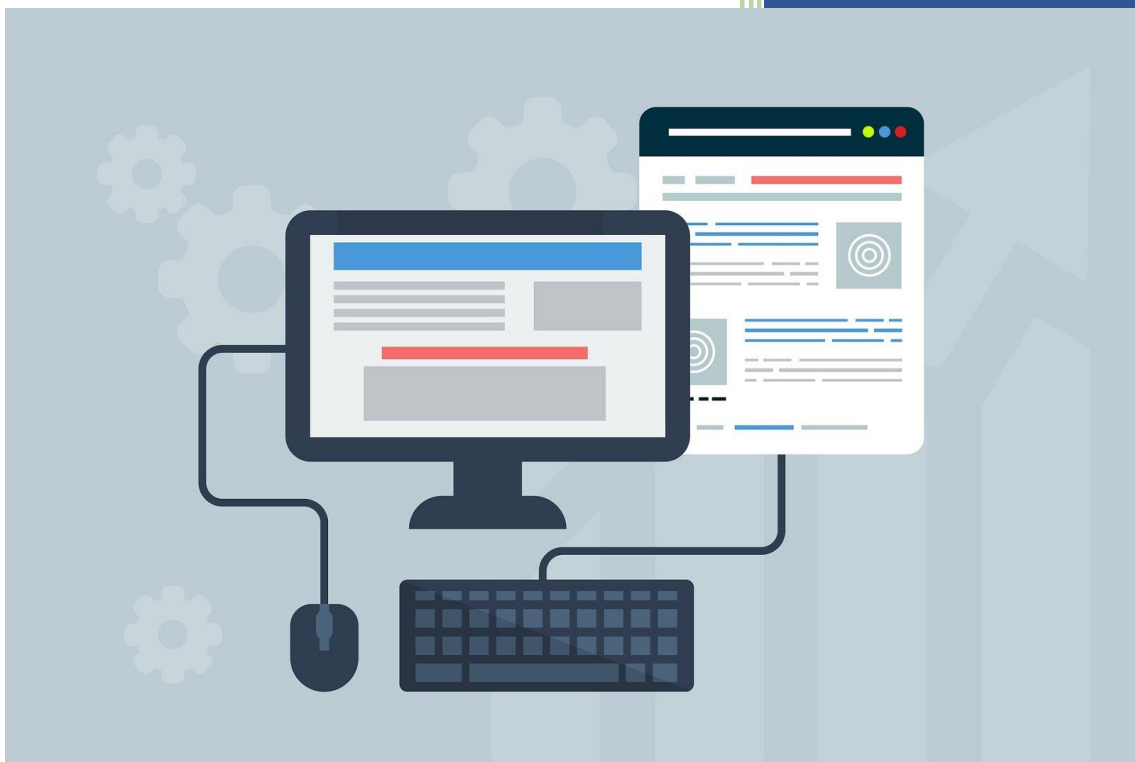


DESARROLLO DE INTERFACES



DESARROLLO DE INTERFACES

15/12/2019

DAVID BORREGO ASENCIO

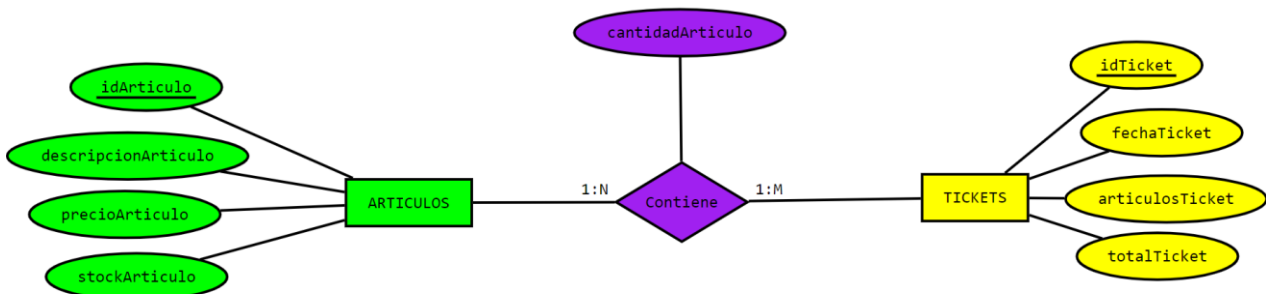
ÍNDICE

Enunciado.....	1
Diagrama Entidad-Relación.....	1
Esquema Relacional	1
Diagrama Workbench	2
Sentencias SQL BBDD	2
Diseños	4
Menú Principal	4
Artículos	5
Tickets	6
Código Java.....	8
Principal.....	8
Nuevo Artículo.....	10
Eliminar Artículo.....	15
Modificar Artículo	19
Ver Artículos	25
Nuevo Ticket.....	27
Ver Tickets.....	39
Pantallazos	45
BIBLIOGRAFÍA.....	62

Enunciado

En esta práctica se deben realizar 3 diseños diferentes para una pantalla principal de un programa de gestión. En dicho programa debemos realizar las operaciones CRUD habituales para Clientes, para Artículos y para Tickets de compra. Acabar de conectar la aplicación con la base de datos y realizar todas las operaciones CRUD, tanto de Artículos como de Tickets. Hay que poner especial cuidado en el alta de los Tickets, donde hay involucrados campos FK.

Diagrama Entidad-Relación



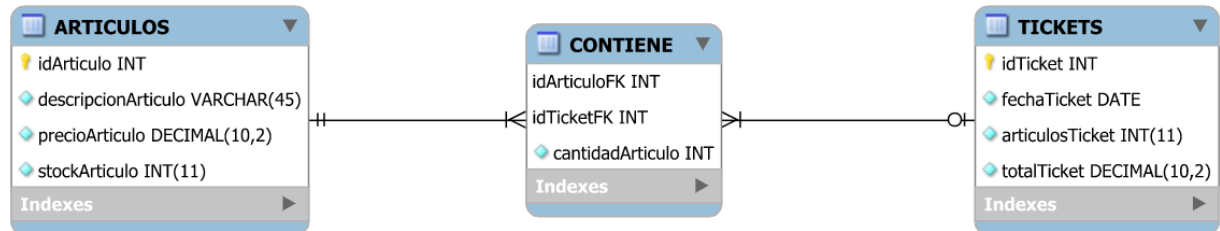
Esquema Relacional

ARTICULOS: idArticulo, descripcionArticulo, precioArticulo, stockArticulo.

TICKETS: idTicket, fechaTicket, articulosTicket, totalTicket.

CONTIENE: idArticuloFK, idTicketFK, cantidadArticulo.

Diagrama Workbench



Sentencias SQL BBDD

-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE,

SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_D
ATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- Schema Tienda

-- Schema Tienda

CREATE SCHEMA IF NOT EXISTS `Tienda` DEFAULT CHARACTER SET utf8 COLLATE
utf8_spanish2_ci ;

USE `Tienda` ;

-- Table `Tienda`.`ARTICULOS`

```
-----  
CREATE TABLE IF NOT EXISTS `Tienda`.`ARTICULOS` (  
  `idArticulo` INT NOT NULL AUTO_INCREMENT,  
  `descripcionArticulo` VARCHAR(45) NOT NULL,  
  `precioArticulo` DECIMAL(10,2) NOT NULL,  
  `stockArticulo` INT(11) NOT NULL,  
  PRIMARY KEY (`idArticulo`),  
  UNIQUE INDEX `descripcionArticulo_UNIQUE` (`descripcionArticulo` ASC) VISIBLE)  
ENGINE = InnoDB;
```

-- Table `Tienda`.`TICKETS`

```
-----  
CREATE TABLE IF NOT EXISTS `Tienda`.`TICKETS` (  
  `idTicket` INT NOT NULL AUTO_INCREMENT,  
  `fechaTicket` DATE NOT NULL,  
  `articulosTicket` INT(11) NOT NULL,  
  `totalTicket` DECIMAL(10,2) NOT NULL,  
  PRIMARY KEY (`idTicket`))  
ENGINE = InnoDB;
```

-- Table `Tienda`.`CONTIENE`

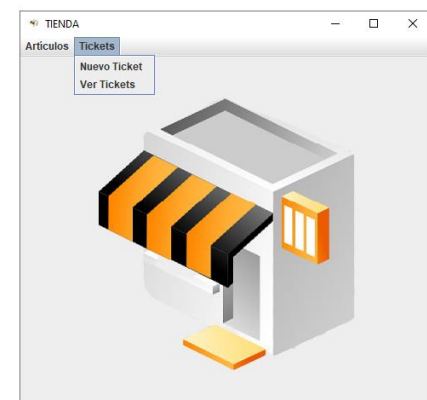
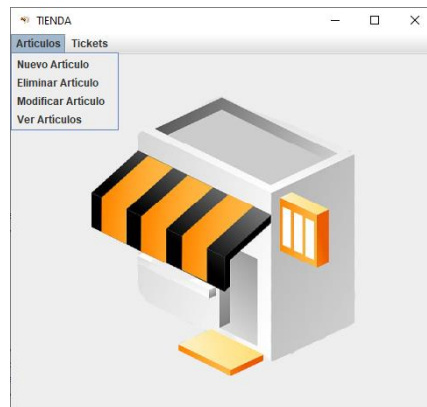
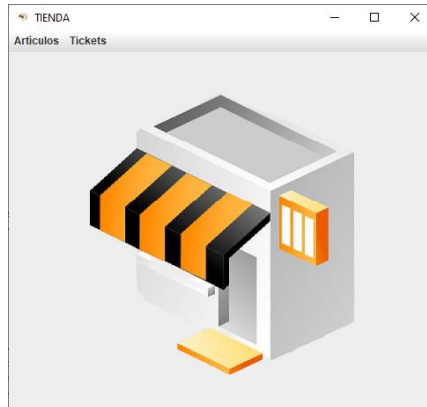
```
-----  
CREATE TABLE IF NOT EXISTS `Tienda`.`CONTIENE` (  
  `idArticuloFK` INT NOT NULL,  
  `idTicketFK` INT NOT NULL,
```

```
`cantidadArticulo` INT NOT NULL,  
  
INDEX `idArticulo_idx` (`idArticuloFK` ASC) VISIBLE,  
  
INDEX `idTicketFK_idx` (`idTicketFK` ASC) VISIBLE,  
  
PRIMARY KEY (`idTicketFK`, `idArticuloFK`),  
  
CONSTRAINT `idArticuloFK`  
    FOREIGN KEY (`idArticuloFK`)  
    REFERENCES `Tienda`.`ARTICULOS` (`idArticulo`)  
  
    ON DELETE NO ACTION  
  
    ON UPDATE NO ACTION,  
  
CONSTRAINT `idTicketFK`  
    FOREIGN KEY (`idTicketFK`)  
    REFERENCES `Tienda`.`TICKETS` (`idTicket`)  
  
    ON DELETE NO ACTION  
  
    ON UPDATE NO ACTION)  
  
ENGINE = InnoDB;
```

