



Programación Orientada a Objetos II

Ingeniería en Desarrollo de Software

Unidad 3. Componentes gráficos

Actividad Complementaria

Francisco Manuel López García

Matrícula: ES1921010154

Docente: Alicia Guadalupe Robertos Pinto

08 de septiembre 2020

Índice

Introducción	3
Contenido	3
<i>Módulos:.....</i>	<i>4</i>
<i>Mapa de navegación.....</i>	<i>6</i>
<i>Módulos Inventario y Recursos Humanos.....</i>	<i>6</i>
<i>Capturas de pantalla</i>	<i>10</i>
<i>Código Fuente</i>	<i>14</i>
<i>Main</i>	<i>14</i>
<i>Clase framePrincipal.....</i>	<i>15</i>
<i>Clase Inventario</i>	<i>19</i>
<i>Clase Articulo.....</i>	<i>28</i>
<i>Clase RecursosHumanos.....</i>	<i>29</i>
<i>Clase Empleado.....</i>	<i>40</i>
<i>Clase Nomina</i>	<i>40</i>
<i>Clase Consultas</i>	<i>44</i>
<i>Clase Conexion</i>	<i>46</i>
<i>Video</i>	<i>49</i>
Bibliografía:	50

Introducción

Con esta actividad terminamos y aplicamos todo lo que vimos a lo largo de las unidades de esta materia. Con lo cual se desarrolla el programa que hemos venido haciendo desde la unidad uno, ahora dándole funcionalidades mediante una base de datos, en donde podremos insertar, consultar, modificar y eliminar los datos de dicha base de datos a través de la aplicación generada en java. Se vera como realizar las conexiones correspondientes, así como las sentencias DLL para SQL.

Todo se integra en un proyecto de NetBeans y en una base de datos de MySql con XAMPP.

Contenido

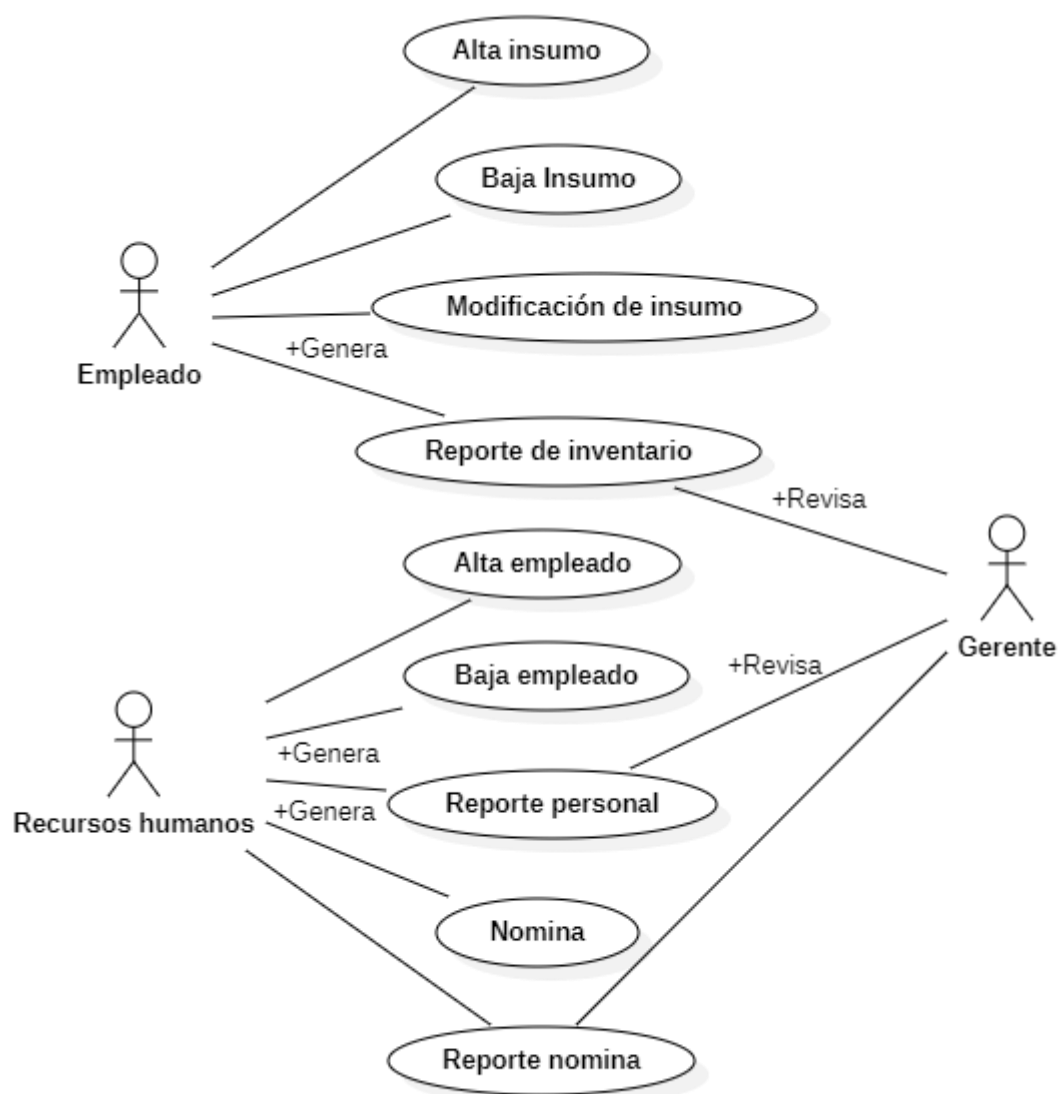
Un restaurante de comida mexicana abre sus puertas por primera vez en el año 2015, con unas cuantas mesas y un espacio reducido, actualmente cuenta con 20 sucursales ubicadas en diferentes puntos estratégicos de la Ciudad de México.

Pero ha tenido diversos problemas con la expansión, ya que, desea tener una oficina central para llevar a cabo actividades administrativas, pero se ha encontrado con ciertas irregularidades, primeramente no existe un control preciso de las ventas, el inventario no se ajusta a las entradas y salidas de mercancía, el área de recursos humanos no tiene un control preciso del personal y se tarda bastante para realizar un reporte y no existe un departamento para la nómina, por lo cual requiere un sistema informático que permite optimizar los recursos de la empresa y de personal.

Por lo cual, el dueño desea contratar los servicios de una consultoría que lo asesore para automatizar todos los procesos de la empresa y sea competitiva a nivel nacional, para lo cual, el sistema informático debe contener los siguientes módulos:

- Un módulo de catálogo de tipos de insumos (altas, bajas, modificaciones).
- Un módulo de inventarios (altas, bajas, modificaciones, existencias). Cuando se da de alta un nuevo registro de inventario, se selecciona del catálogo el tipo de insumo.
- Un módulo de Recursos humanos (altas, bajas modificaciones).
- Un módulo de nómina (número de empleado, salario base, impuestos, imprimir recibo)

- Un módulo para generar reportes (Inventarios, Personal, Nomina)



Módulos:

Catálogo de insumos

En este módulo se podrá realizar el alta, baja y modificación de los insumos que se utilizan en el restaurante, con lo cual llevara el control del catálogo de insumos que se utilizan, con la información necesaria de cada insumo, como lo es su nombre, tipo, precio etc.

Se utilizarán tres formularios:

- Alta
- Baja
- Modificación

Inventarios

Este módulo se encargará de llevar el inventario de los insumos, con lo cual será el encargado de las entradas y salidas de ellos, con el podemos controlar la cantidad de existencia de los insumos y saber cuáles están por terminarse y poder comprar más.

Se utilizarán cuatro formularios:

- Alta
- Baja
- Modificación
- Existencia (este te mostrara la existencia de un insumo en específico)

Recursos humanos

Este módulo se encarga de administrar al personal de los restaurantes, con lo cual se integrarán los datos de los empleados como lo es su nombre, dirección, teléfono, puesto, salario etc.

Se utilizarán tres formularios:

- Alta
- Baja
- Modificación

Nómina

Este módulo se vinculará con recursos humanos para obtener los datos de los empleados, con lo cual este módulo se encargará de generar los pagos de los empleados. También se encargará de calcular los impuestos y de imprimir los recibos correspondientes.

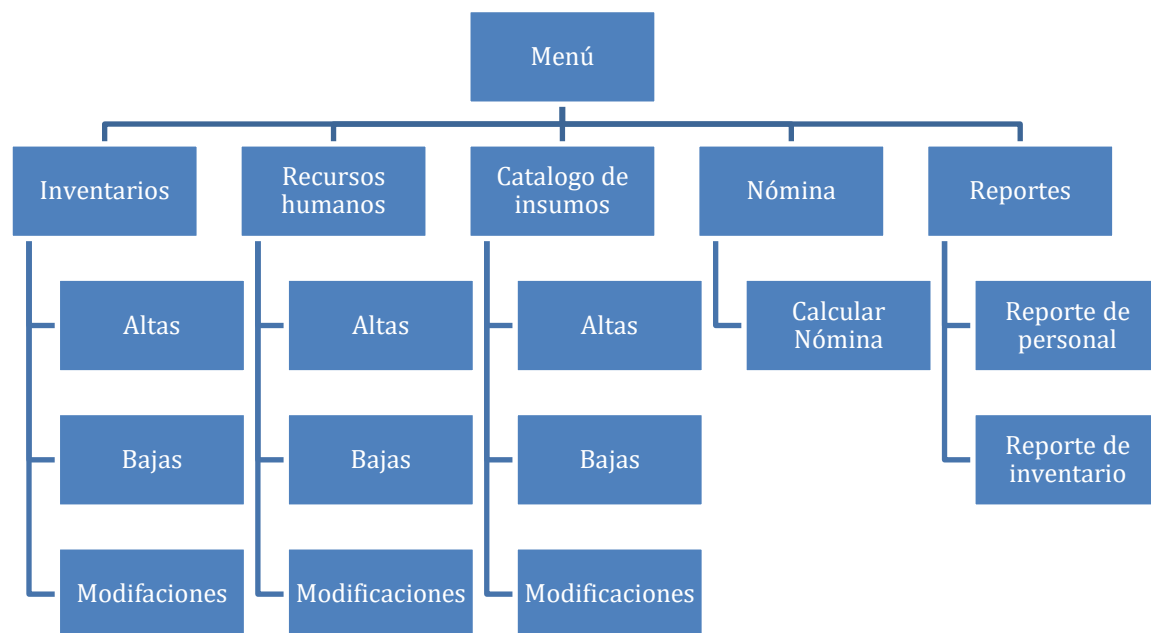
Se utilizará un formulario el cual seleccionara al empleado, pedirá las horas trabajadas en este periodo nominal, calculara los impuestos e imprimirá su recibo.

Reportes

Este módulo se conectará con los otros módulos para poder generar reportes de inventarios, nóminas y de personal. Con lo cual contara con tres formularios:

- Reporte inventario
- Reporte nomina
- Reporte personal

Mapa de navegación



Módulos Inventario y Recursos Humanos

Modulo inventario

Contenedor Alta

- Etiqueta: Código
- Campo texto: Código
- Etiqueta: Artículo
- Campo texto: Artículo
- Etiqueta: Sucursal
- Lista desplegable: Sucursal
- Etiqueta: Tipo de insumo
- Lista desplegable: Tipo de insumo
- Etiqueta: Existencia
- Spinner: Existencia
- Etiqueta: Marca
- Campo texto: Marca
- Botón: Guardar
- Botón: Limpiar
- Botón: Salir

Contenedor Baja

- Etiqueta: Código
- Campo texto: Código
- Etiqueta: Artículo
- Campo texto: Artículo
- Etiqueta: Sucursal
- Lista desplegable: Sucursal

- Etiqueta: Tipo de insumo
- Lista desplegable: Tipo de insumo
- Etiqueta: Existencia
- Spinner: Existencia
- Etiqueta: Marca
- Campo texto: Marca
- Botón: Nueva Búsqueda
- Botón: Eliminar
- Botón: Salir

Contenedor Modificaciones

- Etiqueta: Código
- Campo texto: Código
- Etiqueta: Artículo
- Campo texto: Artículo
- Etiqueta: Sucursal
- Lista desplegable: Sucursal
- Etiqueta: Tipo de insumo
- Lista desplegable: Tipo de insumo
- Etiqueta: Existencia
- Spinner: Existencia
- Etiqueta: Marca
- Campo texto: Marca
- Botón: Búsqueda
- Botón: Guardar
- Botón: Salir

Modulo Recursos Humanos

Contenedor Alta

- Etiqueta: Numero de empleado
- Campo texto: Numero de empleado
- Etiqueta: Nombre
- Campo texto: Nombre
- Etiqueta: Apellido
- Campo texto: Apellido
- Etiqueta: Fecha de nacimiento
- Campo texto: Fecha de nacimiento
- Etiqueta: CURP
- Campo texto: CURP
- Etiqueta: RFC
- Campo texto: RFC
- Etiqueta: Sueldo
- Spinner: Sueldo
- Etiqueta: Puesto
- Lista desplegable: Puesto
- Etiqueta: Sucursal

- Lista desplegable: Sucursal
- Etiqueta: Fecha de ingreso
- Campo texto: Fecha de ingreso
- Botón: Guardar
- Botón: Limpiar
- Botón: Salir

