

RETO SEMANA 7 – FUNCIONALIDADES, REPORTE GRAFICO Y EXPORTACION DEL DOCUMENTO A ARCHIVO EXCEL.

Dargy Julieth Mogollón Lobo

FUNCIONALIDADES

La tienda de Misión TIC, requiere los servicios del equipo desarrolladora Dargy Mogollón, le ha solicitado una aplicación Java de escritorio (local) para el registro de las ventas diarias en su tienda. Actualmente la problemática consiste en que no cuenta con el balance general de las ventas al final del día cuando se hace el cierre de la caja. En muchas ocasiones los valores ingresados no corresponden con lo registrado en la caja físicamente, esto implica que los registros no se están controlando bien ya que se hacen de forma manual.

Se sigue con la continuación del proyecto anterior por lo que en primer lugar, se realiza la creación de FACTURACION en la base de datos, se crean la codificación del CRUD. Para ello se crea el modelo, vista y controlador (MVC) se crean las clases que contiene constructores, getter, setter y las mismas variables de las bases de datos (con su nombre y tipo de dato), luego se pasa a los botones, cajas de texto los cuales se editan en el jframe de la vista de ahí pasar al controlador donde se crea cada función que pueda conectarse con el modelo y tener acceso a la base de datos para registrar y leer los registros. A continuación, unos pantallazos del proceso.

A continuación, se adjuntan pantallazos de la interfaz gráfica, del código y de la base de datos.

Facturacion | Enter a SQL expression to filter results (use Ctrl+Space)

	FacturacionID	ClientID	EmpleadosID	ProductoID	CantidadProducto	Subtotal
1	1		1	1	3	3,600
2	2	2	2	2	2	1,200
3	3	1,098	123	3	4	8,000

1 - 1 - 3600.0
2 - 2 - 1200.0
3 - 123 - 8000.0

Actualizar Crear Editar Borrar

FacturacionID ClientID EmpleadosID
4 1098 2

ProductoID Cantidad Prod Subtotal
4 2 8000.0

Total Total Graficar

Cerrar Exportar

Id facturacion - Id Cliente - Subtot...

Mensaje

Nuevo registro agregado

Aceptar

	FacturacionID	ClientID	EmpleadosID	ProductoID	CantidadProducto	Subtotal
1	1		1	1	3	3,600
2	2	2	2	2	2	1,200
3	3	1,098	123	3	4	8,000
4	4	1,098	2	4	2	2,600

Se revisa que el nuevo registro haya sido correctamente almacenado en la base de datos.

1 - 1 - 3600.0
2 - 2 - 1200.0
3 - 123 - 8000.0
4 - 2 - 2600.0

Actualizar Crear Editar Borrar

FacturacionID ClientID EmpleadosID
4 1098 2

ProductoID Cantidad Prod Subtotal
4 2 8000.0

Total 15400 Total Graficar

Cerrar Exportar

Id facturacion - Id Cliente - Subtot...

Se revisa que el botón del total realice la operación aritmética de la sumatoria de todos los productos elegidos correctamente.

Se grafica

1 - 1 - 3600.0

2 - 2 - 1200.0

3 - 123 - 8000.0

4 - 2 - 2600.0

Actualizar

Crear

Editar

Borrar

FacturacionID

4

ClienteID

1098

EmpleadoID

2

ProductoID

4

Cantidad Prod

2

Subtotal

8000.0

Total

15400

Total

Graficar

Cerrar

Exportar

Id facturacion - Id Cliente - Subtot...



Se exporta como archivo Excel

1 - 1 - 3600.0
2 - 2 - 1200.0
3 - 123 - 8000.0
4 - 2 - 2600.0

Actualizar
Crear
Editar
Borrar

FacturacionID
4

ClienteID
1098

EmpleadoID
2

ProductID
4

Cantidad Prod
2

Subtotal
8000.0

Total
15400

Total

Graficar

Cerrar
Exportar

Id facturacion - Id Cliente - Subtot...

