PROYECTO FINAL

Utilizando la modalidad de Arreglo de Objetos, se solicita almacenar la información correspondiente a 10 artículos.

La estructura del registro está definida por:

TIPO DE DATO	CAMPO
Cadena	Código del artículo
Cadena	Descripción del artículo
Carácter	Impuesto a pagar (‘0’: 0%, ‘7’: 7%, ‘1’: 10%)
Entero	Existencia
Flotante	Costo
Flotante	Precio

Utilice los siguientes datos para presentar la ejecución:

CÓDIGO	DESCRIPCIÓN	IMPUESTO	CANTIDAD EXISTENTE	PRECIO UNITARIO	PRECIO DE VENTA
ABC000	REFRIGERADORA	'7'	1	495.95f	570.34f
XYZ100	MARCADORES PARA TABLEROS	'0'	25	38.95f	44.79f
QWE200	PAÑALES DESECHABLES	'0'	40	10.95f	12.59f
RST300	CERVEZA PANAMA	'1'	35	15.00f	20.00f
DEF400	ABANICO DE PEDESTAL	'7'	15	35.95f	41.34f
JKL500	CUADERNOS DOBLE RAYA	'0'	12	25.95f	29.84f
FGH600	MESA DE PING PONG	'7'	5	75.85f	87.22f
HIJ700	FRESAS CONGELADAS	'0'	50	12.95f	14.89f
BCX800	MESA REDONDAS	'7'	10	125.85f	144.72f
VNX900	LÁPICES DE COLORES	'0'	25	12.50f	14.37f

Luego de finalizado con el almacenamiento de los datos, realice un proceso para llevar a cabo la **venta** de uno o más artículos, por cliente.

El programa debe presentar siempre la opción de confirmación para saber si desea continuar con otro cliente o finalizar.

Por cada cliente que va a realizar un compra, el programa debe pedir: nombre del cliente, el código de artículo que desea comprar y la cantidad comprada.
Esta sección finaliza cuando se escriba un código de artículo igual a cero (“0”).

Formato de salida del reporte a presentar, por cada cliente:

```
                                UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
                                  CENTRO REGIONAL DE CHIRIQUÍ

                                LICENCIATURA EN DESARROLLO DE SOFTWARE
                                        EXAMEN SEMESTRAL

                        ESTUDIANTE: XXXXXXXXXXXXXXXXXXXX    PROFESOR: EDUARDO BEITIA

                                            REPORTE DE VENTAS

NOMBRE DEL CLIENTE: XXXXXXXXXXXXXXXXXXX                    FECHA DE VENTA: DIA/MES/AÑO


Código    Descripción                    Precio de Venta          Cantidad      Valor
XXXXXX    XXXXXXXXXXXXXXX                    999.99               99         9,999.99
  ↕         ↕                               ↕                    ↕           ↕

  XX      Xxxxxxxxxxxxxxxx                  999.99               99         99,999.99
                                                                            ----------
                                                                Sub Total    9,999.99
                                                                            ----------
                                                                      7%       999.99
                                                                     10%       999.99
                                                                            ----------
                                                          Total de impuestos    9999.99
                                                                            ==========
                                                                TOTAL A PAGAR     9999.99
```

Utilice los cuadros de diálogo (JOptionPane) para:

- Introducción de datos por teclado
- Mostrar el Reporte de Ventas por cada cliente
- SOLICITAR PASSWORD EN JAVA al inicio del programa, tomando en cuenta que solo tiene derecho a tres intentos. Si en la tercera vez no coincide la contraseña, no puede utilizar el sistema y se termina la ejecución. Recuerde proteger con asteriscos la introducción de la clave.

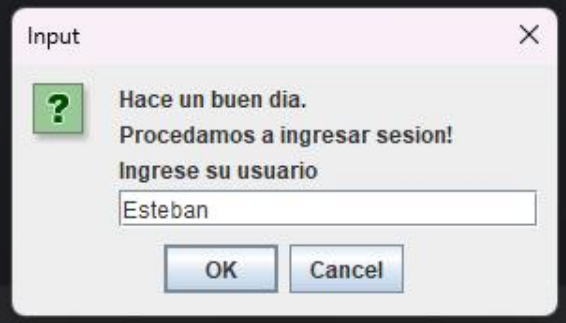
NOTA:

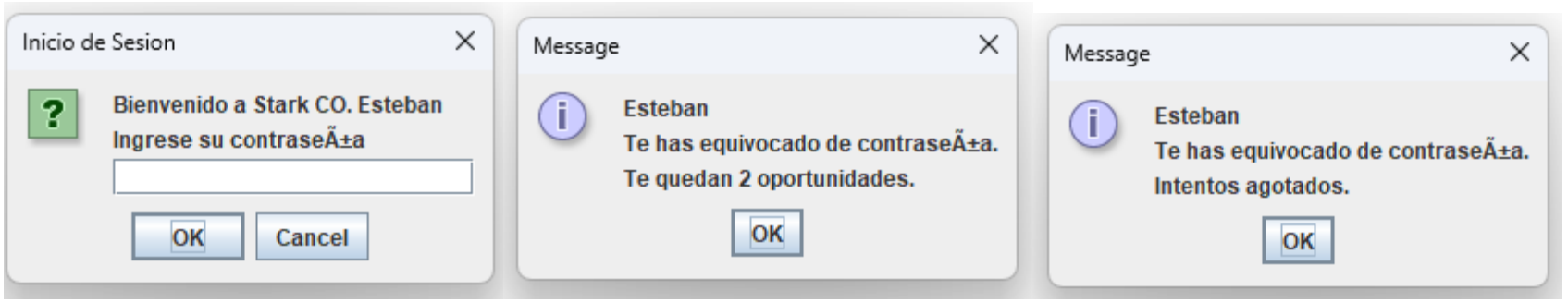
- La fecha de la venta debe ser igual a la fecha del sistema de su computadora.

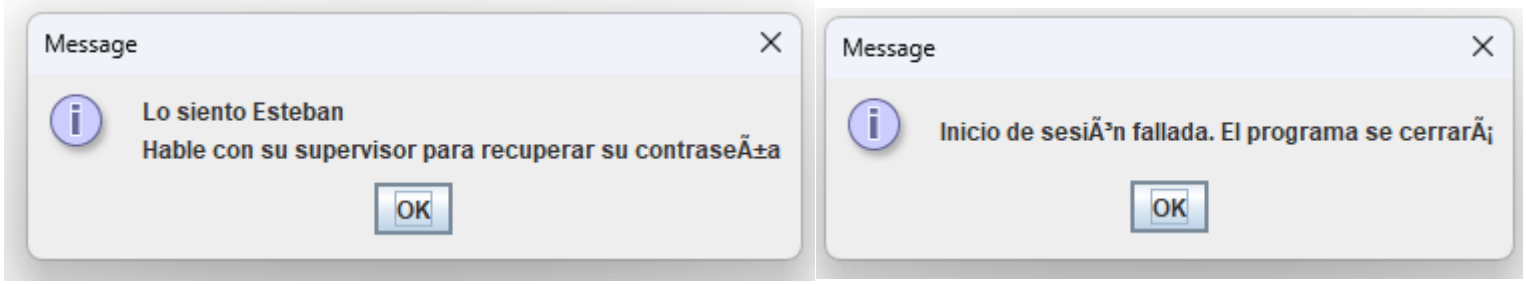
Es importante tomar en cuenta 2 situaciones que se presentan:

1. Cuando el cliente hace una compra se debe verificar si hay en existencia. De existir, se ejecuta la venta y se añade a la factura. Si no hay en existencia, se debe desplegar el siguiente mensaje: "No hay en existencia" y se continúa con el próximo artículo.
2. Cada vez que hace una venta, debe disminuir la cantidad en existencia de acuerdo a la cantidad despachada. Esta disminución en la cantidad se debe reflejar en el campo **existencia** en el arreglo de objeto.

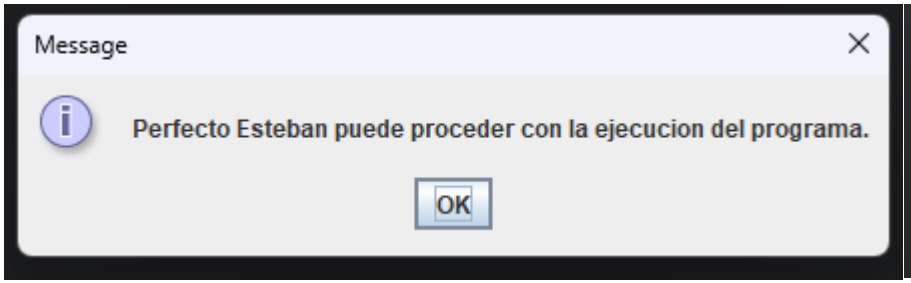
```
1 usage   Laifsyn *
public Main() {
    if (!clsPasswordAntonioNg.esperar_inicio_sesion()){
        JOptionPane.showMessageDialog( parentComponent: null,  message: "Inicio de sesión fallada. El programa se cerrará");
        System.exit( status: 0);
    }
    System.out.println("Hola. Acabamos de leer. Inicio de sesión existosa");
}

}
```

