

Evidencia 1151076_AE1_CD1_I1

Práctica No. 5 -Tablas

- **Descripción del Problema**

Una empresa premia a sus clientes cada que le compran 10 artículos (independientemente del producto), por lo que le ha pedido que le muestre, en una tabla, la información del cliente (nombre y apellidos).

- Esta información se encuentra en el archivo (**cliente.dat**) en el formato:
No_cliente, Nombre ,Primer Apellido, Segundo Apellido
- Los datos de los productos se encuentran en el archivo **productos.dat** en el siguiente formato:
No_producto, Nombre Producto
- Los datos de las compras se encuentran en el archivo **compras_cliente_producto.dat** en el siguiente formato:
No_cliente, No_producto, Cantidad
- La empresa le ha pedido **mostrar una tabla** con la siguiente información:

variables y parámetros involucrados en resolver los requerimientos de la práctica

Cliente	Producto 1	Producto 2	Producto 3	Faltante
Nombre y Apellidos	Cantidad comprada	Cantidad comprada	Cantidad comprada	Cantidad faltante

En donde:

Cantidad faltante es la **cantidad de productos (de cualquier tipo) que le faltan para llegar a 10.**

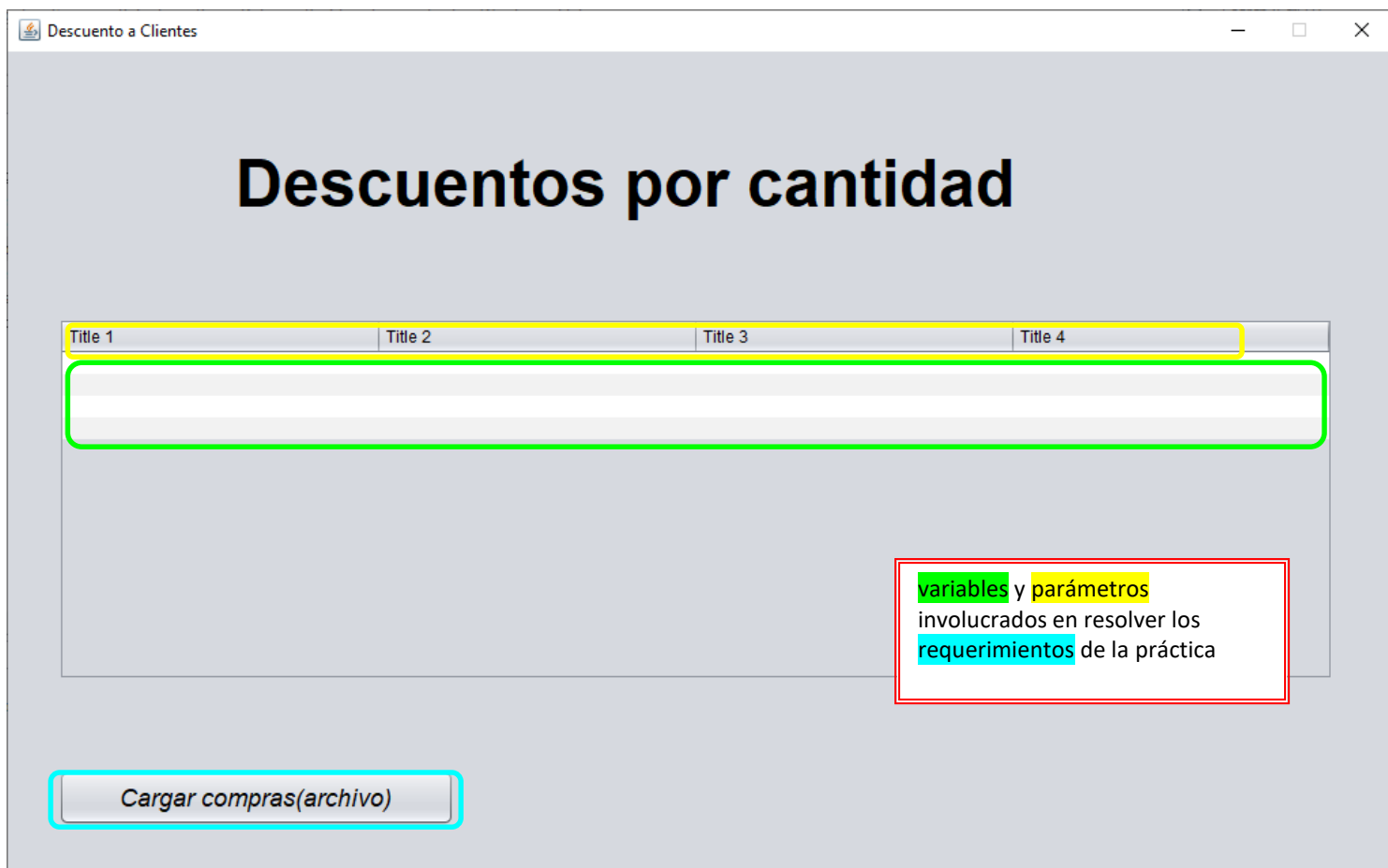
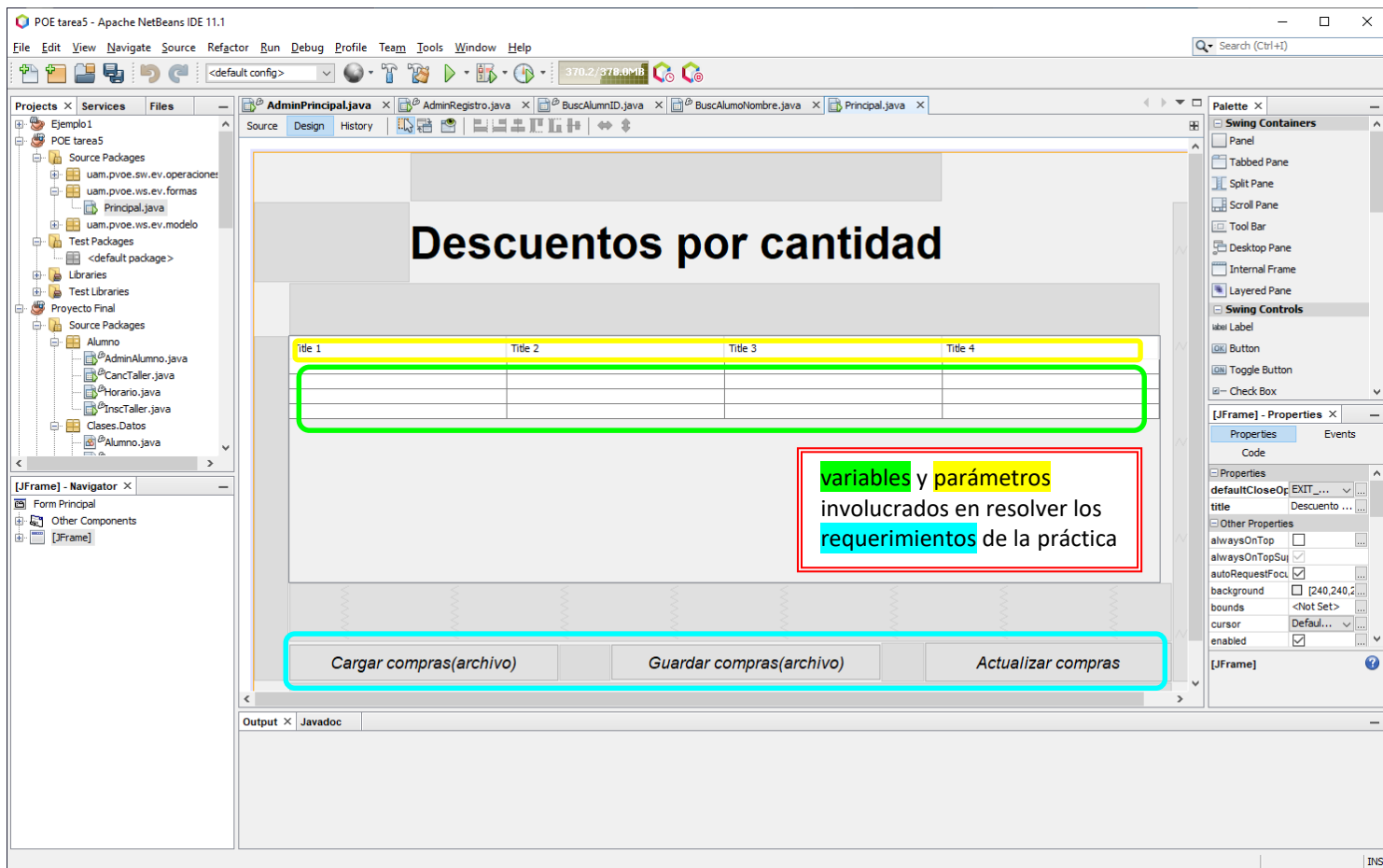
Ejemplo:

Si el cliente 1 ha comprado 2 artículos del producto 1, 3 artículos del producto 2 y 1 del producto 3, la tabla debería mostrar:

Cliente	Producto 1	Producto 2	Producto 3	Faltante
Datos del Cliente 1	2	3	1	4

En **la tabla debe ser posible ajustar los valores de la cantidad comprada**, se deberá **tener un botón** para realizar nuevamente el cálculo **que actualizará** la columna **Faltante**.