Contenedor Baja

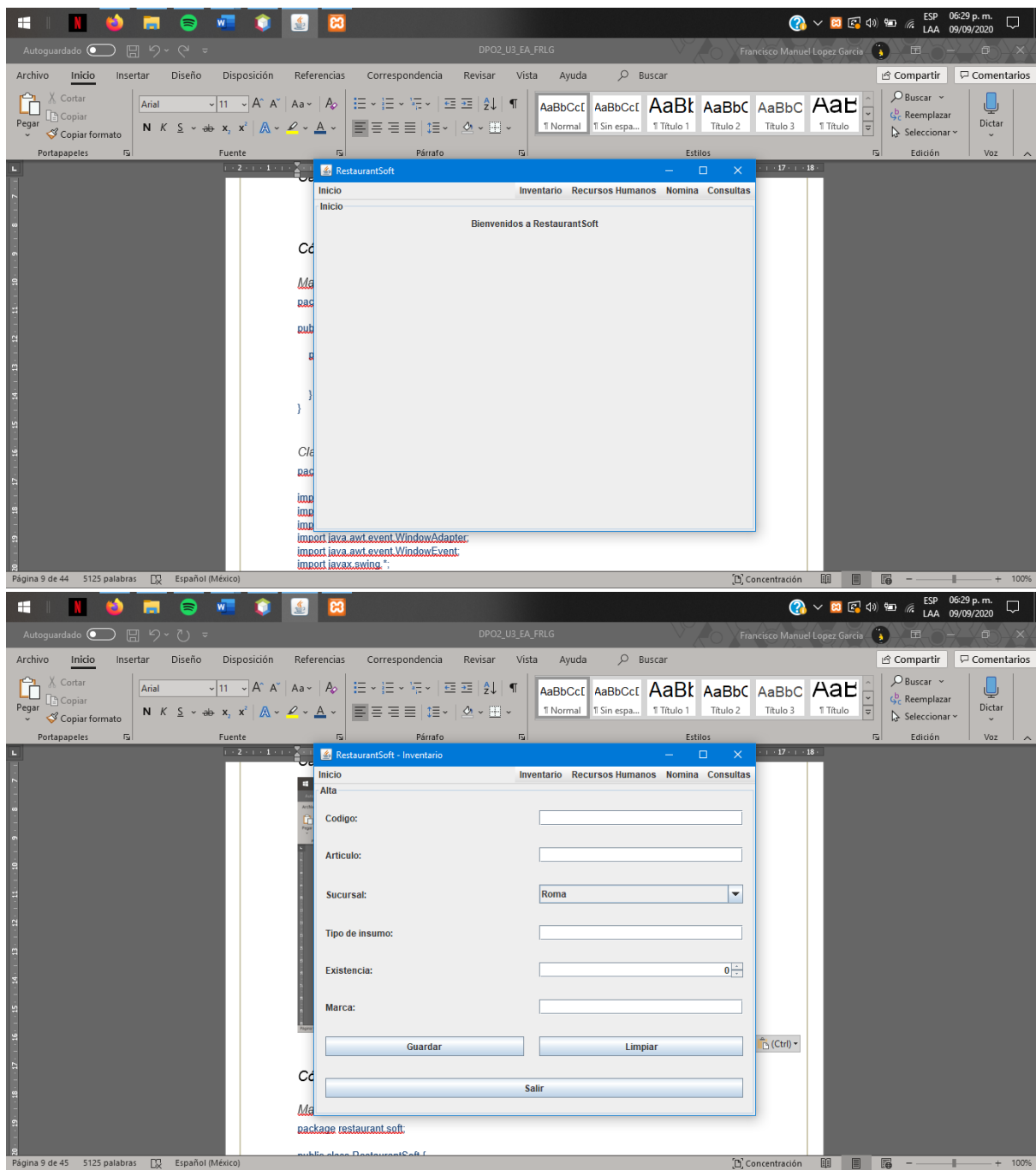
- Etiqueta: Numero de empleado
- Campo texto: Numero de empleado
- Etiqueta: Nombre
- Campo texto: Nombre
- Etiqueta: Apellido
- Campo texto: Apellido
- Etiqueta: Fecha de nacimiento
- Campo texto: Fecha de nacimiento
- Etiqueta: CURP
- Campo texto: CURP
- Etiqueta: RFC
- Campo texto: RFC
- Etiqueta: Sueldo
- Spinner: Sueldo
- Etiqueta: Puesto
- Lista desplegable: Puesto
- Etiqueta: Sucursal
- Lista desplegable: Sucursal
- Etiqueta: Fecha de ingreso
- Campo texto: Fecha de ingreso
- Botón: Nueva búsqueda
- Botón: Limpiar
- Botón: Salir

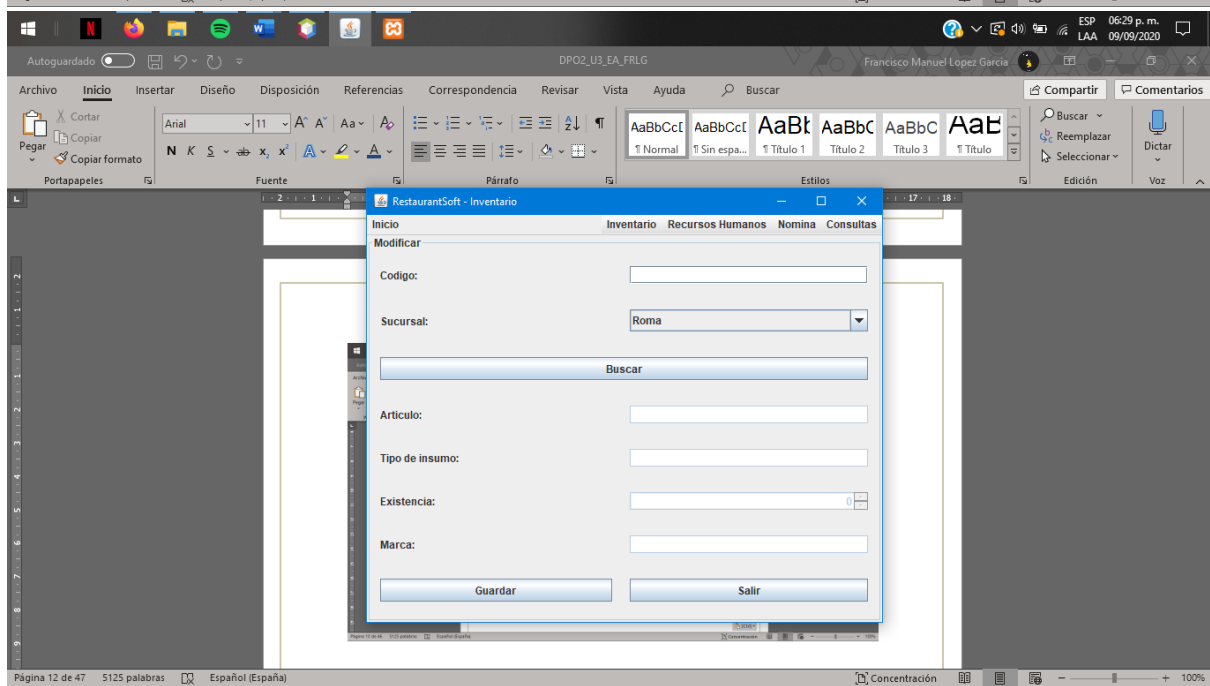
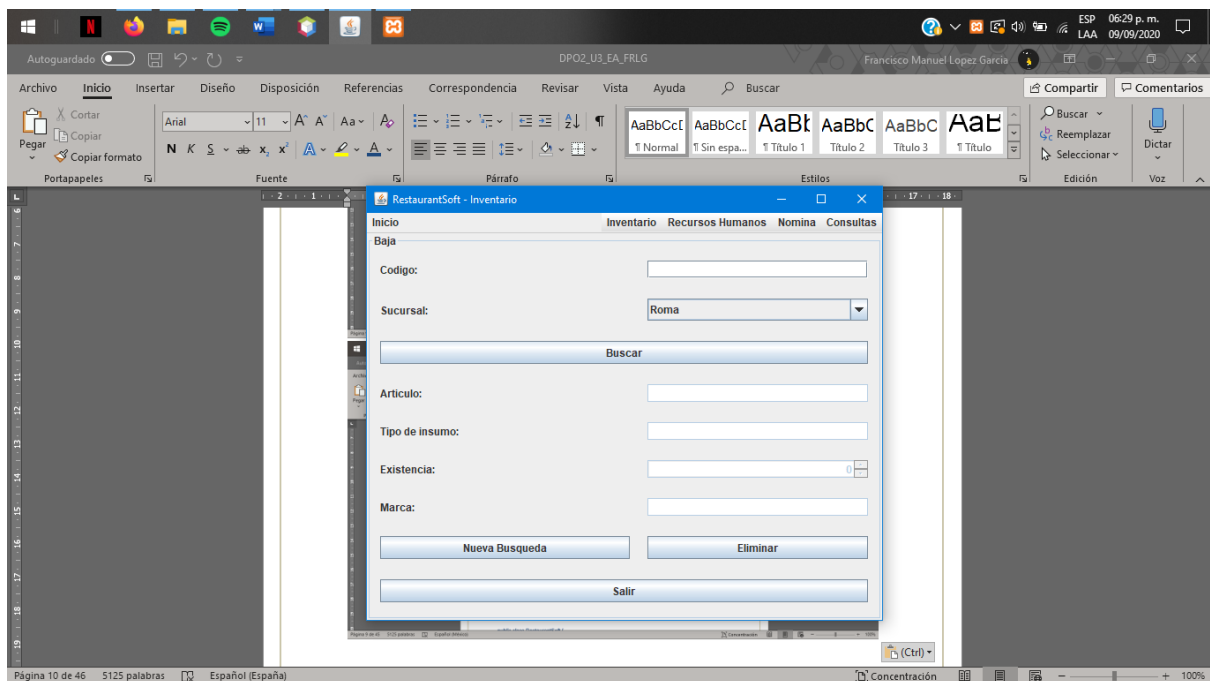
Contenedor Modificaciones

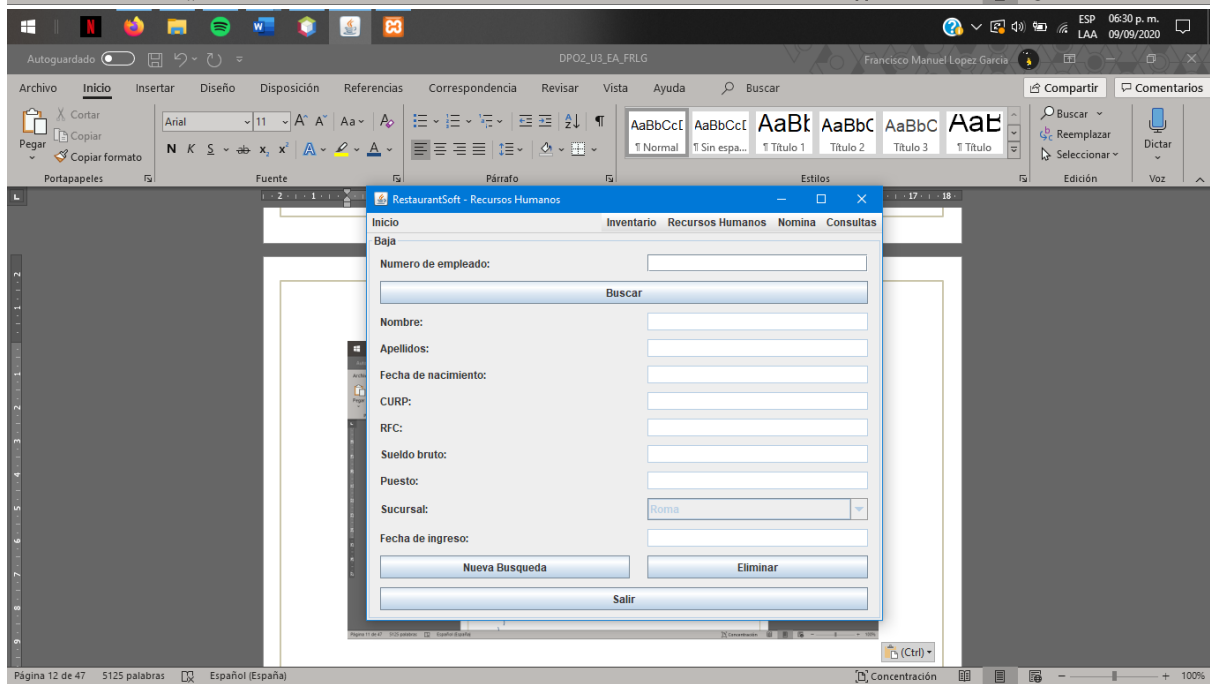
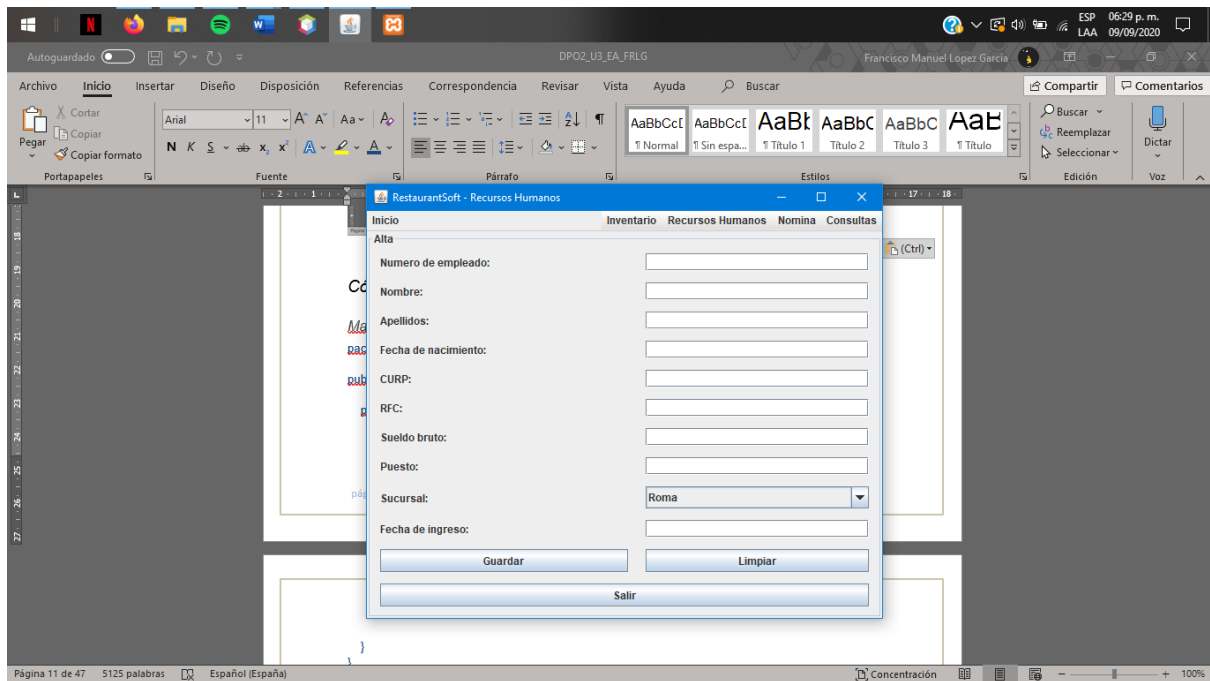
- Etiqueta: Numero de empleado
- Campo texto: Numero de empleado
- Etiqueta: Nombre
- Campo texto: Nombre
- Etiqueta: Apellido
- Campo texto: Apellido
- Etiqueta: Fecha de nacimiento
- Campo texto: Fecha de nacimiento
- Etiqueta: CURP
- Campo texto: CURP
- Etiqueta: RFC
- Campo texto: RFC
- Etiqueta: Sueldo
- Spinner: Sueldo