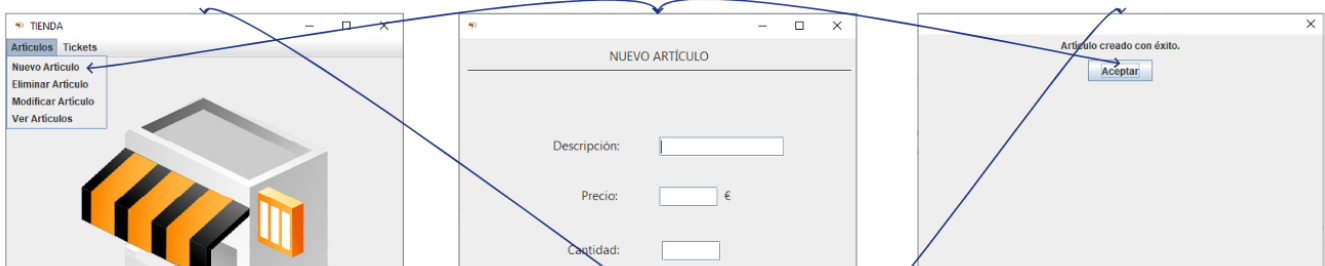
```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

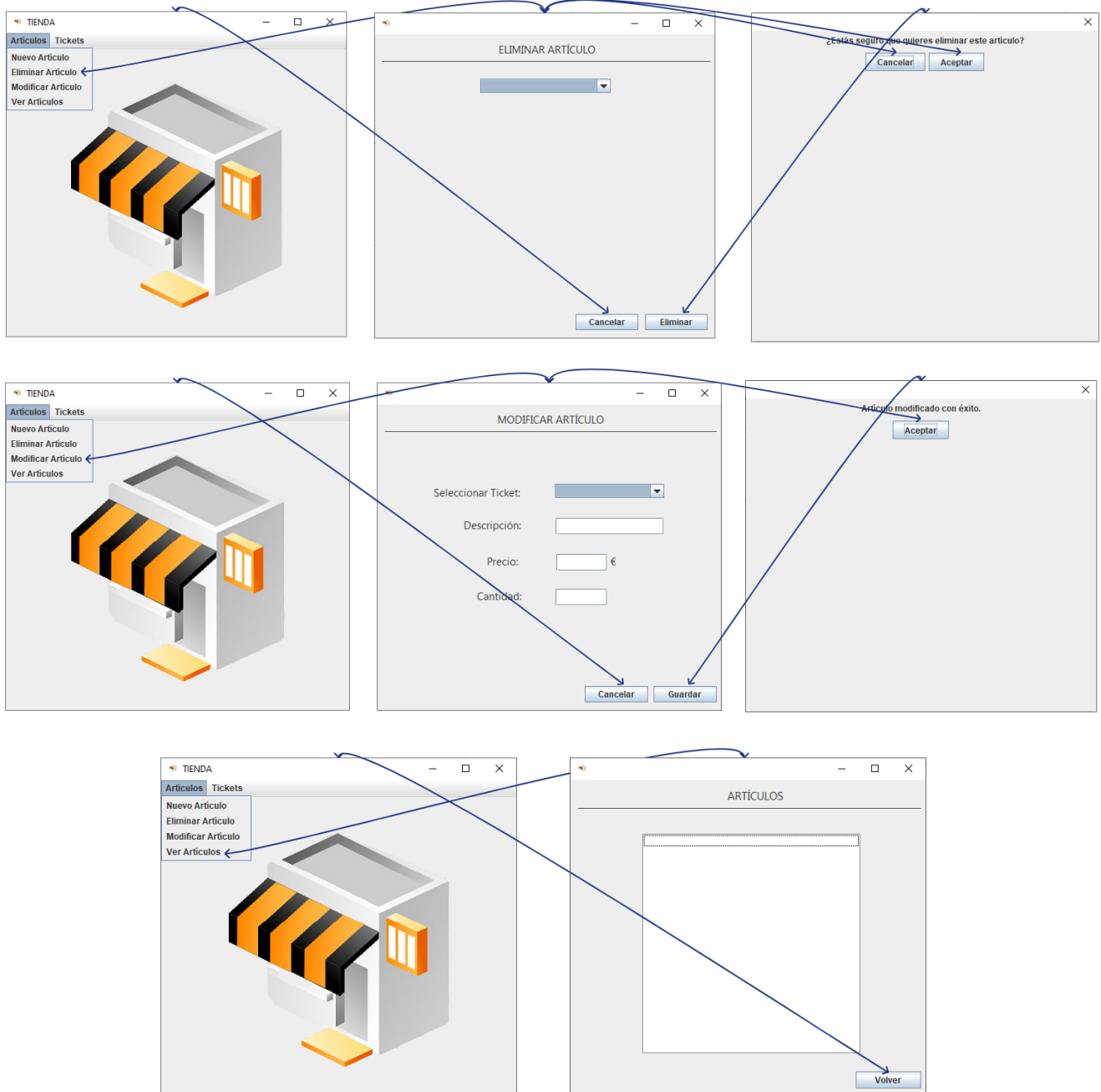
Diseños

Menú Principal

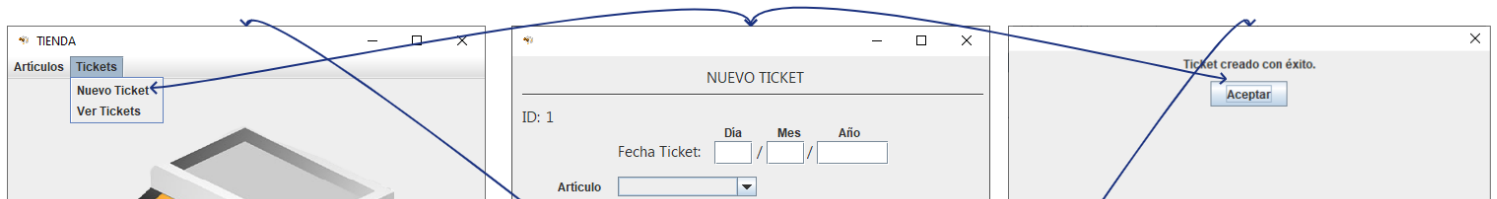


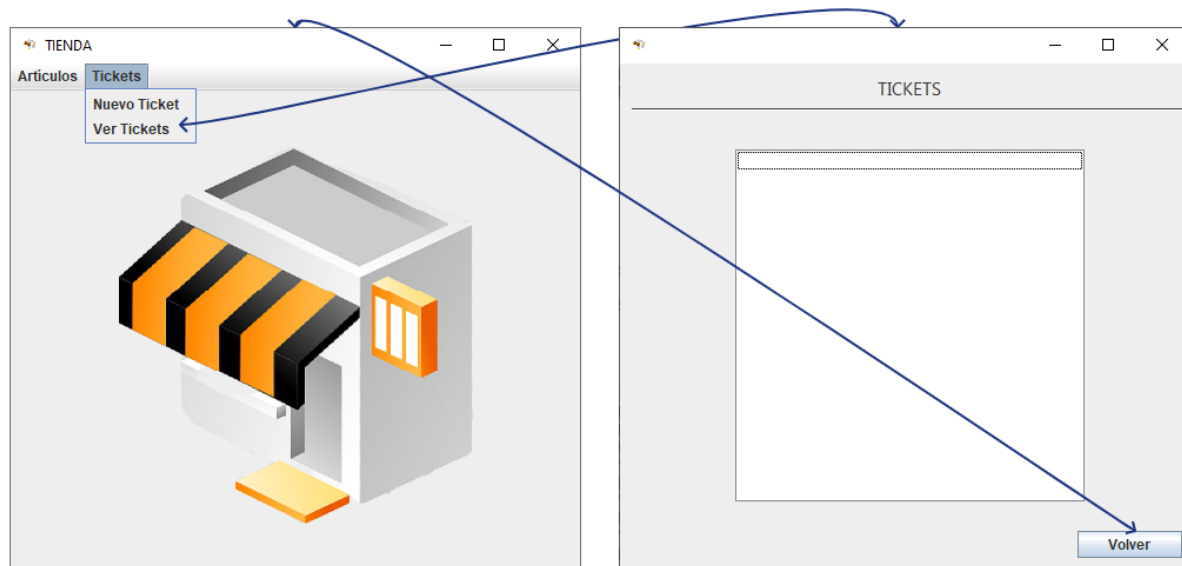
Artículos





Tickets





Código Java

Principal

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;

public class Principal extends JFrame implements ActionListener
{

    private static final long serialVersionUID = 1L;
    private JPanel contentPane = new JPanel() {

        private static final long serialVersionUID = 1L;
        public void paintComponent(Graphics g)
        {
            super.paintComponent(g);
            Dimension tamano = getSize();
            ImageIcon imagenFondo = new
ImageIcon(getClass().getResource("SHOP.png"));
            g.drawImage(imagenFondo.getImage(), 0, 0,
tamano.width, tamano.height, null);
            setOpaque(false);
        }
    };

    private Nuevo_Articulo nuevo_articulo;
    private Eliminar_Articulo eliminar_articulo = new
Eliminar_Articulo();
    private Modificar_Articulo modificar_articulo;
    private Ver_Articulos ver_articulos = new Ver_Articulos();
    private Nuevo_Ticket nuevo_ticket = new Nuevo_Ticket();
    private Ver_Tickets ver_tickets = new Ver_Tickets();

    private JMenuBar menuBar = new JMenuBar();
    private JMenu mnNewMenu = new JMenu("Art\u00E9culos");
    private JMenuItem mntmNewMenuItem = new JMenuItem("Nuevo
Art\u00E9culo");
    private JMenuItem mntmNewMenuItem_1 = new JMenuItem("Eliminar
Art\u00E9culo");
    private JMenuItem mntmNewMenuItem_2 = new JMenuItem("Modificar
Art\u00E9culo");
    private JMenuItem mntmNewMenuItem_3 = new JMenuItem("Ver
Art\u00E9culos");
    private JMenu mnTickets = new JMenu("Tickets");
    private JMenuItem mntmNewMenuItem_4 = new JMenuItem("Nuevo
Ticket");
    private JMenuItem mntmNewMenuItem_5 = new JMenuItem("Ver
Tickets");

    /**
     * Launch the application.
     */
    public static void main(String[] args)

```

```

    {
        EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {
                try
                {
                    Principal frame = new Principal();
                    frame.setVisible(true);
                } catch (Exception e)
                {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public Principal()
    {
        setIconImage(Toolkit.getDefaultToolkit().getImage(Principal.class
s.getResource("/disel\u00Flo_interfaces/SHOP.png")));
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 512, 471);

        this.setTitle("TIENDA");
        setJMenuBar(menuBar);
        menuBar.add(mnNewMenu);
        mnNewMenu.add(mntmNewMenuItem);
        mnNewMenu.add(mntmNewMenuItem_1);
        mnNewMenu.add(mntmNewMenuItem_2);
        mnNewMenu.add(mntmNewMenuItem_3);
        menuBar.add(mnTickets);
        mnTickets.add(mntmNewMenuItem_4);
        mnTickets.add(mntmNewMenuItem_5);

        mntmNewMenuItem.addActionListener(this);
        mntmNewMenuItem_1.addActionListener(this);
        mntmNewMenuItem_2.addActionListener(this);
        mntmNewMenuItem_3.addActionListener(this);
        mntmNewMenuItem_4.addActionListener(this);
        mntmNewMenuItem_5.addActionListener(this);

        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        contentPane.setLayout(new BorderLayout(0, 0));
        setContentPane(contentPane);
        this.setLocationRelativeTo(null);
    }

    @Override
    public void actionPerformed(ActionEvent ae)
    {
        if(ae.getSource().equals(mntmNewMenuItem))

```

```

        {
            nuevo_articulo = new Nuevo_Articulo();
            nuevo_articulo.setVisible(true);
        }else
        {
            if(ae.getSource().equals(mntmNewMenuItem_1))
            {
                eliminar_articulo = new Eliminar_Articulo();
                eliminar_articulo.setVisible(true);
            }else
            {
                if(ae.getSource().equals(mntmNewMenuItem_2))
                {
                    modificar_articulo = new
Modificar_Articulo();
                    modificar_articulo.setVisible(true);
                }else
                {
                    if(ae.getSource().equals(mntmNewMenuItem_3))
                    {
                        ver_articulos = new
Ver_Articulos();
                        ver_articulos.setVisible(true);
                    }else
                    {
                        if(ae.getSource().equals(mntmNewMenuItem_4))
                        {
                            nuevo_ticket = new
Nuevo_Ticket();
                            nuevo_ticket.setVisible(true);
                        }else
                        {
                            if(ae.getSource().equals(mntmNewMenuItem_5))
                            {
                                ver_tickets = new
Ver_Tickets();
                                ver_tickets.setVisible(true);
                            }
                        }
                    }
                }
            }
        }
    }
}

```

Nuevo Artículo

```

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.Panel;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JSeparator;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;

public class Nuevo_Articulo extends JFrame implements WindowListener
{
    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JTextField textfieldDescripcion;
    private JTextField textfieldPrecio;
    private JLabel labelPrecio;
    private JLabel labelMoneda;
    private JLabel labelCantidad;
    private JTextField textfieldCantidad;
    private JLabel labelNuevoArticulo;
    private JButton buttonCancelar;
    private JButton buttonGuardar;
    private JDialog dialog01;
    private Panel panel1;
    private Panel panel2;
    private JLabel mensaje1;
    private JButton btn_dialogo_1;
    private JSeparator separator;
    private Conecta_BBDD base_datos;

    /**
     * Create the frame.
     */
    public Nuevo_Articulo()
    {
        base_datos = new Conecta_BBDD();

        setIconImage(Toolkit.getDefaultToolkit().getImage(Nuevo_Articulo
.class.getResource("/disel\u00F1o_interfaces/SHOP.png")));
        setBounds(100, 100, 500, 470);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
    }

```

```

JLabel labelDescripcion = new JLabel("Descripci\u00F3n:");
labelDescripcion.setFont(new Font("Microsoft New Tai Lue",
Font.PLAIN, 15));
labelDescripcion.setBounds(114, 120, 88, 24);
contentPane.add(labelDescripcion);

textfieldDescripcion = new JTextField();
textfieldDescripcion.setBounds(242, 120, 153, 24);
contentPane.add(textfieldDescripcion);
textfieldDescripcion.setColumns(10);

textfieldPrecio = new JTextField();
textfieldPrecio.setColumns(10);
textfieldPrecio.setBounds(244, 181, 72, 24);
contentPane.add(textfieldPrecio);

labelPrecio = new JLabel("Precio:");
labelPrecio.setFont(new Font("Microsoft New Tai Lue",
Font.PLAIN, 15));
labelPrecio.setBounds(149, 181, 53, 24);
contentPane.add(labelPrecio);

labelMoneda = new JLabel("\u20AC");
labelMoneda.setFont(new Font("Microsoft New Tai Lue",
Font.PLAIN, 15));
labelMoneda.setBounds(323, 181, 25, 24);
contentPane.add(labelMoneda);

labelCantidad = new JLabel("Cantidad:");
labelCantidad.setFont(new Font("Microsoft New Tai Lue",
Font.PLAIN, 15));
labelCantidad.setBounds(132, 247, 63, 24);
contentPane.add(labelCantidad);

textfieldCantidad = new JTextField();
textfieldCantidad.setColumns(10);
textfieldCantidad.setBounds(244, 247, 72, 24);
contentPane.add(textfieldCantidad);

labelNuevoArticulo = new JLabel("NUEVO ART\u00CDCULO");
labelNuevoArticulo.setFont(new Font("Microsoft New Tai
Lue", Font.BOLD, 15));
labelNuevoArticulo.setBounds(183, 11, 146, 24);
contentPane.add(labelNuevoArticulo);

separator = new JSeparator();
separator.setForeground(Color.DARK_GRAY);
separator.setBounds(10, 38, 468, 2);
contentPane.add(separator);

buttonCancelar = new JButton("Cancelar");
buttonCancelar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        base_datos.cierra_conexion();
        Nuevo_Articulo.this.setVisible(false);
        Nuevo_Articulo.this.dispose();
    }
});