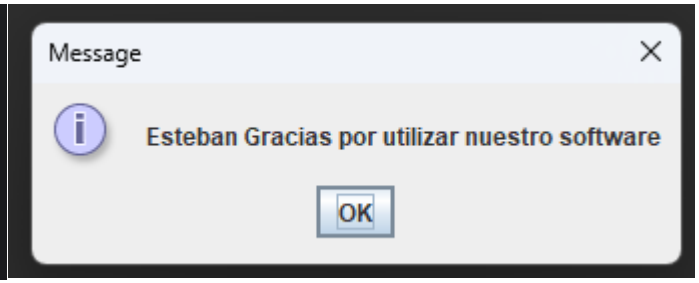


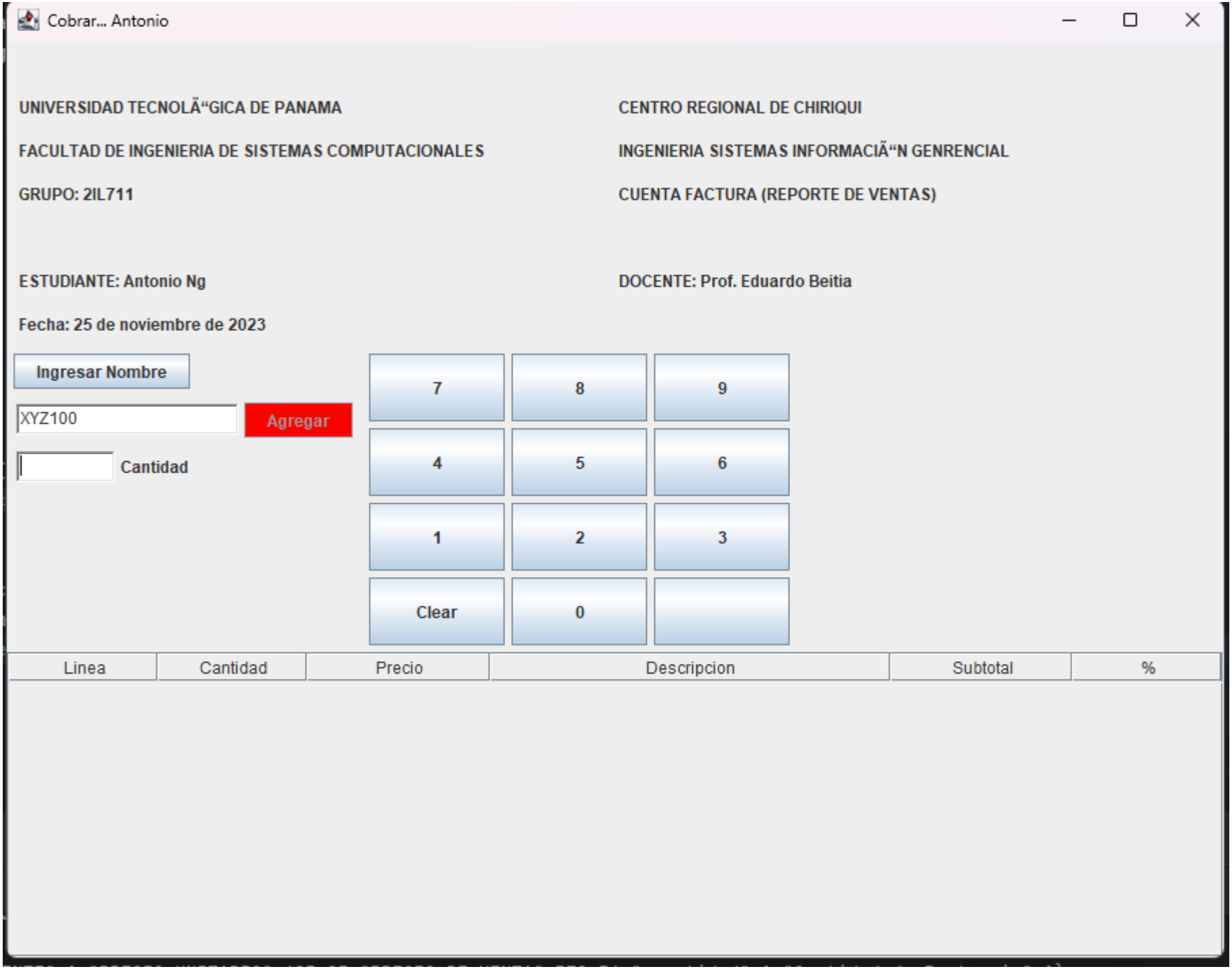


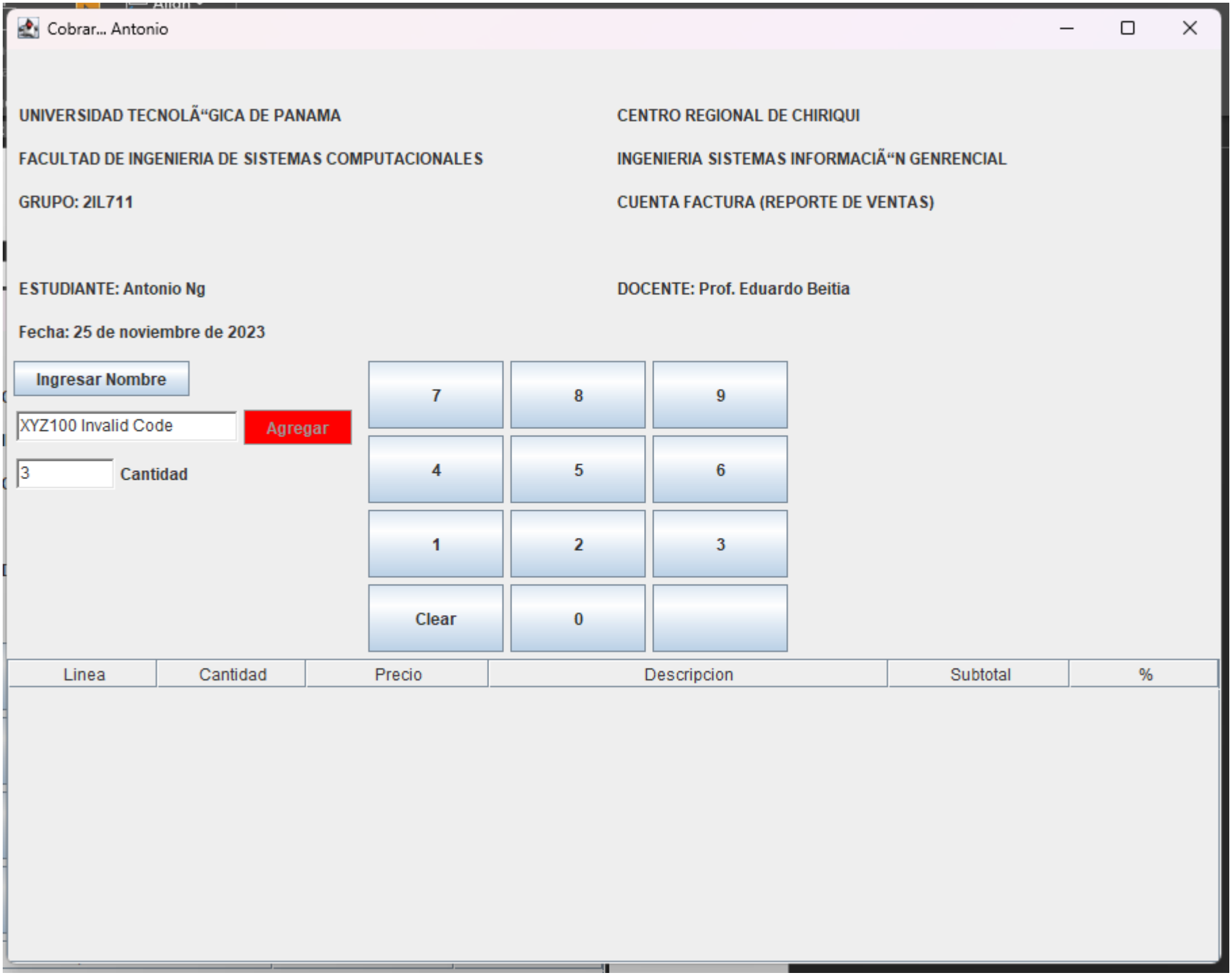


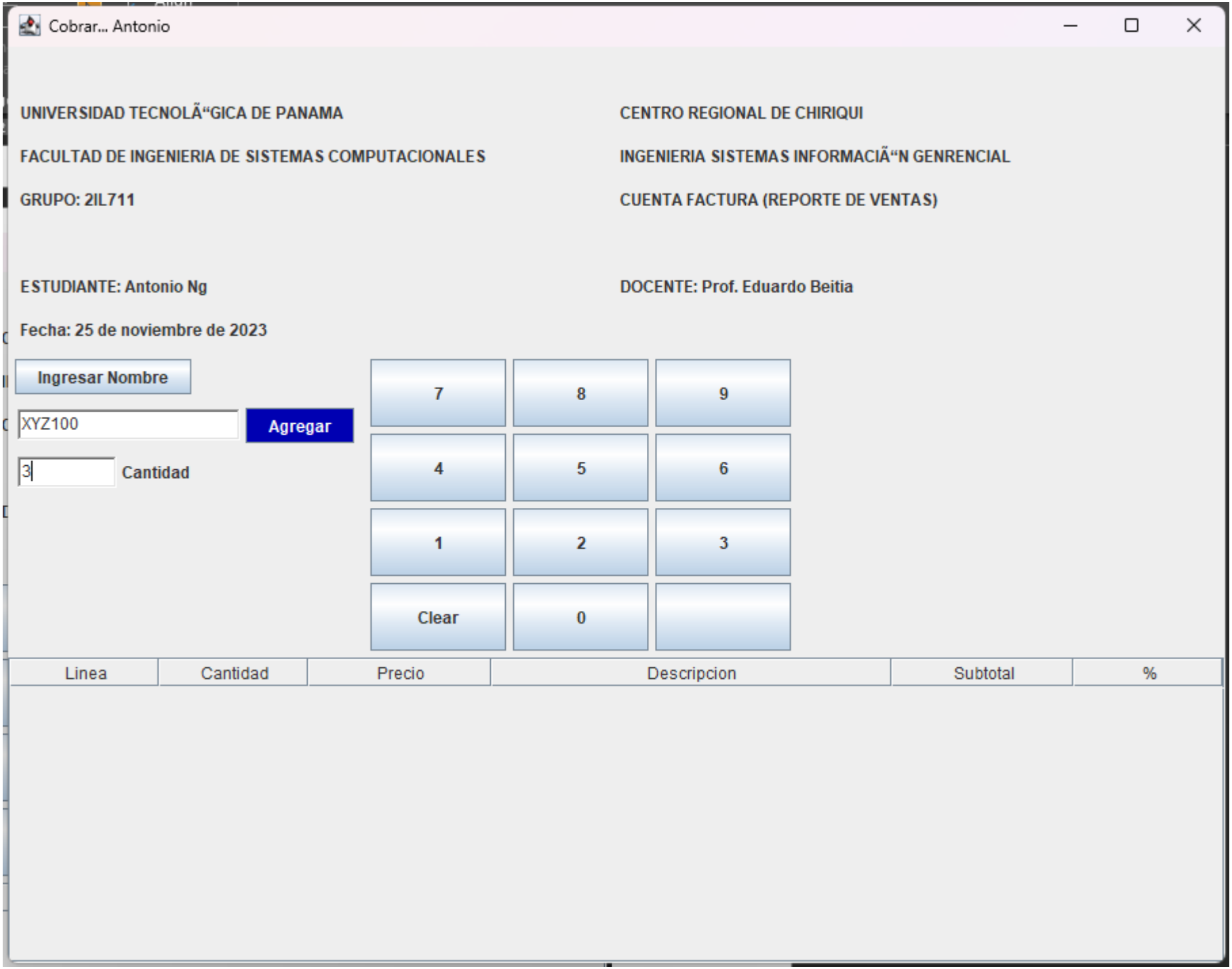
Si se ingresa la contraseña correcta:

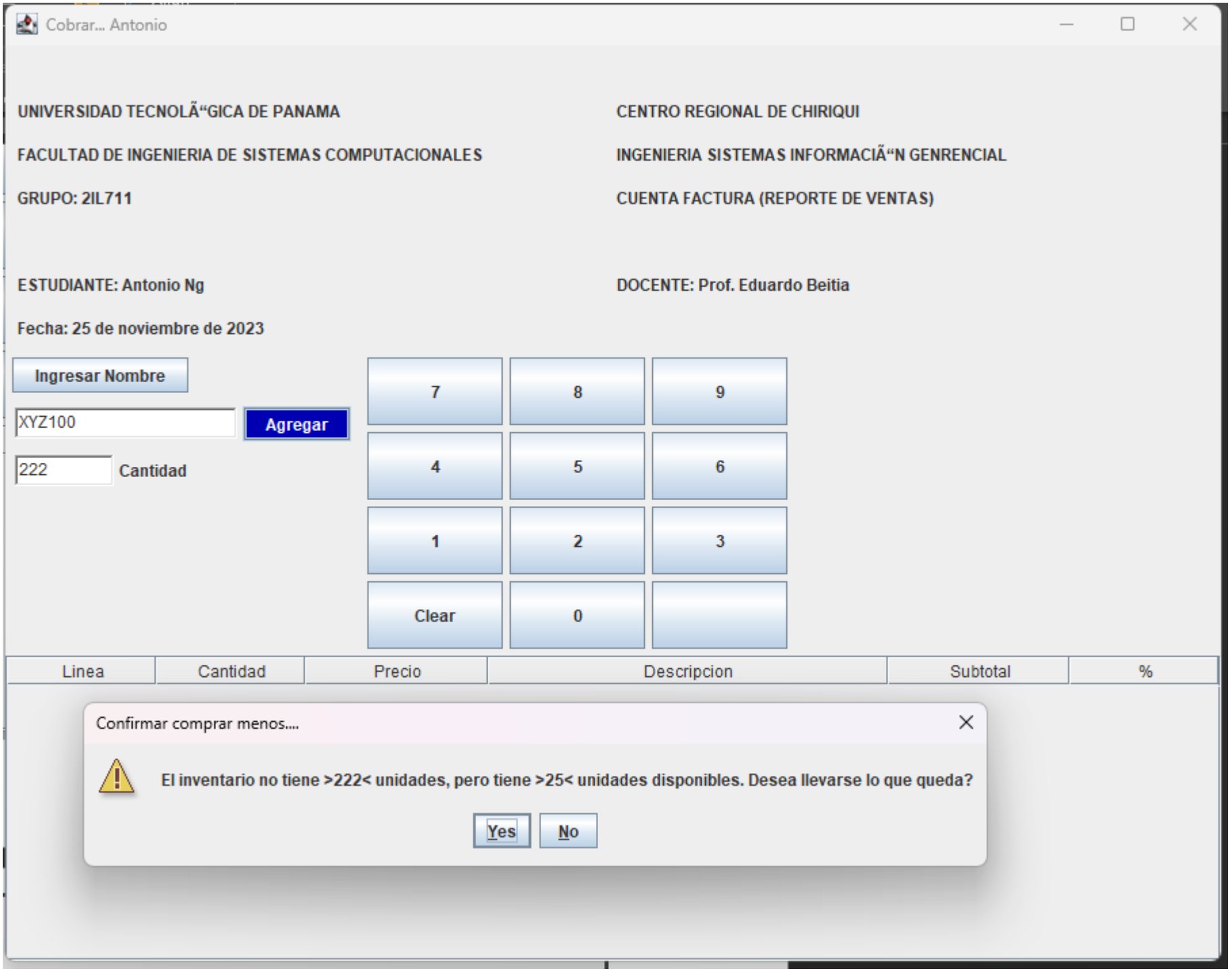

Message
Perfecto Esteban puede proceder con la ejecucion del programa.
OK

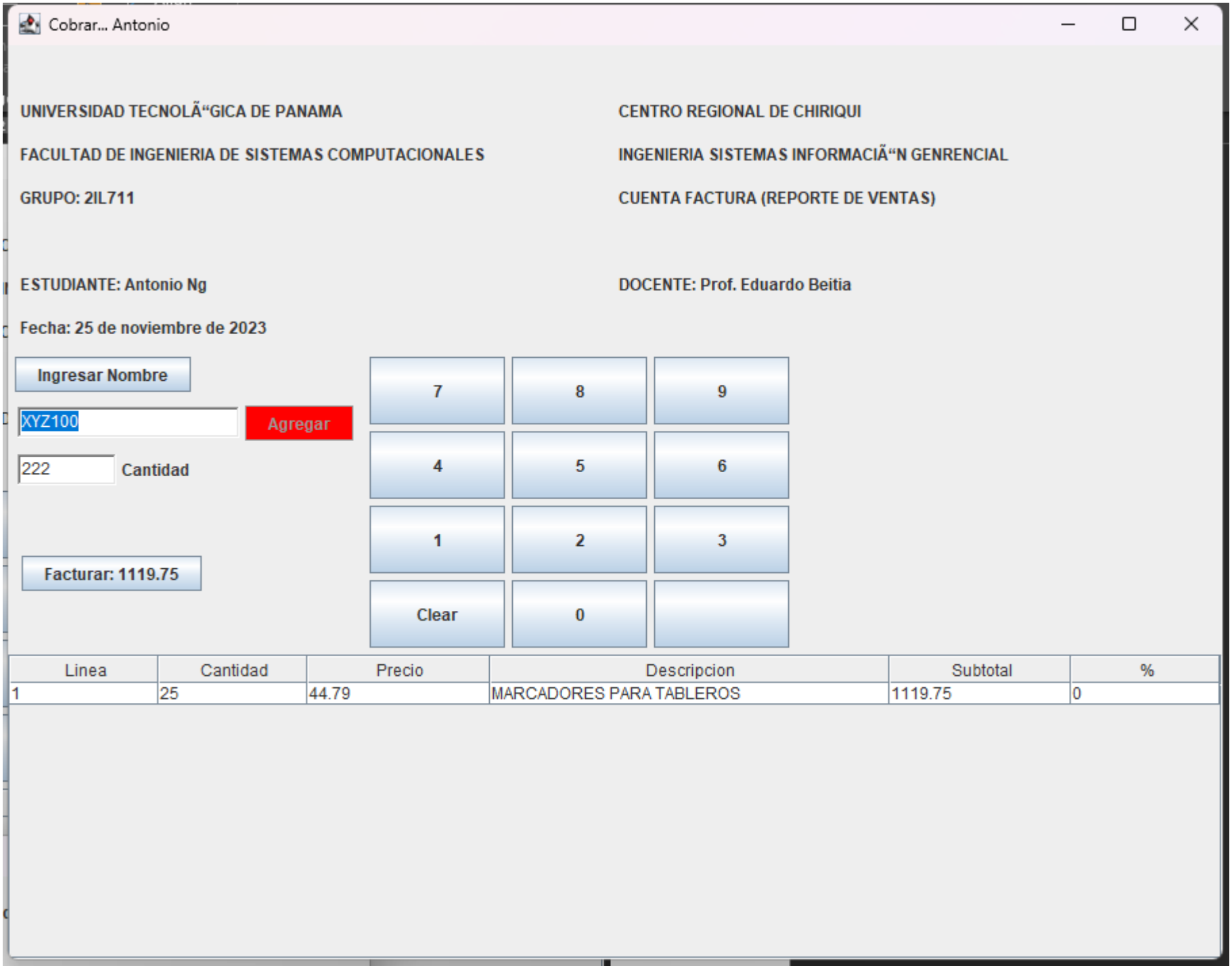

Message
Esteban Gracias por utilizar nuestro software
OK

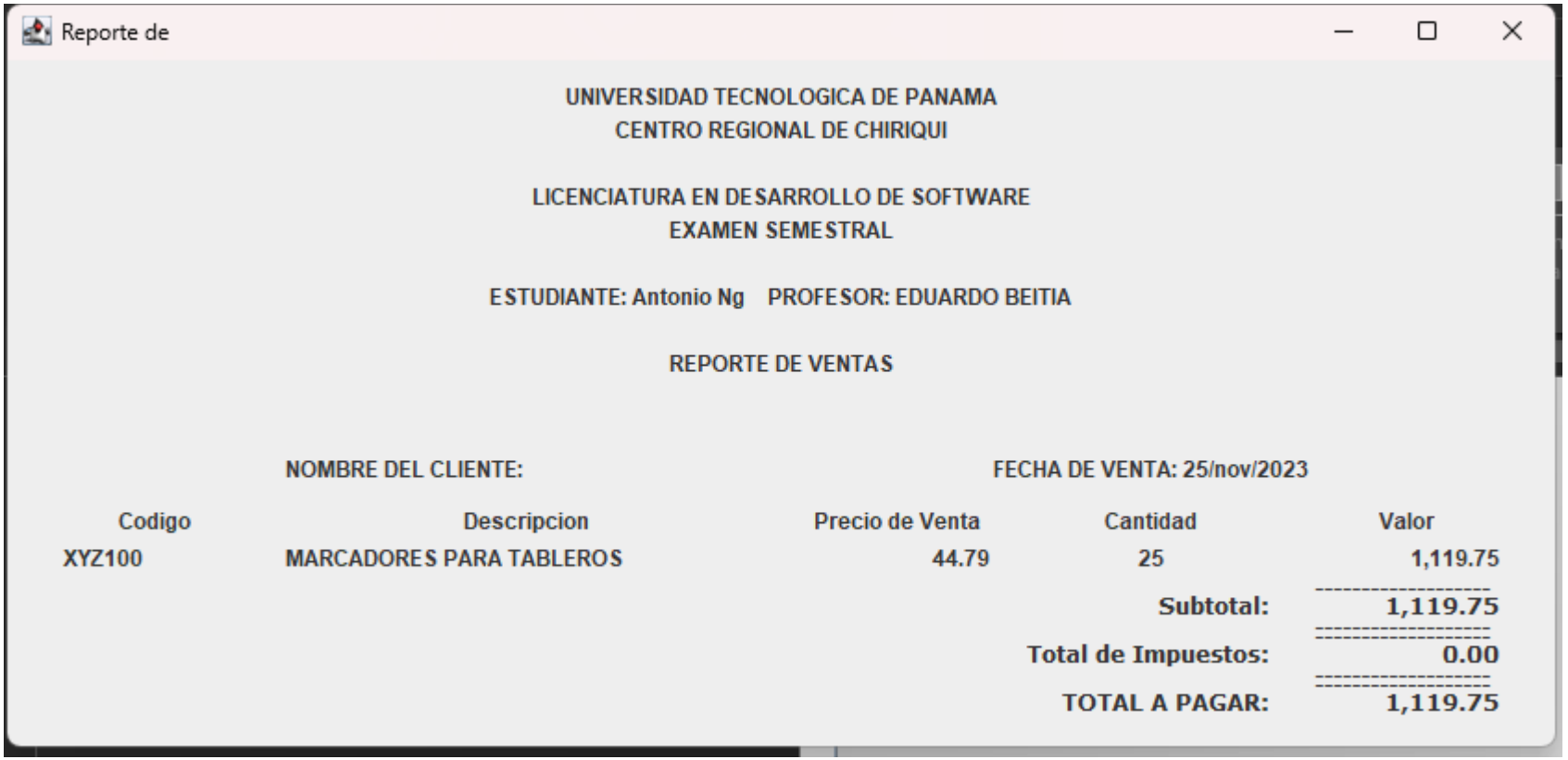

Cobrar... Antonio
UNIVERSIDAD TECNOLÃ“GICA DE PANAMA
CENTRO REGIONAL DE CHIRIQUI
FACULTAD DE INGENIERIA DE SISTEMAS COMPUTACIONALES
INGENIERIA SISTEMAS INFORMACIÃ“N GENRENCIAL
GRUPO: 2IL711
CUENTA FACTURA (REPORTE DE VENTAS)
ESTUDIANTE: Antonio Ng
DOCENTE: Prof. Eduardo Beitia
Fecha: 25 de noviembre de 2023
Ingresar Nombre
XYZ100
Agregar
Cantidad
7
8
9
4
5
6
1
2
3
Clear
0
Linea
Cantidad
Precio
Descripcion
Subtotal
%

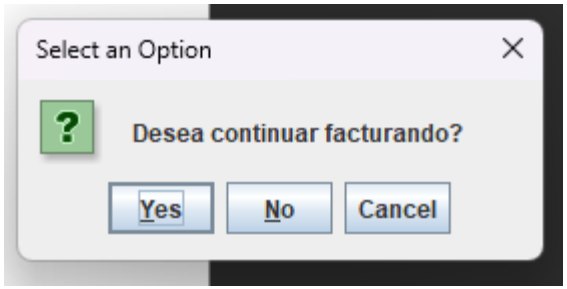

Cobrar... Antonio
UNIVERSIDAD TECNOLÃ“GICA DE PANAMA
CENTRO REGIONAL DE CHIRIQUI
FACULTAD DE INGENIERIA DE SISTEMAS COMPUTACIONALES
INGENIERIA SISTEMAS INFORMACIÃ“N GENRENCIAL
GRUPO: 2IL711
CUENTA FACTURA (REPORTE DE VENTAS)
ESTUDIANTE: Antonio Ng
DOCENTE: Prof. Eduardo Beitia
Fecha: 25 de noviembre de 2023
Ingresar Nombre
XYZ100 Invalid Code
Agregar
3
Cantidad
7
8
9
4
5
6
1
2
3
Clear
0
Linea
Cantidad
Precio
Descripcion
Subtotal
%


Cobrar... Antonio
UNIVERSIDAD TECNOLÃ“GICA DE PANAMA
CENTRO REGIONAL DE CHIRIQUI
FACULTAD DE INGENIERIA DE SISTEMAS COMPUTACIONALES
INGENIERIA SISTEMAS INFORMACIÃ“N GENRENCIAL
GRUPO: 2IL711
CUENTA FACTURA (REPORTE DE VENTAS)
ESTUDIANTE: Antonio Ng
DOCENTE: Prof. Eduardo Beitia
Fecha: 25 de noviembre de 2023
Ingresar Nombre
XYZ100
Agregar
3
Cantidad
7
8
9
4
5
6
1
2
3
Clear
0
Linea
Cantidad
Precio
Descripcion
Subtotal
%

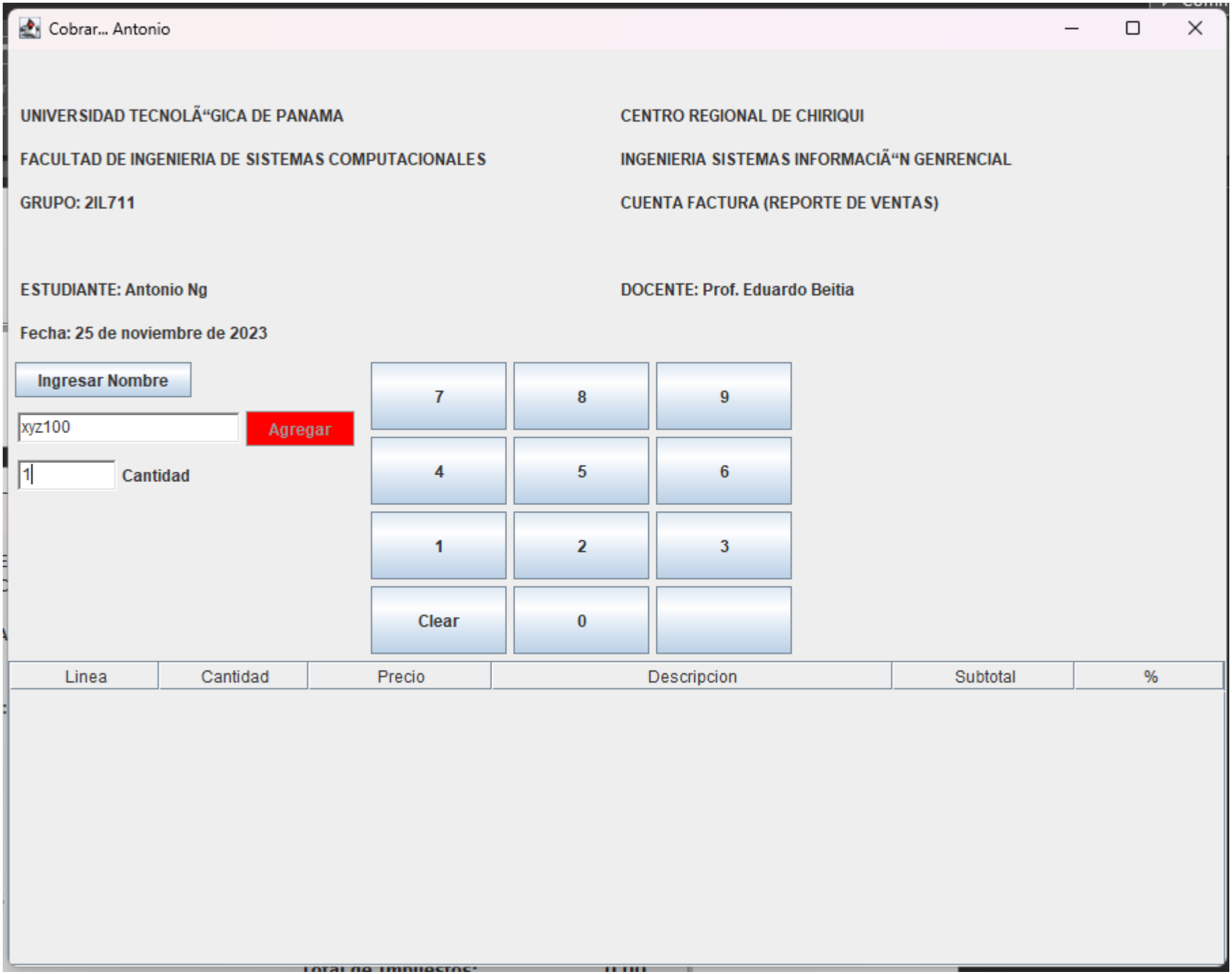

Cobrar... Antonio
UNIVERSIDAD TECNOLÃ“GICA DE PANAMA
CENTRO REGIONAL DE CHIRIQUI
FACULTAD DE INGENIERIA DE SISTEMAS COMPUTACIONALES
INGENIERIA SISTEMAS INFORMACIÃ“N GENRENCIAL
GRUPO: 2IL711
CUENTA FACTURA (REPORTE DE VENTAS)
ESTUDIANTE: Antonio Ng
DOCENTE: Prof. Eduardo Beitia
Fecha: 25 de noviembre de 2023
Ingresar Nombre
XYZ100
Agregar
222
Cantidad
7
8
9
4
5
6
1
2
3
Clear
0
Linea
Cantidad
Precio
Descripcion
Subtotal
%
Confirmar comprar menos....
El inventario no tiene >222< unidades, pero tiene >25< unidades disponibles. Desea llevarse lo que queda?
Yes
No