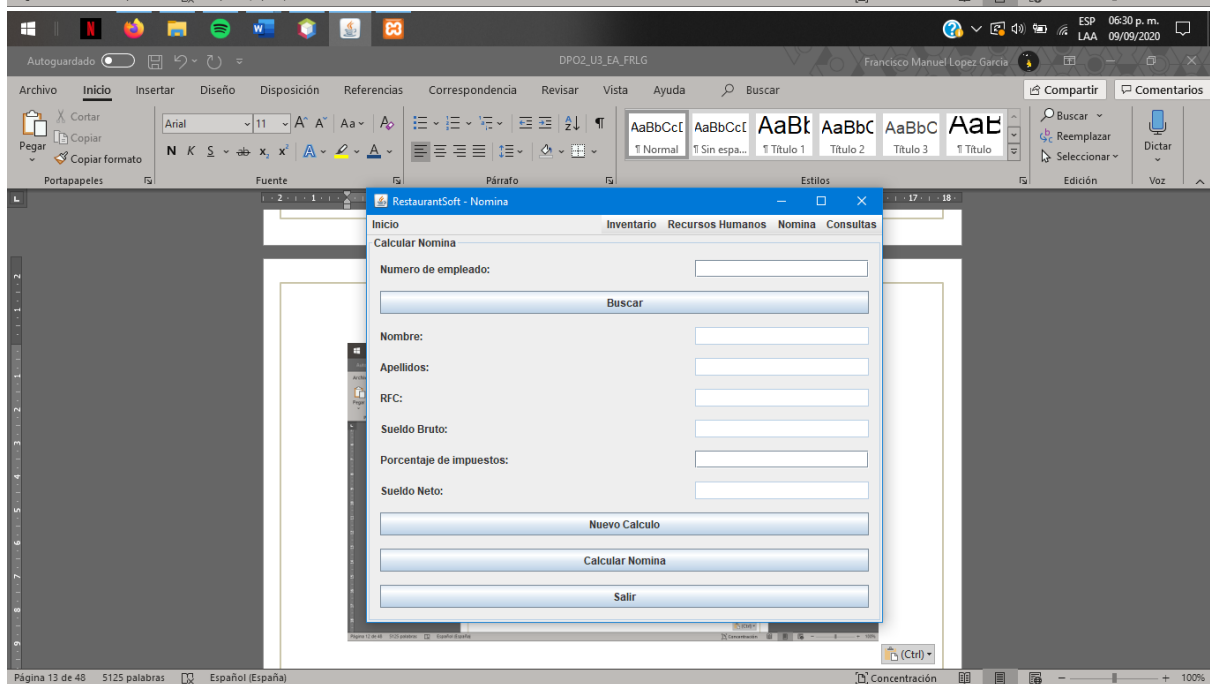
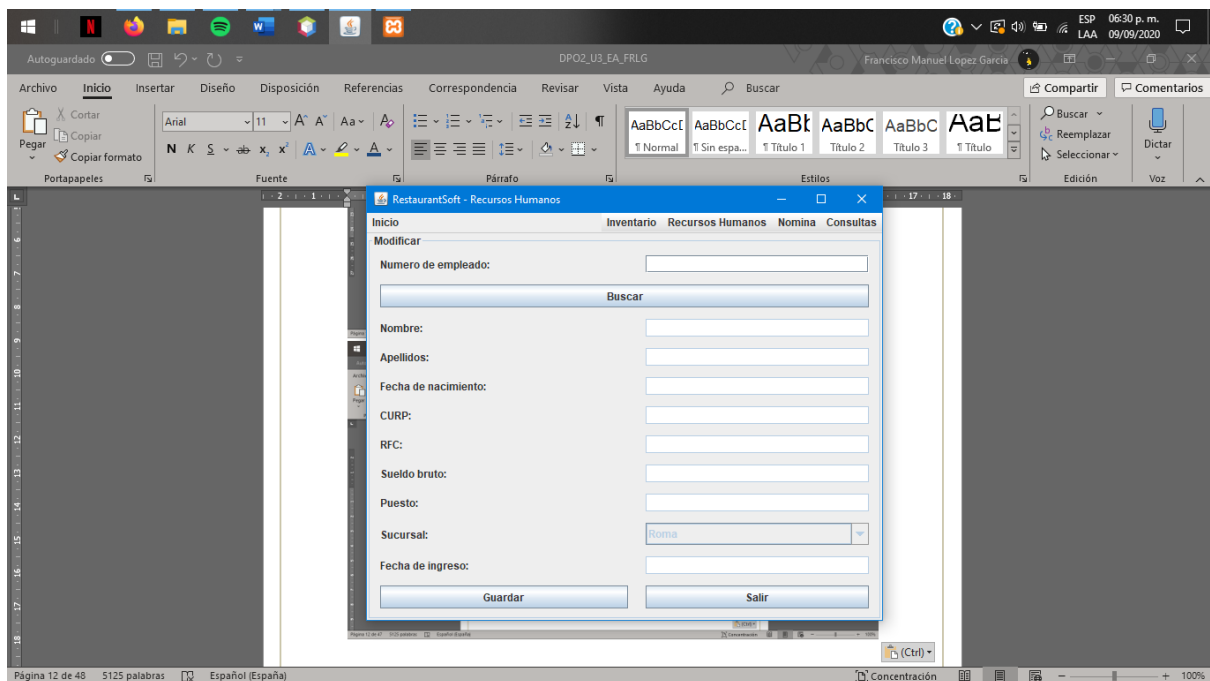
- Etiqueta: Puesto
- Lista desplegable: Puesto
- Etiqueta: Sucursal
- Lista desplegable: Sucursal
- Etiqueta: Fecha de ingreso
- Campo texto: Fecha de ingreso
- Botón: Buscar
- Botón: Guardar
- Botón: Salir

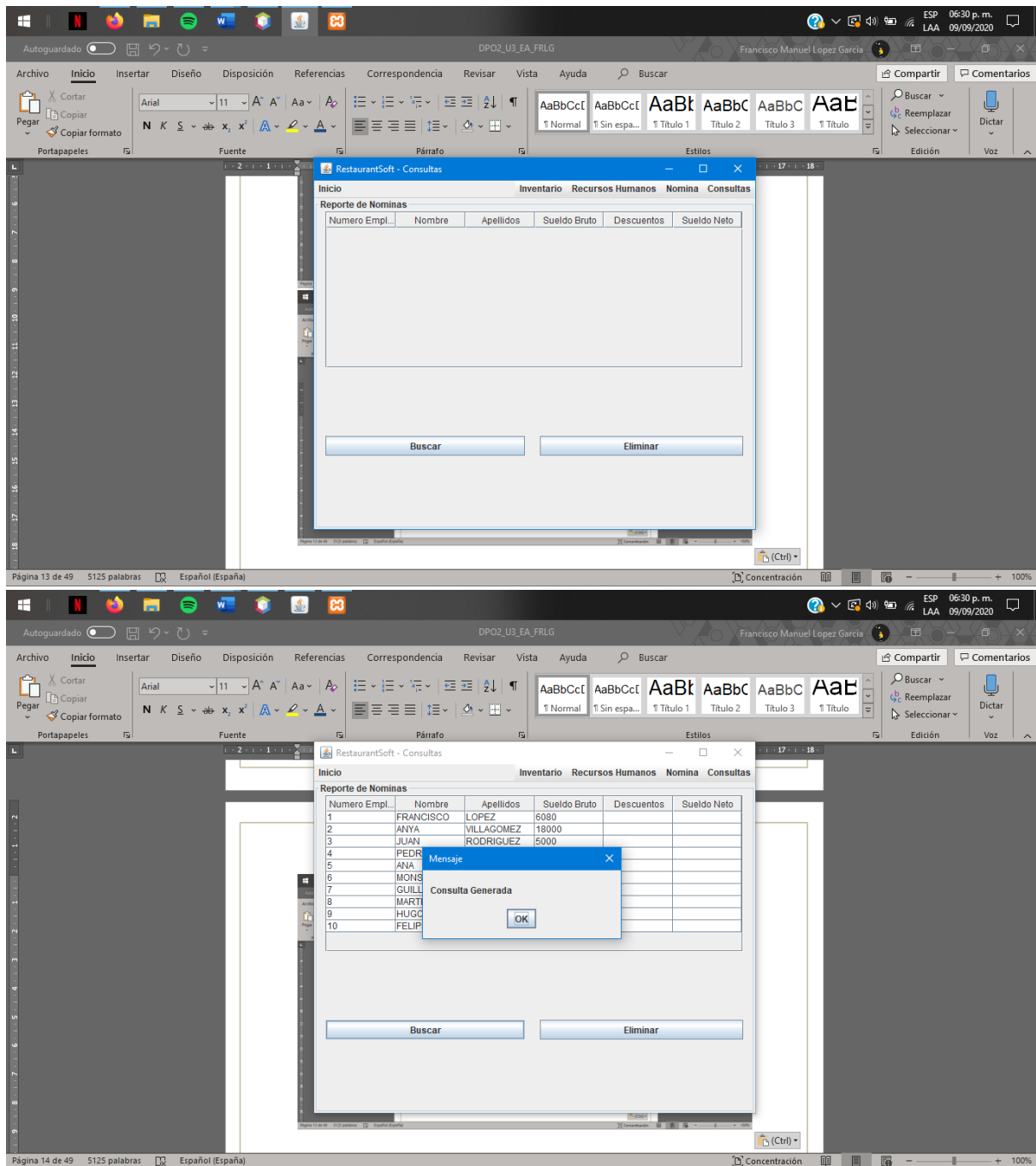
Capturas de pantalla











Código Fuente

Main

package restaurant.soft;

```
public class RestaurantSoft {
```

```
    public static void main(String[] args) {
        framePrincipal fp = new framePrincipal();
        fp.aparienciaTotal();
    }
}
```

```
}  
}
```