```



```

buttonCancelar.setBounds(288, 397, 89, 23);
contentPane.add(buttonCancelar);

buttonGuardar = new JButton("Guardar");
buttonGuardar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        String sentencia_nuevo_articulo = "INSERT INTO
articulos (descripcionArticulo, precioArticulo, stockArticulo) VALUES
('" + textfieldDescripcion.getText() + "', " +
comprueba_cantidad(textfieldPrecio.getText()) + ", " +
textfieldCantidad.getText() + ");";
        System.out.println(sentencia_nuevo_articulo);

        base_datos.agregar_objeto(sentencia_nuevo_articulo);

        dialogo1.setVisible(true);
    }
});
buttonGuardar.setBounds(385, 397, 89, 23);
contentPane.add(buttonGuardar);

//-----Dialog-----
dialogo1 = new JDialog(this, "", true);
mensaje1 = new JLabel("Artículo creado con éxito.");
dialogo1.getContentPane().setLayout(new BorderLayout());
dialogo1.setSize(500, 470);
dialogo1.setResizable(false);
dialogo1.setLocationRelativeTo(null);
dialogo1.addWindowListener(this);
panel1 = new Panel();
panel2 = new Panel();
panel1.setLayout(new FlowLayout());
panel2.setLayout(new FlowLayout());
panel1.add(mensaje1);
btn_dialogo_1 = new JButton("Aceptar");
panel2.add(btn_dialogo_1);
dialogo1.getContentPane().add(BorderLayout.NORTH, panel1);
dialogo1.getContentPane().add(BorderLayout.CENTER, panel2);
btn_dialogo_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        base_datos.cierra_conexion();
        Nuevo_Articulo.this.dispose();
        new Nuevo_Articulo().setVisible(true);
    }
});
//-----
this.addWindowListener(this);
this.setVisible(false);
this.setResizable(false);
this.setLocationRelativeTo(null);
}

public float comprueba_cantidad(String precio)
{
    String precio_articulo = precio;
    char[] array_caracteres = precio_articulo.toCharArray();
    String precio_factorizado = "";
    String string_precio_dos_decimales = "";

```

```

        boolean llave = true;

        for (int i = 0; i < array_caracteres.length; i++)
        {
            if (array_caracteres[i] == ',')
            {
                precio_factorizado += ".";
            }
            else
            {
                precio_factorizado += array_caracteres[i];
            }
        }

        for (int i = 0; i < precio_factorizado.length(); i++)
        {
            if(llave)
            {
                if(precio_factorizado.charAt(i) == '.')
                {
                    string_precio_dos_decimales +=
precio_factorizado.charAt(i);
                    string_precio_dos_decimales +=
precio_factorizado.charAt(i+1);
                    string_precio_dos_decimales +=
precio_factorizado.charAt(i+2); //ESTO PUEDE DAR PROBLEMAS SI EL
NUMERO INTRODUCIDO POR EL USUARIO TIENE MENOS DE DOS DECIMALES,
CORREGIR EN FUTURAS VERSIONES
                    llave = false;
                }
                else
                {
                    string_precio_dos_decimales +=
precio_factorizado.charAt(i);
                }
            }
        }

        return Float.valueOf(string_precio_dos_decimales);
    }

    public void windowActivated(WindowEvent e)
    {}

    public void windowClosed(WindowEvent e)
    {}

    public void windowClosing(WindowEvent e)
    {
        base_datos.cierra_conexion();
        this.dispose();
    }

    public void windowDeactivated(WindowEvent e)
    {}

    public void windowDeiconified(WindowEvent e)
    {}

    public void windowIconified(WindowEvent e)

```

```

    {}

    public void windowOpened(WindowEvent e)
    {}
}

```

Eliminar Artículo

```

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.Panel;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JSeparator;
import javax.swing.border.EmptyBorder;
import javax.swing.SwingConstants;

public class Eliminar_Articulo extends JFrame implements
WindowListener
{

    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JLabel labelEliminarArticulo;
    private JButton buttonCancelar;
    private JButton buttonEliminar;
    private JDialog dialog01;
    private Panel panel1;
    private Panel panel2;
    private JLabel mensaje1;
    private JButton btn_dialogo_1;
    private JButton btn_dialogo_2;
    private ResultSet resultset1;
    private JComboBox<String> comboArticulo;
    private Conecta_BBDD base_datos;
    private JLabel label;
    /**
     * Create the frame.
     */
    public Eliminar_Articulo()
    {

```

```

        base_datos = new Conecta_BBDD();

        setIconImage(Toolkit.getDefaultToolkit().getImage(Nuevo_Articulo
.class.getResource("/dise\u00F1o_interfaces/SHOP.png")));
        setBounds(100, 100, 500, 470);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        labelEliminarArticulo = new JLabel("ELIMINAR
ART\u00C9CULO");
        labelEliminarArticulo.setFont(new Font("Microsoft New Tai
Lue", Font.BOLD, 15));
        labelEliminarArticulo.setBounds(164, 11, 176, 24);
        contentPane.add(labelEliminarArticulo);

        comboArticulo = new JComboBox<String>();
        comboArticulo.setBounds(150, 63, 186, 20);
        contentPane.add(comboArticulo);

        JSeparator separator = new JSeparator();
        separator.setForeground(Color.DARK_GRAY);
        separator.setBounds(10, 38, 468, 2);
        contentPane.add(separator);

        buttonCancelar = new JButton("Cancelar");
        buttonCancelar.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                base_datos.cierra_conexion();
                Eliminar_Articulo.this.dispose();
                new Eliminar_Articulo();
            }
        });
        buttonCancelar.setBounds(286, 397, 89, 23);
        contentPane.add(buttonCancelar);

        buttonEliminar = new JButton("Eliminar");
        buttonEliminar.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                if(comboArticulo.getSelectedItem().toString().equals("Elegir
uno..."))
                {
                    label.setText("Seleccione un art\u00EDculo a
eliminar.");
                }
                else
                {
                    dialogo1.setVisible(true);
                }
            }
        });
        buttonEliminar.setBounds(385, 397, 89, 23);
        contentPane.add(buttonEliminar);

```

```

        label = new JLabel("", SwingConstants.CENTER);
        label.setForeground(Color.RED);
        label.setFont(new Font("Microsoft New Tai Lue", Font.BOLD,
12));

        label.setBounds(10, 94, 468, 24);
        contentPane.add(label);

        //-----Dialog-----
        dialog1 = new JDialog(this, "", true);
        mensaje = new JLabel("¿Estás seguro que quieres eliminar
este artículo?");
        dialog1.getContentPane().setLayout(new BorderLayout());
        dialog1.setSize(500, 470);
        dialog1.setResizable(false);
        dialog1.setLocationRelativeTo(null);
        dialog1.addWindowListener(this);
        panel1 = new Panel();
        panel2 = new Panel();
        panel1.setLayout(new FlowLayout());
        panel2.setLayout(new FlowLayout());
        panel1.add(mensaje);
        btn_dialogo_1 = new JButton("Cancelar");
        btn_dialogo_2 = new JButton("Aceptar");
        panel2.add(btn_dialogo_1);
        panel2.add(btn_dialogo_2);
        dialog1.getContentPane().add(BorderLayout.NORTH, panel1);
        dialog1.getContentPane().add(BorderLayout.CENTER, panel2);
        btn_dialogo_1.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                dialog1.setVisible(false);
            }
        });
        btn_dialogo_2.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                try
                {
                    //----- ELIMINAR ARTÍCULO DE TODOS
LOS TICKETS EN DONDE ES FK -----

                    ResultSet idArticulo =
base_datos.obtener_objetos("SELECT idArticulo FROM articulos WHERE
descripcionArticulo = '"+ comboArticulo.getSelectedItem().toString() +
";");

                    idArticulo.next();
                    base_datos.agregar_objeto("DELETE FROM
contiene WHERE idArticuloFK = '" + idArticulo.getInt(1) + ";");

                    //----- ELIMINAR ARTÍCULO -----

                    String sentencia_eliminar_articulo =
"DELETE FROM articulos WHERE descripcionArticulo = '"+
comboArticulo.getSelectedItem().toString() + ";";

                    System.out.println(sentencia_eliminar_articulo);

```

```

base_datos.agregar_objeto(sentencia_eliminar_articulo);

        base_datos.cierra_conexion();
        dialogo1.setVisible(false);
        Eliminar_Articulo.this.dispose();
        new Eliminar_Articulo().setVisible(true);

    } catch (SQLException e1)
    {
        System.out.println(e1);
        label.setText("Error al eliminar el
artículo");
    }
}

});
//-----

//-----RELLENA JCOMBOBOX DE ARTICULOS-----
comboArticulo.addItem("Elegir uno...");
rellena_jcombobox_articulos();
//-----

this.setVisible(false);
this.setResizable(false);
this.setLocationRelativeTo(null);
}

public void windowActivated(WindowEvent e)
{}

public void windowClosed(WindowEvent e)
{
    base_datos.cierra_conexion();
    this.dispose();
}

public void windowClosing(WindowEvent e)
{}

public void windowDeactivated(WindowEvent e)
{}

public void windowDeiconified(WindowEvent e)
{}

public void windowIconified(WindowEvent e)
{}

public void windowOpened(WindowEvent e)
{}

public void rellena_jcombobox_articulos()
{
    resultset1 = base_datos.obtener_objetos("SELECT
descripcionArticulo FROM articulos ORDER BY 1;");

```

```

        try //USAMOS UN WHILE PARA RELLENAR EL JCOMBOX CON LOS
RESULTADOS DEL RESULTSET
        {
            while(resultSet1.next())
            {

                comboArticulo.addItem(resultSet1.getString("descripcionArticulo"
));
            }
        } catch (SQLException e)
        {
            e.printStackTrace();
        }
    }
}

```

Modificar Artículo

```

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.Panel;
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JSeparator;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JDialog;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.awt.event.ActionEvent;

public class Modificar_Articulo extends JFrame implements
WindowListener, ItemListener
{

    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private Conecta_BBDD base_datos;
    private JPanel contentPane;
    private JTextField textfieldDescripcion;
    private JTextField textfieldPrecio;
    private JTextField textfieldCantidad;
    private JLabel labelPrecio;
    private JLabel label;
}

```

```

private JLabel labelCantidad;
private JLabel labelModificarArticulo;
private JButton buttonCancelar;
private JButton buttonGuardar;
private JLabel labelSeleccionarArticulo;
private JComboBox<String> comboArticulo;
private JDialog dialogo1;
private Panel panel1;
private Panel panel2;
private JLabel mensaje1;
private JButton btn_dialogo_1;
private ResultSet resultset1;
private ResultSet resultset2;
private String nombre_articulo;
/**
 * Create the frame.
 */
public Modificar_Articulo()
{
    base_datos = new Conecta_BBDD();

    setIconImage(Toolkit.getDefaultToolkit().getImage(Nuevo_Articulo
.class.getResource("/dise\u00F1o_interfaces/SHOP.png")));
    setBounds(100, 100, 500, 470);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel labelDescripcin = new JLabel("Descripci\u00F3n:");
    labelDescripcin.setFont(new Font("Microsoft New Tai Lue",
Font.PLAIN, 15));
    labelDescripcin.setBounds(121, 160, 88, 24);
    contentPane.add(labelDescripcin);

    textfieldDescripcion = new JTextField();
    textfieldDescripcion.setBounds(251, 160, 153, 24);
    contentPane.add(textfieldDescripcion);
    textfieldDescripcion.setColumns(10);

    textfieldPrecio = new JTextField();
    textfieldPrecio.setColumns(10);
    textfieldPrecio.setBounds(252, 211, 72, 24);
    contentPane.add(textfieldPrecio);

    labelPrecio = new JLabel("Precio:");
    labelPrecio.setFont(new Font("Microsoft New Tai Lue",
Font.PLAIN, 15));
    labelPrecio.setBounds(154, 211, 53, 24);
    contentPane.add(labelPrecio);

    label = new JLabel("\u20AC");
    label.setFont(new Font("Microsoft New Tai Lue", Font.PLAIN,
15));
    label.setBounds(328, 211, 25, 24);
    contentPane.add(label);

    labelCantidad = new JLabel("Cantidad:");

```



```

        labelCantidad.setFont(new Font("Microsoft New Tai Lue",
Font.PLAIN, 15));
        labelCantidad.setBounds(140, 260, 63, 24);
        contentPane.add(labelCantidad);

        textfieldCantidad = new JTextField();
        textfieldCantidad.setColumns(10);
        textfieldCantidad.setBounds(251, 260, 73, 24);
        contentPane.add(textfieldCantidad);

        labelModificarArticulo = new JLabel("MODIFICAR
ART\u00CDCULO");
        labelModificarArticulo.setFont(new Font("Microsoft New Tai
Lue", Font.BOLD, 15));
        labelModificarArticulo.setBounds(169, 11, 170, 24);
        contentPane.add(labelModificarArticulo);

        JSeparator separator = new JSeparator();
        separator.setForeground(Color.DARK_GRAY);
        separator.setBounds(10, 38, 468, 2);
        contentPane.add(separator);

        buttonCancelar = new JButton("Cancelar");
        buttonCancelar.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e)
            {
                Modificar_Articulo.this.setVisible(false);
                Modificar_Articulo.this.dispose();
            }
        });
        buttonCancelar.setBounds(292, 397, 89, 23);
        contentPane.add(buttonCancelar);

        buttonGuardar = new JButton("Guardar");
        buttonGuardar.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e)
            {
                if(!comboArticulo.getSelectedItem().equals("Elegir uno..."))
                {
                    String sentencia_modificar_podologo =
"UPDATE articulos SET descripcionArticulo = '" +
textfieldDescripcion.getText() + "', precioArticulo = " +
comprueba_cantidad(textfieldPrecio.getText()) + ", stockArticulo = " +
textfieldCantidad.getText() + " WHERE descripcionArticulo = '" +
nombre_articulo + "';";

                    System.out.println(sentencia_modificar_podologo);

                    base_datos.agregar_objeto(sentencia_modificar_podologo);
                    dialogo1.setVisible(true);
                }
            }
        });
        buttonGuardar.setBounds(389, 397, 89, 23);
        contentPane.add(buttonGuardar);

        labelSeleccionarArticulo = new JLabel("Seleccionar
Art\u00EDculo: ");