Mensaje

Exportado

Aceptar

build	16/08/2021 5:32 p. m.	Carpeta de archivos	
nbproject	09/08/2021 5:02 p. m.	Carpeta de archivos	
src	09/08/2021 7:09 p. m.	Carpeta de archivos	
test	26/07/2021 4:32 p. m.	Carpeta de archivos	
build.xml	16/08/2021 5:20 p. m.	Archivo XML	4 KB
Facturacion.xlsx	17/08/2021 8:47 a. m.	Hoja de cálculo d...	4 KB
facturacion_fecha_2021_08_17 08-58-22-hora.xls	17/08/2021 8:58 a. m.	Hoja de cálculo d...	4 KB
facturacion_fecha_2021_08_17 08-58-51-hora.xls	17/08/2021 8:58 a. m.	Hoja de cálculo d...	4 KB
facturacion_fecha_2021_08_17 08-59-06-hora.xls	17/08/2021 8:59 a. m.	Hoja de cálculo d...	4 KB
facturacion_fecha_2021_08_17 12-07-17-hora.xls	17/08/2021 12:07 p. m.	Hoja de cálculo d...	5 KB
manifest.mf	26/07/2021 4:29 p. m.	Archivo MF	1 KB
Productos.xlsx	16/08/2021 1:58 p. m.	Hoja de cálculo d...	4 KB
Users.xlsx	12/08/2021 6:59 p. m.	Hoja de cálculo d...	4 KB

1 - 1 - 3600.0
2 - 2 - 1200.0
3 - 123 - 8000.0
4 - 2 - 2600.0

Actualizar
Crear
Editar
Borrar

FacturacionID
4

ClienteID
1098

EmpleadoID
2

ProductID
4

Cantidad Prod
2

Subtotal
8000.0

Total
15400

Total

Graficar

Cerrar
Exportar

Id facturacion - Id Cliente - Subtot...

	Facturacion ID	Clinte ID	ID Empleados	ID Producto	Cantidad de Producto	Subtotal	
	1	1	1	1	3	3600	
	2	2	2	2	2	1200	
	3	1098	123	3	4	8000	
	4	1098	2	4	2	2600	

	FacturacionID	ClienteID	EmpleadosID	ProductoID	CantidadProducto	Subtotal
1	1		1	1	3	3.600
2	2	2	2	2	2	1.200
3	3	1.098	123	3	4	8.000
4	4	1.098	2	4	2	2.600

VIDEO

<https://drive.google.com/file/d/1AJ3tnq4KxTnLDWwkehNNlgqd6BZaJp3V/view?usp=sharing>

El software cumple con:

- ✓ CRUD (empleados). Para este registro, el usuario debe ingresar a la plataforma y luego seleccionar la opción registrar, en donde debería suministrar la siguiente información para registrar un nuevo empleado: nombre de usuario, contraseña y email.
- ✓ Proveer un portal de acceso donde los usuarios puedan acceder al sistema, si se autentican (usando login y password), exitosamente.
- ✓ Ofrecer una vista para el registro de productos por el usuario, cuando esté autenticado. Es decir, el usuario puede hacer CRUD (productos). En la creación de un producto, se debería ingresar una referencia (un identificador para el producto), nombre del producto, cantidad del producto, y precio por unidad.
- ✓ Ofrecer una vista de los productos existentes.
- ✓ Para el registro de una venta, el usuario cajero deberá poder registrar un producto de una venta, finalmente mostrando el valor de la facturación a pagar

1. Código

BASE DE DATOS

```
CREATE TABLE `productos` (
```

```

        `ProductoID` int(11) NOT NULL AUTO_INCREMENT,
        `Nombre` varchar(50) NOT NULL DEFAULT '0',
        `PrecioUnitario` double NOT NULL DEFAULT 0,
        `Cantidad` int(11) NOT NULL DEFAULT 0,
        PRIMARY KEY (`ProductoID`)
    ) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=latin1;

```

```

DROP TABLE IF EXISTS `empleados`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `empleados` (
    `ID` int(11) NOT NULL,
    `NombreDelEmpleado` varchar(100) NOT NULL,
    `Contraseña` double NOT NULL,
    `Email` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

