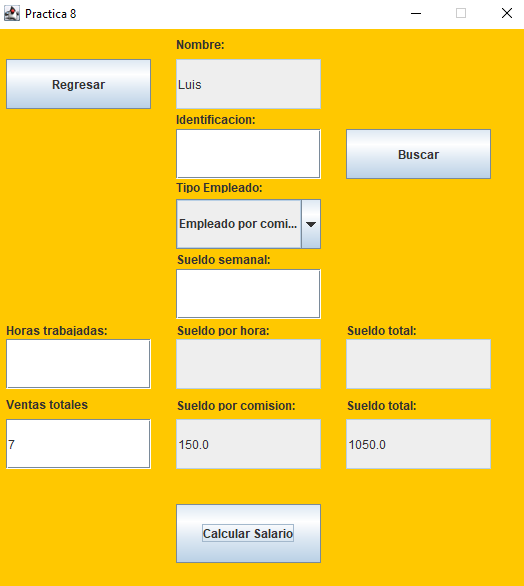
1. Ventana calcular 1



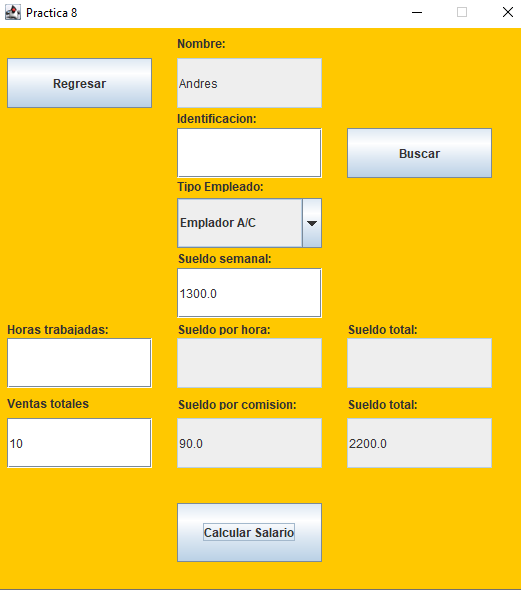
1. Ventana calcular 2



1. Ventana calcular 3



1. Ventana calcular 4



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);

                panelCalcular.setVisible(true);

                frame.add(panelCalcular);

            }

        });

        panelCrear.btnBack.addActionListener(new ActionListener(){

            @Override

            public void actionPerformed(ActionEvent e) {

                panelMenu.setVisible(true);

                panelCrear.setVisible(false);

                frame.add(panelMenu);

            }

        });

        panelCalcular.btnBack.addActionListener(new ActionListener(){

            @Override

            public void actionPerformed(ActionEvent e) {

                panelMenu.setVisible(true);

                panelCalcular.setVisible(false);

                frame.add(panelMenu);

            }

        });

    }

}

1. PanelMenu

import java.awt.Color;

import javax.swing.\*;

public class PanelMenu extends JPanel{

    public JButton btnCrear;

    public JButton btnCalcular;

    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);

        btnCalcular =  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);

    }

}

1. 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 {

    //Creaccion 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 creaccion 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, Float.parseFloat(txtSueldo.getText()));

                            i++;

                        } catch (Exception er) {

                            JOptionPane.showMessageDialog(txtId, "Datos no estan completos", "Error", ERROR\_MESSAGE,

                                    null);

                        }

                        break;

                    case "Empleado por hora":

                        try {

                            Practica8.empleados[i] = new EmpleadosH(txtName.getText(),

                                    Integer.parseInt(txtId.getText()), 2, Float.parseFloat(txtHoras.getText()));

                            i++;

                        } catch (Exception er) {

                            JOptionPane.showMessageDialog(txtId, "Datos no estan completos", "Error", ERROR\_MESSAGE,

                                    null);

                        }

                        break;

                    case "Empleado por comision":

                        try {

                            Practica8.empleados[i] = new EmpleadosC(txtName.getText(),

                                    Integer.parseInt(txtId.getText()), 3, Float.parseFloat(txtComision.getText()));

                            i++;