Clase framePrincipal

```
package restaurant.soft;
```

```
import java.awt.BorderLayout;  
import java.awt.Image;  
import java.awt.event.ActionEvent;  
import java.awt.event.WindowAdapter;  
import java.awt.event.WindowEvent;  
import javax.swing.*;
```

```
/**
```

```
*
```

```
* @author Manne25
```

```
*/
```

```
public class framePrincipal {  
    static JFrame frame = new JFrame("RestaurantSoft");  
    static Inventario c= new Inventario();  
    static RecursosHumanos rh = new RecursosHumanos();  
    static CatalogoTipodeInsumo ti = new CatalogoTipodeInsumo();  
    static Nomina n = new Nomina();  
    static Reportes r = new Reportes();  
    static Consultas con = new Consultas();  
    JPanel pnlInicio = new javax.swing.JPanel();  
    public void limpiarFrame(){  
        frame.remove(c.pnlAlta);  
        frame.remove(c.pnlBaja);  
        frame.remove(c.pnlMod);  
        frame.remove(rh.pnlAlta);  
        frame.remove(rh.pnlBaja);  
        frame.remove(rh.pnlMod);  
        frame.remove(ti.pnlAlta);  
        frame.remove(ti.pnlBaja);  
        frame.remove(ti.pnlMod);  
        frame.remove(n.pnlCalc);  
        frame.remove(r.pnlRPer);  
        frame.remove(r.pnlRInv);  
        frame.remove(pnlInicio);  
        frame.remove(con.pnlNominas);  
    }  
  
    public void aparienciaTotal(){  
        pnlInicio.setBorder(javax.swing.BorderFactory.createTitledBorder("Inicio"));  
        JLabel ini = new javax.swing.JLabel();  
        ini.setText("Bienvenidos a RestaurantSoft");  
        pnlInicio.add(ini);  
        frame.setLocation(400,200);  
        frame.setVisible(true);  
        frame.setSize(600,500);  
    }  
}
```

```

frame.addWindowListener(new WindowAdapter() {
    @Override
    public void windowClosing(WindowEvent e){
        System.exit(0);
    }
});
JMenuBar menu = new javax.swing.JMenuBar();
JMenuItem inicio = new javax.swing.JMenuItem("Inicio");
inicio.addActionListener((ActionEvent e) -> {
    if(e.getSource()==inicio){
        frame.setTitle("RestaurantSoft");
        limpiarFrame();
        frame.add(pnlInicio,BorderLayout.CENTER);
        frame.repaint();
        frame.validate();
    }
});
JMenu inventario = new javax.swing.JMenu("Inventario");
JMenuItem alta = new javax.swing.JMenuItem("Alta");
alta.addActionListener((ActionEvent e) -> {
    if(e.getSource()==alta){
        frame.setTitle("RestaurantSoft - Inventario");
        c.aparienciaAlta();
        limpiarFrame();
        frame.add(c.pnlAlta,BorderLayout.CENTER);
        frame.repaint();
        frame.validate();
    }
});
JMenuItem baja = new javax.swing.JMenuItem("Baja");
baja.addActionListener((ActionEvent e) -> {
    if(e.getSource()==baja){
        frame.setTitle("RestaurantSoft - Inventario");
        c.aparienciaBaja();
        limpiarFrame();
        frame.add(c.pnlBaja,BorderLayout.CENTER);
        frame.repaint();
        frame.validate();
    }
});
JMenuItem modificar = new javax.swing.JMenuItem("Modificar");
modificar.addActionListener((ActionEvent e) -> {
    if(e.getSource()==modificar){
        frame.setTitle("RestaurantSoft - Inventario");
        c.aparienciaModificar();
        limpiarFrame();
        frame.add(c.pnlMod,BorderLayout.CENTER);
        frame.repaint();
        frame.validate();
    }
});

```



```

JMenu recursosHumanos = new javax.swing.JMenu("Recursos Humanos");
JMenuItem altaR = new javax.swing.JMenuItem("Alta");
altaR.addActionListener((ActionEvent e) -> {
    if(e.getSource()==altaR){
        frame.setTitle("RestaurantSoft - Recursos Humanos");
        rh.aparienciaAlta();
        limpiarFrame();
        frame.add(rh.pnlAlta,BorderLayout.CENTER);
        frame.repaint();
        frame.validate();
    }
});
JMenuItem bajaR = new javax.swing.JMenuItem("Baja");
bajaR.addActionListener((ActionEvent e) -> {
    if(e.getSource()==bajaR){
        frame.setTitle("RestaurantSoft - Recursos Humanos");
        rh.aparienciaBaja();
        limpiarFrame();
        frame.add(rh.pnlBaja,BorderLayout.CENTER);
        frame.repaint();
        frame.validate();
    }
});
JMenuItem modificarR = new javax.swing.JMenuItem("Modificar");
modificarR.addActionListener((ActionEvent e) -> {
    if(e.getSource()==modificarR){
        frame.setTitle("RestaurantSoft - Recursos Humanos");
        rh.aparienciaModificar();
        limpiarFrame();
        frame.add(rh.pnlMod,BorderLayout.CENTER);
        frame.repaint();
        frame.validate();
    }
});

JMenu catalogoTipoDeInsumo = new javax.swing.JMenu("Catalogo tipo de insumo");
JMenuItem altaTI = new javax.swing.JMenuItem("Alta");
altaTI.addActionListener((ActionEvent e) -> {
    if(e.getSource()==altaTI){
        frame.setTitle("RestaurantSoft - Catalogo Tipo de Insumo");
        ti.aparienciaAlta();
        limpiarFrame();
        frame.add(ti.pnlAlta,BorderLayout.CENTER);
        frame.repaint();
        frame.validate();
    }
});
JMenuItem bajaTI = new javax.swing.JMenuItem("Baja");
bajaTI.addActionListener((ActionEvent e) -> {
    if(e.getSource()==bajaTI){
        frame.setTitle("RestaurantSoft - Catalogo Tipo de Insumo");
        ti.aparienciaBaja();
    }
});

```

```

        limpiarFrame();
        frame.add(ti.pnlBaja,BorderLayout.CENTER);
        frame.repaint();
        frame.validate();
    }
});
JMenuItem modificarTI = new javax.swing.JMenuItem("Modificar");
modificarTI.addActionListener((ActionEvent e) -> {
    if(e.getSource()==modificarTI){
        frame.setTitle("RestaurantSoft - Catalogo Tipo de Insumo");
        ti.aparienciaModificar();
        limpiarFrame();
        frame.add(ti.pnlMod,BorderLayout.CENTER);
        frame.repaint();
        frame.validate();
    }
});

JMenu nomina = new javax.swing.JMenu("Nomina");
JMenuItem calcN = new javax.swing.JMenuItem("Calculo Nomina");
calcN.addActionListener((ActionEvent e) -> {
    if(e.getSource()==calcN){
        frame.setTitle("RestaurantSoft - Nomina");
        n.aparienciaCalc();
        limpiarFrame();
        frame.add(n.pnlCalc,BorderLayout.CENTER);
        frame.repaint();
        frame.validate();
    }
});
JMenu reportes = new javax.swing.JMenu("Reportes");
JMenuItem rPer = new javax.swing.JMenuItem("Reporte de Personal");
rPer.addActionListener((ActionEvent e) -> {
    if(e.getSource()==rPer){
        frame.setTitle("RestaurantSoft - Reportes");
        r.aparienciaRPer();
        limpiarFrame();
        frame.add(r.pnlRPer,BorderLayout.CENTER);
        frame.repaint();
        frame.validate();
    }
});
JMenuItem rInv = new javax.swing.JMenuItem("Reporte de Inventario");
rInv.addActionListener((ActionEvent e) -> {
    if(e.getSource()==rInv){
        frame.setTitle("RestaurantSoft - Reportes");
        r.aparienciaRInv();
        limpiarFrame();
        frame.add(r.pnlRInv,BorderLayout.CENTER);
        frame.repaint();
        frame.validate();
    }
});

```

```

});
JMenu consultas = new javax.swing.JMenu("Consultas");
JMenuItem nominas = new javax.swing.JMenuItem("Nominas");
nominas.addActionListener((ActionEvent e) -> {
    if(e.getSource()==nominas){
        frame.setTitle("RestaurantSoft - Consultas");
        con.aparienciaNominas();
        limpiarFrame();
        frame.add(con.pnlNominas,BorderLayout.CENTER);
        frame.repaint();
        frame.validate();
    }
});
menu.add(inicio);
menu.add(inventario);
inventario.add(alta);
inventario.add(baja);
inventario.add(modificar);
menu.add(recursosHumanos);
recursosHumanos.add(altaR);
recursosHumanos.add(bajaR);
recursosHumanos.add(modificarR);
menu.add(catalogoTipoDeInsumo);
catalogoTipoDeInsumo.add(altaTI);
catalogoTipoDeInsumo.add(bajaTI);
catalogoTipoDeInsumo.add(modificarTI);
menu.add(nomina);
nomina.add(calcN);
//menu.add(reportes);
//reportes.add(rPer);
//reportes.add(rInv);
menu.add(consultas);
consultas.add(nominas);
frame.add(menu, BorderLayout.PAGE_START);
frame.add(con.pnlInicio,BorderLayout.CENTER);
frame.repaint();
frame.validate();
}
}

```

Clase Inventario

```

package restaurant.soft;

/**
 *
 * @author Manne25
 */
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.Insets;
import java.awt.event.*;

```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.*.*;

public class Inventario {
    Conexion BD = new Conexion();

    //Componentes graficos Alta
    JPanel pnlAlta = new javax.swing.JPanel();
    JLabel lblCodigo = new javax.swing.JLabel();
    JTextField txtCodigo = new javax.swing.JTextField();
    JLabel lblArticulo = new javax.swing.JLabel();
    JTextField txtArticulo = new javax.swing.JTextField();
    JLabel lblSucursal = new javax.swing.JLabel();
    JComboBox cmbSucursal = new javax.swing.JComboBox();
    JLabel lblTipo = new javax.swing.JLabel();
    JTextField txtTipo = new javax.swing.JTextField();
    JLabel lblExistencia = new javax.swing.JLabel();
    JSpinner spnExistencia = new javax.swing.JSpinner();
    JLabel lblMarca = new javax.swing.JLabel();
    JTextField txtMarca = new javax.swing.JTextField();
    JButton btnGuardar = new javax.swing.JButton();
    JButton btnLimpiar = new javax.swing.JButton();
    JButton btnSalir = new javax.swing.JButton();
    //Componentes graficos Baja
    JPanel pnlBaja = new javax.swing.JPanel();
    JLabel lblCodigoB = new javax.swing.JLabel();
    JTextField txtCodigoB = new javax.swing.JTextField();
    JLabel lblSucursalB = new javax.swing.JLabel();
    JComboBox cmbSucursalB = new javax.swing.JComboBox();
    JLabel lblArticuloB = new javax.swing.JLabel();
    JTextField txtArticuloB = new javax.swing.JTextField();
    JLabel lblTipoB = new javax.swing.JLabel();
    JTextField txtTipoB = new javax.swing.JTextField();
    JLabel lblExistenciaB = new javax.swing.JLabel();
    JSpinner spnExistenciaB = new javax.swing.JSpinner();
    JLabel lblMarcaB = new javax.swing.JLabel();
    JTextField txtMarcaB = new javax.swing.JTextField();
    JButton btnBuscar = new javax.swing.JButton();
    JButton btnNuevaBusqueda = new javax.swing.JButton();
    JButton btnEliminar = new javax.swing.JButton();
    JButton btnSalirB = new javax.swing.JButton();
    //Componentes graficos Modificar
    JPanel pnlMod = new javax.swing.JPanel();
    JLabel lblCodigoM = new javax.swing.JLabel();
    JTextField txtCodigoM = new javax.swing.JTextField();
    JLabel lblSucursalM = new javax.swing.JLabel();
    JComboBox cmbSucursalM = new javax.swing.JComboBox();

```

```

JLabel lblArticuloM = new javax.swing.JLabel();
JTextField txtArticuloM = new javax.swing.JTextField();
JLabel lblTipoM = new javax.swing.JLabel();
JTextField txtTipoM = new javax.swing.JTextField();
JLabel lblExistenciaM = new javax.swing.JLabel();
JSpinner spnExistenciaM = new javax.swing.JSpinner();
JLabel lblMarcaM = new javax.swing.JLabel();
JTextField txtMarcaM = new javax.swing.JTextField();
JButton btnBuscarM = new javax.swing.JButton();
JButton btnGuardarM = new javax.swing.JButton();
JButton btnSalirM = new javax.swing.JButton();

public void aparienciaAlta(){
    pnlAlta.setBorder(javax.swing.BorderFactory.createTitledBorder("Alta"));
    lblCodigo.setText("Codigo:");
    lblArticulo.setText("Articulo:");
    lblSucursal.setText("Sucursal:");
    lblTipo.setText("Tipo de insumo:");
    lblExistencia.setText("Existencia:");
    lblMarca.setText("Marca:");
    cmbSucursal.setModel( new javax.swing.DefaultComboBoxModel(new
String[]{"Roma", "Insurgentes", "Condesa", "Centro"}));
    spnExistencia.setModel(new javax.swing.SpinnerNumberModel(0,0,9999,1));
    btnGuardar.setText("Guardar");
    btnGuardar.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){
            guardar();
        }
    });
    btnLimpiar.setText("Limpiar");
    btnLimpiar.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){
            limpiar();
        }
    });
    btnSalir.setText("Salir");
    btnSalir.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){
            System.exit(0);
        }
    });
    GridBagLayout gbl = new GridBagLayout();
    GridBagConstraints gbc = new GridBagConstraints();
    pnlAlta.setLayout(gbl);
    gbc.weightx = 1;
    gbc.weighty = 1;
    gbc.fill = GridBagConstraints.HORIZONTAL;
    gbc.anchor = GridBagConstraints.CENTER;
    gbc.gridwidth = 1;

```

```

gbc.insets = new Insets(0,10,0,10);
pnlAlta.add(lblCodigo, gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlAlta.add(txtCodigo, gbc);
gbc.gridwidth = 1;
pnlAlta.add(lblArticulo,gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlAlta.add(txtArticulo,gbc);
gbc.gridwidth = 1;
pnlAlta.add(lblSucursal, gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlAlta.add(cmbSucursal, gbc);
gbc.gridwidth = 1;
pnlAlta.add(lblTipo,gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlAlta.add(txtTipo,gbc);
gbc.gridwidth = 1;
pnlAlta.add(lblExistencia,gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlAlta.add(spнExistencia,gbc);
gbc.gridwidth = 1;
pnlAlta.add(lblMarca,gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlAlta.add(txtMarca,gbc);
gbc.gridwidth = 1;
pnlAlta.add(btnGuardar,gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlAlta.add(btnLimpiar,gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlAlta.add(btnSalir,gbc);