```

```

CREATE TABLE `facturacion` (
    `FacturacionID` int(11) NOT NULL AUTO_INCREMENT,
    `ClienteID` int(11) NOT NULL,
    `EmpleadosID` int(11) NOT NULL,
    `ProductoID` int(11) NOT NULL,
    `CantidadProducto` int(11) NOT NULL,
    `Subtotal` double NOT NULL,
    PRIMARY KEY (`FacturacionID`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1

```

NETBEANS

```

/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package pkgController;

```

```

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.DefaultListModel;
import javax.swing.JOptionPane;
import pkgModel.clsDAOProducto;

```

```

import pkgModel.clsProducto;
import pkgView.frmProducto;
import pkgView.graph.graphProducts;

/**
 *
 * @author Dargy Lobo
 */
public class ctrProducto implements ActionListener {

    clsDAOProducto modeloproducto;
    frmProducto vista_prods;

    public ctrProducto(frmProducto prod) {
        this.modeloproducto = new clsDAOProducto();
        this.vista_prods = prod;
        this.vista_prods.getBtnClose().addActionListener(this);
        this.vista_prods.getBtnNew().addActionListener(this);
        this.vista_prods.getBtnSave().addActionListener(this);
        this.vista_prods.getBtnBusc().addActionListener(this);
        this.vista_prods.getBtnBorrar().addActionListener(this);
        this.vista_prods.getBtnEditProd().addActionListener(this);

        this.vista_prods.getListProds().addMouseListener(new
java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                list(evt);
            }
        });
        this.list();
    }

    public void list() {
        DefaultListModel<clsProducto> dlm = new DefaultListModel();
        java.sql.ResultSet rs = modeloproducto.select();
        try {
            while (rs.next()) {
                dlm.addElement(new clsProducto(rs.getInt(1),
rs.getString(2), rs.getDouble(3), rs.getInt(4)));
            }
        } catch (Exception e) {
        }
        vista_prods.getListProds().setModel(dlm);
    }

    public void list(java.awt.event.MouseEvent evt) {
        clsProducto result = (clsProducto)
this.vista_prods.getListProds().getSelectedValue();
        if (result instanceof clsProducto) {
            vista_prods.getTxtID().setText("" + result.getIdProducto());
            vista_prods.getTxtName().setText(result.getNombProd());
            vista_prods.getTxtPrecioUnit().setText("" +
result.getPrecioUnit());
            vista_prods.getTxtCantidad().setText("" +
result.getCantidad());
        }
    }

```

```

    }

    @Override
    public void actionPerformed(ActionEvent asd) {
        switch (asd.getActionCommand()) {
            case "Actualizar": {
                vista_prods.getTxtID().setText("");
                vista_prods.getTxtName().setText("");
                vista_prods.getTxtPrecioUnit().setText("");
                vista_prods.getTxtCantidad().setText("");
                vista_prods.getTxtBuscProd().setText("");
                break;
            }
            case "Crear": {

modeloprod.getProds().setIdProducto(Integer.parseInt(vista_prods.getTxtID()
.getText()));

modeloprod.getProds().setNombProd(vista_prods.getTxtName().getText());

modeloprod.getProds().setPrecioUnit(Double.parseDouble(vista_prods.getTxtPr
ecioUnit().getText()));

modeloprod.getProds().setCantidad(Integer.parseInt(vista_prods.getTxtCantid
ad().getText()));

                if (modeloprod.insert()) {
                    JOptionPane.showMessageDialog(vista_prods, "Nuevo
producto agregado");
                    this.list();
                } else {
                    JOptionPane.showMessageDialog(vista_prods, "No se puedo
agregar producto");
                }
                break;
            }
            case "Editar": {

modeloprod.getProds().setIdProducto(Integer.parseInt(vista_prods.getTxtID()
.getText()));

modeloprod.getProds().setNombProd(vista_prods.getTxtName().getText());

modeloprod.getProds().setPrecioUnit(Double.parseDouble(vista_prods.getTxtPr
ecioUnit().getText()));

modeloprod.getProds().setCantidad(Integer.parseInt(vista_prods.getTxtCantid
ad().getText()));

                if (modeloprod.update()) {
                    JOptionPane.showMessageDialog(vista_prods, "Producto
actualizado");
                } else {
                    JOptionPane.showMessageDialog(vista_prods, "Producto no
actualizado");
                }
            }
        }
    }
}