```

```

        labelSeleccionarArticulo.setFont(new Font("Microsoft New
Tai Lue", Font.PLAIN, 15));
        labelSeleccionarArticulo.setBounds(66, 110, 153, 24);
        contentPane.add(labelSeleccionarArticulo);

        comboArticulo = new JComboBox<String>();
        comboArticulo.setBounds(250, 112, 154, 20);
        contentPane.add(comboArticulo);

        //-----Dialog-----
        dialogol = new JDialog(this, "", true);
        mensaje1 = new JLabel("Artículo modificado con éxito.");
        dialogol.getContentPane().setLayout(new BorderLayout());
        dialogol.setSize(500, 470);
        dialogol.setResizable(false);
        dialogol.setLocationRelativeTo(null);
        dialogol.addWindowListener(this);
        panel1 = new Panel();
        panel2 = new Panel();
        panel1.setLayout(new FlowLayout());
        panel2.setLayout(new FlowLayout());
        panel1.add(mensaje1);
        btn_dialogo_1 = new JButton("Aceptar");
        panel2.add(btn_dialogo_1);
        dialogol.getContentPane().add(BorderLayout.NORTH, panel1);
        dialogol.getContentPane().add(BorderLayout.CENTER, panel2);
        btn_dialogo_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                base_datos.cierra_conexion();
                dialogol.setVisible(false);
                Modificar_Articulo.this.dispose();
                new Modificar_Articulo().setVisible(true);

            }
        });
        //-----
        //-----RELLENA JCOMBOBOX DE ARTICULOS-----
        comboArticulo.addItem("Elegir uno...");
        rellena_jcombobox_articulos();
        //-----

        comboArticulo.addItemListener(this);
        this.addWindowListener(this);
        this.setResizable(false);
        this.setVisible(false);
        this.setLocationRelativeTo(null);
    }

    public void windowActivated(WindowEvent e)
    {}

    public void windowClosed(WindowEvent e)
    {}

    public void windowClosing(WindowEvent e)
    {
        base_datos.cierra_conexion();
        this.dispose();
    }

```

```

    public void windowDeactivated(WindowEvent e)
    {}

    public void windowDeiconified(WindowEvent e)
    {}

    public void windowIconified(WindowEvent e)
    {}

    public void windowOpened(WindowEvent e)
    {}

    public void rellena_jcombobox_articulos()
    {
        resultset1 = base_datos.obtener_objetos("SELECT
descripcionArticulo FROM articulos ORDER BY 1;");

        try //USAMOS UN WHILE PARA RELLENAR EL JCOMBOX CON LOS
RESULTADOS DEL RESULTSET
        {
            while(resultset1.next())
            {

                comboArticulo.addItem(resultset1.getString("descripcionArticulo"
));
            }
        } catch (SQLException e)
        {
            e.printStackTrace();
        }
    }

    public void rellena_todos_los_campos()
    {

        try //USAMOS UN WHILE PARA RELLENAR LOS CAMPOS CON LOS
DATOS DEL RESULTSET
        {
            while(resultset2.next())
            {

                textfieldDescripcion.setText(resultset2.getString("descripcionAr
ticulo"));

                textfieldPrecio.setText(resultset2.getString("precioArticulo"));

                textfieldCantidad.setText(resultset2.getString("stockArticulo"))

;

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

    @Override
    public void itemStateChanged(ItemEvent ie)
    {

```

```

        if(ie.getStateChange() == 1 &&
(!comboArticulo.getSelectedItemAt().equals("Elegir uno..."))) // LA
PRIMERA CONDICIÓN SIRVE PARA SOLUCIONAR EL PROBLEMA DE QUE EL
JCOMBOBOX SE LLAME DOS VECES CON EL ITEMListener, CON LOS CHOICE NO
OCURRE ESTE PROBLEMA
        {
            nombre_articulo = (String)ie.getItem();
            System.out.println("SELECT * FROM articulos WHERE
descripcionArticulo = '" + nombre_articulo + "';");
            resultSet2 = base_datos.obtener_objetos("SELECT *
FROM articulos WHERE descripcionArticulo = '" + (String)ie.getItem()
+ "';"); //ACTUALIZA EL RESULTSET
            rellena_todos_los_campos();
        }else
        {
            textfieldDescripcion.setText("");
            textfieldPrecio.setText("");
            textfieldCantidad.setText("");
        }
    }

    public float comprueba_cantidad(String precio)
    {
        String precio_articulo = precio;
        char[] array_caracteres = precio_articulo.toCharArray();
        String precio_factorizado = "";
        String string_precio_dos_decimales = "";
        boolean llave = true;

        for (int i = 0; i < array_caracteres.length; i++)
        {
            if (array_caracteres[i] == ',')
            {
                precio_factorizado += ".";
            }
            else
            {
                precio_factorizado += array_caracteres[i];
            }
        }

        for (int i = 0; i < precio_factorizado.length(); i++)
        {
            if(llave)
            {
                if(precio_factorizado.charAt(i) == '.')
                {
                    string_precio_dos_decimales +=
precio_factorizado.charAt(i);
                    string_precio_dos_decimales +=
precio_factorizado.charAt(i+1);
                    string_precio_dos_decimales +=
precio_factorizado.charAt(i+2); //ESTO PUEDE DAR PROBLEMAS SI EL
NUMERO INTRODUCIDO POR EL USUARIO TIENE MENOS DE DOS DECIMALES,
CORREGIR EN FUTURAS VERSIONES
                    llave = false;
                }
            }
            else
            {

```

```

                                string_precio_dos_decimales +=
precio_factorizado.charAt(i);

                                }
                                }
                                }
                                return Float.valueOf(string_precio_dos_decimales);
                                }
                                }

```

Ver Artículos

```

import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JSeparator;
import javax.swing.JTable;
import javax.swing.SwingConstants;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableCellRenderer;
import javax.swing.table.DefaultTableModel;

public class Ver_Articulos extends JFrame implements WindowListener
{

    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JTable tabla;
    private ResultSet resultset;
    private Conecta_BBDD base_datos;
    private DefaultTableModel modelo;
    private DefaultTableCellRenderer tcr;
    /**
     * Create the frame.
     */
    public Ver_Articulos()
    {
        base_datos = new Conecta_BBDD();

        setBounds(100, 100, 500, 470);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
    }

```

```

//ARRAY DE STRING CON LAS COLUMNAS DE LA TABLA
String[] columnNames = {"Descripción", "Precio", "Stock"};

//CREAMOS UN MODELO DE DATOS Y LE PASAMOS EL OBJETO QUE
CONTIENE EL NOMBRE DE LAS COLUMNAS
modelo = new DefaultTableModel(null, columnNames);

//SE CREA LA TABLA Y SE LE PASA EL MODELO EN EL CONSTRUCTOR
tabla = new JTable(modelo);

//SE DEFINE LE TAMAÑO
tabla.setPreferredScrollableViewportSize(new Dimension(500,
70));

//CREAMOS UN JSCROLLPANE Y LE AÑADIMOS UN JTABLE
JScrollPane scrollPane = new JScrollPane(tabla);
scrollPane.setBounds(37, 68, 417, 311);

//AGREGAMOS EL SCROLL
getContentPane().add(scrollPane);

//HACEMOS QUE EL TEXTO DENTRO DE LA TABLA SALGA CENTRADO
tcr = new DefaultTableCellRenderer();
tcr.setHorizontalAlignment(SwingConstants.CENTER); //ALINEA
LOS STRINGS EN EL CENTRO DE LA CELDA
tabla.getColumnModel().getColumn(0).setCellRenderer(tcr);
tabla.getColumnModel().getColumn(1).setCellRenderer(tcr);
tabla.getColumnModel().getColumn(2).setCellRenderer(tcr);

JSeparator separator = new JSeparator();
separator.setForeground(Color.DARK_GRAY);
separator.setBounds(10, 35, 468, 2);
contentPane.add(separator);

JLabel lblArticulos = new JLabel("ART\u00CDCULOS");
lblArticulos.setFont(new Font("Microsoft New Tai Lue",
Font.BOLD, 15));
lblArticulos.setBounds(206, 11, 90, 24);
contentPane.add(lblArticulos);

JButton button = new JButton("Volver");
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        base_datos.cierra_conexion();
        Ver_Articulos.this.dispose();
    }
});
button.setBounds(365, 390, 89, 23);
contentPane.add(button);

rellena_tabla();

this.setResizable(false);
this.addWindowListener(this);
this.setVisible(false);
this.setLocationRelativeTo(null);
}

```

```

public void windowActivated(WindowEvent e)
{}

public void windowClosed(WindowEvent e)
{
    base_datos.cierra_conexion();
    this.dispose();
}

public void windowClosing(WindowEvent e)
{}

public void windowDeactivated(WindowEvent e)
{}

public void windowDeiconified(WindowEvent e)
{}

public void windowIconified(WindowEvent e)
{}

public void windowOpened(WindowEvent e)
{}

public void rellena_tabla()
{
    resultset = base_datos.obtener_objetos("SELECT * FROM
articulos ORDER BY 2;");

    try
    {
        while(resultset.next())//USAMOS UN WHILE PARA
RELLENAR LA TABLA
        {
            Object [] fila = new Object[3]; // HAY 3
COLUMNAS

            // SE RELLENA CADA POSICIÓN DEL ARRAY CON UNA
DE LAS COLUMNAS DE LA TABLA EN LA BASE DE DATOS
            for (int i = 0; i < 3; i++)
            {
                fila[i] = resultset.getObject(i+2); // EL
PRIMER INDICE EN EL RS ES EL 1, NO EL 0, POR ESO SE SUMA 1.
            }
            modelo.addRow(fila); // SE AÑADE AL MODELO LA
FILA COMPLETA
        }
    } catch (SQLException e)
    {
        e.printStackTrace();
    }
}
}

```

[Nuevo Ticket](#)

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.List;
import java.awt.Panel;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JSeparator;
import javax.swing.SwingConstants;
import javax.swing.border.EmptyBorder;

public class Nuevo_Ticket extends JFrame implements WindowListener
{

    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JLabel lblCantidad;
    private JLabel lblNuevoArticulo;
    private JDialog dialog01;
    private Panel panel1;
    private Panel panel2;
    private JLabel mensaje1;
    private JButton btn_dialogo_1;
    private JSeparator separator;
    private JLabel lblFechaTicket;
    private JLabel cantidadTotalTicket;
    private List lista;
    private JComboBox<String> combo_articulo;
    private JButton btnAgregarArticulo;
    private JComboBox<String> combo_cantidad;
    private JComboBox<String> combo_dia;
    private JComboBox<String> combo_mes;
    private JComboBox<String> combo_anyo;
    private ResultSet resultset1;
    private ResultSet resultset2;
    private ResultSet resultset_idTicket;
    long idTicket;
    private double cantidad_total;
    private Conecta_BBDD base_datos;
    private ArrayList<String> lista_de_articulos;
    private ArrayList<Integer> lista_de_cantidades_de_articulos;
    private JLabel mensaje_aviso;
```



```

    boolean llave = false;
    private int cantidad_int;
    private JLabel label_idTicket;
    /**
     * Create the frame.
     */
    public Nuevo_Ticket()
    {

        base_datos = new Conecta_BBDD();

        lista_de_cantidades_de_articulos = new
ArrayList<Integer>();

        setIconImage(Toolkit.getDefaultToolkit().getImage(Nuevo_Articulo
.class.getResource("/disse\u00F1o/interfaces/SHOP.png")));
        setBounds(100, 100, 512, 471);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        separator = new JSeparator();
        separator.setForeground(Color.DARK_GRAY);
        separator.setBounds(28, 38, 468, 2);
        contentPane.add(separator);

        lblNuevoArticulo = new JLabel("NUEVO TICKET");
        lblNuevoArticulo.setFont(new Font("Microsoft New Tai Lue",
Font.BOLD, 15));
        lblNuevoArticulo.setBounds(204, 11, 125, 24);
        contentPane.add(lblNuevoArticulo);

        lblFechaTicket = new JLabel("Fecha Ticket:");
        lblFechaTicket.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblFechaTicket.setBounds(28, 60, 88, 24);
        contentPane.add(lblFechaTicket);

        lblCantidad = new JLabel("Precio Total: ",
SwingConstants.CENTER);
        lblCantidad.setForeground(Color.BLUE);
        lblCantidad.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblCantidad.setBounds(42, 397, 125, 24);
        contentPane.add(lblCantidad);

        cantidadTotalTicket = new JLabel("0€",
SwingConstants.LEFT);
        cantidadTotalTicket.setForeground(Color.BLUE);
        cantidadTotalTicket.setFont(new Font("Tahoma", Font.BOLD,
15));

        cantidadTotalTicket.setBounds(157, 397, 135, 24);
        contentPane.add(cantidadTotalTicket);

        lista = new List();
        lista.setBounds(53, 210, 414, 162);
        contentPane.add(lista);

        combo_articulo = new JComboBox<String> ();
        combo_articulo.setBounds(124, 112, 168, 20);

```

```

        contentPane.add(combo_articulo);

        btnAgregarArticulo = new JButton("A\u00F1adir");
        btnAgregarArticulo.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e)
            {
                agrega_articulo_a_la_lista();
            }
        });
        btnAgregarArticulo.setBounds(290, 181, 86, 23);
        contentPane.add(btnAgregarArticulo);

        JLabel label = new JLabel("/");
        label.setFont(new Font("Microsoft New Tai Lue", Font.PLAIN,
15));
        label.setBounds(176, 60, 13, 24);
        contentPane.add(label);

        JLabel label_1 = new JLabel("/");
        label_1.setFont(new Font("Microsoft New Tai Lue",
Font.PLAIN, 15));
        label_1.setBounds(297, 60, 13, 24);
        contentPane.add(label_1);

        JLabel lblDia = new JLabel("D\u00E9");
        lblDia.setBounds(143, 44, 18, 14);
        contentPane.add(lblDia);