}

public void guardar(){
    if("").equals(txtCodigo.getText()))
    {
        JOptionPane.showMessageDialog(null,"Alguno de los campos esta vacio ","Error",
JOptionPane.ERROR_MESSAGE);
    }
    Artículo art = new Artículo();
    art.codigo = txtCodigo.getText();
    art.articulo = txtArticulo.getText();
    art.sucursal = cmbSucursal.getSelectedItem().toString();
    art.idTipoInsumo = txtTipo.getText();
    art.existencia = spнExistencia.getValue().toString();
    art.marca = txtMarca.getText();

    BD.insertarInventario(art.codigo,art.articulo,art.sucursal,art.idTipoInsumo,art.existencia,art.marca);
    limpiar();

}

public void limpiar(){

```

```

txtCodigo.setText("");
txtArticulo.setText("");
txtMarca.setText("");
txtTipo.setText("");
spnExistencia.setValue(0);
txtCodigoB.setText("");
txtArticuloB.setText("");
txtMarcaB.setText("");
txtTipoB.setText("");
spnExistenciaB.setValue(0);
JOptionPane.showMessageDialog( null,"Limpiar con exito", "Mensaje",
JOptionPane.PLAIN_MESSAGE);
}
public void aparienciaBaja(){
    pnlBaja.setBorder(javax.swing.BorderFactory.createTitledBorder("Baja"));
    lblCodigoB.setText("Codigo:");
    lblArticuloB.setText("Articulo:");
    lblSucursalB.setText("Sucursal:");
    lblTipoB.setText("Tipo de insumo:");
    lblExistenciaB.setText("Existencia:");
    lblMarcaB.setText("Marca:");
    cmbSucursalB.setModel( new javax.swing.DefaultComboBoxModel(new
String[]{"Roma","Insurgentes","Condesa","Centro"}));
    spnExistenciaB.setModel(new javax.swing.SpinnerNumberModel(0,0,9999,1));
    txtArticuloB.setEnabled(false);
    txtMarcaB.setEnabled(false);
    txtTipoB.setEnabled(false);
    spnExistenciaB.setEnabled(false);
    btnBuscar.setText("Buscar");
    btnBuscar.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){
            Buscar();
        }
    });
    btnNuevaBusqueda.setText("Nueva Búsqueda");
    btnNuevaBusqueda.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){
            limpiar();
        }
    });
    btnEliminar.setText("Eliminar");
    btnEliminar.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){
            Eliminar();
        }
    });
    btnSalirB.setText("Salir");
    btnSalirB.addMouseListener(new MouseAdapter(){
        @Override

```

```

        public void mouseClicked(MouseEvent e){
            System.exit(0);
        }
    });
    GridBagLayout gbl = new GridBagLayout();
    GridBagConstraints gbc = new GridBagConstraints();
    pnlBaja.setLayout(gbl);
    gbc.weightx = 1;
    gbc.weighty = 1;
    gbc.fill = GridBagConstraints.HORIZONTAL;
    gbc.anchor = GridBagConstraints.CENTER;
    gbc.gridwidth = 1;
    gbc.insets = new Insets(0,10,0,10);
    pnlBaja.add(lblCodigoB, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlBaja.add(txtCodigoB, gbc);
    gbc.gridwidth = 1;
    pnlBaja.add(lblSucursalB, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlBaja.add(cmbSucursalB, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlBaja.add(btnBuscar, gbc);
    gbc.gridwidth = 1;
    pnlBaja.add(lblArticuloB, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlBaja.add(txtArticuloB, gbc);
    gbc.gridwidth = 1;
    pnlBaja.add(lblTipoB, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlBaja.add(txtTipoB, gbc);
    gbc.gridwidth = 1;
    pnlBaja.add(lblExistenciaB, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlBaja.add(spnrExistenciaB, gbc);
    gbc.gridwidth = 1;
    pnlBaja.add(lblMarcaB, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlBaja.add(txtMarcaB, gbc);
    gbc.gridwidth = 1;
    pnlBaja.add(btnNuevaBusqueda, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlBaja.add(btnEliminar, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlBaja.add(btnSalirB, gbc);
}

public void Buscar(){
    Articulo art = new Articulo();

    try{
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection conexion =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/restaurant_soft", "manne", "Fco-1402");
    }
}

```



```

        Statement st = conexion.createStatement();
        String consulta = "SELECT * FROM restaurant_soft.articulo WHERE
codigo='"+txtCodigoB.getText()+" AND sucursal='"+cmbSucursalB.getSelectedItem().toString()+"';";
        ResultSet rs=null;
        rs = st.executeQuery(consulta);
        if(rs.next()){
            art.articulo=rs.getString("articulo");
            art.idTipoInsumo=rs.getString("idTipoInsumo");
            art.marca=rs.getString("marca");
            art.existencia=rs.getString("existencia");
            txtArticuloB.setText(art.articulo);
            txtTipoB.setText(art.idTipoInsumo);
            txtMarcaB.setText(art.marca);
            spnExistenciaB.setValue(Integer.parseInt(art.existencia));
        }

        rs.close();
        st.close();
        conexion.close();
    }
    catch(ClassNotFoundException | SQLException e){
        JOptionPane.showMessageDialog(null,"Error al conectar con la Base de Datos","Error",
JOptionPane.ERROR_MESSAGE);
    }

    JOptionPane.showMessageDialog( null,"Insumo encontrado", "Mensaje",
JOptionPane.PLAIN_MESSAGE);
}

public void BuscarM(){
    Articulo art = new Articulo();

    try{
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection conexion =
DriverManager.getConnection("jdbc:mysql://localhost:3306/restaurant_soft","manne","Fco-1402");
        Statement st = conexion.createStatement();
        String consulta = "SELECT * FROM restaurant_soft.articulo WHERE
codigo='"+txtCodigoM.getText()+" AND sucursal='"+cmbSucursalM.getSelectedItem().toString()+"';";
        ResultSet rs=null;
        rs = st.executeQuery(consulta);
        if(rs.next()){
            art.articulo=rs.getString("articulo");
            art.idTipoInsumo=rs.getString("idTipoInsumo");
            art.marca=rs.getString("marca");
            art.existencia=rs.getString("existencia");
            txtArticuloM.setText(art.articulo);
            txtTipoM.setText(art.idTipoInsumo);
            txtMarcaM.setText(art.marca);
            spnExistenciaM.setValue(Integer.parseInt(art.existencia));
            txtArticuloM.setEnabled(true);
            txtMarcaM.setEnabled(true);
            txtTipoM.setEnabled(true);

```

```

        spnExistenciaM.setEnabled(true);
        txtCodigoM.setEnabled(false);
        cmbSucursalM.setEnabled(false);
    }
    rs.close();
    st.close();
    conexion.close();
}
catch(ClassNotFoundException | SQLException e){
    JOptionPane.showMessageDialog(null,"Error al conectar con la Base de Datos","Error",
JOptionPane.ERROR_MESSAGE);
}

JOptionPane.showMessageDialog( null,"Insumo encontrado", "Mensaje",
JOptionPane.PLAIN_MESSAGE);
}
public void Eliminar(){
    if("").equals(txtCodigoB.getText())
    {
        JOptionPane.showMessageDialog(null,"Alguno de los campos esta vacio ","Error",
JOptionPane.ERROR_MESSAGE);
    }
    Articulo art = new Articulo();
    art.codigo = txtCodigoB.getText();
    art.sucursal = cmbSucursalB.getSelectedItem().toString();
    BD.eliminarInventario(art.codigo,art.sucursal);
    limpiar();
}
public void aparienciaModificar(){
    pnlMod.setBorder(javax.swing.BorderFactory.createTitledBorder("Modificar"));
    lblCodigoM.setText("Codigo:");
    lblArticuloM.setText("Articulo:");
    lblSucursalM.setText("Sucursal:");
    lblTipoM.setText("Tipo de insumo:");
    lblExistenciaM.setText("Existencia:");
    lblMarcaM.setText("Marca:");
    cmbSucursalM.setModel( new javax.swing.DefaultComboBoxModel(new
String[]{"Roma","Insurgentes","Condesa","Centro"}));
    spnExistenciaM.setModel(new javax.swing.SpinnerNumberModel(0,0,9999,1));
    txtArticuloM.setEnabled(false);
    txtMarcaM.setEnabled(false);
    txtTipoM.setEnabled(false);
    spnExistenciaM.setEnabled(false);
    btnBuscarM.setText("Buscar");
    btnBuscarM.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){
            BuscarM();
        }
    });
    btnGuardarM.setText("Guardar");
    btnGuardarM.addMouseListener(new MouseAdapter(){

```

```

        @Override
        public void mouseClicked(MouseEvent e){
            actualizar();
        }
    });

    btnSalirM.setText("Salir");
    btnSalirM.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){
            System.exit(0);
        }
    });

    GridBagLayout gbl = new GridBagLayout();
    GridBagConstraints gbc = new GridBagConstraints();
    pnlMod.setLayout(gbl);
    gbc.weightx = 1;
    gbc.weighty = 1;
    gbc.fill = GridBagConstraints.HORIZONTAL;
    gbc.anchor = GridBagConstraints.CENTER;
    gbc.gridwidth = 1;
    gbc.insets = new Insets(0,10,0,10);
    pnlMod.add(lblCodigoM, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlMod.add(txtCodigoM, gbc);
    gbc.gridwidth = 1;
    pnlMod.add(lblSucursalM, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlMod.add(cmbSucursalM, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlMod.add(btnBuscarM, gbc);
    gbc.gridwidth = 1;
    pnlMod.add(lblArticuloM, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlMod.add(txtArticuloM, gbc);
    gbc.gridwidth = 1;
    pnlMod.add(lblTipoM, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlMod.add(txtTipoM, gbc);
    gbc.gridwidth = 1;
    pnlMod.add(lblExistenciaM, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlMod.add(spnrExistenciaM, gbc);
    gbc.gridwidth = 1;
    pnlMod.add(lblMarcaM, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlMod.add(txtMarcaM, gbc);
    gbc.gridwidth = 1;
    pnlMod.add(btnGuardarM, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlMod.add(btnSalirM, gbc);
}

```

```

public void actualizar(){
    if("").equals(txtCodigoM.getText())
    {
        JOptionPane.showMessageDialog(null,"Alguno de los campos esta vacio ","Error",
JOptionPane.ERROR_MESSAGE);
    }
    Artículo art = new Artículo();
    art.codigo = txtCodigoM.getText();
    art.articulo = txtArticuloM.getText();
    art.sucursal = cmbSucursalM.getSelectedItem().toString();
    art.idTipoInsumo = txtTipoM.getText();
    art.existencia = spnExistenciaM.getValue().toString();
    art.marca = txtMarcaM.getText();

    BD.modificarInventario(art.codigo,art.articulo,art.sucursal,art.idTipoInsumo,art.existencia,art.marca);
    limpiar();
    txtArticuloM.setEnabled(false);
    txtMarcaM.setEnabled(false);
    txtTipoM.setEnabled(false);
    spnExistenciaM.setEnabled(false);
    txtCodigoM.setEnabled(true);
    cmbSucursalM.setEnabled(true);
    limpiar();
}
}

```

Clase Artículo

```
package restaurant.soft;
```

```

/**
 *
 * @author Manne25
 */
public class Artículo {
    String codigo;
    String articulo;
    String sucursal;
    String idTipoInsumo;
    String existencia;
    String marca;

    public void Artículo(String cod,String art, String suc, String id, String exi, String mar){
        codigo=cod;
        articulo=art;
        sucursal=suc;
        idTipoInsumo=id;
        existencia=exi;
        marca=mar;
    }
}

```

```
}
```

Clase RecursosHumanos

```
package restaurant.soft;
```

```
/**
```

```
*
```

```
* @author Manne25
```

```
*/
```

```
import java.awt.GridBagConstraints;
```

```
import java.awt.GridBagLayout;
```

```
import java.awt.Insets;
```

```
import java.awt.event.*;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.ResultSet;
```