```



```

        }
        this.list();
        break;
    }
    case "Cerrar": {
        this.vista_prods.dispose();

        break;
    }

    case "Buscar": {

modeloprod.getProds().setIdProducto(Integer.parseInt(vista_prods.getTxtBusc
Prod().getText()));
        java.sql.ResultSet resultado;
        resultado =
modeloprod.search(Integer.parseInt(vista_prods.getTxtBuscProd().getText()))
;
        try {
            if (resultado.next()) {

vista_prods.getTxtID().setText(resultado.getString(1));

vista_prods.getTxtName().setText(resultado.getString(2));

vista_prods.getTxtPrecioUnit().setText(resultado.getString(3));

vista_prods.getTxtCantidad().setText(resultado.getString(4));

            }
        } catch (SQLException ex) {
            System.out.println("error");
            System.out.println("error buscando" + ex);

Logger.getLogger(ctrProducto.class.getName()).log(Level.SEVERE, null, ex);
        }
        break;
    }
    case "Graficar": {
        String[] args = null;
        graphProducts grafica= new graphProducts();
        grafica.init(args);
        break;
    }
    case "Exportar": {
        new
pkgModel.export.ExcelWriterFactura().write(modeloprod.select());
        break;
    }

    case "Borrar": {

modeloprod.getProds().setIdProducto(Integer.parseInt(vista_prods.getTxtID()
.getText()));
        modeloprod.delete();
        this.list();
    }
}

```

```

        break;
    }

}

public void show_frame() {
    vista_prods.setVisible(true);
}
}

```

* To change this license header, choose License Headers in Project Properties.

* To change this template file, choose Tools | Templates

* and open the template in the editor.

*/

```
package pkgController;
```

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.DefaultListModel;
import javax.swing.JOptionPane;
import pkgModel.clsDAOEmpleados;
import pkgModel.clsEmpleados;
import pkgView.frmEmpleados;
```

```
/**
```

```
 *
```

```
 * @author Dargy Lobo
```

```
 */
```

```
public class ctrEmpleados implements ActionListener {
```

```
    clsDAOEmpleados modeloEmpleados;
```

```
    frmEmpleados vista_emple;
```

```
    public ctrEmpleados(frmEmpleados emple){
```

```
        this.modeloEmpleados = new clsDAOEmpleados();
```

```
        this.vista_emple = emple;
```

```
        this.vista_emple.getBtnClose().addActionListener(this);
```

```
        this.vista_emple.getBtnNew().addActionListener(this);
```

```
        this.vista_emple.getBtnSave().addActionListener(this);
```

```
        this.vista_emple.getBtnBusc().addActionListener(this);
```

```
        this.vista_emple.getBtnBorrar().addActionListener(this);
```

```
        this.vista_emple.getBtnEditProd().addActionListener(this);
```

```
        this.vista_emple.getListProds().addMouseListener(new
        java.awt.event.MouseAdapter() {
```

```
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                list(evt);
            }

```

```
        });
    }
}

```

```

        this.list();
    }

    public void list() {
        DefaultListModel<clsEmpleados> dlm = new DefaultListModel();
        java.sql.ResultSet rs = modeloEmpleados.select();
        try {
            while (rs.next()) {
                dlm.addElement(new clsEmpleados(rs.getInt(1),
rs.getString(2), rs.getDouble(3), rs.getString(4)));
            }
        } catch (Exception e) {
        }
        vista_emple.getListProds().setModel(dlm);
    }

    public void list(java.awt.event.MouseEvent evt) {
        clsEmpleados result = (clsEmpleados)
this.vista_emple.getListProds().getSelectedValue();
        if (result instanceof clsEmpleados) {
            vista_emple.getTxtID().setText("" + result.getIdEmpleado());
            vista_emple.getTxtName().setText(result.getNombreDeEmpleado());
            vista_emple.getTxtPrecioUnit().setText("" +
result.getContraseña());
            vista_emple.getTxtCantidad().setText("" + result.getEmail());
        }
    }