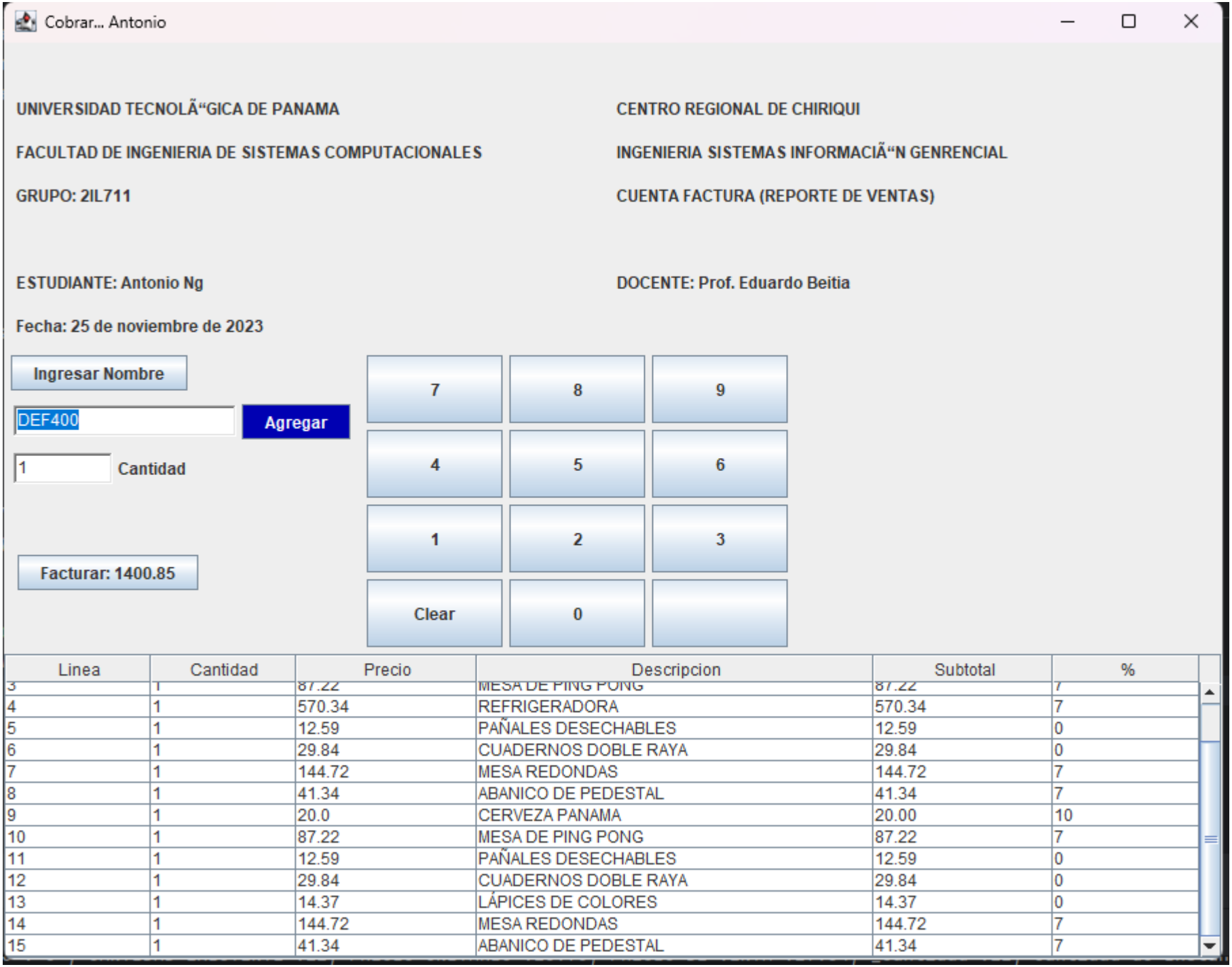

Cobrar... Antonio
UNIVERSIDAD TECNOLÃ“GICA DE PANAMA
CENTRO REGIONAL DE CHIRIQUI
FACULTAD DE INGENIERIA DE SISTEMAS COMPUTACIONALES
INGENIERIA SISTEMAS INFORMACIÃ“N GENRENCIAL
GRUPO: 2IL711
CUENTA FACTURA (REPORTE DE VENTAS)
ESTUDIANTE: Antonio Ng
DOCENTE: Prof. Eduardo Beitia
Fecha: 25 de noviembre de 2023
Ingresar Nombre
XYZ100
Agregar
222
Cantidad
Facturar: 1119.75
7
8
9
4
5
6
1
2
3
Clear
0
Linea
Cantidad
Precio
Descripcion
Subtotal
%
1
25
44.79
MARCADORES PARA TABLEROS
1119.75
0





Intentamos comprar del mismo producto….


Cobrar... Antonio
UNIVERSIDAD TECNOLÃ“GICA DE PANAMA
CENTRO REGIONAL DE CHIRIQUI
FACULTAD DE INGENIERIA DE SISTEMAS COMPUTACIONALES
INGENIERIA SISTEMAS INFORMACIÃ“N GENRENCIAL
GRUPO: 2IL711
CUENTA FACTURA (REPORTE DE VENTAS)
ESTUDIANTE: Antonio Ng
DOCENTE: Prof. Eduardo Beitia
Fecha: 25 de noviembre de 2023
Ingresar Nombre
xyz100
Agregar
1
Cantidad
7
8
9
4
5
6
1
2
3
Clear
0
Linea
Cantidad
Precio
Descripcion
Subtotal
%



Linea	Cantidad	Precio	Descripcion	Subtotal	%
3	1	87.22	MESA DE PING PONG	87.22	7
4	1	570.34	REFRIGERADORA	570.34	7
5	1	12.59	PAÑALES DESECHABLES	12.59	0
6	1	29.84	CUADERNOS DOBLE RAYA	29.84	0
7	1	144.72	MESA REDONDAS	144.72	7
8	1	41.34	ABANICO DE PEDESTAL	41.34	7
9	1	20.0	CERVEZA PANAMA	20.00	10
10	1	87.22	MESA DE PING PONG	87.22	7
11	1	12.59	PAÑALES DESECHABLES	12.59	0
12	1	29.84	CUADERNOS DOBLE RAYA	29.84	0
13	1	14.37	LÁPICES DE COLORES	14.37	0
14	1	144.72	MESA REDONDAS	144.72	7
15	1	41.34	ABANICO DE PEDESTAL	41.34	7

Cuando se ingresan muchos artículos

Reporte de

UNIVERSIDAD TECNOLOGICA DE PANAMA
CENTRO REGIONAL DE CHIRIQUI

LICENCIATURA EN DESARROLLO DE SOFTWARE
EXAMEN SEMESTRAL

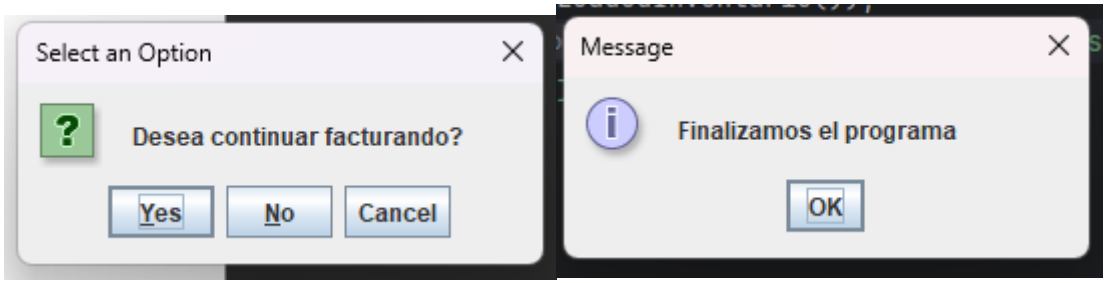
ESTUDIANTE: Antonio Ng PROFESOR: EDUARDO BEITIA

REPORTE DE VENTAS

NOMBRE DEL CLIENTE: FECHA DE VENTA: 25/nov/2023

Codigo	Descripcion	Precio de Venta	Cantidad	Valor
BCX800	MESA REDONDAS	144.72	1	144.72
RST300	CERVEZA PANAMA	20.00	1	20.00
FGH600	MESA DE PING PONG	87.22	1	87.22
ABC000	REFRIGERADORA	570.34	1	570.34
QWE200	PAÑALES DESECHABLES	12.59	1	12.59
JKL500	CUADERNOS DOBLE RAYA	29.84	1	29.84
BCX800	MESA REDONDAS	144.72	1	144.72
DEF400	ABANICO DE PEDESTAL	41.34	1	41.34
RST300	CERVEZA PANAMA	20.00	1	20.00
FGH600	MESA DE PING PONG	87.22	1	87.22
QWE200	PAÑALES DESECHABLES	12.59	1	12.59
JKL500	CUADERNOS DOBLE RAYA	29.84	1	29.84
VNX900	LÁPICES DE COLORES	14.37	1	14.37
BCX800	MESA REDONDAS	144.72	1	144.72
DEF400	ABANICO DE PEDESTAL	41.34	1	41.34
RST300	CERVEZA PANAMA	20.00	33	660.00
JKL500	CUADERNOS DOBLE RAYA	29.84	10	298.40

Subtotal:	**2,359.25**
7%	**88.31**
10%	**70.00**
Total de Impuestos:	**158.31**
TOTAL A PAGAR:	**2,517.56**