```
import java.sql.SQLException;
```

```
import java.sql.Statement;
```

```
import javax.swing.*;
```

```
public class RecursosHumanos {
```

```
    Conexion BD = new Conexion();
```

```
    //Componentes graficos Alta
```

```
    JPanel pnlAlta = new javax.swing.JPanel();
```

```
    JLabel lblNumeroEmpleado = new javax.swing.JLabel();
```

```
    JTextField txtNumeroEmpleado = new javax.swing.JTextField();
```

```
    JLabel lblNombre = new javax.swing.JLabel();
```

```
    JTextField txtNombre = new javax.swing.JTextField();
```

```
    JLabel lblApellidos = new javax.swing.JLabel();
```

```
    JTextField txtApellidos= new javax.swing.JTextField();
```

```
    JLabel lblFechaNac = new javax.swing.JLabel();
```

```
    JTextField txtFechaNac = new javax.swing.JTextField();
```

```
    JLabel lblCURP = new javax.swing.JLabel();
```

```
    JTextField txtCURP = new javax.swing.JTextField();
```

```
    JLabel lblRFC = new javax.swing.JLabel();
```

```
    JTextField txtRFC = new javax.swing.JTextField();
```

```
    JLabel lblSueldo = new javax.swing.JLabel();
```

```
    JTextField txtSueldo= new javax.swing.JTextField();
```

```
    JLabel lblPuesto = new javax.swing.JLabel();
```

```
    JTextField txtPuesto = new javax.swing.JTextField();
```

```
    JLabel lblSucursal = new javax.swing.JLabel();
```

```
    JComboBox cmbSucursal = new javax.swing.JComboBox();
```

```
    JLabel lblFechaIng = new javax.swing.JLabel();
```

```
    JTextField txtFechaIng = new javax.swing.JTextField();
```

```
    JButton btnGuardar = new javax.swing.JButton();
```

```
    JButton btnLimpiar = new javax.swing.JButton();
```

```
    JButton btnSalir = new javax.swing.JButton();
```

```
    //Componentes graficos Baja
```

```
    JPanel pnlBaja = new javax.swing.JPanel();
```

```
    JTextField txtNumeroEmpleadoB = new javax.swing.JTextField();
```

```
    JButton btnBuscar = new javax.swing.JButton();
```

```
    JTextField txtNombreB = new javax.swing.JTextField();
```

```

JTextField txtApellidosB= new javax.swing.JTextField();
JTextField txtFechaNacB = new javax.swing.JTextField();
JTextField txtCURPB = new javax.swing.JTextField();
JTextField txtRFCB = new javax.swing.JTextField();
JTextField txtSueldoB= new javax.swing.JTextField();
JTextField txtPuestoB = new javax.swing.JTextField();
JComboBox cmbSucursalB = new javax.swing.JComboBox();
JTextField txtFechaIngB = new javax.swing.JTextField();
JButton btnNuevaBusqueda = new javax.swing.JButton();
JButton btnEliminar= new javax.swing.JButton();
JButton btnSalirB = new javax.swing.JButton();
//Componentes graficos Modificar
JPanel pnlMod = new javax.swing.JPanel();
JTextField txtNumeroEmpleadoM = new javax.swing.JTextField();
JButton btnBuscarM = new javax.swing.JButton();
JTextField txtNombreM = new javax.swing.JTextField();
JTextField txtApellidosM= new javax.swing.JTextField();
JTextField txtFechaNacM = new javax.swing.JTextField();
JTextField txtCURPM = new javax.swing.JTextField();
JTextField txtRFCM = new javax.swing.JTextField();
JTextField txtSueldoM= new javax.swing.JTextField();
JTextField txtPuestoM = new javax.swing.JTextField();
JComboBox cmbSucursalM = new javax.swing.JComboBox();
JTextField txtFechaIngM = new javax.swing.JTextField();
JButton btnGuardarM = new javax.swing.JButton();
JButton btnSalirM = new javax.swing.JButton();

public void aparienciaAlta(){
    pnlAlta.setBorder(javax.swing.BorderFactory.createTitledBorder("Alta"));
    lblNumeroEmpleado.setText("Numero de empleado:");
    lblNombre.setText("Nombre:");
    lblApellidos.setText("Apellidos:");
    lblFechaNac.setText("Fecha de nacimiento:");
    lblCURP.setText("CURP:");
    lblRFC.setText("RFC:");
    lblSueldo.setText("Sueldo bruto:");
    lblPuesto.setText("Puesto:");
    lblSucursal.setText("Sucursal:");
    lblFechaIng.setText("Fecha de ingreso:");
    cmbSucursal.setModel( new javax.swing.DefaultComboBoxModel(new
String[]{"Roma", "Insurgentes", "Condesa", "Centro"}));
    btnGuardar.setText("Guardar");
    btnGuardar.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){
            guardar();
        }
    });
    btnLimpiar.setText("Limpiar");
    btnLimpiar.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){

```

```

        limpiar();
    }
});
btnSalir.setText("Salir");
btnSalir.addMouseListener(new MouseAdapter(){
    @Override
    public void mouseClicked(MouseEvent e){
        System.exit(0);
    }
});

GridBagLayout gbl = new GridBagLayout();
GridBagConstraints gbc = new GridBagConstraints();
pnlAlta.setLayout(gbl);
gbc.weightx = 1;
gbc.weighty = 1;
gbc.fill = GridBagConstraints.HORIZONTAL;
gbc.anchor = GridBagConstraints.CENTER;
gbc.gridwidth = 1;
gbc.insets = new Insets(0,10,0,10);
pnlAlta.add(lblNumeroEmpleado, gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlAlta.add(txtNumeroEmpleado, gbc);
gbc.gridwidth = 1;
pnlAlta.add(lblNombre,gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlAlta.add(txtNombre,gbc);
gbc.gridwidth = 1;
pnlAlta.add(lblApellidos,gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlAlta.add(txtApellidos,gbc);
gbc.gridwidth = 1;
pnlAlta.add(lblFechaNac,gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlAlta.add(txtFechaNac,gbc);
gbc.gridwidth = 1;
pnlAlta.add(lblCURP,gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlAlta.add(txtCURP,gbc);
gbc.gridwidth = 1;
pnlAlta.add(lblRFC,gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlAlta.add(txtRFC,gbc);
gbc.gridwidth = 1;
pnlAlta.add(lblSueldo,gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlAlta.add(txtSueldo,gbc);
gbc.gridwidth = 1;
pnlAlta.add(lblPuesto,gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlAlta.add(txtPuesto,gbc);
gbc.gridwidth = 1;

```

```

        pnlAlta.add(lblSucursal,gbc);
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        pnlAlta.add(cmbSucursal,gbc);
        gbc.gridwidth = 1;
        pnlAlta.add(lblFechaIng,gbc);
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        pnlAlta.add(txtFechaIng,gbc);
        gbc.gridwidth = 1;
        pnlAlta.add(btnGuardar,gbc);
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        pnlAlta.add(btnLimpiar,gbc);
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        pnlAlta.add(btnSalir,gbc);

    }

    public void guardar(){
        if("").equals(txtNumeroEmpleado.getText()))
        {
            JOptionPane.showMessageDialog(null,"Alguno de los campos esta vacio ","Error",
JOptionPane.ERROR_MESSAGE);
        }
        Empleado emp = new Empleado();
        emp.numeroEmpleado = txtNumeroEmpleado.getText();
        emp.nombre = txtNombre.getText();
        emp.apellidos = txtApellidos.getText();
        emp.fechaNacimiento = txtFechaNac.getText();
        emp.curp = txtCURP.getText();
        emp.rfc = txtRFC.getText();
        emp.sueldo = txtSueldo.getText();
        emp.puesto = txtPuesto.getText();
        emp.sucursal = cmbSucursal.getSelectedItem().toString();
        emp.fechaIngreso = txtFechaIng.getText();

        BD.insertarEmpleado(emp.numeroEmpleado,emp.nombre,emp.apellidos,emp.fechaNacimiento,emp.
curp,emp.rfc,emp.sueldo,emp.puesto,emp.sucursal,emp.fechaIngreso);
        //limpiar();
    }

    public void limpiar(){

        txtNumeroEmpleado.setText("");
        txtNombre.setText("");
        txtApellidos.setText("");
        txtFechaNac.setText("");
        txtCURP.setText("");
        txtRFC.setText("");
        txtSueldo.setText("");
        txtPuesto.setText("");
        txtFechaIng.setText("");
        txtNumeroEmpleadoB.setText("");
        txtNombreB.setText("");
        txtApellidosB.setText("");

```



```

txtFechaNacB.setText("");
txtCURPB.setText("");
txtRFCB.setText("");
txtSueldoB.setText("");
txtPuestoB.setText("");
txtFechaIngB.setText("");
txtNumeroEmpleadoM.setText("");
txtNombreM.setText("");
txtApellidosM.setText("");
txtFechaNacM.setText("");
txtCURPM.setText("");
txtRFCM.setText("");
txtSueldoM.setText("");
txtPuestoM.setText("");
txtFechaIngM.setText("");
JOptionPane.showMessageDialog( null,"Limpiar con exito", "Mensaje",
JOptionPane.PLAIN_MESSAGE);
}
public void aparienciaBaja(){
    pnlBaja.setBorder(javax.swing.BorderFactory.createTitledBorder("Baja"));
    lblNumeroEmpleado.setText("Numero de empleado:");
    lblNombre.setText("Nombre:");
    lblApellidos.setText("Apellidos:");
    lblFechaNac.setText("Fecha de nacimiento:");
    lblCURP.setText("CURP:");
    lblRFC.setText("RFC:");
    lblSueldo.setText("Sueldo bruto:");
    lblPuesto.setText("Puesto:");
    lblSucursal.setText("Sucursal:");
    lblFechaIng.setText("Fecha de ingreso:");
    cmbSucursalB.setModel( new javax.swing.DefaultComboBoxModel(new
String[]{"Roma", "Insurgentes", "Condesa", "Centro"}));
    txtNombreB.setEnabled(false);
    txtApellidosB.setEnabled(false);
    txtFechaNacB.setEnabled(false);
    txtCURPB.setEnabled(false);
    txtRFCB.setEnabled(false);
    txtSueldoB.setEnabled(false);
    txtPuestoB.setEnabled(false);
    cmbSucursalB.setEnabled(false);
    txtFechaIngB.setEnabled(false);
    btnBuscar.setText("Buscar");
    btnBuscar.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){
            Buscar();
        }
    });
    btnNuevaBusqueda.setText("Nueva Busqueda");
    btnNuevaBusqueda.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){

```

```

        limpiar();
    }
});
btnEliminar.setText("Eliminar");
btnEliminar.addMouseListener(new MouseAdapter(){
    @Override
    public void mouseClicked(MouseEvent e){
        Eliminar();
    }
});
btnSalirB.setText("Salir");
btnSalirB.addMouseListener(new MouseAdapter(){
    @Override
    public void mouseClicked(MouseEvent e){
        System.exit(0);
    }
});
GridBagLayout gbl = new GridBagLayout();
GridBagConstraints gbc = new GridBagConstraints();
pnlBaja.setLayout(gbl);
gbc.weightx = 1;
gbc.weighty = 1;
gbc.fill = GridBagConstraints.HORIZONTAL;
gbc.anchor = GridBagConstraints.CENTER;
gbc.gridwidth = 1;
gbc.insets = new Insets(0,10,0,10);
pnlBaja.add(lblNumeroEmpleado, gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlBaja.add(txtNumeroEmpleadoB, gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlBaja.add(btnBuscar, gbc);
gbc.gridwidth = 1;
pnlBaja.add(lblNombre, gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlBaja.add(txtNombreB, gbc);
gbc.gridwidth = 1;
pnlBaja.add(lblApellidos, gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlBaja.add(txtApellidosB, gbc);
gbc.gridwidth = 1;
pnlBaja.add(lblFechaNac, gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlBaja.add(txtFechaNacB, gbc);
gbc.gridwidth = 1;
pnlBaja.add(lblCURP, gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlBaja.add(txtCURPB, gbc);
gbc.gridwidth = 1;
pnlBaja.add(lblRFC, gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
pnlBaja.add(txtRFCB, gbc);
gbc.gridwidth = 1;

```

```

        pnlBaja.add(lblSueldo,gbc);
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        pnlBaja.add(txtSueldoB,gbc);
        gbc.gridwidth = 1;
        pnlBaja.add(lblPuesto,gbc);
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        pnlBaja.add(txtPuestoB,gbc);
        gbc.gridwidth = 1;
        pnlBaja.add(lblSucursal,gbc);
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        pnlBaja.add(cmbSucursalB,gbc);
        gbc.gridwidth = 1;
        pnlBaja.add(lblFechaIng,gbc);
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        pnlBaja.add(txtFechaIngB,gbc);
        gbc.gridwidth = 1;
        pnlBaja.add(btnNuevaBusqueda,gbc);
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        pnlBaja.add(btnEliminar,gbc);
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        pnlBaja.add(btnSalirB,gbc);
    }
    public void Buscar(){
        Empleado emp = new Empleado();
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conexion =
DriverManager.getConnection("jdbc:mysql://localhost:3306/restaurant_soft","manne","Fco-1402");
            Statement st = conexion.createStatement();
            String consulta = "SELECT * FROM restaurant_soft.empleado WHERE
numeroEmpleado='"+txtNumeroEmpleadoB.getText()+"'";
            ResultSet rs=null;
            rs = st.executeQuery(consulta);
            if(rs.next()){
                emp.numeroEmpleado = rs.getString("numeroEmpleado");
                emp.nombre = rs.getString("nombre");
                emp.apellidos = rs.getString("apellidos");
                emp.fechaNacimiento = rs.getString("fechaNacimiento");
                emp.curp = rs.getString("curp");
                emp.rfc = rs.getString("rfc");
                emp.sueldo = rs.getString("sueldo");
                emp.puesto = rs.getString("puesto");
                emp.sucursal = rs.getString("sucursal");
                emp.fechaIngreso = rs.getString("fechaIngreso");
                txtNombreB.setText(emp.nombre);
                txtApellidosB.setText(emp.apellidos);
                txtFechaNacB.setText(emp.fechaNacimiento);
                txtCURPB.setText(emp.curp);
                txtRFCB.setText(emp.rfc);
                txtSueldoB.setText(emp.sueldo);
                txtPuestoB.setText(emp.puesto);
                cmbSucursalB.setSelectedItem(emp.sucursal);
            }
        }
    }
}

```

```

        txtFechaIngB.setText(emp.fechaIngreso);
        cmbSucursalB.setSelectedItem(emp.sucursal);
    }
    rs.close();
    st.close();
    conexion.close();
}
catch(ClassNotFoundException | SQLException e){
    JOptionPane.showMessageDialog(null,"Error al conectar con la Base de Datos","Error",
JOptionPane.ERROR_MESSAGE);
}