    @Override
    public void actionPerformed(ActionEvent asd) {
        switch (asd.getActionCommand()) {
            case "Actualizar": {
                vista_emple.getTxtID().setText("");
                vista_emple.getTxtName().setText("");
                vista_emple.getTxtPrecioUnit().setText("");
                vista_emple.getTxtCantidad().setText("");
                vista_emple.getTxtBuscProd().setText("");
                break;
            }
            case "Crear": {

modeloEmpleados.getEmple().setIdEmpleado(Integer.parseInt(vista_emple.getTx
tID().getText()));

modeloEmpleados.getEmple().setNombreDeEmpleado(vista_emple.getTxtName().get
Text());

modeloEmpleados.getEmple().setContraseña(Double.parseDouble(vista_emple.get
TxtPrecioUnit().getText()));

modeloEmpleados.getEmple().setEmail(vista_emple.getTxtCantidad().getText())
;

                if (modeloEmpleados.insert()) {
                    JOptionPane.showMessageDialog(vista_emple, "Nuevo

```

```

Empleado agregado");
                this.list();
            } else {
                JOptionPane.showMessageDialog(vista_emple, "No se puedo
agregar el Empleado");
            }
            break;
        }
        case "Editar": {

modeloEmpleados.getEmple().setIdEmpleado(Integer.parseInt(vista_emple.getTx
tID().getText()));

modeloEmpleados.getEmple().setNombreDeEmpleado(vista_emple.getTxtName().get
Text());

modeloEmpleados.getEmple().setContraseña(Double.parseDouble(vista_emple.get
TxtPrecioUnit().getText()));

modeloEmpleados.getEmple().setEmail(vista_emple.getTxtCantidad().getText())
;

                if (modeloEmpleados.update()) {
                    JOptionPane.showMessageDialog(vista_emple, "Empleado
actualizado");
                } else {
                    JOptionPane.showMessageDialog(vista_emple, "Empleado no
actualizado");
                }
                this.list();
                break;
            }
        case "Cerrar": {
            this.vista_emple.dispose();

            break;
        }

        case "Buscar": {

modeloEmpleados.getEmple().setIdEmpleado(Integer.parseInt(vista_emple.getTx
tBuscProd().getText()));
                java.sql.ResultSet resultado;

resultado=modeloEmpleados.search(Integer.parseInt(vista_emple.getTxtBuscPro
d().getText()));
                try {
                    if (resultado.next()){
                        vista_emple.getTxtID().setText(resultado.getString(1));

vista_emple.getTxtName().setText(resultado.getString(2));

vista_emple.getTxtPrecioUnit().setText(resultado.getString(3));

vista_emple.getTxtCantidad().setText(resultado.getString(4));

```

```

        }
    } catch (SQLException ex) {
        System.out.println("error");
        System.out.println("error buscando"+ex);

        Logger.getLogger(ctrProducto.class.getName()).log(Level.SEVERE, null, ex);
    }
    break;
}
case "Borrar": {

modeloEmpleados.getEmple().setIdEmpleado(Integer.parseInt(vista_emple.getTx
tID().getText()));
        modeloEmpleados.delete();
        this.list();
        break;
    }

}

}

public void show_frame() {
    vista_emple.setVisible(true);
}
}

```

```

CREATE TABLE `productos` (
  `ProductoID` int(11) NOT NULL AUTO_INCREMENT,
  `Nombre` varchar(50) NOT NULL DEFAULT '0',
  `PrecioUnitario` double NOT NULL DEFAULT 0, `Cantidad` int(11)
  NOT NULL DEFAULT 0, PRIMARY KEY (`ProductoID`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `direcciones` (
  `DireccionID` int(11) NOT NULL,
  `CalleCarrera` varchar(50) NOT NULL DEFAULT '0',
  `Numero` int(11) NOT NULL DEFAULT 0,
  `Barrio` varchar(50) NOT NULL DEFAULT '0',
  `Ciudad` varchar(50) NOT NULL DEFAULT '0',
  PRIMARY KEY (`DireccionID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1

```