        JLabel lblMes = new JLabel("Mes");
        lblMes.setBounds(228, 44, 28, 14);
        contentPane.add(lblMes);

        JLabel lblAnyo = new JLabel("A\u00F1o");
        lblAnyo.setBounds(328, 44, 28, 14);
        contentPane.add(lblAnyo);

        JButton btnGuardar = new JButton("Guardar");
        btnGuardar.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                mensaje_aviso.setText(""); //LIMPIA EL MENSAJE
                if(lista_de_articulos.size() > 0) //COMPROBAMOS
QUE HAYA ALGÚN ARTÍCULO EN LA LISTA
                {
                    //----- AGREGAR EL TICKET A LA BBDD
                    -----
                    try
                    {
                        String sentencia_agregar_ticket =
"INSERT INTO tickets (fechaTicket, articulosTicket, totalTicket)
VALUES ('"+ combo_anyo.getSelectedItemAt().toString() + "-" +
devuelve_mes(combo_mes.getSelectedItemAt().toString()) + "-" +
combo_dia.getSelectedItemAt().toString() + "', " +
lista_de_articulos.size() + ", " + (float)cantidad_total + ");";
                        if
(base_datos.agregar_fecha_ticket(sentencia_agregar_ticket) ==
true)//AGREGA EL TICKET A LA BBDD SI LA FECHA ES CORRECTA
                        {

```

```

//----- OBTIENE EL ID
DE TODOS LOS OJETOS DE LA LISTA LA CANTIDAD Y LOS AGREGA AL TICKET EN
LA BD -----
for (int i = 0; i <
lista_de_articulos.size(); i++)
{
    System.out.println("SELECT idArticulo FROM articulos WHERE
descripcionArticulo = '"+ lista_de_articulos.get(i) + "';"); //OBTIENE
ID DEL ARTICULO
    ResultSet
resultset_insertaArticulos = base_datos.obtener_objetos("SELECT
idArticulo FROM articulos WHERE descripcionArticulo = '"+
lista_de_articulos.get(i) + "';");

    try
    {

        resultset_insertaArticulos.first();

        System.out.println("El id del articulo es: " +
resultset_insertaArticulos.getInt(1) + " Y la cantidad de dicho
artículo es: " + lista_de_cantidades_de_articulos.get(i) );

//-----
AGREGA LOS ARTICULOS AL TICKET CORRESPONDIENTE -----
        String
sentencia_agregar_articulos_al_ticket = "INSERT INTO contiene (
idArticuloFK, idTicketFK, cantidadArticulo) VALUES (" +
resultset_insertaArticulos.getInt(1) + ", " + idTicket + ", " +
lista_de_cantidades_de_articulos.get(i) + ");";

        base_datos.agregar_objeto(sentencia_agregar_articulos_al_ticket)
;

        base_datos.agregar_objeto("UPDATE articulos SET stockArticulo =
stockArticulo - " + lista_de_cantidades_de_articulos.get(i) + " WHERE
idArticulo = " + resultset_insertaArticulos.getInt(1) + " "); //BAJA
EL STOCK DEL ARTÍCULO

    } catch (SQLException
e2)
    {

        mensaje_aviso.setText("Error al agregar los artículos al ticket
en la BBDD.");

        e2.printStackTrace();

    }

}
dialogo1.setVisible(true);
}
else
{
    mensaje_aviso.setText("La
fecha elegida no es correcta.");
}

```

```

        } catch (Exception e1)
        {
            mensaje_aviso.setText("Introduzca
una fecha válida para el ticket.");
            e1.printStackTrace();
        }
    }
    else
    {
        mensaje_aviso.setText("Debe introducir al
menos un artículo en el ticket.");
    }
}

});
btnGuardar.setBounds(386, 400, 81, 23);
contentPane.add(btnGuardar);

JButton btnCancelar = new JButton("Cancelar");
btnCancelar.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        Nuevo_Ticket.this.dispose();
    }
});
btnCancelar.setBounds(290, 400, 88, 23);
contentPane.add(btnCancelar);

JButton btnEliminarArticulo = new JButton("Eliminar");
btnEliminarArticulo.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        elimina_articulo_de_lista();
    }
});
btnEliminarArticulo.setBounds(386, 181, 81, 23);
contentPane.add(btnEliminarArticulo);

combo_cantidad = new JComboBox<String>();
combo_cantidad.setBounds(126, 143, 60, 20);
contentPane.add(combo_cantidad);

JLabel lblArticulo = new JLabel("Artículo:");
lblArticulo.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblArticulo.setBounds(63, 111, 69, 19);
contentPane.add(lblArticulo);

JLabel lblCantidad_1 = new JLabel("Cantidad:");
lblCantidad_1.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblCantidad_1.setBounds(54, 142, 69, 20);
contentPane.add(lblCantidad_1);

combo_dia = new JComboBox<String>();
combo_dia.setBounds(126, 62, 45, 22);
contentPane.add(combo_dia);

combo_mes = new JComboBox<String>();
combo_mes.setBounds(190, 62, 102, 22);

```

```

contentPane.add(combo_mes);

combo_anyo = new JComboBox<String>();
combo_anyo.setBounds(308, 62, 81, 22);
contentPane.add(combo_anyo);

mensaje_avisos = new JLabel("", SwingConstants.CENTER);
mensaje_avisos.setForeground(Color.RED);
mensaje_avisos.setFont(new Font("Microsoft New Tai Lue",
Font.BOLD, 12));
mensaje_avisos.setBounds(64, 372, 391, 24);
contentPane.add(mensaje_avisos);

label_idTicket = new JLabel("", SwingConstants.LEFT);
label_idTicket.setForeground(Color.RED);
label_idTicket.setFont(new Font("Microsoft New Tai Lue",
Font.BOLD | Font.ITALIC, 12));
label_idTicket.setBounds(28, 11, 139, 24);
contentPane.add(label_idTicket);

//-----Dialog-----
dialogo1 = new JDialog(this, "", true);
mensaje1 = new JLabel("Ticket creado con éxito.");
dialogo1.addWindowListener(new WindowListener()
{

    public void windowActivated(WindowEvent e)
    {}
    public void windowClosed(WindowEvent e)
    {}
    public void windowClosing(WindowEvent e)
    {
        dialogo1.setVisible(false);
        base_datos.cierra_conexion();
        Nuevo_Ticket.this.dispose();
    }

    public void windowDeactivated(WindowEvent e)
    {}
    public void windowDeiconified(WindowEvent e)
    {}
    public void windowIconified(WindowEvent e)
    {}
    public void windowOpened(WindowEvent e)
    {}
});

dialogo1.getContentPane().setLayout(new BorderLayout());
dialogo1.setSize(500, 470);
dialogo1.setResizable(false);
dialogo1.setLocationRelativeTo(null);
dialogo1.addWindowListener(this);
panel1 = new Panel();
panel2 = new Panel();
panel1.setLayout(new FlowLayout());
panel2.setLayout(new FlowLayout());
panel1.add(mensaje1);
btn_dialogo_1 = new JButton("Aceptar");
panel2.add(btn_dialogo_1);

```

```

dialogo1.getContentPane().add(BorderLayout.NORTH, panel1);
dialogo1.getContentPane().add(BorderLayout.CENTER, panel2);
btn_dialogo_1.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e) {

        dialogo1.setVisible(false);
        base_datos.cierra_conexion();
        Nuevo_Ticket.this.dispose();
        new Nuevo_Ticket().setVisible(true);

    }

});

//-----

cantidad_total = 0.0;
lista_de_articulos = new ArrayList<String>();

idTicket = dame_idTicke();
label_idTicket.setText("REF. TICKET: " + idTicket);

JLabel lblListaCompra = new JLabel("--LISTA DE
ART\u00CDCULOS--", SwingConstants.CENTER);
lblListaCompra.setForeground(Color.GRAY);
lblListaCompra.setFont(new Font("Microsoft New Tai Lue",
Font.BOLD | Font.ITALIC, 12));
lblListaCompra.setBounds(124, 190, 139, 24);
contentPane.add(lblListaCompra);

rellena_dias();
rellena_meses();
rellena_anyos();
rellena_articulos();
rellena_cantidad();

this.setVisible(false);
this.setLocationRelativeTo(null);
}

public void windowActivated(WindowEvent e)
{}

public void windowClosed(WindowEvent e)
{}

public void windowClosing(WindowEvent e)
{
    base_datos.cierra_conexion();
    this.dispose();
}

public void windowDeactivated(WindowEvent e)
{}

public void windowDeiconified(WindowEvent e)
{}

```

```

public void windowIconified(WindowEvent e)
{}

public void windowOpened(WindowEvent e)
{}

public void rellena_dias()
{
    for (int i = 0; i < 31; i++)
    {
        combo_dia.addItem(""+(i+1));
    }
}

public void rellena_meses()
{
    String [] meses =
{"Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto", "Se
ptiembre", "Octubre", "Noviembre", "Diciembre"};

    for (int i = 0; i < 12; i++)
    {
        combo_mes.addItem(meses[i]);
    }
}

public void rellena_anyos()
{
    for (int i = 1900; i < 2050; i++)
    {
        combo_anyo.addItem(""+i);
    }
    combo_anyo.setSelectedIndex(119);
}

public void rellena_cantidad()
{
    for (int i = 0; i < 20; i++)
    {
        combo_cantidad.addItem(""+(i+1));
    }
}

public void rellena_articulos()
{
    resultset1 = base_datos.obtener_objetos("SELECT
descripcionArticulo FROM articulos ORDER BY 1;");

    try //USAMOS UN WHILE PARA RELLENAR EL JCOMBOX CON LOS
RESULTADOS DEL RESULTSET
    {
        while(resultset1.next())
        {
            combo_articulo.addItem(resultset1.getString("descripcionArticulo
"));
        }
    } catch (SQLException e)
    {
        e.printStackTrace();
    }
}

```

```

    }

    public double dame_precio_articulo(String articulo_seleccionado)
    {
        String articulo = articulo_seleccionado;
        double precio_articulo = 0.0;
        resultset2 = base_datos.obtener_objetos("SELECT
precioArticulo FROM articulos WHERE descripcionArticulo = '" +
articulo + "';");
        try
        {
            while(resultset2.next())
            {
                precio_articulo = resultset2.getDouble(1);
            }
        } catch (SQLException e)
        {
            System.out.println(e);
        }
        return precio_articulo;
    }

    public String devuelve_articulo_seleccionado_de_la_lista()
    {
        String texto_lista = lista.getSelectedItem().toString();
        String [] descompone_string;
        descompone_string = texto_lista.split(" -- ");
        return descompone_string[0];
    }

    public void agrega_articulo_a_la_lista()
    {
        mensaje_aviso.setText("");
        cantidad_int =
Integer.valueOf(combo_cantidad.getSelectedItem().toString());

        if(lista_de_articulos.contains(combo_articulo.getSelectedItem().
toString()))
        {
            mensaje_aviso.setText("El artículo que está agregando
ya existe.");
        }
        else
        {
            //----- AGREGA A LA LISTA EL ELEMENTO
SELECCIONADO -----
            lista.add(combo_articulo.getSelectedItem().toString()
+ " -- " + combo_cantidad.getSelectedItem().toString() + " -- " +
(String.format("%.2f",
(dame_precio_articulo(combo_articulo.getSelectedItem().toString()) *
cantidad_int))+ "€"));
            //----- AGREGA AL ARRAYLIST EL ELEMENTO
SELECCIONADO -----
            lista_de_articulos.add(combo_articulo.getSelectedItem().toString
());

```



```

//----- AGREGA AL ARRAYLIST LA CANTIDAD DEL
ELEMENTO -----
        lista_de_cantidades_de_articulos.add(cantidad_int);

//-----AUMENTA TOTAL DE TICKET-----
-----
        cantidad_total +=
dame_precio_articulo(combo_articulo.getSelectedItem().toString()) *
cantidad_int ; //AUMENTA EL VALOR DEL TICKET CON EL PRECIO
SELECCIONADO, TENIENDO EN CUENTA PRECIO Y CANTIDAD SELECCIONADA DEL
ARTÍCULO
        cantidadTotalTicket.setText((String.format("%.2f",
cantidad_total)+"€"));
//-----
-----
    }

}

public void elimina_articulo_de_lista()
{

    try
    {
        mensaje_aviso.setText("");
        int indice = 0;
        indice = lista.getSelectedIndex(); //RECOGEMOS EL
INDICE DEL LA LÍNEA QUE TENEMOS SELECCIONADA EN EL LIST, YA QUE ESTÉ
MÉTODO SI SE USA UNA VEZ, LUEGO YA NO DEVUELVE EL ÍNDICE QUE SE
ENCUENTRA SELECCIONADO, POR LO QUE LO ALMACENAMOS PARA USARLO VARIAS
VECES

        //----- DISMINUYE TICKET -----
        disminuye_ticket(indice);

        //----- ELIMINA DEL ARRAYLIST EL ELEMENTO
SELECCIONADO -----
        try
        {
            for (int i = 0; i < lista_de_articulos.size();
i++)
            {

                if(devuelve_articulo_seleccionado_de_la_lista().equals(lista_de_
articulos.get(i).toString()))
                {
                    lista_de_articulos.remove(indice);

                    lista_de_cantidades_de_articulos.remove(indice); // ELIMINA DEL
ARRAYLIST LA CANTIDAD DEL ARTÍCULO SELECCIONADO
                }
            }
        }
        catch (NullPointerException e)
        {
            System.err.println("Error al eliminar el
elemento del ArrayList.");
        }
    }
}

```

```

//----- ELIMINA DE LA LISTA EL ELEMENTO
SELECCIONADO -----
    try
    {
        lista.remove(indice);