El **usar** una tabla de **cadenas (String)** o de **Objetos** será elección del alumno.



Descuento a Clientes

Descuentos por cantidad

variables y parámetros involucrados en resolver los requerimientos (cargar los datos de los archivos correspondientes) de la práctica

Cliente	Producto 1	Producto 2	Producto 3	Faltante
Nombre 01 Primer Apellido 01...	2	3	1	4
Nombre 02 Primer Apellido 02...	0	4	5	1
Nombre 03 Primer Apellido 03...	2	4	3	1
Nombre 04 Primer Apellido 04...	9	0	1	0
Nombre 05 Primer Apellido 05...	1	2	4	3
Nombre 06 Primer Apellido 06...	0	0	7	3
Nombre 07 Primer Apellido 07...	1	1	1	7
Nombre 08 Primer Apellido 08...	3	4	1	2
Nombre 09 Primer Apellido 09...	1	1	2	6

Cargar compras(archivo) Guardar compras(archivo) Actualizar compras

Descuento a Clientes

Descuentos por cantidad

variables y parámetros involucrados en resolver los requerimientos (ajustar los datos directamente en la tabla) de la práctica

Cliente	Producto 1	Producto 2	Producto 3	Faltante
Nombre 01 Primer Apellido 01 Segundo Apellido 01	4	1	5	4
Nombre 02 Primer Apellido 02 Segundo Apellido 02	1	3	6	1
Nombre 03 Primer Apellido 03 Segundo Apellido 03	3	2	4	1
Nombre 04 Primer Apellido 04 Segundo Apellido 04	9	0	1	0
Nombre 05 Primer Apellido 05 Segundo Apellido 05	3	5	2	3
Nombre 06 Primer Apellido 06 Segundo Apellido 06	2	6	1	3
Nombre 07 Primer Apellido 07 Segundo Apellido 07	3	2	5	7
Nombre 08 Primer Apellido 08 Segundo Apellido 08	2	5	0	2
Nombre 09 Primer Apellido 09 Segundo Apellido 09	4	1	3	6

Cargar compras(archivo) Guardar compras(archivo) Actualizar compras

Descuento a Clientes

Descuentos por cantidad

variables y parámetros involucrados en resolver los requerimientos (actualizar el cálculo de los faltantes, de acuerdo a los ajustes realizados) de la práctica

Cliente	Producto 1	Producto 2	Producto 3	Faltante
Nombre 01 Primer Apellido 01...	4	1	5	0
Nombre 02 Primer Apellido 02...	1	3	6	0
Nombre 03 Primer Apellido 03...	3	2	4	1
Nombre 04 Primer Apellido 04...	9	0	1	0
Nombre 05 Primer Apellido 05...	3	5	2	0
Nombre 06 Primer Apellido 06...	2	6	1	1
Nombre 07 Primer Apellido 07...	3	2	5	0
Nombre 08 Primer Apellido 08...	2	5	0	3
Nombre 09 Primer Apellido 09...	4	1	2	3

Cargar compras(archivo) Guardar compras(archivo) Actualizar compras

Código de la práctica

Escribir.java

```

package uam.pvoe.sw.ev.operaciones;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.LinkedList;
import java.util.StringTokenizer;
import uam.pvoe.ws.ev.modelo.Cliente;
import uam.pvoe.ws.ev.modelo.Compra;
import uam.pvoe.ws.ev.modelo.Producto;

/**
 *
 * @author [redacted]
 */
public class Escribir {

    public void escribirProducto (String nombreArchivo,
        LinkedList<Producto> listProd){
        System.out.println("Escribiendo información de productos");

        try {

```

```

FileWriter fw = new FileWriter(nombreArchivo);
//FileWriter fw = new FileWriter (nombreArchivo,true);
BufferedWriter archivoEscritura = new BufferedWriter(fw);
String almacenar;

for ( Producto x: listProd){
    almacenar= x.getNumProd()+ ","+ x.getNombre();
    archivoEscritura.write(almacenar);
    archivoEscritura.write('\n');
    archivoEscritura.flush();

}
archivoEscritura.close();

} catch (IOException e) {
    System.out.println("Error al escribir en el archivo");
    e.printStackTrace();
}
System.out.println("El archivo se guardo como:\t"+ nombreArchivo+ "\n");
}

public void escribirCliente (String nombreArchivo,
    LinkedList<Cliente> listClien){
    System.out.println("Escribiendo información de clientes");

    try {
        FileWriter fw = new FileWriter(nombreArchivo);
        //FileWriter fw = new FileWriter (nombreArchivo,true);
        BufferedWriter archivoEscritura = new BufferedWriter(fw);
        String almacenar;

        for ( Cliente x: listClien){
            almacenar= x.getNumCliente()+ ","+ x.getNombre()+ ","+
                x.getApePat()+ ","+ x.getApeMat();
            archivoEscritura.write(almacenar);
            archivoEscritura.write('\n');
            archivoEscritura.flush();

        }
        archivoEscritura.close();

    } catch (IOException e) {
        System.out.println("Error al escribir en el archivo");
        e.printStackTrace();
    }
    System.out.println("El archivo se guardo como:\t"+ nombreArchivo+ "\n");
}

public void escribirCompra (String nombreArchivo,
    LinkedList<Cliente> listClien){
    System.out.println("Escribiendo información de compras");