```

CREATE TABLE `tbl_user` (
  `id` varchar(20) DEFAULT NULL,
  `email` varchar(20) DEFAULT NULL,
  `nickname` varchar(20) DEFAULT NULL,
  `password` varchar(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1

```

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package pkgModel;
import java.sql.ResultSet;
import java.sql.SQLException;
/**
 *
 * @author Dargy Lobo
 */
public class clsDAOFacturacion {
    private clsConnection connFact;
    private clsFacturacion fact;

    public clsDAOFacturacion() {
        connFact = new clsConnection();
        fact = new clsFacturacion();
    };
}

    public ResultSet select() {
        try {
            String SQL = "SELECT * FROM facturacion";
            return getConnFact().processSQL(SQL);
        } catch (Exception e) {
            System.out.println("Error de result set " + e);
        }
        return null;
    }

    public ResultSet select(int id) {
        try {
            String SQL = "SELECT * FROM facturacion WHERE
facturacionID='" + getFact().getFacturacionID() + "'";
            ResultSet rs = connFact.processSQL(SQL);
            if (rs.next()) {
                return rs;
            }
        } catch (Exception e) {
            System.out.println("Error de select " + e);
        }
        return null;
    }

    public ResultSet search(int num){
        String SQL;
        SQL="SELECT * FROM facturacion WHERE facturacionID='"
+num+ "'";
        ResultSet rss=getConnFact().processSQL(SQL);
        return rss;
    }

```

```

    }

    public boolean insert_Factturacion() {
        if (this.select(getFact().getFacturacionID()) == null)
        {
            return this.insert();
        } else {
            return this.update();
        }
    }

    public boolean insert() {
        String SQL = "INSERT INTO facturacion VALUES ('" +
getFact().getFacturacionID()+
                                "',''" +
getFact().getEmpleadosID() + "',''" + getFact().getProductoID()
+ "',''" + getFact().getCantidadProducto() + "',''" +
getFact().getSubtotal() + "')";
        if (getConnFact().processIUDSQL(SQL) == 0) {
            return false;
        } else {
            return true;
        }
    }

    public boolean update() {
        String SQL = "UPDATE facturacion SET EmpleadosID = '" +
getFact().getEmpleadosID()+
                                "','', ProductoID = '" +
getFact().getProductoID() + "','', CantidadProducto = '" +
getFact().getCantidadProducto() + "','', Subtotal = '" +
getFact().getSubtotal()+
                                "'" WHERE facturacionID='" +
getFact().getFacturacionID()+
                                "'";
        if (getConnFact().processIUDSQL(SQL) == 0) {
            return false;
        } else {
            return true;
        }
    }

    public boolean delete() {
        String SQL = "DELETE from facturacion WHERE
facturacionID='" + getFact().getFacturacionID() + "'";
        if (getConnFact().processIUDSQL(SQL) == 0) {
            return false;
        } else {
            return true;
        }
    }
}

/**
 * @return the connFact
 */
public clsConnection getConnFact() {
    return connFact;
}

```

```

    /**
     * @param connFact the connFact to set
     */
    public void setConnFact(clsConnection connFact) {
        this.connFact = connFact;
    }

    /**
     * @return the fact
     */
    public clsFacturacion getFact() {
        return fact;
    }

    /**
     * @param fact the fact to set
     */
    public void setFact(clsFacturacion fact) {
        this.fact = fact;
    }
}

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package pkgController;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.DefaultListModel;
import javax.swing.JOptionPane;
import pkgModel.clsDAOFacturacion;
import pkgModel.clsFacturacion;
import pkgView.frmFacturacion;
/**
 *
 * @author Dargy Lobo
 */
public class ctrFacturacion implements ActionListener {
    clsDAOFacturacion ModeloFacturacion;
    frmFacturacion vista_fac;

    public ctrFacturacion (frmFacturacion fac) {