JOptionPane.showMessageDialog( null,"Empleado encontrado", "Mensaje",
JOptionPane.PLAIN_MESSAGE);
}
public void Eliminar(){
    if("").equals(txtNumeroEmpleadoB.getText())
    {
        JOptionPane.showMessageDialog(null,"Alguno de los campos esta vacio ","Error",
JOptionPane.ERROR_MESSAGE);
    }
    Empleado emp = new Empleado();
    emp.numeroEmpleado = txtNumeroEmpleadoB.getText();
    BD.eliminarEmpleado(emp.numeroEmpleado);
    limpiar();
}
public void aparienciaModificar(){
    pnlMod.setBorder(javax.swing.BorderFactory.createTitledBorder("Modificar"));
    lblNumeroEmpleado.setText("Numero de empleado:");
    lblNombre.setText("Nombre:");
    lblApellidos.setText("Apellidos:");
    lblFechaNac.setText("Fecha de nacimiento:");
    lblCURP.setText("CURP:");
    lblRFC.setText("RFC:");
    lblSueldo.setText("Sueldo bruto:");
    lblPuesto.setText("Puesto:");
    lblSucursal.setText("Sucursal:");
    lblFechaIng.setText("Fecha de ingreso:");
    cmbSucursalM.setModel( new javax.swing.DefaultComboBoxModel(new
String[]{"Roma","Insurgentes","Condesa","Centro"}));
    txtNombreM.setEnabled(false);
    txtApellidosM.setEnabled(false);
    txtFechaNacM.setEnabled(false);
    txtCURPM.setEnabled(false);
    txtRFCM.setEnabled(false);
    txtSueldoM.setEnabled(false);
    txtPuestoM.setEnabled(false);
    cmbSucursalM.setEnabled(false);
    txtFechaIngM.setEnabled(false);
    btnBuscarM.setText("Buscar");
    btnBuscarM.addMouseListener(new MouseAdapter(){
        @Override

```

```

        public void mouseClicked(MouseEvent e){
            BuscarM();
        }
    });
    btnGuardarM.setText("Guardar");
    btnGuardarM.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){
            actualizar();
        }
    });

    btnSalirM.setText("Salir");
    btnSalirM.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){
            System.exit(0);
        }
    });
    GridBagLayout gbl = new GridBagLayout();
    GridBagConstraints gbc = new GridBagConstraints();
    pnlMod.setLayout(gbl);
    gbc.weightx = 1;
    gbc.weighty = 1;
    gbc.fill = GridBagConstraints.HORIZONTAL;
    gbc.anchor = GridBagConstraints.CENTER;
    gbc.gridwidth = 1;
    gbc.insets = new Insets(0,10,0,10);
    pnlMod.add(lblNumeroEmpleado, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlMod.add(txtNumeroEmpleadoM, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlMod.add(btnBuscarM,gbc);
    gbc.gridwidth = 1;
    pnlMod.add(lblNombre,gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlMod.add(txtNombreM,gbc);
    gbc.gridwidth = 1;
    pnlMod.add(lblApellidos,gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlMod.add(txtApellidosM,gbc);
    gbc.gridwidth = 1;
    pnlMod.add(lblFechaNac,gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlMod.add(txtFechaNacM,gbc);
    gbc.gridwidth = 1;
    pnlMod.add(lblCURP,gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlMod.add(txtCURPM,gbc);
    gbc.gridwidth = 1;
    pnlMod.add(lblRFC,gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;

```

```

        pnlMod.add(txtRFCM,gbc);
        gbc.gridwidth = 1;
        pnlMod.add(lblSueldo,gbc);
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        pnlMod.add(txtSueldoM,gbc);
        gbc.gridwidth = 1;
        pnlMod.add(lblPuesto,gbc);
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        pnlMod.add(txtPuestoM,gbc);
        gbc.gridwidth = 1;
        pnlMod.add(lblSucursal,gbc);
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        pnlMod.add(cmbSucursalM,gbc);
        gbc.gridwidth = 1;
        pnlMod.add(lblFechaIng,gbc);
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        pnlMod.add(txtFechaIngM,gbc);
        gbc.gridwidth = 1;
        pnlMod.add(btnGuardarM,gbc);
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        pnlMod.add(btnSalirM,gbc);
    }
    public void actualizar(){
        if("").equals(txtNumeroEmpleadoM.getText())
        {
            JOptionPane.showMessageDialog(null,"Alguno de los campos esta vacio ","Error",
JOptionPane.ERROR_MESSAGE);
        }
        Empleado emp = new Empleado();
        emp.numeroEmpleado = txtNumeroEmpleadoM.getText();
        emp.nombre = txtNombreM.getText();
        emp.apellidos = txtApellidosM.getText();
        emp.fechaNacimiento = txtFechaNacM.getText();
        emp.curp = txtCURPM.getText();
        emp.rfc = txtRFCM.getText();
        emp.sueldo = txtSueldoM.getText();
        emp.puesto = txtPuestoM.getText();
        emp.sucursal = cmbSucursalM.getSelectedItem().toString();
        emp.fechaIngreso = txtFechaIngM.getText();

        BD.modificarEmpleado(emp.numeroEmpleado,emp.nombre,emp.apellidos,emp.fechaNacimiento,em
p.curp,emp.rfc,emp.sueldo,emp.puesto,emp.sucursal,emp.fechaIngreso);

        txtNumeroEmpleadoM.setEnabled(true);
        txtNombreM.setEnabled(false);
        txtApellidosM.setEnabled(false);
        txtFechaNacM.setEnabled(false);
        txtCURPM.setEnabled(false);
        txtRFCM.setEnabled(false);
        txtSueldoM.setEnabled(false);
        txtPuestoM.setEnabled(false);
        cmbSucursalM.setEnabled(false);

```

```

        txtFechaIngM.setEnabled(false);
        limpiar();
    }
    public void BuscarM(){
        Empleado emp = new Empleado();
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conexion =
DriverManager.getConnection("jdbc:mysql://localhost:3306/restaurant_soft","manne","Fco-1402");
            Statement st = conexion.createStatement();
            String consulta = "SELECT * FROM restaurant_soft.empleado WHERE
numeroEmpleado="+"'"+txtNumeroEmpleadoM.getText()+"'";
            ResultSet rs=null;
            rs = st.executeQuery(consulta);
            if(rs.next()){
                emp.numeroEmpleado = rs.getString("numeroEmpleado");
                emp.nombre = rs.getString("nombre");
                emp.apellidos = rs.getString("apellidos");
                emp.fechaNacimiento = rs.getString("fechaNacimiento");
                emp.curp = rs.getString("curp");
                emp.rfc = rs.getString("rfc");
                emp.sueldo = rs.getString("sueldo");
                emp.puesto = rs.getString("puesto");
                emp.sucursal = rs.getString("sucursal");
                emp.fechaIngreso = rs.getString("fechaIngreso");
                txtNombreM.setText(emp.nombre);
                txtApellidosM.setText(emp.apellidos);
                txtFechaNacM.setText(emp.fechaNacimiento);
                txtCURPM.setText(emp.curp);
                txtRFCM.setText(emp.rfc);
                txtSueldoM.setText(emp.sueldo);
                txtPuestoM.setText(emp.puesto);
                cmbSucursalM.setSelectedItem(emp.sucursal);
                txtFechaIngM.setText(emp.fechaIngreso);
                txtNumeroEmpleadoM.setEnabled(false);
                txtNombreM.setEnabled(true);
                txtApellidosM.setEnabled(true);
                txtFechaNacM.setEnabled(true);
                txtCURPM.setEnabled(true);
                txtRFCM.setEnabled(true);
                txtSueldoM.setEnabled(true);
                txtPuestoM.setEnabled(true);
                cmbSucursalM.setEnabled(true);

                cmbSucursalM.setSelectedItem(emp.sucursal);
                txtFechaIngM.setEnabled(true);
            }
            rs.close();
            st.close();
            conexion.close();
        }
        catch(ClassNotFoundException | SQLException e){

```

```

        JOptionPane.showMessageDialog(null,"Error al conectar con la Base de Datos","Error",
JOptionPane.ERROR_MESSAGE);
    }

    JOptionPane.showMessageDialog( null,"Empleado encontrado", "Mensaje",
JOptionPane.PLAIN_MESSAGE);
}
}

```

Clase Empleado

```

package restaurant.soft;

/**
 *
 * @author Manne25
 */
public class Empleado {
    String numeroEmpleado;
    String nombre;
    String apellidos;
    String fechaNacimiento;
    String curp;
    String rfc;
    String sueldo;
    String puesto;
    String sucursal;
    String fechaIngreso;
    String sueldoNeto;
    String descuentos;
}

```

Clase Nomina

```

package restaurant.soft;

import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.Insets;
import java.awt.event.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.*;

/**
 *
 * @author Manne25
 */
public class Nomina {
    Conexion BD = new Conexion();
}

```



```

Empleado emp = new Empleado();
//Componentes graficos Calculo Nomina
JPanel pnlCalc = new javax.swing.JPanel();
JLabel lblNumeroEmpleado = new javax.swing.JLabel();
JTextField txtNumeroEmpleado = new javax.swing.JTextField();
JButton btnBuscar = new javax.swing.JButton();
JLabel lblNombre= new javax.swing.JLabel();
JTextField txtNombre = new javax.swing.JTextField();
JLabel lblApellidos = new javax.swing.JLabel();
JTextField txtApellidos= new javax.swing.JTextField();
JLabel lblRFC= new javax.swing.JLabel();
JTextField txtRFC = new javax.swing.JTextField();
JLabel lblSueldoBruto = new javax.swing.JLabel();
JTextField txtSueldoBruto= new javax.swing.JTextField();
JLabel lblImpuestos= new javax.swing.JLabel();
JTextField txtImpuestos = new javax.swing.JTextField();
JLabel lblSueldoNeto = new javax.swing.JLabel();
JTextField txtSueldoNeto= new javax.swing.JTextField();
JButton btnNuevoCalculo = new javax.swing.JButton();
JButton btnCalcularNomina = new javax.swing.JButton();
JButton btnSalir = new javax.swing.JButton();

public void aparienciaCalc(){
    pnlCalc.setBorder(javax.swing.BorderFactory.createTitledBorder("Calcular Nomina"));
    lblNumeroEmpleado.setText("Numero de empleado:");
    lblNombre.setText("Nombre:");
    lblApellidos.setText("Apellidos:");
    lblRFC.setText("RFC:");
    lblSueldoBruto.setText("Sueldo Bruto:");
    lblImpuestos.setText("Porcentaje de impuestos:");
    lblSueldoNeto.setText("Sueldo Neto:");
    txtNombre.setEnabled(false);
    txtApellidos.setEnabled(false);
    txtRFC.setEnabled(false);
    txtSueldoBruto.setEnabled(false);
    txtSueldoNeto.setEnabled(false);
    btnBuscar.setText("Buscar");
    btnBuscar.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){
            Buscar();
        }
    });
    btnNuevoCalculo.setText("Nuevo Calculo");
    btnNuevoCalculo.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){
            limpiar();
        }
    });
    btnCalcularNomina.setText("Calcular Nomina");
    btnCalcularNomina.addMouseListener(new MouseAdapter(){

```

```

        @Override
        public void mouseClicked(MouseEvent e){
            CalcularNomina();
        }
    });
    btnSalir.setText("Salir");
    btnSalir.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){
            System.exit(0);
        }
    });
    GridBagLayout gbl = new GridBagLayout();
    GridBagConstraints gbc = new GridBagConstraints();
    pnlCalc.setLayout(gbl);
    gbc.weightx = 1;
    gbc.weighty = 1;
    gbc.fill = GridBagConstraints.HORIZONTAL;
    gbc.anchor = GridBagConstraints.CENTER;
    gbc.gridwidth = 1;
    gbc.insets = new Insets(0,10,0,10);
    pnlCalc.add(lblNumeroEmpleado, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlCalc.add(txtNumeroEmpleado, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlCalc.add(btnBuscar,gbc);
    gbc.gridwidth = 1;
    pnlCalc.add(lblNombre,gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlCalc.add(txtNombre,gbc);
    gbc.gridwidth = 1;
    pnlCalc.add(lblApellidos,gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlCalc.add(txtApellidos,gbc);
    gbc.gridwidth = 1;
    pnlCalc.add(lblRFC,gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlCalc.add(txtRFC,gbc);
    gbc.gridwidth = 1;
    pnlCalc.add(lblSueldoBruto,gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlCalc.add(txtSueldoBruto,gbc);
    gbc.gridwidth = 1;
    pnlCalc.add(lblImpuestos,gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlCalc.add(txtImpuestos,gbc);
    gbc.gridwidth = 1;
    pnlCalc.add(lblSueldoNeto,gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    pnlCalc.add(txtSueldoNeto,gbc);
    pnlCalc.add(btnNuevoCalculo,gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;

```