```

```
try {
    FileWriter fw = new FileWriter(nombreArchivo);
    //FileWriter fw = new FileWriter (nombreArchivo,true);
    BufferedWriter archivoEscritura = new BufferedWriter(fw);
    String almacenar;

    for ( Cliente x: listClien){

        for ( int i= 0; i<3; i++){
            almacenar= x.getNumCliente()+ ", "+ ( i+ 1)+ ", "+
                x.ObtenerCompra( i);
            archivoEscritura.write(almacenar);
            archivoEscritura.write('\n');
            archivoEscritura.flush();
        }

    }
    archivoEscritura.close();

} catch (IOException e) {
    System.out.println("Error al escribir en el archivo");
    e.printStackTrace();
}
System.out.println("El archivo se guardo como:\t"+ nombreArchivo+ "\n");

}

}
```

Leer.java

```
package uam.pvoe.sw.ev.operaciones;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.LinkedList;
import java.util.StringTokenizer;
import uam.pvoe.ws.ev.modelo.Cliente;
import uam.pvoe.ws.ev.modelo.Compra;
import uam.pvoe.ws.ev.modelo.Producto;

/**
 *
 * @author [REDACTED]
 */
public class Leer {

    public LinkedList<Producto> LeerProdctos( String nombreArchivo) {
        LinkedList<Producto> listaProductos= new LinkedList();
        String cadenaLeida = "";
        FileReader fr;

        try {
            fr = new FileReader( nombreArchivo);
            BufferedReader archivoLectura = new BufferedReader( fr);
            Producto produc;

            //System.out.println("Los productos son:");

            cadenaLeida = archivoLectura.readLine();
            while (cadenaLeida != null) {
                produc = new Producto();
                StringTokenizer st = new StringTokenizer(cadenaLeida, ",");

                produc.setNumProdt( Integer.parseInt(st.nextToken()));
                produc.setNombre( st.nextToken());

                //System.out.println( produc);
                listaProductos.add( produc);

                cadenaLeida = archivoLectura.readLine();
            }
            archivoLectura.close();
        } catch (FileNotFoundException e) {
            System.out.println("No se pudo encontrar el archivo");
            e.printStackTrace();
        } catch (IOException e) {
            System.out.println("No se pudo leer del archivo");
            e.printStackTrace();
        }
        return listaProductos;
    }
}
```

```
}
```

```
public LinkedList<Cliente> LeerClientes( String nombreArchivo) {  
    LinkedList<Cliente> listaClientes= new LinkedList();  
    String cadenaLeida = "";  
    FileReader fr;  
  
    try {  
        fr = new FileReader( nombreArchivo);  
        BufferedReader archivoLectura = new BufferedReader( fr);  
        Cliente clien;  
  
        //System.out.println("Los clientes son:");  
  
        cadenaLeida = archivoLectura.readLine();  
        while (cadenaLeida != null) {  
            clien = new Cliente();  
            StringTokenizer st = new StringTokenizer(cadenaLeida, ",");  
  
            clien.setNumCliente( Integer.parseInt(st.nextToken()));  
            clien.setNombre( st.nextToken());  
            clien.setApePat( st.nextToken());  
            clien.setApeMat( st.nextToken());  
  
            //System.out.println( clien);  
            listaClientes.add( clien);  
  
            cadenaLeida = archivoLectura.readLine();  
        }  
        archivoLectura.close();  
    } catch (FileNotFoundException e) {  
        System.out.println("No se pudo encontrar el archivo");  
        e.printStackTrace();  
    } catch (IOException e) {  
        System.out.println("No se pudo leer del archivo");  
        e.printStackTrace();  
    }  
    return listaClientes;  
}
```

```
public LinkedList<Compra> LeerCompras( String nombreArchivo) {  
    LinkedList<Compra> listaCompras= new LinkedList();  
    String cadenaLeida = "";  
    FileReader fr;  
  
    try {  
        fr = new FileReader( nombreArchivo);  
        BufferedReader archivoLectura = new BufferedReader( fr);  
        Compra comp;  
  
        cadenaLeida = archivoLectura.readLine();
```



```
while (cadenaLeida != null) {
    comp = new Compra();
    StringTokenizer st = new StringTokenizer(cadenaLeida, ",");

    comp.setNumCliente( Integer.parseInt(st.nextToken()));
    comp.setNumproducto( Integer.parseInt(st.nextToken()));
    comp.setCantidad( Integer.parseInt(st.nextToken()));

    listaCompras.add( comp);

    cadenaLeida = archivoLectura.readLine();
}
archivoLectura.close();
} catch (FileNotFoundException e) {
    System.out.println("No se pudo encontrar el archivo");
    e.printStackTrace();
} catch (IOException e) {
    System.out.println("No se pudo leer del archivo");
    e.printStackTrace();
}
return listaCompras;
}
}
```