```



```
this.ModeloFacturacion = new clsDAOFacturacion();
this.vista_fac = fac;
this.vista_fac.getBtnClose().addActionListener(this);
this.vista_fac.getBtnNew().addActionListener(this);
this.vista_fac.getBtnSave().addActionListener(this);
this.vista_fac.getBtnBusc().addActionListener(this);
this.vista_fac.getBtnBorrar().addActionListener(this);
this.vista_fac.getBtnEdit().addActionListener(this);
this.vista_fac.getBtnTotal().addActionListener(this);

this.vista_fac.getBtnSubtotal().addActionListener(this);

this.vista_fac.getBtnexportar().addActionListener(this);

this.vista_fac.getBtngraficar().addActionListener(this);

        this.vista_fac.getListProds().addMouseListener(new
java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent
evt) {
                list(evt);
        }
});
```

```

        this.list();
    }
    public void list() {
        DefaultListModel<clsFacturacion> dlm = new
DefaultListModel();
        java.sql.ResultSet rs = ModeloFacturacion.select();
        try {
            while (rs.next()) {
                dlm.addElement(new
clsFacturacion(rs.getInt(1), rs.getInt(2), rs.getInt(3),
rs.getInt(4), rs.getDouble(5)));
            }
        } catch (Exception e) {
        }
        vista_fac.getListProds().setModel(dlm);
    }

    public void list(java.awt.event.MouseEvent evt) {
        clsFacturacion result = (clsFacturacion)
this.vista_fac.getListProds().getSelectedValue();
        if (result instanceof clsFacturacion) {
            vista_fac.getTxtID().setText("") +
result.getFacturacionID());
            vista_fac.getTxtIDEmpleado().setText("") +
result.getEmpleadosID());
            vista_fac.getTxtPrecioUnit().setText("") +
result.getProductoID());
            vista_fac.getTxtCantidad().setText("") +
result.getCantidadProducto());
            vista_fac.getTxtsubtotal().setText("") +
result.getSubtotal());
        }
    }

    @Override
    public void actionPerformed(ActionEvent asd) {
        //throw new UnsupportedOperationException("Not
supported yet."); //To change body of generated methods, choose
Tools | Templates.
        switch (asd.getActionCommand()) {
            case "Actualizar": {
                vista_fac.getTxtID().setText("");
                vista_fac.getTxtIDEmpleado().setText("");
                vista_fac.getTxtPrecioUnit().setText("");
                vista_fac.getTxtCantidad().setText("");
                vista_fac.getTxtsubtotal().setText("");
                break;
            }

            case "Crear": {

```

```
ModeloFacturacion.getEmple().setFacturacionID(Integer.parseInt(vista_fac.getTxtID().getText()));
```

```
ModeloFacturacion.getEmple().setEmpleadosID(vista_fac.getTxtIDEmpleado().getText());
```

```
ModeloFacturacion.getEmple().setProductoID(Double.parseDouble(vista_fac.getTxtPrecioUnit().getText()));
```

```
ModeloFacturacion.getEmple().setCantidadProducto(vista_fac.getTxtCantidad().getText());
```

```
        if (ModeloFacturacion.insert()) {
            JOptionPane.showMessageDialog(vista_fac,
"Nuevo producto agregado");
            this.list();
        } else {
            JOptionPane.showMessageDialog(vista_fac,
"No se puedo agregar producto");
        }
        break;
    }
    case "Editar": {
```

```
ModeloFacturacion.getEmple().setFacturacionID(Integer.parseInt(vista_fac.getTxtID().getText()));
```

```
ModeloFacturacion.getEmple().setEmpleadosID(vista_fac.getTxtIDEmpleado().getText());
```

```
ModeloFacturacion.getEmple().setProductoID(Double.parseDouble(vista_fac.getTxtPrecioUnit().getText()));
```

```
ModeloFacturacion.getEmple().setCantidadProducto(vista_fac.getTxtCantidad().getText());
```

```
        if (ModeloFacturacion.update()) {
            JOptionPane.showMessageDialog(vista_fac,
"Producto actualizado");
        } else {
            JOptionPane.showMessageDialog(vista_fac,
"Producto no actualizado");
        }
        this.list();
        break;
    }
    case "Cerrar": {
        this.vista_fac.dispose();

        break;
    }
```

```

        }
        case "Buscar": {

ModeloFacturacion.getFact().setFacturacionID(Integer.parseInt(
vista_fac.getTxtBusc().getText()));
        java.sql.ResultSet resultado;

resultado=ModeloFacturacion.search(Integer.parseInt(vista_fac.
getTxtBusc().getText()));
        try {
            if (resultado.next()){

vista_fac.getTxtID().setText(resultado.getString(1));

vista_fac.getTxtIDempleado().setText(resultado.getString(2));

vista_fac.getTxtPrecioUnit().setText(resultado.getString(3));

vista_fac.getTxtCantidad().setText(resultado.getString(4));
            }
        } catch (SQLException ex) {
            System.out.println("error");
            System.out.println("error buscando"+ex);

Logger.getLogger(ctrFacturacion.class.getName()).log(Level.SEVERE,
null, ex);
        }
        break;
    }
    case "Borrar": {

ModeloFacturacion.getFact().setFacturacionID(Integer.parseInt(
vista_fac.getTxtBusc().getText()));
        ModeloFacturacion.delete();
        this.list();
        break;
    }

}

package pkgView.graph;

import javafx.application.Application;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.chart.PieChart;
import javafx.stage.Stage;
import pkgModel.clsDAOFactura;

/**