                        } catch (Exception er) {

                            JOptionPane.showMessageDialog(txtId, "Datos no estan completos", "Error", ERROR\_MESSAGE,

                                    null);

                        }

                        break;

                    case "Emplador A/C":

                        try {

                            Practica8.empleados[i] = new EmpleadosAC(txtName.getText(),

                                    Integer.parseInt(txtId.getText()), 4, Float.parseFloat(txtComision.getText()),

                                    Float.parseFloat(txtSueldo.getText()));

                            i++;

                        } catch (Exception er) {

                            JOptionPane.showMessageDialog(txtId, "Datos no estan completos", "Error", ERROR\_MESSAGE,

                                    null);

                        }

                        break;

                    default:

                        break;

                }

                limpiar();

            }

        });

        ///Dependeiendo 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);

    }

}

1. PanelCalcular

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class PanelCalcular extends JPanel {

    int posicion;

    //creaccion 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 refleajarlos

                        if (Practica8.empleados[i].getId().toString().equals(txtId.getText())) {

                            switch (Practica8.empleados[i].typeE) {

                                case 1:

                                    limpiar();

                                    txtName.setText(Practica8.empleados[i].getName());

                                    txtSueldo.setText(

                                            ((EmpleadosA) Practica8.empleados[i]).getSueldoSemanal().toString());

                                    comboTypeE.setSelectedIndex(0);

                                    posicion = i;

                                    break;

                                case 2:

                                    limpiar();

                                    txtName.setText(Practica8.empleados[i].getName());

                                    txtHoras.setText(((EmpleadosH) Practica8.empleados[i]).getSueldoHoras().toString());

                                    comboTypeE.setSelectedIndex(1);

                                    posicion = i;

                                    break;

                                case 3:

                                    limpiar();

                                    txtName.setText(Practica8.empleados[i].getName());

                                    txtComision.setText(((EmpleadosC) Practica8.empleados[i]).getComision().toString());

                                    comboTypeE.setSelectedIndex(2);

                                    posicion = i;

                                    break;

                                case 4:

                                    limpiar();

                                    txtName.setText(((EmpleadosAC) Practica8.empleados[i]).getName());

                                    txtComision.setText(((EmpleadosAC) Practica8.empleados[i]).getSueldoC().toString());

                                    txtSueldo.setText(((EmpleadosAC) Practica8.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]).calSalario(Float.parseFloat(txtSueldo.getText()));

                    } else if (Practica8.empleados[posicion].typeE == 2) {

                        txtCalHoras.setText(((EmpleadosH) Practica8.empleados[posicion])

                                .calSalario(Float.parseFloat(txtHorasT.getText())).toString());

                    } else if (Practica8.empleados[posicion].typeE == 3) {

                        txtCalComision.setText(((EmpleadosC) Practica8.empleados[posicion])

                                .calSalario(Float.parseFloat(txtComisionT.getText())).toString());

                    } else if (Practica8.empleados[posicion].typeE == 4) {

                        txtCalComision.setText(((EmpleadosAC) Practica8.empleados[posicion])

                                .calSalario(Float.parseFloat(txtComisionT.getText())).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);

*}*

*}*

1. interfazE

public interface InterfazE {

    Float calSalario(float hola);

}

1. 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;

    }

}

1. 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;

    }

}

1. EmpleadosA

public class EmpleadosA extends Empleados {

    public Float sueldoSemenal;

    public EmpleadosA(String name, Integer id, int typeE, Float sueldoSemenal) {

        super(name,id,typeE);

        this.sueldoSemenal=sueldoSemenal;

    }

    public Float getSueldoSemanal() {

        return this.sueldoSemenal;

    }

    public Float calSalario(float sueldoSemanal) {

        return this.sueldoSemenal;

    }

}

1. EmpleadosAC

public class EmpleadosAC extends Empleados{

    Float sueldoA;

    Float sueldoC;

    public EmpleadosAC(String name, Integer id, int typeE, Float sueldoC, Float 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;

    }

}

1. 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 calSalario(float numVentas){

        return comision \* numVentas;

    }

}