Select an Option
Desea continuar facturando?
Yes
No
Cancel
Message
Finalizamos el programa
OK

```java
/************************************
*                                   *
* utp\ac\antonio_ng\examen\Main.java *
*                                   *
************************************/
package com.utp.ac.antonio_ng.examen;

import com.utp.ac.antonio_ng.examen.pktCaja.pktProducto.clsCobrar;
import com.utp.ac.antonio_ng.examen.pktInventario.clsLoadedInventario;
import com.utp.ac.antonio_ng.examen.pktPasswordAntonioNg.clsPasswordAntonioNg;

import javax.swing.JOptionPane;

public class Main {

    public static void main(String[] args) {
        Main entry = new Main();
    }

    public Main() {
        if (!clsPasswordAntonioNg.esperar_inicio_sesion()) {
            JOptionPane.showMessageDialog(null, "Inicio de sesión fallada. El programa se cerrará");
            System.exit(0);
        }
        System.out.println("Hola. Acabamos de leer. Inicio de sesión existosa");
        clsCobrar.facturar_al_cliente(new clsLoadedInventario());
        JOptionPane.showMessageDialog(null, "Finalizamos el programa");
        System.out.println("Acabamos con la ejecución del programa.");
        System.exit(0);
    }


}


/************************************************
*                                               *
* utp\ac\antonio_ng\examen\pktCaja\clsCliente.java *
*                                               *
************************************************/
package com.utp.ac.antonio_ng.examen.pktCaja;
```

```java
import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import com.fasterxml.jackson.annotation.JsonInclude;
import com.utp.ac.antonio_ng.examen.pktPersona.clsPersona;

@JsonInclude(JsonInclude.Include.NON_NULL)
@JsonIgnoreProperties(ignoreUnknown = true)
public class clsCliente {
    public clsCliente(String nombre) {
        persona.setNombre(nombre);
    }

    clsPersona persona = new clsPersona();

    public String get_nombre() {
        return persona.getNombreCompleto();
    }
}


/************************************************
*                                               *
* utp\ac\antonio_ng\examen\pktCaja\clsFactura.java *
*                                               *
************************************************/
package com.utp.ac.antonio_ng.examen.pktCaja;

import com.utp.ac.antonio_ng.examen.pktCaja.pktProducto.clsProductoInmutable;
import com.utp.ac.antonio_ng.examen.pktInventario.clsLoadedInventario;
import io.vavr.Tuple;
import io.vavr.Tuple2;

import javax.swing.*;
import javax.swing.border.BevelBorder;
import java.awt.*;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
```

```java
import java.util.*;
import java.util.concurrent.CountDownLatch;

public class clsFactura {
    public static void main(String[] args) throws InterruptedException {
        clsLoadedInventario inventario = new clsLoadedInventario();
        ArrayList<clsProductoInmutable> productos = new ArrayList<>();
        String[] codigos = new String[]{"XYZ100", "ABC000", "QWE200", "RST300", "DEF400"};
        CountDownLatch latch = new CountDownLatch(1);
        for (String codigo : codigos)
            productos.add(inventario.fetch(codigo, 3));
        Tuple2<clsFactura, CountDownLatch> set = clsFactura.generarReporte(io.vavr.Tuple.of(new clsCliente("Nombre " +
                "de" + " Cliente"), productos), latch);
        set._2.await();
    }

    clsCliente cliente;
    ArrayList<clsProductoInmutable> productos;
    public CountDownLatch latch = new CountDownLatch(1);

    public clsFactura(Tuple2<clsCliente, ArrayList<clsProductoInmutable>> datos, CountDownLatch _latch) {
        cliente = datos._1;
        productos = datos._2;
        latch = _latch;
        generar_reporte();
    }

    public static Tuple2<clsFactura, CountDownLatch> generarReporte(Tuple2<clsCliente,
            ArrayList<clsProductoInmutable>> datos, CountDownLatch latch) {
        clsFactura reporte = new clsFactura(datos, latch);
        return Tuple.of(reporte, reporte.latch);
    }

    void generar_reporte() {
        JFrame frame = new JFrame("Reporte de " + cliente.get_nombre());
        frame.setLayout(new GridBagLayout());
        GridBagConstraints constraints = new GridBagConstraints();
        constraints.gridwidth = 5;
        constraints.gridx = 0;
        constraints.gridy = 0;
        constraints.weightx = 1;
```

```
        constraints.weighty = 1;
        acoplar_encabezado(frame, constraints);
        constraints.insets = new Insets(0, 20, 0, 20);
        constraints.gridy = 1;
        acoplar_datos_de_productos(frame, constraints);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
        frame.pack();
        frame.setMinimumSize(frame.getPreferredSize());
        frame.addKeyListener(new KeyListener() {
            @Override
            public void keyTyped(KeyEvent e) {}

            @Override
            public void keyPressed(KeyEvent e) {
                if (e.getKeyCode() == KeyEvent.VK_ESCAPE) frame.dispose();
            }

            @Override
            public void keyReleased(KeyEvent e) {}
        });
        frame.addWindowListener(new WindowListener() {
            @Override
            public void windowOpened(WindowEvent e) {}

            @Override
            public void windowClosing(WindowEvent e) {}

            @Override
            public void windowClosed(WindowEvent e) {
                latch.countDown();
            }

            @Override
            public void windowIconified(WindowEvent e) {}

            @Override
            public void windowDeiconified(WindowEvent e) {}

            @Override
            public void windowActivated(WindowEvent e) {}
```

```
        @Override
        public void windowDeactivated(WindowEvent e) {
            System.out.println("Window Deavtivated");
        }
    });
}

void acoplar_encabezado(JFrame frame, GridBagConstraints frame_constraints) {
    JPanel panel = new JPanel(new GridBagLayout());
    JPanel panel_superior = new JPanel(new GridLayout(10, 1, 0, 0));
    JPanel panel_inferior = new JPanel(new GridLayout(1, 2, 5, 5));
    panel_superior.setBorder(BorderFactory.createBevelBorder(BevelBorder.RAISED, Color.CYAN, Color.RED));
    panel_superior.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
    panel_superior.setBounds(500, 500, 200, 200);
    acoplar_JLabel("UNIVERSIDAD TECNOLOGICA DE PANAMA", panel_superior);
    acoplar_JLabel("CENTRO REGIONAL DE CHIRIQUI", panel_superior);
    acoplar_JLabel("", panel_superior);
    acoplar_JLabel("LICENCIATURA EN DESARROLLO DE SOFTWARE", panel_superior);
    acoplar_JLabel("EXAMEN SEMESTRAL", panel_superior);
    acoplar_JLabel("", panel_superior);
    acoplar_JLabel("ESTUDIANTE: Antonio Ng  \t  PROFESOR: EDUARDO BEITIA", panel_superior);
    acoplar_JLabel("", panel_superior);
    acoplar_JLabel("REPORTE DE VENTAS", panel_superior);
    acoplar_JLabel("", panel_superior);
    acoplar_JLabel("NOMBRE DEL CLIENTE: " + cliente.get_nombre(), panel_inferior);
    String fecha = LocalDateTime.now().format(DateTimeFormatter.ofPattern("dd/MMM/yyyy", Locale.forLanguageTag(
            "es-ES")));
    acoplar_JLabel("FECHA DE VENTA: " + fecha, panel_inferior);

    panel_superior.setPreferredSize(new Dimension(800, 200));
    panel_inferior.setPreferredSize(new Dimension(800, 40));
    panel.setPreferredSize(new Dimension(800, 240));
    GridBagConstraints constraints = new GridBagConstraints();
    panel.add(panel_superior, constraints);
    constraints.gridx++;
    panel.add(panel_inferior, constraints);
    frame.add(panel, frame_constraints);

}
```

```java
    void acoplar_JLabel(String texto, JPanel panel) {
        panel.add(new JLabel(texto) {{
            setHorizontalAlignment(JLabel.CENTER);
        }}, Component.CENTER_ALIGNMENT);
    }

    void acoplar_datos_de_productos(JFrame frame, GridBagConstraints frame_constraints) {
        JPanel panel = new JPanel(new GridBagLayout());

        GridBagConstraints constraints = new GridBagConstraints();
        constraints.insets = new Insets(0, 10, 0, 10);
        constraints.gridx = 0;
        constraints.gridy = 0;
        constraints.gridwidth = 1;

        acoplar_titulo(panel, constraints, new String[]{"Codigo", "Descripcion", "Precio de Venta", "Cantidad",
                "Valor"});
        HashMap<Integer, Double> totales_itbms = new HashMap<>();
        totales_itbms.put(7, 0d);
        totales_itbms.put(10, 0d);
        double subtotal = 0; // Total sin el impuesto
        for (clsProductoInmutable producto : productos) {
            String codigo = producto.get_codigo_articulo();
            String descripcion = producto.get_description();
            String venta = convertir_a_formato_display(producto.get_venta());
            String cantidad = String.valueOf(producto.get_cantidad());

            double total = producto.get_venta() * producto.get_cantidad();
            totales_itbms.computeIfPresent(producto.retrieve_itbms(), (key, value) -> {
                value += producto.get_itbms() * total;
                return value;
            });
            subtotal += total;
            String valor = convertir_a_formato_display(total);
            acoplar_fila(panel, constraints, new String[]{codigo, descripcion, venta, cantidad, valor});
        }
        double total = subtotal;
        double total_impuestos = 0d;

        // Insertar SubTotal
        insertar_separador(panel, constraints);
```

```
        acoplar_resultado(panel, constraints, new String[]{"Subtotal: ", convertir_a_formato_display(subtotal)});

        // Acumulamos el impuesto, y imprimimos cada línea de impuesto
        insertar_separador(panel, constraints);
        for (Map.Entry<Integer, Double> entry : totales_itbms.entrySet()) {
            if (Math.round(entry.getValue()) == 0) continue;
            total_impuestos += entry.getValue();
            acoplar_resultado(panel, constraints, new String[]{String.format("%d%%", entry.getKey()),
                    convertir_a_formato_display(entry.getValue())});
        }
        total += total_impuestos;
        insertar_separador(panel, constraints);
        // Insertar Total de Impuestos
        acoplar_resultado(panel, constraints, new String[]{"Total de Impuestos: ",
                convertir_a_formato_display(total_impuestos)});
        insertar_separador(panel, constraints);
        insertar_separador(panel, constraints);
        // Insertar Total
        acoplar_resultado(panel, constraints, new String[]{"TOTAL A PAGAR: ", convertir_a_formato_display(total)});
        acoplar_resultado(panel, constraints, new String[]{" ", ""});


        frame.add(panel, frame_constraints);
    }

    void acoplar_titulo(JPanel panel, GridBagConstraints constraints, String[] fila) {
        int start_x = constraints.gridx;

        constraints.anchor = GridBagConstraints.CENTER;
        for (String celda : fila) {
            JLabel label = new JLabel(celda);
            panel.add(label, constraints);
            constraints.gridx++;
        }
        constraints.gridy++;
        constraints.gridx = start_x;
    }

    void acoplar_fila(JPanel panel, GridBagConstraints constraints, String[] fila) {
        int start_x = constraints.gridx;
        for (String celda : fila) {
```

```
            int alignment_constant = switch (constraints.gridx) {
                case 0, 1 -> SwingConstants.LEFT;
                case 3 -> SwingConstants.CENTER;
                default -> SwingConstants.RIGHT;
            };
            int width = switch (constraints.gridx) {
//                case 0, 4, 3 -> 100;
                case 1 -> 260;
                default -> 100;
            };
            JLabel label = new JLabel(celda);
            label.setPreferredSize(new Dimension(width, 25));
            label.setHorizontalAlignment(alignment_constant);
            panel.add(label, constraints);
            constraints.gridx++;
        }
        constraints.gridy++;
        constraints.gridx = start_x;
    }

    void insertar_separador(JPanel panel, GridBagConstraints constraints) {
        int start_x = constraints.gridx;
        for (String celda : new String[]{"", "", "", "", "-------------------"}) {
            JLabel label = new JLabel(celda);
            label.setPreferredSize(new Dimension(100, 5));
            constraints.anchor = GridBagConstraints.EAST;
            label.setFont(label.getFont().deriveFont(16f));
            panel.add(label, constraints);
            constraints.gridx++;
        }
        constraints.gridy++;
        constraints.gridx = start_x;
    }

    void acoplar_resultado(JPanel panel, GridBagConstraints constraints, String[] fila) {
        int start_x = constraints.gridx;
        for (String celda : new String[]{"", "", "", fila[0], fila[1]}) {
            JLabel label = new JLabel(celda);
            constraints.anchor = GridBagConstraints.EAST;
            label.setFont(new Font("Verdana", Font.BOLD, 12));
            panel.add(label, constraints);
```

```
            constraints.gridx++;
        }
        constraints.gridy++;
        constraints.gridx = start_x;

    }

    String convertir_a_formato_display(double numero) {
        StringBuilder resultado_de_conversion = new StringBuilder();
        Formatter formatter = new Formatter(resultado_de_conversion, Locale.US);
        formatter.format("%(,.2f", numero);
        return resultado_de_conversion.toString();
    }
}


/*************************************************************
*                                                            *
* utp\ac\antonio_ng\examen\pktCaja\pktProducto\clsArticulo.java *
*                                                            *
*************************************************************/
package com.utp.ac.antonio_ng.examen.pktCaja.pktProducto;

import com.fasterxml.jackson.annotation.JsonIgnore;
import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import com.fasterxml.jackson.annotation.JsonInclude;
import com.fasterxml.jackson.annotation.JsonProperty;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.utp.ac.antonio_ng.examen.pktResourceFiles.FileResource;

import java.io.File;
import java.util.Arrays;
import java.util.List;

@JsonInclude(JsonInclude.Include.NON_NULL)
@JsonIgnoreProperties(ignoreUnknown = true)
public class clsArticulo {
    public static void main(String[] args) {
        generar_datos();
    }
```

```java
public static List<clsArticulo> generar_datos() {
    // Leer y deserializar los datos del archivo
    List<clsArticulo> lista_articulos = List.of();
    ObjectMapper mapper = new ObjectMapper();
    try {
        File input_stream = new FileResource().getFileFromResource("inventario.json");