    } catch (ArrayIndexOutOfBoundsException e) {

        System.err.println("Error al eliminar el
elemento seleccionado de la lista.");
        mensaje_aviso.setText("Selecciona el archivo
que quieres eliminar");
    }

    } catch (Exception e)
    {
        mensaje_aviso.setText("Elemento a eliminar no
seleccionado.");
    }
}

public int devuelve_mes(String mes)
{

    String mes_del_anyo = mes;
    switch (mes_del_anyo)
    {
    case "Enero":
        return 1;
    case "Febrero":
        return 2;
    case "Marzo":
        return 3;
    case "Abril":
        return 4;
    case "Mayo":
        return 5;
    case "Junio":
        return 6;
    case "Julio":
        return 7;
    case "Agosto":
        return 8;
    case "Septiembre":
        return 9;
    case "Octubre":
        return 10;
    case "Noviembre":
        return 11;
    case "Diciembre":
        return 12;
    default:
        return 0;
    }
}

public void disminuye_ticket(int linea_seleccionada)

```

```

        {
            int indice = linea_seleccionada;

            String array = lista.getItem(indice);
            String[] string_cantidad_con_simbolo = array.split(" --
");
            String string_cantidad_sin_simbolo =
string_cantidad_con_simbolo[2];
            string_cantidad_sin_simbolo =
string_cantidad_sin_simbolo.substring(0,
string_cantidad_sin_simbolo.length()-1);
            String cantidad_con_punto =
string_cantidad_sin_simbolo.replaceAll(",", ".");

            double cantidad_double =
Double.parseDouble(cantidad_con_punto);

            cantidad_total -= cantidad_double;

            if(cantidad_total<0)
            {
                cantidad_total = 0;
            }

            cantidadTotalTicket.setText((String.format("%.2f",
cantidad_total)+"€"));
        }

        public long dame_idTicke() {

            resultset_idTicket = base_datos.obtener_objetos("SELECT
idTicket FROM tickets ORDER BY idTicket DESC;");
            long id_nuevo_ticket = 0;

            try
            {
                resultset_idTicket.first();
                id_nuevo_ticket = resultset_idTicket.getInt(1);

            } catch (SQLException e)
            {
                System.out.println("Error al obtener el id del ticket
de la base de datos.");
                System.out.println("O bien no existe todavía ningún
ticket en la base de datos.");
            }

            return id_nuevo_ticket+1;

        }
    }
}

```

Ver Tickets

```
import java.awt.Color;
```

```

import java.awt.Dimension;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JSeparator;
import javax.swing.JTable;
import javax.swing.SwingConstants;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableCellRenderer;
import javax.swing.table.DefaultTableModel;

public class Ver_Tickets extends JFrame implements WindowListener,
ItemListener
{

    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JTable tabla;
    private ResultSet resultset;
    private Conecta_BBDD base_datos ;
    private DefaultTableModel modelo;
    private DefaultTableCellRenderer tcr;
    private JComboBox<String>comboTicket;
    private ResultSet resulset_datos_ticket;
    private ResultSet descripcionArticulo;
    private JLabel label_Ticket;
    private JLabel label_Articulos;
    private JLabel label_Total;
    private int id_TicketSeleccionado;
    /**
     * Create the frame.
     */
    public Ver_Tickets ()
    {

        base_datos = new Conecta_BBDD();

        setBounds(100, 100, 500, 470);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        //ARRAY DE STRING CON LAS COLUMNAS DE LA TABLA

```

```

String[] columnNames = {"Artículo", "Cantidad"};

//CREAMOS UN MODELO DE DATOS Y LE PASAMOS EL OBJETO QUE
CONTIENE EL NOMBRE DE LAS COLUMNAS
modelo = new DefaultTableModel(null, columnNames);

//SE CREA LA TABLA Y SE LE PASA EL MODELO EN EL CONSTRUCTOR
tabla = new JTable(modelo);

//SE DEFINE LE TAMAÑO
tabla.setPreferredScrollableViewportSize(new Dimension(500,
70));

//CREAMOS UN JSCROLLPANE Y LE AÑADIMOS UN JTABLE
JScrollPane scrollPane = new JScrollPane(tabla);
scrollPane.setBounds(37, 91, 423, 246);

//AGREGAMOS EL SCROLL
getContentPane().add(scrollPane);

//HACEMOS QUE EL TEXTO DENTRO DE LA TABLA SALGA CENTRADO
tcr = new DefaultTableCellRenderer();
tcr.setHorizontalAlignment(SwingConstants.CENTER); //ALINEA
LOS STRINGS EN EL CENTRO DE LA CELDA
tabla.getColumnModel().getColumn(0).setCellRenderer(tcr);
tabla.getColumnModel().getColumn(1).setCellRenderer(tcr);

JSeparator separator = new JSeparator();
separator.setForeground(Color.DARK_GRAY);
separator.setBounds(10, 35, 468, 2);
contentPane.add(separator);

JLabel lblArticulos = new JLabel("TICKETS",
SwingConstants.CENTER);
lblArticulos.setFont(new Font("Microsoft New Tai Lue",
Font.BOLD, 15));
lblArticulos.setBounds(199, 11, 90, 24);
contentPane.add(lblArticulos);

JButton button = new JButton("Volver");
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        Ver_Tickets.this.dispose();
    }
});
button.setBounds(371, 400, 89, 23);
contentPane.add(button);

comboTicket = new JComboBox<String>();
comboTicket.setBounds(243, 48, 103, 20);
contentPane.add(comboTicket);

JLabel lblRefTicket = new JLabel("REF. TICKET:",
SwingConstants.CENTER);
lblRefTicket.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblRefTicket.setBounds(130, 48, 103, 19);
contentPane.add(lblRefTicket);

```

```

        label_Ticket = new JLabel("Fecha Ticket:");
        label_Ticket.setForeground(Color.BLUE);
        label_Ticket.setFont(new Font("Tahoma", Font.BOLD, 15));
        label_Ticket.setBounds(37, 348, 217, 24);
        contentPane.add(label_Ticket);

        label_Articulos = new JLabel("N\u00F3mero de
Art\u00E9culos: ");
        label_Articulos.setForeground(Color.BLUE);
        label_Articulos.setFont(new Font("Tahoma", Font.BOLD, 15));
        label_Articulos.setBounds(37, 373, 217, 24);
        contentPane.add(label_Articulos);

        label_Total = new JLabel("Total:");
        label_Total.setForeground(Color.BLUE);
        label_Total.setFont(new Font("Tahoma", Font.BOLD, 15));
        label_Total.setBounds(37, 397, 217, 24);
        contentPane.add(label_Total);

        this.setResizable(false);
        this.addWindowListener(this);
        this.setVisible(false);
        this.setLocationRelativeTo(null);

        comboTicket.addItem("Elegir uno...");
        rellena_tickets();
        comboTicket.addItemListener(this);
    }

    public void windowActivated(WindowEvent e)
    {}

    public void windowClosed(WindowEvent e)
    {
        this.dispose();
    }

    public void windowClosing(WindowEvent e)
    {}

    public void windowDeactivated(WindowEvent e)
    {}

    public void windowDeiconified(WindowEvent e)
    {}

    public void windowIconified(WindowEvent e)
    {}

    public void windowOpened(WindowEvent e)
    {}

    public void rellena_tabla()
    {

        resultset =base_datos.obtener_objetos("SELECT * FROM
contiene WHERE idTicketFK = "+ id_TicketSeleccionado + ";" );

        try
        {

```

```

        while(resultSet.next())//USAMOS UN WHILE PARA
RELLENAR LA TABLA
        {
            Object [] fila = new Object[2]; // HAY 3
COLUMNAS

            descripcionArticulo =
base_datos.obtener_objetos("SELECT descripcionArticulo FROM articulos
WHERE idArticulo="+ resultSet.getInt(1) +";");
            descripcionArticulo.next();
            fila[0] = descripcionArticulo.getString(1);

            fila[1] = resultSet.getObject(3);

            modelo.addRow(fila); // SE AÑADE AL MODELO LA
FILA COMPLETA
        }
    } catch (SQLException e)
    {
        e.printStackTrace();
    }
}

public void rellena_tickets()
{
    String sentencia = "SELECT * FROM tickets;";
    resulset_datos_ticket =
base_datos.obtener_objetos(sentencia);

    try
    {
        while(resulset_datos_ticket.next())
        {

            comboTicket.addItem(resulset_datos_ticket.getString(1));

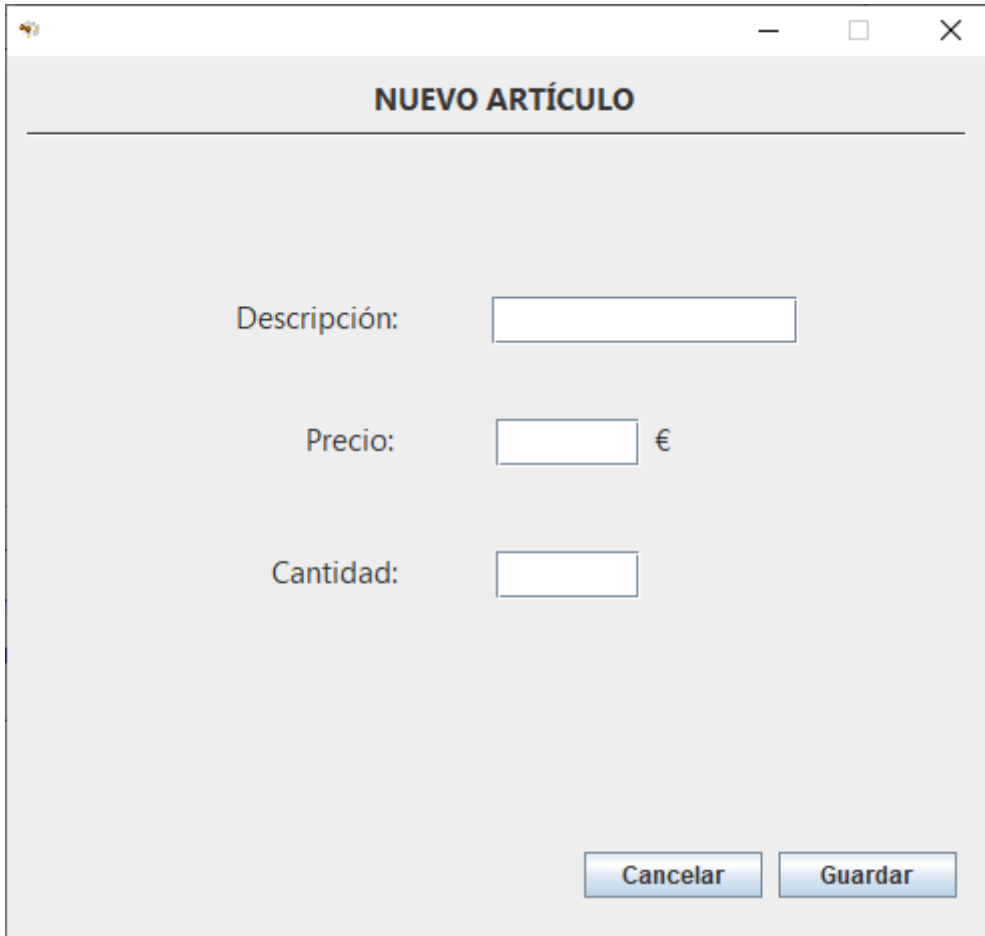
        }
    } catch (SQLException e)
    {
        System.out.println("Error al rellena los tickets en
el JComboBox.");
    }
}

public void itemStateChanged(ItemEvent ie)
{
    if(ie.getStateChange()==1 &&
(!comboTicket.getSelectedItem().toString().equals("Elegir uno...")))
// LA PRIMERA CONDICIÓN SIRVE PARA SOLUCIONAR EL PROBLEMA DE QUE EL
JCOMBOBOX SE LLAME DOS VECES CON EL ITEMLISTENER, CON LOS CHOICE NO
OCURRE ESTE PROBLEMA
    {
        id_TicketSeleccionado =
Integer.parseInt(comboTicket.getSelectedItem().toString());

```

```
String sentencia = "SELECT * FROM tickets WHERE  
idTicket = " + id_TicketSeleccionado + ";";  
resultSet_datos_ticket =  
base_datos.obtener_objetos(sentencia);  
  
try  
{  
    resultSet_datos_ticket.next();  
  