```

        pnlCalc.add(btnCalcularNomina,gbc);
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        pnlCalc.add(btnSalir,gbc);
    }
    public void Buscar(){
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conexion =
DriverManager.getConnection("jdbc:mysql://localhost:3306/restaurant_soft","manne","Fco-1402");
            Statement st = conexion.createStatement();
            String consulta = "SELECT * FROM restaurant_soft.empleado WHERE
numeroEmpleado='"+txtNumeroEmpleado.getText()+"'";
            ResultSet rs=null;
            rs = st.executeQuery(consulta);
            if(rs.next()){
                emp.numeroEmpleado = rs.getString("numeroEmpleado");
                emp.nombre = rs.getString("nombre");
                emp.apellidos = rs.getString("apellidos");
                emp.fechaNacimiento = rs.getString("fechaNacimiento");
                emp.curp = rs.getString("curp");
                emp.rfc = rs.getString("rfc");
                emp.sueldo = rs.getString("sueldo");
                emp.puesto = rs.getString("puesto");
                emp.sucursal = rs.getString("sucursal");
                emp.fechaIngreso = rs.getString("fechaIngreso");
                txtNombre.setText(emp.nombre);
                txtApellidos.setText(emp.apellidos);
                txtRFC.setText(emp.rfc);
                txtSueldoBruto.setText(emp.sueldo);
            }
            rs.close();
            st.close();
            conexion.close();
        }
        catch(ClassNotFoundException | SQLException e){
            JOptionPane.showMessageDialog(null,"Error al conectar con la Base de Datos","Error",
JOptionPane.ERROR_MESSAGE);
        }

        JOptionPane.showMessageDialog( null,"Empleado encontrado", "Mensaje",
JOptionPane.PLAIN_MESSAGE);
    }

    public void limpiar(){

        txtNumeroEmpleado.setText("");
        txtNombre.setText("");
        txtApellidos.setText("");
        txtRFC.setText("");
        txtSueldoBruto.setText("");
        txtImpuestos.setText("");
        txtSueldoNeto.setText("");
    }

```

```

        JOptionPane.showMessageDialog( null,"Limpiar con exito", "Mensaje",
JOptionPane.PLAIN_MESSAGE);
    }
    public void CalcularNomina(){
        if("").equals(txtNumeroEmpleado.getText()))
        {
            JOptionPane.showMessageDialog(null,"Alguno de los campos esta vacio ", "Error",
JOptionPane.ERROR_MESSAGE);
        }
        else
        {
            int imp = Integer.parseInt(txtImpuestos.getText());
            int des = (int) ((imp*.01)*Integer.parseInt(emp.sueldo));
            emp.descuentos = Integer.toString(des);
            int tot = Integer.parseInt(emp.sueldo)-des;
            emp.sueldoNeto = Integer.toString(tot);
            txtSueldoNeto.setText(emp.sueldoNeto);
            BD.actualizarNomina(emp.numeroEmpleado, emp.descuentos, emp.sueldoNeto);
        }
    }
}

```

Clase Consultas

```
package restaurant.soft;
```

```

import java.awt.BorderLayout;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.Insets;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.io.FileOutputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import javax.swing.text.Document;

```

```

/**
 *
 * @author Manne25
 */
public class Consultas {
    JPanel pnlNominas = new javax.swing.JPanel();
    DefaultTableModel modelo= new DefaultTableModel();

```

```

JScrollPane scroll=new JScrollPane();
JTable tabla = new JTable();
JButton btnBuscar = new javax.swing.JButton();
JButton btnEliminar = new javax.swing.JButton();

public void aparienciaNominas(){
    pnlNominas.setBorder(javax.swing.BorderFactory.createTitledBorder("Reporte de Nominas"));
    modelo.addColumn("Numero Empleado");
    modelo.addColumn("Nombre");
    modelo.addColumn("Apellidos");
    modelo.addColumn("Sueldo Bruto");
    modelo.addColumn("Descuentos");
    modelo.addColumn("Sueldo Neto");
    tabla.setModel(modelo);
    scroll.setViewportView(tabla);
    btnBuscar.setText("Buscar");
    btnBuscar.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){
            Buscar();
        }
    });
    btnEliminar.setText("Eliminar");
    btnEliminar.addMouseListener(new MouseAdapter(){
        @Override
        public void mouseClicked(MouseEvent e){
            limpiar();
        }
    });
    GridBagLayout gbl = new GridBagLayout();
    GridBagConstraints gbc = new GridBagConstraints();
    pnlNominas.setLayout(gbl);
    gbc.weightx = 1;
    gbc.weighty = 1;
    gbc.fill = GridBagConstraints.BOTH;
    gbc.anchor = GridBagConstraints.CENTER;
    gbc.gridwidth = 0;
    gbc.insets = new Insets(0,10,0,10);
    pnlNominas.add(scroll,gbc);
    gbc.fill = GridBagConstraints.HORIZONTAL;
    gbc.gridwidth = 1;
    pnlNominas.add(btnBuscar,gbc);
    gbc.gridwidth =1;
    pnlNominas.add(btnEliminar,gbc);
}

public void Buscar(){
    try{
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection conexion =
DriverManager.getConnection("jdbc:mysql://localhost:3306/restaurant_soft","manne","Fco-1402");
        Statement st = conexion.createStatement();
        String consulta = "SELECT * FROM restaurant_soft.empleado ";
    }
}

```

```

        ResultSet rs=null;
        rs = st.executeQuery(consulta);
        while(rs.next()){
            Object[] data
            ={rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(7),rs.getString(12),rs.getString(11)};
            modelo.addRow(data);
        }
        rs.close();
        st.close();
        conexion.close();
    }
    catch(ClassNotFoundException | SQLException e){
        JOptionPane.showMessageDialog(null,"Error al conectar con la Base de Datos","Error",
        JOptionPane.ERROR_MESSAGE);
    }

    JOptionPane.showMessageDialog( null,"Consulta Generada", "Mensaje",
    JOptionPane.PLAIN_MESSAGE);
}
public void limpiar(){
    for(int x=0;x<modelo.getRowCount();x++){
        modelo.removeRow(x);
    }
}
}
}

```

Clase Conexion

```
package restaurant.soft;
```

```

/**
 *
 * @author Manne25
 */
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
public class Conexion {
    public void insertarInventario(String cod,String art, String suc, String id, String exi, String mar){
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conexion =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/restaurant_soft","manne","Fco-1402");
            Statement st = conexion.createStatement();
            String insercion ="INSERT INTO restaurant_soft.articulo VALUES
            (""+cod+"",""+art+"",""+suc+"",""+id+"",""+exi+"",""+mar+"");";
            st.executeUpdate(insercion);
            st.close();
        }
    }
}

```

```

        conexion.close();
        JOptionPane.showMessageDialog( null,"Guardado con exito", "Mensaje",
JOptionPane.PLAIN_MESSAGE);
    }
    catch(ClassNotFoundException | SQLException e){
        JOptionPane.showMessageDialog(null,"Error al conectar con la Base de Datos","Error",
JOptionPane.ERROR_MESSAGE);
    }

}

public void eliminarInventario(String cod, String suc){
    try{
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection conexion =
DriverManager.getConnection("jdbc:mysql://localhost:3306/restaurant_soft","manne","Fco-1402");
        Statement st = conexion.createStatement();
        String eliminar ="DELETE FROM restaurant_soft.articulo WHERE codigo='"+cod+"' AND
sucursal='"+suc+"';";
        st.executeUpdate(eliminar);
        st.close();
        conexion.close();
        JOptionPane.showMessageDialog( null,"Insumo eliminado", "Mensaje",
JOptionPane.PLAIN_MESSAGE);
    }
    catch(ClassNotFoundException | SQLException e){
        JOptionPane.showMessageDialog(null,"Error al conectar con la Base de Datos","Error",
JOptionPane.ERROR_MESSAGE);
    }

}

public void modificarInventario(String cod,String art, String suc, String id, String exi, String mar){
    try{
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection conexion =
DriverManager.getConnection("jdbc:mysql://localhost:3306/restaurant_soft","manne","Fco-1402");
        Statement st = conexion.createStatement();
        String actualizar ="UPDATE restaurant_soft.articulo SET codigo='"+cod+",
articulo='"+art+", sucursal='"+suc+", idTipoInsumo='"+id+", existencia='"+exi+",marca='"+mar+"
WHERE codigo='"+cod+"' AND sucursal='"+suc+"';";
        st.executeUpdate(actualizar);
        st.close();
        conexion.close();
        JOptionPane.showMessageDialog( null,"Insumo actualizado", "Mensaje",
JOptionPane.PLAIN_MESSAGE);
    }
    catch(ClassNotFoundException | SQLException e){
        JOptionPane.showMessageDialog(null,"Error al conectar con la Base de Datos","Error",
JOptionPane.ERROR_MESSAGE);
    }

}

```

```

    public void insertarEmpleado(String num,String nom, String ap, String fn, String curp, String rfc,
String sue, String pue, String suc, String fi){
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conexion =
DriverManager.getConnection("jdbc:mysql://localhost:3306/restaurant_soft","manne","Fco-1402");
            Statement st = conexion.createStatement();
            String insercion ="INSERT INTO restaurant_soft.empleado VALUES
("+num+"",""+nom+"",""+ap+"",""+fn+"",""+curp+"",""+rfc+"",""+sue+"",""+pue+"",""+suc+"",""+fi+"",null,null);";
            st.executeUpdate(insercion);
            st.close();
            conexion.close();
            JOptionPane.showMessageDialog( null,"Guardado con exito", "Mensaje",
JOptionPane.PLAIN_MESSAGE);
        }
        catch(ClassNotFoundException | SQLException e){
            JOptionPane.showMessageDialog(null,"Error al conectar con la Base de Datos","Error",
JOptionPane.ERROR_MESSAGE);
        }
    }

```

```

    public void modificarEmpleado(String num,String nom, String ap, String fn, String curp, String rfc,
String sue, String pue, String suc, String fi){
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conexion =
DriverManager.getConnection("jdbc:mysql://localhost:3306/restaurant_soft","manne","Fco-1402");
            Statement st = conexion.createStatement();
            String actualizar ="UPDATE restaurant_soft.empleado SET numeroEmpleado="+num+",
nombre="+nom+", apellidos="+ap+", fechaNacimiento="+fn+",
curp="+curp+",rfc="+rfc+",sueldo="+sue+",puesto="+pue+",sucursal="+suc+"
,fechaIngreso="+fi+" WHERE numeroEmpleado="+num+"";";
            st.executeUpdate(actualizar);
            st.close();
            conexion.close();
            JOptionPane.showMessageDialog( null,"Empleado actualizado", "Mensaje",
JOptionPane.PLAIN_MESSAGE);
        }
        catch(ClassNotFoundException | SQLException e){
            JOptionPane.showMessageDialog(null,"Error al conectar con la Base de Datos","Error",
JOptionPane.ERROR_MESSAGE);
        }
    }

```

```

    }
    public void eliminarEmpleado(String num){
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conexion =
DriverManager.getConnection("jdbc:mysql://localhost:3306/restaurant_soft","manne","Fco-1402");
            Statement st = conexion.createStatement();
            String eliminar ="DELETE FROM restaurant_soft.empleado WHERE
numeroEmpleado="+num+"";";

```



```

        st.executeUpdate(eliminar);
        st.close();
        conexion.close();
        JOptionPane.showMessageDialog( null,"Empleado eliminado", "Mensaje",
JOptionPane.PLAIN_MESSAGE);
    }
    catch(ClassNotFoundException | SQLException e){
        JOptionPane.showMessageDialog(null,"Error al conectar con la Base de Datos","Error",
JOptionPane.ERROR_MESSAGE);
    }

}

public void actualizarNomina(String num, String des, String sueN){
    try{
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection conexion =
DriverManager.getConnection("jdbc:mysql://localhost:3306/restaurant_soft","manne","Fco-1402");
        Statement st = conexion.createStatement();
        String actualizar ="UPDATE restaurant_soft.empleado SET descuentos='"+des+"",
sueldoNeto="'+sueN+"' WHERE numeroEmpleado='"+num+"';";
        st.executeUpdate(actualizar);
        st.close();
        conexion.close();
        JOptionPane.showMessageDialog( null,"Nomina generada", "Mensaje",
JOptionPane.PLAIN_MESSAGE);
    }
    catch(ClassNotFoundException | SQLException e){
        JOptionPane.showMessageDialog(null,"Error al conectar con la Base de Datos","Error",
JOptionPane.ERROR_MESSAGE);
    }

}

}

```

Video

<https://drive.google.com/file/d/1Rtsb1vkiEIJKLFsIs2lQ4zxkRNPH6C19/view?usp=sharing>

Conclusión:

Con todo lo aprendido y aplicado en este proyecto RestaurantSoft, pude aprender como realizar una aplicación en jaco, con la cual puedo realizar consultas, inserciones, modificaciones y eliminaciones de una base de datos, con ello pienso que es un gran avance en mi desarrollo profesional ya que hoy en día las bases de datos se utilizan en todas partes y con ello puedo desempeñarme de una manera eficaz en el campo laboral del área de un ingeniero en desarrollo de software.

Bibliografía:

- Aguilar, L. J., Azuela, M. F., & Baena, L. R. (1988). Fundamentos de programación. McGraw Hill.
- Barnes, D. J., Kölling, M., & Brenta, B. I. (2007). *Programación orientada a objetos con Java*. Pearson Educación.
- García, J., Rodríguez, J., Mingo, I., Imaz, A., Brazález, A., Larzabal, A. et ál. (2000). Aprende Java como si estuviera en primero. Recuperado de: <http://ocw.uc3m.es/cursos-archivados/programacion-java/manuales/java2-U-Navarra.pdf/view>
- Gosling, J., Holmes, D. y Arnold, K. (2001). El lenguaje de programación Java TM.
- Luis, J. A. (2008). Fundamentos de programación. *Algoritmos, Estructura de Datos y Objetos. 4ta Edición*. Editorial Mc Graw-Hill Interamericana.