```

```

*
* @author Dargy Lobo
*/
public class graphFactura extends Application {

    @Override
    public void start(Stage stage) {
        Scene scene = new Scene(new Group());
        stage.setTitle("Grafica %Cantidad Productos");
        stage.setWidth(500);
        stage.setHeight(500);

        java.util.ArrayList list=new
clsDAOFactura().statistic();
        ObservableList<PieChart.Data> pieChartData =
FXCollections.observableArrayList(list);
        final PieChart chart = new
PieChart(pieChartData);
        chart.setTitle("Productos %Cantidad");

        ((Group) scene.getRoot()).getChildren().add(chart);
        stage.setScene(scene);
        stage.show();
    }

    public static void init(String[] args) {
        launch(args);
    }

}

/*
 * To change this license header, choose License Headers
in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
*/
package pkgModel.export;

import java.io.File;
import java.io.FileOutputStream;
import java.sql.ResultSet;
import javax.swing.JOptionPane;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
/**

```

```

*
* @author Dargy Lobo
*/
public class ExcelWriterFactura {
    XSSFWorkbook workbook = new XSSFWorkbook();
    XSSFSheet sheet = workbook.createSheet("Lista de
    Productos Facturas");
    public void write(ResultSet rs){
        try {
            Object[][] data=new Object[6][];
            int rowCount=0;

            while(rs.next()){
                data[rowCount]=new Object[100];
                data[0][0]="Facturacion ID";
                data[0][1]="Ciente ID";
                data[0][2]="ID Producto";
                data[0][3]="otracosa";
                data[0][4]="Cantidad de Producto";
                data[0][5]="Subtotal";
                data[rowCount][0]=rs.getString(1);
                data[rowCount][1]=rs.getString(2);
                data[rowCount][2]=rs.getString(3);
                data[rowCount][3]=rs.getString(4);
                data[rowCount][4]=rs.getString(5);
                data[rowCount][5]=rs.getString(6);
                rowCount++;
            }
            rowCount=0;
            for(Object[] reg: data){
                Row row = sheet.createRow(++rowCount);
                int columnCount=0;

                if(reg!=null){
                    for (Object field: reg){
                        Cell cell =row.createCell(++columnCount);
                        cell.setCellValue((String) field);
                    }
                }
            }
            // File initial_file =new
            File("Facturacion.xlsx");

            // FileOutputStream outputStream =new
            FileOutputStream(initial_file);
            // workbook.write(outputStream);
            DateTimeFormatter dtf =
            DateTimeFormatter.ofPattern("yyyy_MM_dd HH-mm-ss");
            FileOutputStream file = new
            FileOutputStream("facturacion_fecha_"+dtf.format(LocalDat
            eTime.now()).toString()+"-hora.xls");

```

```
        workbook.write(file);
        file.close();
        JOptionPane.showMessageDialog(null,
"Exportado");
    } catch (Exception e) {
    }
}
}
```