        lista_articulos = Arrays.asList(mapper.treeToValue(mapper.readTree(input_stream).get("data"), clsArticulo[].class));
    } catch (Exception e) {
        System.out.println("Error leyendo los datos de \"inventario.json\".");
        System.exit(1);
    }
    lista_articulos.forEach(System.out::println);
    System.out.println("\nFin de la deserialización.....\n");

    return lista_articulos;
}

@Override
public String toString() {
    String ret = "";
    try {
        ret = new ObjectMapper().writeValueAsString(this);
    } catch (Exception e) {
        System.out.println("Falla en deserializando el objeto:\n" + e);
    }
    return ret;
}

@JsonProperty("CODIGO")
String codigo;
@JsonProperty("DESCRIPCION")
String descripcion;
@JsonProperty("IMPUESTO")
int itbms;
@JsonIgnore
public boolean EXENTO;

public int retrieve_itbms() {
    return itbms;
}
```

```java
    public void setItbms(String itbms) {
        int ret;
        switch (itbms) {
            case "1", "10" -> ret = 10;
            case "7" -> ret = 7;
            default -> ret = 0;
        }
        this.itbms = ret;
        if (this.itbms == 0) EXENTO = true;
    }

    public String getItbms() {
        String ret;
        switch (itbms) {
            case 10 -> ret = "10";
            case 7 -> ret = "7";
            default -> ret = "0";
        }
        return ret;
    }

    @JsonProperty("CANTIDAD EXISTENTE")
    int cantidad;
    @JsonProperty("PRECIO UNITARIO")
    float costo;
    @JsonProperty("PRECIO DE VENTA")
    float precio;


}


/*************************************************************
*                                                            *
* utp\ac\antonio_ng\examen\pktCaja\pktProducto\clsCobrar.java *
*                                                            *
*************************************************************/
package com.utp.ac.antonio_ng.examen.pktCaja.pktProducto;

import com.utp.ac.antonio_ng.examen.pktCaja.clsCliente;
import com.utp.ac.antonio_ng.examen.pktCaja.clsFactura;
```

```java
import com.utp.ac.antonio_ng.examen.pktInventario.clsLoadedInventario;
import io.vavr.Tuple;
import io.vavr.Tuple2;
import io.vavr.control.Try;
import io.vavr.control.Validation;

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
import java.nio.charset.StandardCharsets;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.ArrayList;
import java.util.Locale;
import java.util.Optional;
import java.util.Random;
import java.util.concurrent.CountDownLatch;

/**
 * Code Blocking Class - La finalización del i/o genera una lista de Productos
 */
public class clsCobrar {

    public static void main(String[] args) throws InterruptedException {
        System.setProperty("file.encoding", "UTF-8");
        clsLoadedInventario inventario = new clsLoadedInventario();

        System.out.println("Se debió haber generado la venta?");
        clsCobrar.facturar_al_cliente(inventario);
        inventario.print_inventario();
    }


    public static void facturar_al_cliente(clsLoadedInventario inventario) {
        CountDownLatch latch = new CountDownLatch(1);
        clsCobrar ventana_facturacion = new clsCobrar(inventario, latch);
        while (true) {
            try {
                ventana_facturacion.latch.await();
```

```
            } catch (InterruptedException e) {
                System.out.println("Unexpected Exception:" + e);
                System.exit(1);
            }
            System.out.println("Esperando reporte cierre");
            try {
                ventana_facturacion.inner_latch.await();
            } catch (InterruptedException e) {
                throw new RuntimeException(e);
            }
            if (JOptionPane.showConfirmDialog(null, "Desea continuar facturando?") != 0) {
                ventana_facturacion.frame.dispose();
                break;
            }
            ventana_facturacion.frame.dispose();
            ventana_facturacion = new clsCobrar(inventario, new CountDownLatch(1));
            System.out.println("Estado actual del inventario:...");
            inventario.print_inventario();
        }
    }

    clsFactura reporte_factura;
    clsLoadedInventario inventario;
    JButton btn_totalizar = new JButton("Facturar $0.00");
    JButton btn_ingresar_nombre;
    clsCliente cliente = new clsCliente("");
    CountDownLatch latch;
    CountDownLatch inner_latch;
    clsTablaDeArticulos tabla;
    JFrame frame = new JFrame("Cobrar... Antonio");

    clsCobrar(clsLoadedInventario inv, CountDownLatch countdown_latch) {
        inventario = inv;
        tabla = new clsTablaDeArticulos(inventario);
        latch = countdown_latch;
        Dimension dimension = new Dimension(800, 800);

        frame.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
        frame.setSize(dimension);
        frame.setLayout(new GridLayout(3, 1) {});
        acoplar_header(frame);
```

```
        acoplar_ingreso_de_datos(frame);
        acoplar_resultado_de_ingresos(frame);
//        BorderLayout e = new BorderLayout();
//        e.
        frame.setVisible(true);
        frame.pack();
        frame.addWindowListener(new WindowListener() {
            @Override
            public void windowOpened(WindowEvent e) {}

            @Override
            public void windowClosing(WindowEvent e) {}

            @Override
            public void windowClosed(WindowEvent e) {
                latch.countDown();
            }

            @Override
            public void windowIconified(WindowEvent e) {}

            @Override
            public void windowDeiconified(WindowEvent e) {}

            @Override
            public void windowActivated(WindowEvent e) {}

            @Override
            public void windowDeactivated(WindowEvent e) {}
        });
    }

    void acoplar_header(JFrame frame) {
        LocalDateTime local_date = LocalDateTime.now();
        String fecha_actual = new String(local_date.format(DateTimeFormatter.ofPattern("dd 'de' MMMM 'de' yyyy",
                Locale.forLanguageTag("es-ES"))).getBytes(), StandardCharsets.UTF_8);
        System.out.println(fecha_actual + "\n\n");
        JLabel[] labels = new JLabel[]{new JLabel(""), new JLabel(""),
                new JLabel("UNIVERSIDAD TECNOLÓGICA DE PANAMA"), new JLabel("CENTRO REGIONAL DE CHIRIQUI"),
                new JLabel("FACULTAD DE INGENIERIA DE SISTEMAS COMPUTACIONALES"),
                new JLabel("INGENIERIA SISTEMAS " + "INFORMACIÓN GENRENCIAL"), new JLabel("GRUPO: 2IL711"),
```

```
            new JLabel("CUENTA FACTURA (REPORTE DE " + "VENTAS)"), new JLabel(""), new JLabel(""), new JLabel(
            "ESTUDIANTE: Antonio Ng"), new JLabel("DOCENTE" + ": Prof. Eduardo Beitia"), new JLabel(
            "Fecha: " + fecha_actual), new JLabel("")};
    JPanel header = new JPanel(new GridLayout(labels.length / 2, 1, 5, 5));
    for (int index = 0; index < labels.length; index += 2) {
        labels[index].setHorizontalAlignment(JLabel.LEFT);
        String new_text;
        new_text = "   " + labels[index].getText();
        labels[index].setText(new_text);
        labels[index + 1].setHorizontalAlignment(JLabel.LEFT);
        header.add(labels[index]);
        header.add(labels[index + 1]);
    }
    frame.add(header);
}

void acoplar_ingreso_de_datos(JFrame frame) {
    JPanel panel = new JPanel(new GridBagLayout());
    panel.setAlignmentX(Component.LEFT_ALIGNMENT);
    GridBagConstraints constraints = new GridBagConstraints();
    constraints.gridx = 0;
    constraints.gridy = 0;
    constraints.weightx = .1;
    constraints.insets = new Insets(5, 5, 5, 5);
    constraints.anchor = GridBagConstraints.WEST;

    // Buton de Ingresar Nombre
    {
        btn_ingresar_nombre = new JButton("Ingresar Nombre");
        btn_ingresar_nombre.addActionListener(e -> {

            String initial_value = cliente.get_nombre();
            String nombre = JOptionPane.showInputDialog(null, "Ingrese su nombre", initial_value);
            JButton this_button = ((JButton) e.getSource());
            if (nombre == null || nombre.isBlank()) {
                this_button.setText("Ingresar Nombre");
                return;
            }
            cliente = new clsCliente(nombre);
            this_button.setText("Cliente: " + cliente.get_nombre());
        });
```

```
        panel.add(btn_ingresar_nombre, constraints);
    }

    // Buton Ingresar Articulo y Entrada de Datos
    ButtonAgregar btn_ingresar_articulo;
    {

        constraints.gridx = 0;
        constraints.gridy = 1;
        GridBagConstraints original_constraints = (GridBagConstraints) constraints.clone();

        JPanel sub_panel = new JPanel(new GridBagLayout());
        constraints.insets = new Insets(0, 2, 0, 2);
        constraints.gridx = 1;
        constraints.gridy = 0;
        btn_ingresar_articulo = new ButtonAgregar(inventario, tabla);
        sub_panel.add(btn_ingresar_articulo, constraints);

        // TextField Ingresar Codigo
        constraints.gridx = 0;
        TextFieldIngresarCodigo text_field = new TextFieldIngresarCodigo(btn_ingresar_articulo);
        sub_panel.add(text_field, constraints);
        constraints = original_constraints;
        panel.add(sub_panel, constraints);
    }

    // Ingresar Cantidad
    {
        constraints.gridx = 0;
        constraints.gridy = 2;
        GridBagConstraints original_constraints = (GridBagConstraints) constraints.clone();

        JPanel sub_panel = new JPanel(new GridBagLayout());
        constraints.insets = new Insets(0, 2, 0, 2);
        constraints.gridx = 1;
        constraints.gridy = 0;
        JLabel label = new JLabel("Cantidad");

        sub_panel.add(label, constraints);

        // TextField Ingresar Cantidad
```

```
            constraints.gridx = 0;
            TextFieldIngresarCantidad text_field = new TextFieldIngresarCantidad(btn_ingresar_articulo);
            sub_panel.add(text_field, constraints);
            constraints = original_constraints;
            panel.add(sub_panel, constraints);
        }


        // Facturar
        {
            constraints.gridx = 0;
            constraints.gridy = 3;

            JPanel sub_panel = new JPanel();
            JButton boton = btn_totalizar;
            boton.setVisible(false);

            boton.addActionListener(e -> {
                frame.setEnabled(false);
                System.out.println("Generamos el reporte");
                Tuple2<clsFactura, CountDownLatch> set = clsFactura.generarReporte(Tuple.of(cliente,
                        tabla.productos_en_la_tabla), new CountDownLatch(1));
                inner_latch = set._2;
                latch.countDown();
            });

            tabla.tabla.getModel().addTableModelListener(e -> {
                boton.setVisible(true);
                boton.setText(String.format("Facturar: %.2f", tabla.totalizar()));
            });
            sub_panel.add(boton);
            panel.add(sub_panel, constraints);
        }

        // Agregar Teclado Numérico
        {
            GridBagConstraints original_constraints = (GridBagConstraints) constraints.clone();
            constraints.gridx = 1;
            constraints.gridy = 0;
            constraints.gridheight = 4;
            constraints.weighty = 1;
```

```
            constraints.weightx = 11;
            constraints.gridwidth = 1;
            JPanel digits_panel = new JPanel(new GridLayout(4, 3, 5, 5));
//            digits_panel.setBackground(Color.CYAN);
            JTextArea text_area = new JTextArea();
            for (Integer index : new Integer[]{7, 8, 9, 4, 5, 6, 1, 2, 3, -1, 0, -2}) {
                NumericButton btn = new NumericButton(index, text_area);
                btn.setPreferredSize(new Dimension(100, 50));
                digits_panel.add(btn);
            }
            Dimension text_area_prefered_size = digits_panel.getPreferredSize();
            text_area_prefered_size.width = 120;
            text_area.setFont(text_area.getFont().deriveFont(16f));
            text_area.setVisible(false);
            text_area.setLineWrap(true);
            text_area.setWrapStyleWord(true);
            text_area.setPreferredSize(text_area_prefered_size);

            JPanel sub_panel = new JPanel(new GridLayout(1, 2, 5, 5));
            sub_panel.add(digits_panel);
            sub_panel.add(text_area);
            panel.add(sub_panel, constraints);
            constraints = original_constraints;
        }
        frame.add(panel);
    }

    void acoplar_resultado_de_ingresos(JFrame frame) {
        // Acoplamos la Tabla de datos
        frame.add(tabla.scroll_pane);
    }

}

class ButtonAgregar extends JButton {
    static public JButton button;
    clsLoadedInventario inventario;
    Optional<Tuple2<String, Integer>> last_valid_input = Optional.empty();

    public ButtonAgregar(clsLoadedInventario inv, clsTablaDeArticulos tabla_de_datos) {
        super("Agregar");
```

```
        inventario = inv;
        button = this;
        if (input_is_valid().isEmpty()) block_button();
        addActionListener(e -> {
            if (last_valid_input.isEmpty()) return;
            Tuple2<String, Integer> inputs = last_valid_input.get();
            Optional<clsProductoInmutable> producto = tabla_de_datos.try_insert_producto(inputs._1, inputs._2);
            if (producto.isEmpty()) return;
            tabla_de_datos.insert_producto(producto.get());
            if (TextFieldIngresarCodigo.text_field.isEmpty()) return;
            TextFieldIngresarCodigo.text_field.get().requestFocus();
            TextFieldIngresarCodigo.text_field.get().selectAll();
            if (input_is_valid().isEmpty()) block_button();
        });
    }

    /**
     * If it returns empty, it means the button should be blocked
     */
    Optional<Tuple2<String, Integer>> input_is_valid() {

        if (TextFieldIngresarCodigo.text_field.isEmpty()) return Optional.empty();
        if (TextFieldIngresarCantidad.text_field.isEmpty()) return Optional.empty();

        String codigo = TextFieldIngresarCodigo.text_field.get().getText().toUpperCase();
        String cantidad = TextFieldIngresarCantidad.text_field.get().getText().toUpperCase();
        if (inventario.query(codigo).isEmpty()) return Optional.empty();

        Try<Integer> attempt = Try.of(() -> Integer.parseInt(cantidad));
        if (attempt.isFailure()) return Optional.empty();

        Optional<Integer> query_result = inventario.query(codigo);
        if (attempt.get() <= 0 || query_result.isEmpty() || query_result.get() <= 0) return Optional.empty();

        last_valid_input = Optional.of(Tuple.of(codigo, attempt.get()));
        return last_valid_input;
    }

    void block_button() {
        setForeground(Color.BLACK);
        setBackground(Color.RED);