Principal.java

```
/*
 * 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 uam.pvoe.ws.ev.formas;
```

```
import java.util.Vector;
import javax.swing.table.DefaultTableModel;
import uam.pvoe.sw.ev.operaciones.Escribir;
import uam.pvoe.ws.ev.modelo.Cliente;
```

```
import uam.pvoe.ws.ev.modelo.ListaClientes;
import uam.pvoe.ws.ev.modelo.ListaCompras;
import uam.pvoe.ws.ev.modelo.ListaProductos;
```

```
/**
 *
 * @author 
 */
```

```
public class Principal extends javax.swing.JFrame {
    int columnas= 5;
    ListaClientes lsClien;
    ListaProductos lsProduc;
    ListaCompras lsCompr;

    public Principal() {
        initComponents();
        Ocultar();
    }

    public void Ocultar(){
        btnActualizarCompras.setVisible(false);
        btnGuardarCompras.setVisible(false);
    }

    public void Mostrar(){
        btnActualizarCompras.setVisible(true);
        btnGuardarCompras.setVisible(true);
    }

    private void llenarTabla(){
        DefaultTableModel modelo = new DefaultTableModel(
            lsCompr.TotalClientes(),5);

        tbCompras.setModel(modelo);
    }
}
```

```

Vector columnas = new Vector();
columnas.add("Cliente");
columnas.add("Producto 1");
columnas.add("Producto 2");
columnas.add("Producto 3");
columnas.add("Faltante");

Vector filas = new Vector();
Vector fila;
int faltantes;

for( Cliente x: lsClien.getListaClientes()){
    faltantes= 10;
    fila = new Vector();

    fila.add(x.getNombre()+ " "+ x.getApePat()+ " "+ x.getApeMat());
    for ( int i=0; i< lsProduc.NumeroTotalProductos(); i++){
        fila.add( Integer.toString( x.ObtenerCompra( i)));
        faltantes-= x.ObtenerCompra( i);
        //System.out.println("\tFaltantes:\t"+ faltantes);
    }
    if (faltantes< 0) faltantes= 0;

    fila.add( Integer.toString( faltantes));
    filas.add(fila);
////////////////////////////////////cambios de int a String
}
modelo.setDataVector(filas, columnas);
tbCompras.setModel(modelo);

}

private void listaClientes(){
    lsClien= new ListaClientes();

    lsClien.LLenarListaProductos( "cliente.dat");
}

private void listaProductos(){
    lsProduc= new ListaProductos();

    lsProduc.LLenarListaProductos( "productos.dat");
}

private void listaCompras(){
    lsCompr= new ListaCompras();

    lsCompr.LLenarListaCompras( "compras_cliente_producto.dat");
    lsCompr.CompraACliente( lsClien.getListaClientes());

}

/**
 * This method is called from within the constructor to initialize the form.

```

```

* WARNING: Do NOT modify this code. The content of this method is always
* regenerated by the Form Editor.
*/
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
private void initComponents() {

    spTablaCompras = new javax.swing.JScrollPane();
    tbCompras = new javax.swing.JTable();
    lbTitulo = new javax.swing.JLabel();
    btnCargaCompras = new javax.swing.JToggleButton();
    btnActualizarCompras = new javax.swing.JToggleButton();
    btnGuardarCompras = new javax.swing.JToggleButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Descuento a Clientes");
    setMinimumSize(new java.awt.Dimension(500, 600));
    setResizable(false);

    tbCompras.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null}
        },
        new String [] {
            "Title 1", "Title 2", "Title 3", "Title 4"
        }
    ));
    spTablaCompras.setViewportViewView(tbCompras);

    lbTitulo.setFont(new java.awt.Font("Dialog", 1, 48)); // NOI18N
    lbTitulo.setText("Descuentos por cantidad");

    btnCargaCompras.setFont(new java.awt.Font("Dialog", 2, 18)); // NOI18N
    btnCargaCompras.setText("Cargar compras(archivo)");
    btnCargaCompras.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnCargaComprasActionPerformed(evt);
        }
    });

    btnActualizarCompras.setFont(new java.awt.Font("Dialog", 2, 18)); // NOI18N
    btnActualizarCompras.setText("Actualizar compras");
    btnActualizarCompras.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnActualizarComprasActionPerformed(evt);
        }
    });

    btnGuardarCompras.setFont(new java.awt.Font("Dialog", 2, 18)); // NOI18N
    btnGuardarCompras.setText("Guardar compras(archivo)");
    btnGuardarCompras.addActionListener(new java.awt.event.ActionListener() {