    // INDICAMOS EL NUEVO FORMATO PARA LA FECHA  
    SimpleDateFormat formatter = new  
SimpleDateFormat("dd/MM/yyyy");  
    String fechaTexto =  
formatter.format(resultSet_datos_ticket.getDate(2));  
  
    label_Ticket.setText("Fecha Ticket:  
"+fechaTexto);  
    label_Articulos.setText("Num. Artículos:  
"+resultSet_datos_ticket.getString(3));  
    label_Total.setText("Total:  
"+resultSet_datos_ticket.getString(4)+"€");  
  
    modelo.setRowCount(0); //LIMPIAMOS LA TABLA  
    rellena_tabla();  
}  
catch (SQLException e)  
{  
    System.out.println("Error al obtener los datos  
del ticket seleccionado.");  
}  
}  
}
```

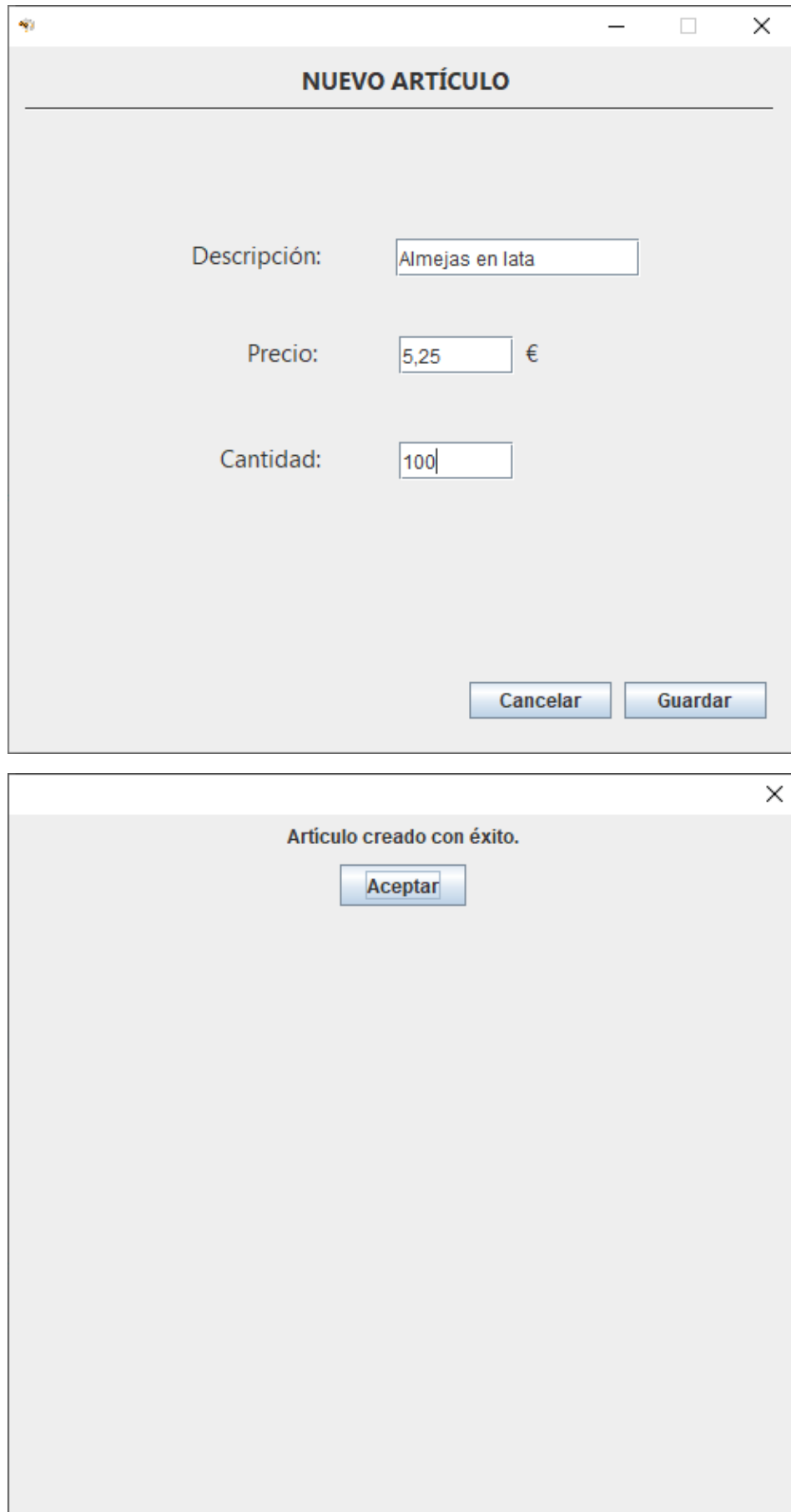

Pantallazos



The screenshot shows a window titled "NUEVO ARTÍCULO" with a light gray background. It contains three input fields: "Descripción:" with a long text box, "Precio:" with a text box followed by a "€" symbol, and "Cantidad:" with a text box. At the bottom right, there are two buttons: "Cancelar" and "Guardar".

En esta pantalla podremos insertar artículos en la BBDD, para ello tendremos que dar una descripción, un precio y una cantidad. El usuario podrá introducir una cantidad decimal usando una coma, o bien un punto. Ejemplo: 10.25 o 10,25. En caso de introducir una cantidad con decimales, un requisito será tener dos cifras decimales, en futuras versiones corregiremos esto.

La cantidad es el stock que tenemos en nuestro almacén de dicho artículo. En caso de que el usuario introduzca un número decimal, se redondea la cifra introducida.



The image shows two overlapping windows from a software application. The top window, titled 'NUEVO ARTÍCULO', is a form for adding a new item. It contains three input fields: 'Descripción:' with the value 'Almejas en lata', 'Precio:' with the value '5,25' followed by a Euro symbol '€', and 'Cantidad:' with the value '100'. At the bottom right of this window are two buttons: 'Cancelar' and 'Guardar'. The bottom window is a smaller dialog box with the title bar 'Artículo creado con éxito.' and a single 'Aceptar' button in the center.

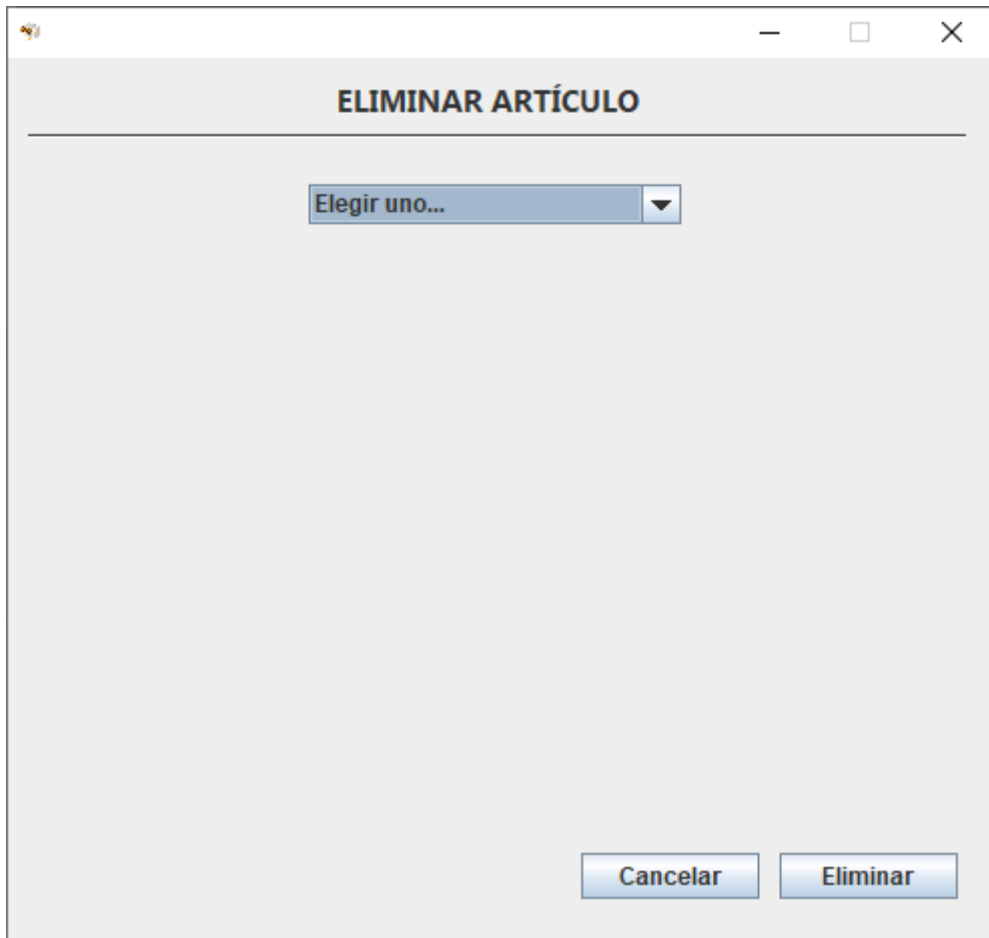
NUEVO ARTÍCULO

Descripción:

Precio: €

Cantidad:

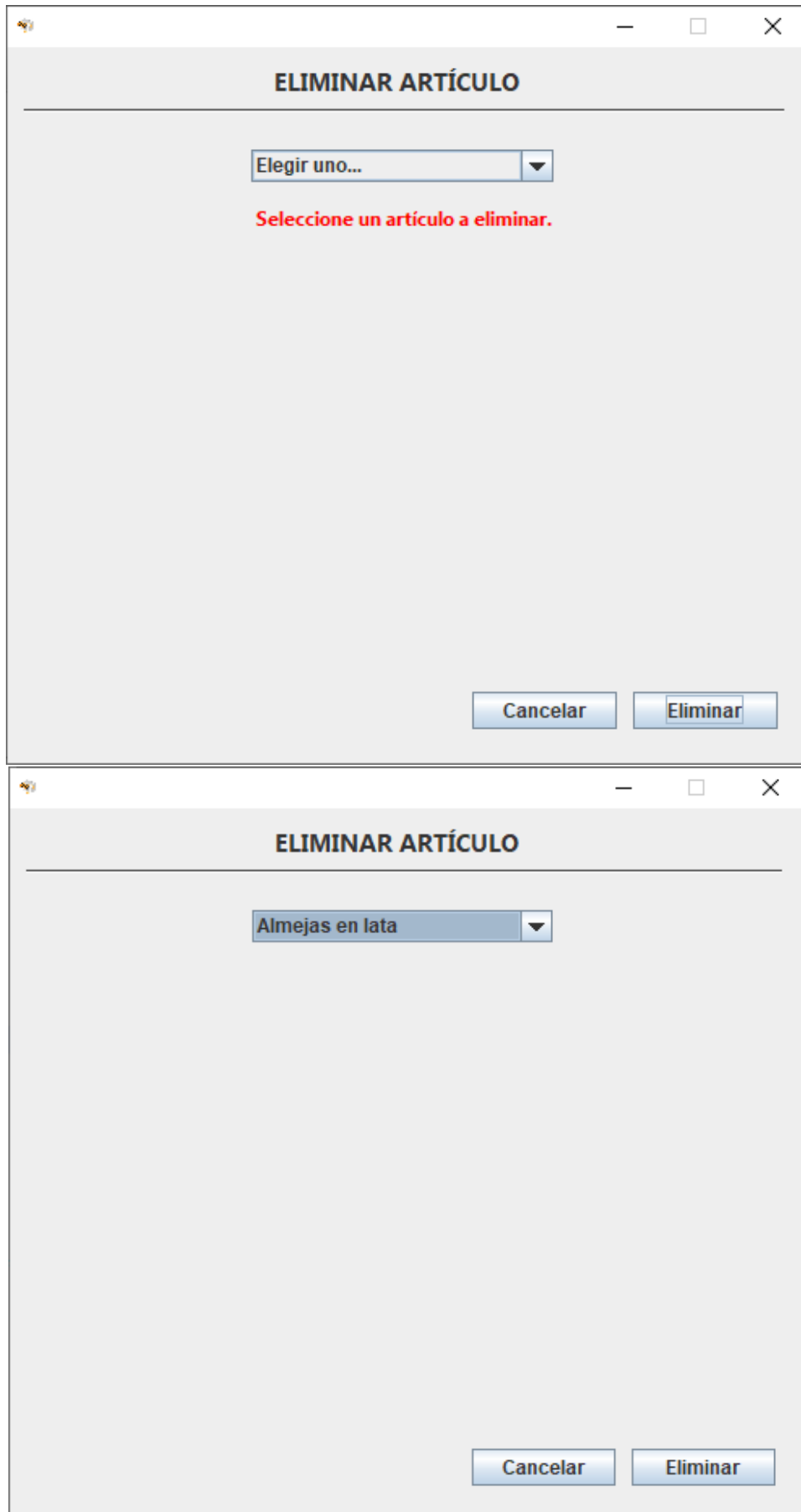
Artículo creado con éxito.