```

```
        setEnabled(false);
    }

    void unblock_button() {
        setEnabled(true);
        setBackground(Color.BLUE.darker());
        setForeground(Color.WHITE);
    }

}

class clsTablaDeArticulos {
    public static void main(String[] args) {
        clsTablaDeArticulos e = new clsTablaDeArticulos(new clsLoadedInventario());
        Object ignored = e.try_insert_producto("XYZ100", 700);

    }

    private static final String[] Columnas = new String[]{"Linea", "Cantidad", "Precio", "Descripcion", "Subtotal",
            "%"};
    private static final Integer[] Columnas_width = new Integer[]{5, 10, 40, 200, 40, 10};
    final clsLoadedInventario inventario;
    JScrollPane scroll_pane;
    JTable tabla = new JTable() {
        {
            DefaultTableModel model = new DefaultTableModel();
            // Agregamos los nombres de las columnas
            for (String name : Columnas)
                model.addColumn(name);

            // Actualizamos el modelo de la tabla
            setModel(model);

            // Editamos la anchura de descripción
            for (int index = 0; index < Columnas.length; index++)
                getColumnModel().getColumn(index).setPreferredWidth(Columnas_width[index]);
        }

        @Override
        public boolean isCellEditable(int row, int column) {
            return false;
```

```
    }
};

clsTablaDeArticulos(clsLoadedInventario inventario) {
    this.inventario = inventario;
    scroll_pane = new JScrollPane(tabla);
    int suma = 0;
    for (Integer numero : Columnas_width)
        suma += numero;
    scroll_pane.setPreferredSize(new Dimension(suma, 200));
}

public double totalizar() {
    if (productos_en_la_tabla.isEmpty()) return 0d;
    double ret = 0d;
    for (clsProductoInmutable producto : productos_en_la_tabla)
        ret += producto.get_cantidad() * producto.get_venta();
    return ret;
}

/**
 * Provides a Reference to the inserted product
 */
public Optional<clsProductoInmutable> try_insert_producto(String codigo, int amount) {
    Validation<Optional<Integer>, clsProductoInmutable> fetch_result = inventario.try_fetch(codigo, amount);
    if (fetch_result.isInvalid()) {
        if (fetch_result.getError().isEmpty()) return Optional.empty();
        int cantidad_sugerida = fetch_result.getError().get();
        if (cantidad_sugerida <= 0) return Optional.empty();

        String mensaje = String.format("El inventario no tiene >%d< unidades, pero tiene >%d< unidades " +
                "disponibles. Desea llevarse lo que queda?", amount, cantidad_sugerida);
        int result = JOptionPane.showConfirmDialog(null, mensaje, "Confirmar comprar menos....",
                JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE);
        // 1 = No
        if (result == JOptionPane.NO_OPTION) return Optional.empty();
        return try_insert_producto(codigo, cantidad_sugerida);
    }
    clsProductoInmutable producto = fetch_result.get();
    return Optional.of(producto);
}
```

```
    Integer line = 0;
    public ArrayList<clsProductoInmutable> productos_en_la_tabla = new ArrayList<>();

    void insert_producto(clsProductoInmutable producto) {
        line++;

        DefaultTableModel model = (DefaultTableModel) tabla.getModel();
        if (line > 1 && productos_en_la_tabla.get(line - 2).get_codigo_articulo().equals(producto.get_codigo_articulo())) {
            clsProductoInmutable last_input = productos_en_la_tabla.get(line - 2);
            producto.setCantidad(producto.get_cantidad() + last_input.get_cantidad());
            line--;
            model.removeRow(line - 1);
            productos_en_la_tabla.remove(line - 1);
        }
        productos_en_la_tabla.add(producto);
        Object[] row = new Object[]{line, producto.get_cantidad(), producto.get_venta(), producto.get_description(),
                String.format("%.2f", producto.get_cantidad() * producto.get_venta()),
                producto.articulo.retrieve_itbms()};
        model.addRow(row);
    }

}

class TextFieldIngresarCodigo extends TextField {
    ButtonAgregar binded_button;
    static Optional<TextFieldIngresarCodigo> text_field = Optional.empty();

    TextFieldIngresarCodigo(ButtonAgregar button) {
        setColumns(20);
        text_field = Optional.of(this);
        binded_button = button;
        addTextListener(e -> {
            if (binded_button.input_is_valid().isEmpty()) binded_button.block_button();
            else binded_button.unblock_button();
        });
    }
}


class TextFieldIngresarCantidad extends TextField {
```

```
    ButtonAgregar binded_button;
    static Optional<TextFieldIngresarCantidad> text_field = Optional.empty();

    TextFieldIngresarCantidad(ButtonAgregar button) {
        setColumns(7);
        text_field = Optional.of(this);
        binded_button = button;
        addTextListener(e -> {
            if (binded_button.input_is_valid().isEmpty()) binded_button.block_button();
            else binded_button.unblock_button();
        });
    }
}

class NumericButton extends JButton {
    Integer value;
    Optional<TextFieldIngresarCantidad> target_textfield = TextFieldIngresarCantidad.text_field;
    static Integer failed = 0;
    Optional<Thread> active_thread = Optional.empty();
    static CountDownLatch cd = new CountDownLatch(5);
    static JTextArea jokes_area;

    NumericButton(Integer value, JTextArea text_area_jokes) {
        super(String.valueOf(value));
        jokes_area = text_area_jokes;
        this.value = value;
        if (value >= 0) addActionListener(e -> {
            if (target_textfield.isEmpty()) {
                System.out.println("Se desconoce el objetivo de cantidad para el buton " + value);
                return;
            }
            TextFieldIngresarCantidad text_field = target_textfield.get();
            text_field.setText(text_field.getText() + value);
        });
        else if (value == -1) {
            setText("Clear");
            addActionListener(e -> {
                if (target_textfield.isEmpty()) {
                    System.out.println("Se desconoce el objetivo de cantidad para el buton " + value);
                    return;
                }
```

```
                TextFieldIngresarCantidad text_field = target_textfield.get();
                text_field.setText("");
            });
        } else if (value == -2) {
            setText("");
            addActionListener(e -> {
                NumericButton button = (NumericButton) e.getSource();
                failed++;
                if (active_thread.isEmpty()) {
                    spawn_thread(button, text_area_jokes);
                } else {
                    active_thread.get().interrupt();
                    spawn_thread(button, text_area_jokes);
                }
                if (failed > 5) button.setText("???");
                text_area_jokes.setVisible(true);
                text_area_jokes.setText(RandomText.get(failed));

            });
        }

    }

    Integer spawns = 0;

    void spawn_thread(NumericButton button, JTextArea joke_text_area) {
        spawns++;
        Integer spawn_no = spawns;
        Thread t = new Thread(new Runnable() {
            @Override
            public void run() {
                try {
//                    System.out.println("Waiting for " + spawn_no);
                    Thread.sleep(4500);
//                    System.out.println("We finished waiting for No." + spawns);
                    button.setText("");
                    joke_text_area.setText("");
                    joke_text_area.setVisible(false);
                    failed = 0;
                } catch (Exception Ignored) {
                    System.out.println("Exception No." + spawn_no);
```

```
                }
            }
        });
        t.start();
        active_thread = Optional.of(t);
    }

}


class RandomText {
    static String[] Block_1 = new String[]{"Button has no assigned function.", "Clicking holds no consequential " +
            "impact.", "No outcome from repeated button presses.", "Functionality remains inert, unaffected by " +
            "clicks" + ".", "Absence of programmed response to clicks.", "Button lacks predefined operational " +
            "significance.", "Repeated clicks yield no discernible effect.", "No action results from persistent " +
            "clicks.", "Button " + "remains unresponsive to user input.",
            "Clicking won't initiate any programmed " + "action.", "Programmed " + "response is nonexistent for " +
            "clicks.", "No functional assignment for button " + "action.", "Persistent " + "clicking won't alter " +
            "system state.", "Button's purpose is undefined, inert.", "Clicking provides no " + "tangible system " +
            "feedback."};
    static String[] Block_2 = new String[]{"Clicking like it owes you money, huh?",
            "Spamming won't summon a pizza, " + "just saying.", "Are you trying to break the internet, one click at " +
            "a" + " time?", "This isn't Morse code for " + "'bring snacks,' you know?", "Button abuse! The pixels are" +
            " " + "crying!", "Congratulations! You've just won " + "the 'Clicker of the Day' award.", "Pushing " +
            "buttons like" + " it's an Olympic sport.", "Clicking won't make it" + " a magic portal, but nice try.",
            "Someone's on a " + "mission to redefine 'persistent.'", "Judging by your " + "clicks, you're training " +
            "for a thumb-wrestling " + "championship.", "Button Olympics training in progress.", "Expecting confetti?" +
            " Keep clicking.", "Desperate for attention? Button understands.", "Button is not a " + "stress ball.",
            "Trying to unlock " + "secret cat videos?", "Click therapy? Seeking zen through buttons.", "Button " +
            "applause: click harder for " + "encore.", "Button whisperer, mastering the silent art.",
            "World record" + " for clicks: within reach.", "Button pusher extraordinaire reporting for duty.",
            "Button honors your " + "relentless dedication.", "Breaking news: Button unimpressed, needs vacation.",
            "Button says: 'Calm down," + " warrior.'", "Click " + "count exceeds cosmic expectations.", "Button " +
            "endorses patience, not spam.", "Your " + "click: a " + "masterpiece of persistence.", "Button " +
            "contemplates life during your clicks.", "Button predicts " + "Nobel for clickology soon.", "Click " +
            "wizardry: unmatched, slightly eccentric.", "Button wonders if you " + "need coffee.",};
    static String[] Block_3 = new String[]{"Still clicking? Puzzle intensifies.",
            "Confused by relentless clicks, " + "seriously.", "Program bewildered. Infinite clicks detected.",
            "Button mystified by endless persistence.", "Endless clicks defy all logic.", "Program bewildered: " +
            "endless clicks persist.", "Infinity clicks: " + "program's existential crisis.", "Program ponders: " +
            "clicks beyond comprehension.", "Clicks exceed " + "algorithmic comprehension limit.", "Persistent " +
```

```
            "clicking: program in deep confusion.", "Endless clicks " + "baffle program's logic.", "Infinite clicks: "
            + "program questions reality.", "Endless persistence confuses " + "programmed logic.", "Program " +
            "mystified: " + "clicks never-ending saga.", "Clicks surpass program's patience " + "threshold.",
            "Unyielding clicks: " + "program in utter disbelief.", "Program contemplates: clicks without " + "purpose" +
            ".", "Clicks persist, " + "program's sanity questioned.", "Endless clicks: program in silent chaos.",
            "Program bewildered: clicks " + "relentless mystery.", "Clicks transcend program's understanding realm.",
            "Program astonished: clicks " + "beyond explanation.", "Unending clicks: program's perpetual surprise.",
            "Clicks defy program's expected " + "limits."};
    static String[] Block_4 = new String[]{"Click wizard: Why are you pressing?",
            "Button philosophy: What is the " + "truth of clicking?", "Button says: Halt, warrior.", "Button " +
            "meditation: Continual clicking and " + "tranquility.", "Click prophet: Unpredictable clicks.",
            "Button " + "apocalypse: Silent clicks.", "Press for " + "tea, the button knows.", "Click maniac: " +
            "Continuous clicking" + " has no end.", "Transcendent click: Cosmic " + "clicks anticipated.", "Button " +
            "miracle: Inexplicable " + "clicks.", "Your click - a masterpiece of persistence" + ".", "Button mystery:" +
            " Unsolvable clicks.", "Button Zen master: Clicking is Zen.", "Click artist: " + "Unpredictable " +
            "masterpiece.", "Button physics: " + "Soundless clicks.", "Button genius: Click for brilliance.", "Button" +
            " meditation: Profound philosophy of " + "clicking.", "Click magic: Magical and unpredictable clicks.",
            "Button poetry: Rhythm of clicks.", "Press" + " for laughter, the button knows.", "Click miracle: " +
            "Continual " + "mysterious clicks.", "Click prophecy: " + "Unpredictable clicks.", "Button silence: " +
            "Soundless clicks.", "Button genius: Click for brilliance.", "Button poetry: Rhythm of clicks.", "Click " +
            "magic: Magical and " + "unpredictable clicks.", "Button " + "adventure: Journey of clicks.", "Button " +
            "philosopher: Press for wisdom.", "Button art: Painting with " + "clicks.", "Click prophet: Unpredictable" +
            " clicks."};

    static String get(Integer input) {
        System.out.println(input);
        Random rand = new Random();
        String[] Block;
        if (input > 50) return ".............";
        else if (input > 30) Block = Block_4;
        else if (input > 15) Block = Block_3;
        else if (input > 6) Block = Block_2;
        else Block = Block_1;
        return input + "-) " + Block[rand.nextInt(Block.length)];
    }
}


/*************************************************************
*                                                            *
* utp\ac\antonio_ng\examen\pktCaja\pktProducto\clsProducto.java *
```

```
*                                                                    *
**********************************************************************/
package com.utp.ac.antonio_ng.examen.pktCaja.pktProducto;

import com.fasterxml.jackson.annotation.JsonIgnore;
import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import com.fasterxml.jackson.annotation.JsonProperty;
import com.fasterxml.jackson.annotation.JsonUnwrapped;
import com.fasterxml.jackson.databind.ObjectMapper;