```

```

    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnGuardarComprasActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(167, 167, 167)
                    .addComponent(lbTitulo))
                .addGroup(layout.createSequentialGroup()
                    .addGap(37, 37, 37)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(btnCargaCompras, javax.swing.GroupLayout.PREFERRED_SIZE, 291,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(55, 55, 55)
                            .addComponent(btnGuardarCompras, javax.swing.GroupLayout.PREFERRED_SIZE, 291,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(46, 46, 46)
                            .addComponent(btnActualizarCompras, javax.swing.GroupLayout.PREFERRED_SIZE, 267,
javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addComponent(spTablaCompras, javax.swing.GroupLayout.PREFERRED_SIZE, 937,
javax.swing.GroupLayout.PREFERRED_SIZE))))
                    .addGap(31, Short.MAX_VALUE))
            .addContainerGap());
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(52, 52, 52)
            .addComponent(lbTitulo, javax.swing.GroupLayout.PREFERRED_SIZE, 88,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(56, 56, 56)
            .addComponent(spTablaCompras, javax.swing.GroupLayout.PREFERRED_SIZE, 266,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 63, Short.MAX_VALUE)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(btnCargaCompras, javax.swing.GroupLayout.PREFERRED_SIZE, 41,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(btnActualizarCompras, javax.swing.GroupLayout.PREFERRED_SIZE, 44,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(btnGuardarCompras, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(35, 35, 35))
            .addContainerGap());

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

```

```

private void btnCargaComprasActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_btnCargaComprasActionPerformed
    listaClientes();
    listaProductos();
    listaCompras();
    llenarTabla();
    Mostrar();
} //GEN-LAST:event_btnCargaComprasActionPerformed

private void btnActualizarComprasActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_btnActualizarComprasActionPerformed
    DefaultTableModel model=
        (DefaultTableModel) tbCompras.getModel();
    int compraActualizada;

    for( Cliente x: lsClien.getListClientes()){

        for ( int i=1; i<= lsProduc.NumeroTotalProductos(); i++){
            String valor = (String)tbCompras.getValueAt(
                (x.getNumCliente()-1), i);

            if( valor!=null){
                //System.out.println("Cliente " + x.getNumCliente()+
                //"", producto( " + i + " )" + ":\t" + valor);

                compraActualizada= Integer.parseInt(valor);
                x.CargarCompra(i-1, compraActualizada);
            }

        }

        llenarTabla();
    }

} //GEN-LAST:event_btnActualizarComprasActionPerformed

private void btnGuardarComprasActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_btnGuardarComprasActionPerformed
    Escribir escr= new Escribir();

    escr.escribirCliente( "cliente_v2.dat",
        lsClien.getListClientes());

    escr.escribirProducto( "productos_v2.dat",
        lsProduc.getListProductos());

    escr.escribirCompra( "compras_cliente_producto_v2.dat",
        lsClien.getListClientes());

} //GEN-LAST:event_btnGuardarComprasActionPerformed

/**
 * @param args the command line arguments
 */

```

```

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

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

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JToggleButton btnActualizarCompras;
private javax.swing.JToggleButton btnCargaCompras;
private javax.swing.JToggleButton btnGuardarCompras;
private javax.swing.JLabel lbTitulo;
private javax.swing.JScrollPane spTablaCompras;
private javax.swing.JTable tbCompras;
// End of variables declaration//GEN-END:variables
}

```

Cliente.java

```
package uam.pvoe.ws.ev.modelo;

/**
 *
 * @author [REDACTED]
 */
public class Cliente {
    private int numCliente;
    private String nombre;
    private String apePat;
    private String apeMat;
    private int compras[];

    public Cliente( int numCliente, String nombre,
        String apePat, String apeMat){
        this.numCliente= numCliente;
        this.nombre= nombre;
        this.apePat= apePat;
        this.apeMat= apeMat;
    }

    public Cliente( ){

    }

    /**
     * @return the numCliente
     */
    public int getNumCliente() {
        return numCliente;
    }

    /**
     * @param numCliente the numCliente to set
     */
    public void setNumCliente(int numCliente) {
        this.numCliente = numCliente;
    }

    /**
     * @return the nombre
     */
    public String getNombre() {
        return nombre;
    }

    /**
     * @param nombre the nombre to set
     */
}
```



```

public void setNombre(String nombre) {
    this.nombre = nombre;
}

/**
 * @return the apePat
 */
public String getApePat() {
    return apePat;
}

/**
 * @param apePat the apePat to set
 */
public void setApePat(String apePat) {
    this.apePat = apePat;
}

/**
 * @return the apeMat
 */
public String getApeMat() {
    return apeMat;
}

/**
 * @param apeMat the apeMat to set
 */
public void setApeMat(String apeMat) {
    this.apeMat = apeMat;
}

public int[] getCompras() {
    return compras;
}

public void setCompras(int[] compras) {
    this.compras= compras;
}

public void InicializaCompra( int numTotalProductos){
    setCompras(new int[numTotalProductos]);
}

public void CargarCompra( int index, int cantidad){
    getCompras()[index]= cantidad;
}

public int ObtenerCompra( int index){
    return getCompras()[index];
}

@Override
public String toString(){