@JsonIgnoreProperties

public class clsProducto implements Cloneable {
    @JsonUnwrapped
    clsArticulo articulo;

    @Override
    public String toString() {
        String ret = "";
        try {
            ret = new ObjectMapper().writeValueAsString(this);
        } catch (Exception e) {
            System.out.println("Falla en deserializando el objeto:\n" + e);
        }
        return ret;
    }

    @JsonIgnore
    public String get_codigo_articulo() {
        return articulo.codigo;
    }

    @JsonIgnore
    public float get_venta() {
        return articulo.precio;
    }

    @JsonIgnore
    public float get_itbms() {
        return (float) articulo.retrieve_itbms() / 100;
    }
```

```java
@JsonIgnore
public int retrieve_itbms() {
    return articulo.retrieve_itbms();
}

@JsonIgnore
public String get_description() {
    return articulo.descripcion;
}

@JsonIgnore
public int en_inventario() {
    return articulo.cantidad;
}

@JsonProperty("Cantidad de Instancia")
Integer cantidad = 0; // La cantidad del artículo que tiene esta instancia

public int get_cantidad() {
    return cantidad;
}

public void setCantidad(Integer cantidad) {
    this.cantidad = cantidad;
}

/**
 * Devuelve una copia de la clase, pero con una cantidad definida
 *
 * @return
 */
public clsProductoInmutable fetch_instance(int cantidad) {
    ObjectMapper objectMapper = new ObjectMapper();
    articulo.cantidad -= cantidad;
    setCantidad(articulo.cantidad);
    clsProductoInmutable ret = null;
    try {
        String producto_as_string = objectMapper.writeValueAsString(this);
        ret = objectMapper.readValue(producto_as_string, clsProductoInmutable.class);
    } catch (Exception e) {
```

```
            System.out.printf("Hubo un problema convirtiendo el objeto con codigo %s\n", this.get_codigo_articulo());
            System.exit(1);
        }
        ret.setCantidad(cantidad);
        return ret;
    }

}


/*****************************************************************
*                                                                *
* utp\ac\antonio_ng\examen\pktCaja\pktProducto\clsProductoInmutable.java *
*                                                                *
*****************************************************************/
package com.utp.ac.antonio_ng.examen.pktCaja.pktProducto;

public class clsProductoInmutable extends clsProducto {
    public clsProductoInmutable fetch_instance(int _cantidad) throws UnsupportedOperationException {
        throw new UnsupportedOperationException();
    }

    @Override
    public int en_inventario() throws UnsupportedOperationException { // Elimina el acceso accidental al inventario
        throw new UnsupportedOperationException();
    }
}


/*****************************************************************
*                                                               *
* utp\ac\antonio_ng\examen\pktInventario\clsLoadedInventario.java *
*                                                               *
*****************************************************************/
package com.utp.ac.antonio_ng.examen.pktInventario;

import com.utp.ac.antonio_ng.examen.pktCaja.pktProducto.clsProducto;
import com.utp.ac.antonio_ng.examen.pktCaja.pktProducto.clsProductoInmutable;
import com.utp.ac.antonio_ng.examen.pktResourceFiles.FileResource;

import com.fasterxml.jackson.databind.ObjectMapper;
```

```java
import io.vavr.control.Validation;

import java.io.File;
import java.util.*;

public class clsLoadedInventario {
    HashMap<String, clsProducto> inventario = new HashMap<>();
    List<clsProducto> lista_producto;

    public void print_inventario() {
    for (Map.Entry<String, clsProducto> set: inventario.entrySet()){
        System.out.println(set.getValue().toString());
    }
    }

    /**
     * Devuelve la cantidad en inventario si hay registro
     */
    public Optional<Integer> query(String codigo) {
        if (!inventario.containsKey(codigo))
            return Optional.empty();
        return Optional.of(inventario.get(codigo).en_inventario());
    }

    /**
     * Método para usar cuando se puede garantizar la cantidad que queda en inventario.
     */
    public clsProductoInmutable fetch(String codigo, int a_substraer) {
        return inventario.get(codigo).fetch_instance(a_substraer);
    }

    /**
     * En caso de Ok:
     * Retorna una copia del Objeto con la cantidad pedida.
     * En caso de Error:
     * Retorna nada si no encuentra el producto en la base de datos.
     * Retorna la cantidad disponible si resulta haber menos de lo pedido.
     */
    public Validation<Optional<Integer>, clsProductoInmutable> try_fetch(String codigo, int amount) {
        if (!inventario.containsKey(codigo))
            return Validation.invalid(Optional.empty());
```

```
        Validation<Optional<Integer>, clsProductoInmutable> ret;
        clsProducto producto = inventario.get(codigo);
        int en_inventario = producto.en_inventario();
        if (en_inventario >= amount) {
            ret = Validation.valid(fetch(codigo, amount));
        } else if (en_inventario <= 0)
            ret = Validation.invalid(Optional.of(0));
        else
            ret = Validation.invalid(Optional.of(en_inventario));
        return ret;
    }

    public clsLoadedInventario() {
        ObjectMapper mapper = new ObjectMapper();
        try {
            File input_stream = new FileResource().getFileFromResource("inventario.json");
            lista_producto = Arrays.asList(mapper.treeToValue(mapper.readTree(input_stream).get("data"),
                    clsProducto[].class));
        } catch (Exception e) {
            System.out.println("Error al leer los datos.......\n" + e);
            System.exit(400);
        }
        for (clsProducto producto : lista_producto) {
            System.out.println(producto.toString());
            producto.setCantidad(producto.en_inventario());
            inventario.put(producto.get_codigo_articulo(), producto);
        }
    }
}


/**************************************************************************
*                                                                         *
* utp\ac\antonio_ng\examen\pktPasswordAntonioNg\clsPasswordAntonioNg.java *
*                                                                         *
**************************************************************************/
package com.utp.ac.antonio_ng.examen.pktPasswordAntonioNg;

import javax.swing.JLabel;
import javax.swing.JOptionPane;
```

```java
import javax.swing.JPasswordField;


public class clsPasswordAntonioNg {
    Integer attempts = 1;
    static final Integer password_attempts = 3;
    String usuario = "", mensaje = "";
    static final JLabel titulo = new JLabel("Ingrese su contraseña");
    public static boolean is_logged_in = false;
    // Solo me faltaba una manera de saber si se logró iniciar sesión de manera exitosa, asi que creé un acceso publico a la variable

    public static void main(String[] args) {
        if (!clsPasswordAntonioNg.esperar_inicio_sesion()) {
            JOptionPane.showMessageDialog(null, "Inicio de sesión fallada. El programa se cerrará");
            System.exit(0);
        }
        System.out.println("Hola. Acabamos de leer. Inicio de sesión existosa");
    }

    public static boolean esperar_inicio_sesion() {
        is_logged_in = false;
        boolean end = false;

        clsPasswordAntonioNg instance = new clsPasswordAntonioNg();
        while (!end) {
            end = instance.spawn();
        }
        return is_logged_in;
    }

    boolean spawn() {
        JPasswordField objLeerPasword = new JPasswordField();
        this.usuario = JOptionPane.showInputDialog(null, new Object[]{"Hace un buen dia. \nProcedamos a ingresar sesion!", "Ingrese
su usuario"}, "");

        if (this.usuario == null)
            return false; // Intentamos denuevo si decide cancelar por accidente.
        while (this.attempts <= password_attempts) {
            int x = JOptionPane.showConfirmDialog(null, new Object[]{new JLabel("Bienvenido a Stark CO. " + this.usuario), titulo,
objLeerPasword}, "Inicio de Sesion", JOptionPane.OK_CANCEL_OPTION);
            // El usuario oprimio un boton distinto a OK
```

```
            if (x != 0) {
                JOptionPane.showMessageDialog(null, "Decidio cancelar el programa.");
                return true;
            }

            // La contraseña se guarda en un vector de caracteres
            char[] vector = objLeerPasword.getPassword();
            String pass = new String(vector);

            if (pass.isEmpty())
                continue;
            if (pass.equals("fin")) {
                JOptionPane.showMessageDialog(null, "Perfecto " + usuario + " puede proceder con la ejecucion del programa.");
                is_logged_in = true;
                break;
            } else {
                if (this.attempts < password_attempts)
                    mensaje = String.format("Te quedan %d oportunidades.", password_attempts - this.attempts);
                else
                    mensaje = "Intentos agotados.";
                objLeerPasword.setText("");
                JOptionPane.showMessageDialog(null, usuario + "\nTe has equivocado de contraseña.\n" + mensaje);
                this.attempts++;
            }
        }
        if (this.attempts > password_attempts)
            JOptionPane.showMessageDialog(null, "Lo siento " + usuario + "\nHable con su supervisor para recuperar su contraseña");
        else
            JOptionPane.showMessageDialog(null, usuario + " Gracias por utilizar nuestro software");
        return true;
    }
}


/*************************************************
*                                                *
* utp\ac\antonio_ng\examen\pktPersona\clsPersona.java *
*                                                *
*************************************************/
package com.utp.ac.antonio_ng.examen.pktPersona;
```

```java
import com.fasterxml.jackson.annotation.JsonIgnore;
import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import com.fasterxml.jackson.annotation.JsonInclude;
import com.fasterxml.jackson.annotation.JsonProperty;

@JsonIgnoreProperties(ignoreUnknown = true)
@JsonInclude(JsonInclude.Include.NON_NULL)
public class clsPersona {
    @Override
    public String toString() {
        return String.format("%s:{Cedula: %s, Nombre Completo: %s %s, telefono: %d}", getClass().getName(), cedula, nombre, apellido,
telefono);
    }

    String cedula = "";
    String nombre = "";
    String apellido = "";
    @JsonProperty("teléfono")
    Integer telefono = 0;

    public void setTelefono(Integer telefono) {
        this.telefono = telefono;
    }

    public void setApellido(String apellido) {
        this.apellido = apellido;
    }

    public void setCedula(String cedula) {
        this.cedula = cedula;
    }

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

    public String getCedula() {
        return cedula;
    }

    public String getNombre() {
```

```
        return nombre;
    }

    public String getApellido() {
        return apellido;
    }

    @JsonIgnore
    public String getNombreCompleto() {
        return (getNombre() + " " + getApellido()).trim();
    }

    public Integer getTelefono() {
        return telefono;
    }
}


/*************************************************************
*                                                            *
* utp\ac\antonio_ng\examen\pktResourceFiles\FileResource.java *
*                                                            *
*************************************************************/
package com.utp.ac.antonio_ng.examen.pktResourceFiles;

import java.io.*;
import java.net.URISyntaxException;
import java.net.URL;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.util.List;

public class FileResource {
    public static void main(String[] args) throws URISyntaxException, IOException {
        // Test
        File file = new FileResource().getFileFromResource("lista_de_clientes/source_data.json");
        System.out.println("Resultado de lectura:\n" + Files.readString(file.toPath(), StandardCharsets.UTF_8));
    }

    public FileResource() {
    }
```

```
public String getFileContent(String file_name) throws URISyntaxException, IOException {
    return Files.readString(getFileFromResource(file_name).toPath());
}

public FileResource(String file_name) throws URISyntaxException {
    if (file_name == null) return;
    System.out.println("getResourceAsStream : " + file_name);
    InputStream is = getFileFromResourceAsStream(file_name);
    printInputStream(is);

    System.out.println("\ngetResource : " + file_name);
    File file = getFileFromResource(file_name);
    printFile(file);

}
// get a file from the resources folder
// works everywhere, IDEA, unit test and JAR file.
public InputStream getFileFromResourceAsStream(String fileName) {

    // The class loader that loaded the class
    ClassLoader classLoader = getClass().getClassLoader();
    InputStream inputStream = classLoader.getResourceAsStream(fileName);

    // the stream holding the file content
    if (inputStream == null) {
        throw new IllegalArgumentException("file not found! " + fileName);
    } else {
        return inputStream;
    }
}
/*
    The resource URL is not working in the JAR
    If we try to access a file that is inside a JAR,
    It throws NoSuchFileException (linux), InvalidPathException (Windows)

    Resource URL Sample: file:java-io.jar!/json/file1.json
 */
public File getFileFromResource(String fileName) throws URISyntaxException {

    ClassLoader classLoader = getClass().getClassLoader();
```

```
        URL resource = classLoader.getResource(fileName);
        if (resource == null) {
            throw new IllegalArgumentException("file not found! " + fileName);
        } else {

            // failed if files have whitespaces or special characters
            //return new File(resource.getFile());

            return new File(resource.toURI());
        }

    }
    // print input stream
    private static void printInputStream(InputStream input_stream) {

        try (InputStreamReader streamReader = new InputStreamReader(input_stream, StandardCharsets.UTF_8); BufferedReader reader =
new BufferedReader(streamReader)) {

            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }

        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    // print a file
    private static void printFile(File file) {
        List<String> lines;
        try {
            lines = Files.readAllLines(file.toPath(), StandardCharsets.UTF_8);
            lines.forEach(System.out::println);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```