```

```
String mensaje = "";
mensaje = "( "+ getNumCliente()+ " )Cliente:\t"+ getNombre()+ " "+ getApePat()+
    " "+ getApeMat());
return mensaje;
}
}
```

Compra.java

```
package uam.pvoe.ws.ev.modelo;

import java.util.Comparator;

/**
 *
 * @author [REDACTED]
 */
public class Compra implements Comparator< Compra>{
    private int numCliente;
    private int numproducto;
    private int cantidad;

    public Compra(){
        numCliente= 0;
        numproducto= 0;
        cantidad= 0;
    }

    /**
     * @return the numCliente
     */
    public int getNumCliente() {
        return numCliente;
    }

    /**
     * @param numCliente the numCliente to set
     */
    public void setNumCliente(int numCliente) {
        this.numCliente = numCliente;
    }

    /**
     * @return the numproducto
     */
    public int getNumproducto() {
        return numproducto;
    }

    /**
     * @param numproducto the numproducto to set
     */
    public void setNumproducto(int numproducto) {
        this.numproducto = numproducto;
    }

    /**
     * @return the cantidad
     */
}
```

```

*/
public int getCantidad() {
    return cantidad;
}

/**
 * @param cantidad the cantidad to set
 */
public void setCantidad(int cantidad) {
    this.cantidad = cantidad;
}

public int numeroClientes(){
    int clientesFinales= 0;

    return clientesFinales;
}

@Override
public String toString(){
    String mensaje = "";
    mensaje = "( "+ numCliente+ " )Producto:\t"+ numproducto+
        "\n\tCantidad:\t"+ cantidad;
    return mensaje;
}

@Override
public int compare(Compra o1, Compra o2) {
    return o1.getNumCliente() - o2.getNumCliente();
}
}

```

ListaClientes.java

```
package uam.pvoe.ws.ev.modelo;

import java.util.LinkedList;
import uam.pvoe.sw.ev.operaciones.Leer;

/**
 *
 * @author [REDACTED]
 */
public class ListaClientes {
    private LinkedList<Cliente> listaClientes;

    public ListaClientes(){
        listaClientes= new LinkedList();
    }

    public void LLenarListaProductos (String nombreArchivo){
        Leer ls= new Leer();

        setListaClientes(ls.LeerClientes( nombreArchivo));
        //System.out.println( this.toString());
    }

    /**
     * @return the listaClientes
     */
    public LinkedList<Cliente> getListaClientes() {
        return listaClientes;
    }

    /**
     * @param listaClientes the listaClientes to set
     */
    public void setListaClientes(LinkedList<Cliente> listaClientes) {
        this.listaClientes = listaClientes;
    }

    @Override
    public String toString(){
        String mensaje= "";
        for ( Cliente x : getListaClientes()){
            mensaje+= x.toString()+ "\n";
        }

        return mensaje;
    }
}
```

ListaCompras.java

```
package uam.pvoe.ws.ev.modelo;

import java.util.ArrayList;
import java.util.LinkedList;
import uam.pvoe.sw.ev.operaciones.Leer;

/**
 *
 * @author [REDACTED]
 */
public class ListaCompras {
    private LinkedList<Compra> listaCompras;
    private ArrayList<Integer> listaClientesCompra;

    public ListaCompras(){
        listaCompras= new LinkedList();
    }

    public ListaCompras( LinkedList<Compra> listaCompras){
        this.listaCompras= listaCompras;
    }

    /**
     * @return the listaCompras
     */
    public LinkedList<Compra> getListaCompras() {
        return listaCompras;
    }

    /**
     * @param listaCompras the listaCompras to set
     */
    public void setListaCompras(LinkedList<Compra> listaCompras) {
        this.listaCompras = listaCompras;
    }

    /**
     * @return the listaClientesCompra
     */
    public ArrayList<Integer> getListaClientesCompra() {
        return listaClientesCompra;
    }

    /**
     * @param listaClientesCompra the listaClientesCompra to set
     */
    public void setListaClientesCompra(ArrayList<Integer> listaClientesCompra) {
        this.listaClientesCompra = listaClientesCompra;
    }
}
```

```

public void LLenarListaCompras ( String nombreArchivo){
    Leer ls= new Leer();

    setListaCompras(ls.LeerCompras( nombreArchivo));
    //System.out.println( this.toString());

}

public int TotalClientes( ){
    int clientesFinales= 1;
    Compra compraCompa= getListaCompras().getFirst();
    setListaClientesCompra(new ArrayList());

    for ( Compra x: getListaCompras()){
        if ( compraCompa.compare(compraCompa, x)!= 0){
            clientesFinales++;
            compraCompa= x;

        }

    }

    return clientesFinales;

}

public void CompraACliente( LinkedList<Cliente> listaC){
    Compra compraCompa=
        getListaCompras().getFirst(),
        compraX=
        getListaCompras().getFirst();
    listaClientesCompra=
        new ArrayList();
    int productosTotales=
        TotalClientes( );

    for( Cliente x: listaC){
        x.InicializaCompra( productosTotales);
        /*
        if ( compraCompa.getNumCliente()== x.getNumCliente()){
            x.CargarCompra(compraCompa.getNumproducto()-1,
                compraCompa.getCantidad());
            System.out.println( (compraCompa.getNumproducto()-1)+ " "+
                compraCompa.getCantidad());
        }*/

    }

    listaClientesCompra.add( compraCompa.getNumCliente());

    for ( Compra x: getListaCompras()){
        if ( compraCompa.compare(compraCompa, x)!= 0){
            compraCompa= x;

```

```

        listaClientesCompra.add( compraCompa.getNumCliente());
    }
    compraX= x;

    for( Cliente y: listaC){
        if ( compraX.getNumCliente()== y.getNumCliente()){
            //System.out.println("\tcliente "+ compraX.getNumCliente());

            y.CargarCompra(compraX.getNumproducto()-1,
                compraX.getCantidad());
            //System.out.println( compraX.getNumproducto()+ " "+
            //    compraX.getCantidad());
        }

    }

}

}

}

@Override
public String toString(){
    String mensaje = "";
    for ( Compra x : getListaCompras()){
        mensaje+= x.toString()+"\n";
    }

    return mensaje;
}

}

```


ListaProductos.java

```
package uam.pvoe.ws.ev.modelo;

import java.util.LinkedList;
import uam.pvoe.sw.ev.operaciones.Leer;
/**
 *
 * @author [REDACTED]
 */
public class ListaProductos {
    private LinkedList<Producto> listaProductos;

    public ListaProductos(){
        listaProductos= new LinkedList();
    }

    public void LLenarListaProductos ( String nombreArchivo){
        Leer ls= new Leer();

        setListaProductos(ls.LeerProdctos( nombreArchivo));
        //System.out.println( this.toString());
    }

    public int NumeroTotalProductos(){
        return listaProductos.size();
    }

    /**
     * @return the listaProductos
     */
    public LinkedList<Producto> getListaProductos() {
        return listaProductos;
    }

    /**
     * @param listaProductos the listaProductos to set
     */
    public void setListaProductos(LinkedList<Producto> listaProductos) {
        this.listaProductos = listaProductos;
    }

    @Override
    public String toString(){
        String mensaje = "";
        for ( Producto x : getListaProductos()){
            mensaje+= x.toString()+ "\n";
        }

        return mensaje;
    }
}
```

Producto.java

```
package uam.pvoe.ws.ev.modelo;

/**
 *
 * @author [REDACTED]
 */
public class Producto {
    private int numProd;
    private int cantidad;
    private String nombre;

    public Producto( int numProd, String nombre){
        this.cantidad= 0;
    }

    public Producto(){
        this.cantidad= 0;
    }

    /**
     * @return the numProd
     */
    public int getNumProd() {
        return numProd;
    }

    /**
     * @param numProd the numProd to set
     */
    public void setNumProd(int numProd) {
        this.numProd = numProd;
    }

    /**
     * @return the nombre
     */
    public String getNombre() {
        return nombre;
    }

    /**
     * @param nombre the nombre to set
     */
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    /**
     * @return the cantidad
     */
}
```

```
*/
public int getCantidad() {
    return cantidad;
}

/**
 * @param cantidad the cantidad to set
 */
public void setCantidad(int cantidad) {
    this.cantidad = cantidad;
}

@Override
public String toString(){
    String mensaje = "";
    mensaje = "( "+ numProdt+ " )Producto:\t"+ nombre+ "\n\tCantidad:\t"+
        cantidad;
    return mensaje;
}
}
```