The screenshot shows a Java Swing window titled "ELIMINAR ARTÍCULO". The window has a title bar with standard OS controls (minimize, maximize, close). Below the title bar, there is a horizontal line. Below the line, there is a JComboBox with the text "Elegir uno..." and a downward arrow. At the bottom right, there are two buttons: "Cancelar" and "Eliminar".

En esta pantalla podremos eliminar los artículos que existan en nuestra base de datos, una vez se abra la venta de cargará un JComboBox con todos los artículos que tengamos registrados.

Para eliminar un artículo tendremos que seleccionarlo y hacer clic en "Eliminar". En caso de no elegir ninguno nos saldrá un aviso. Hay que tener en cuenta que si eliminar un artículo, este artículo se eliminará también de los tickets que lo contiene.



ELIMINAR ARTÍCULO

Elegir uno...

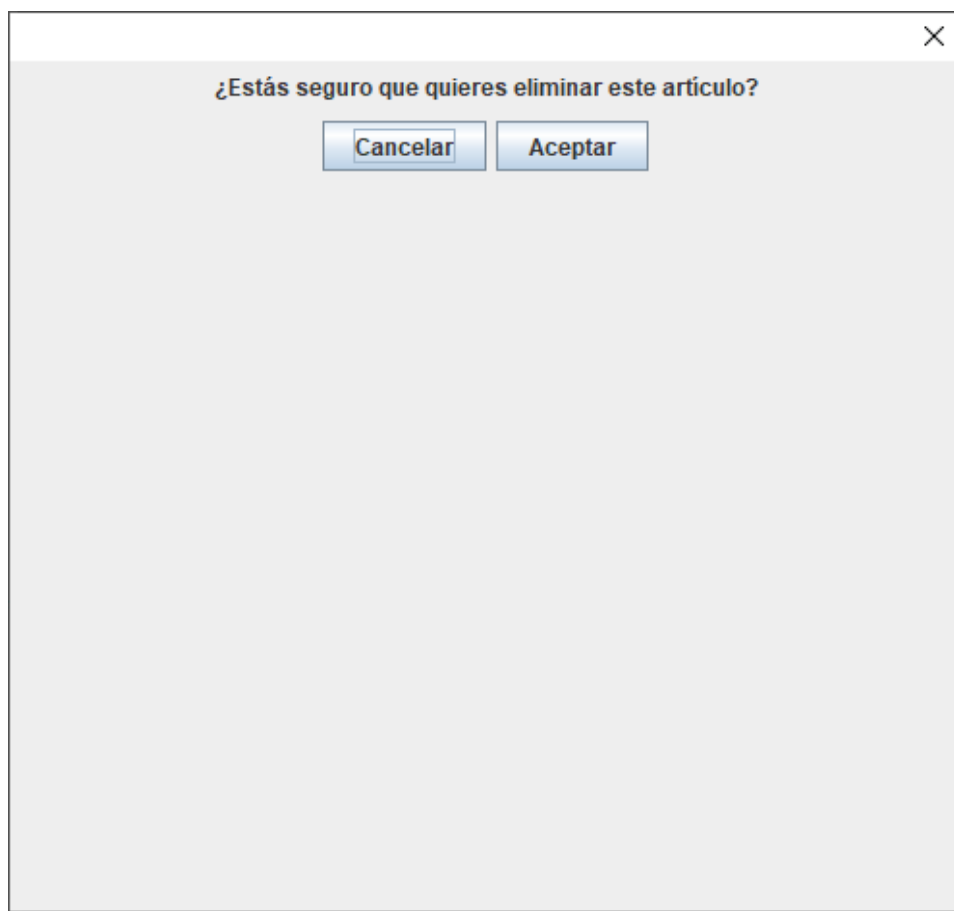
Seleccione un artículo a eliminar.

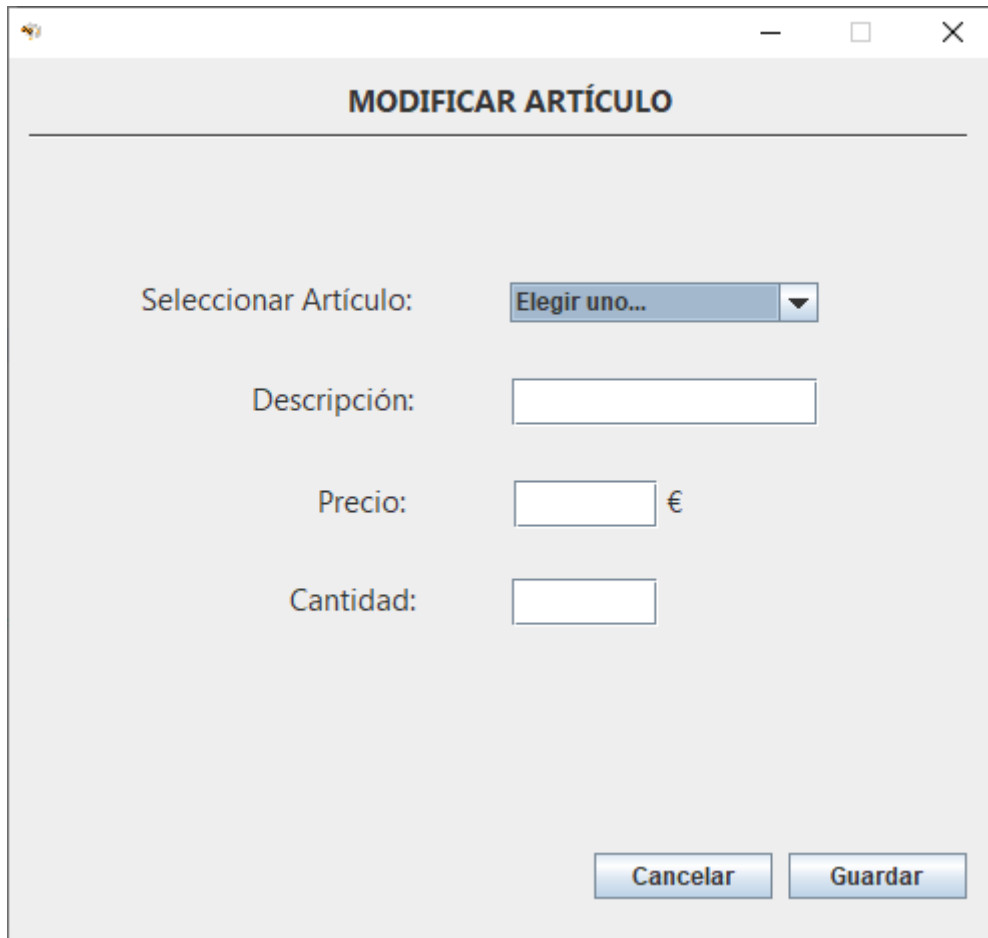
Cancelar Eliminar

ELIMINAR ARTÍCULO

Almejas en lata

Cancelar Eliminar





MODIFICAR ARTÍCULO

Seleccionar Artículo:

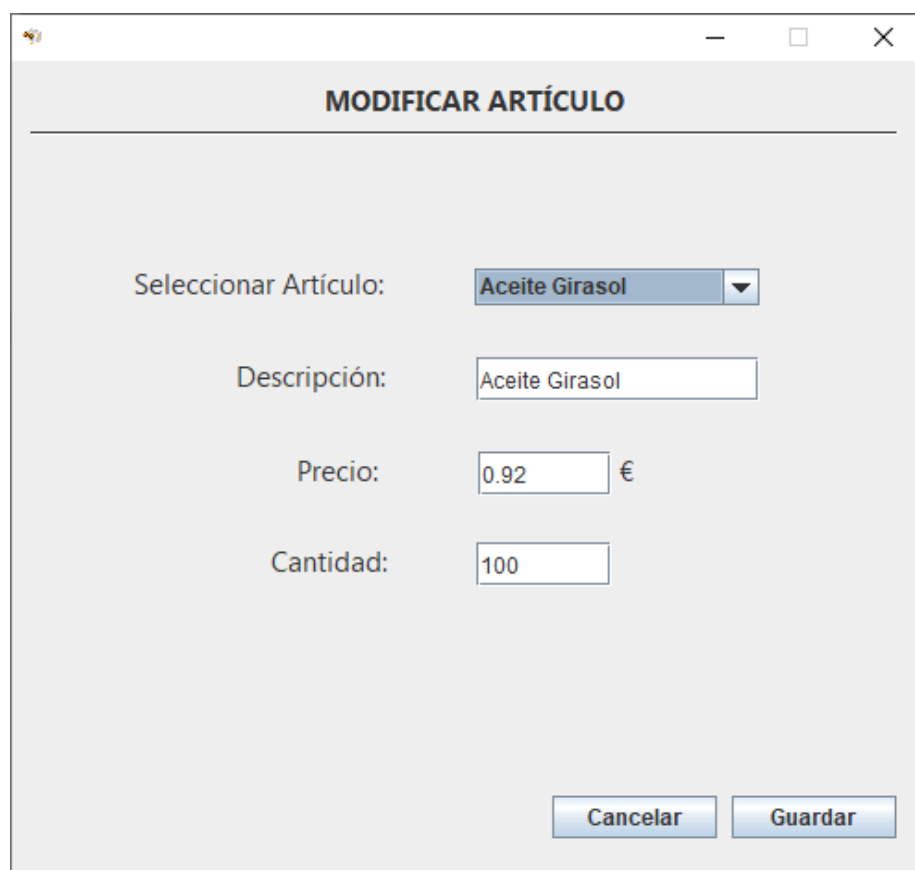
Descripción:

Precio: €

Cantidad:

En esta ventana podremos modificar los artículos que tengamos registrados. Disponemos de un JComboBox a la escucha, una vez que seleccionemos un artículo se cargará automáticamente todos los campos con sus correspondientes registros en la base de datos.

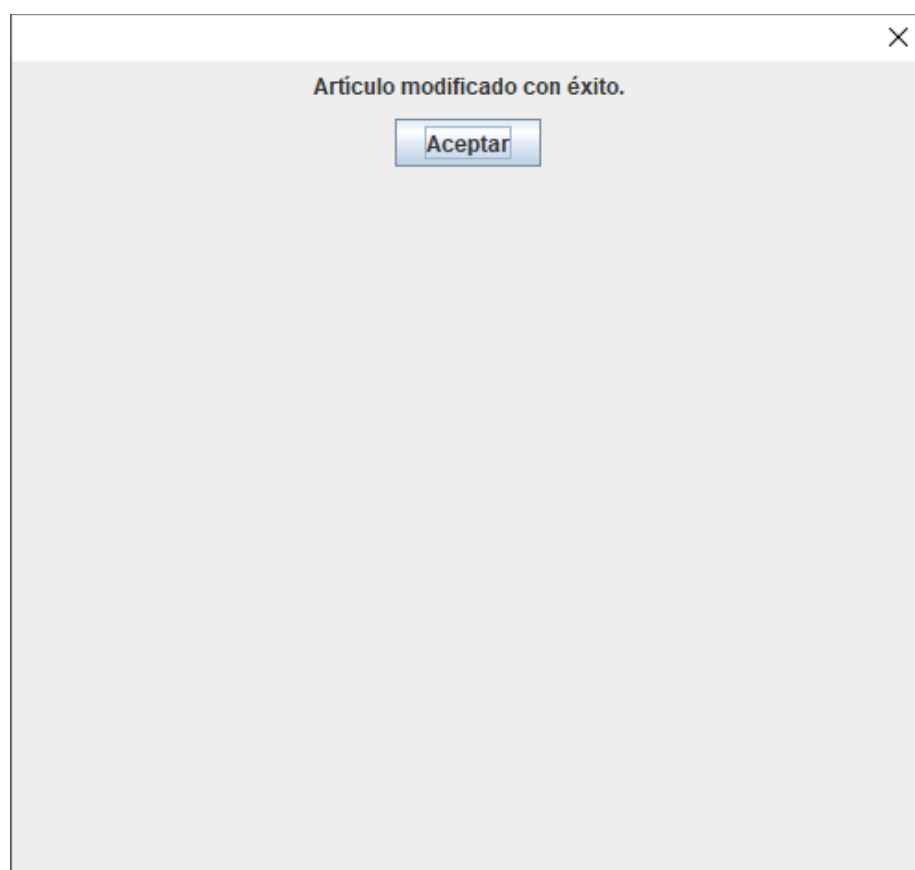
Para modificar el artículo, modificamos los campos que deseemos y haremos clic en “Guardar”.



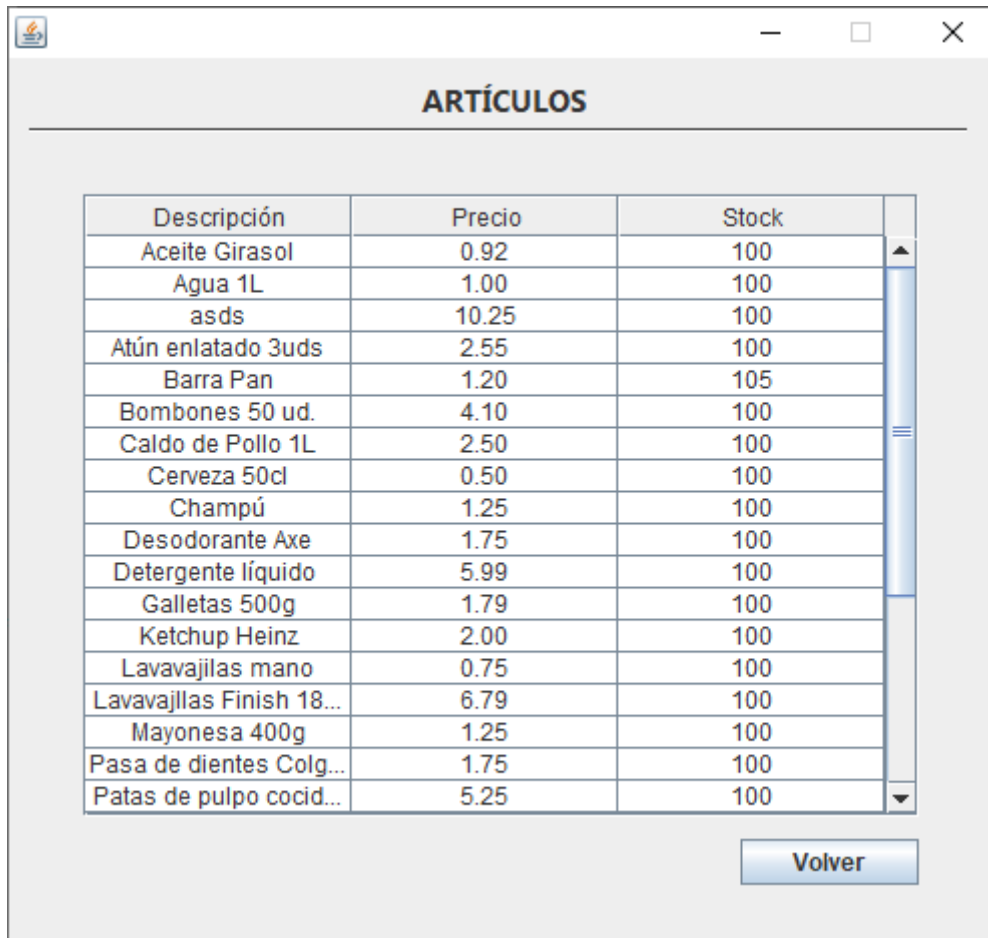
A screenshot of a software window titled "MODIFICAR ARTÍCULO". The window has a standard Windows-style title bar with a minimize button, a maximize button, and a close button. The main area is light gray and contains the following fields:

- Seleccionar Artículo:** A dropdown menu with "Aceite Girasol" selected.
- Descripción:** A text input field containing "Aceite Girasol".
- Precio:** A text input field containing "0.92", followed by a "€" symbol.
- Cantidad:** A text input field containing "100".

At the bottom right of the window are two buttons: "Cancelar" and "Guardar".



A screenshot of a software window showing a confirmation message. The window has a title bar with a close button. The main area is light gray and contains the text "Artículo modificado con éxito." centered. Below the text is a single button labeled "Aceptar".



Descripción	Precio	Stock
Aceite Girasol	0.92	100
Agua 1L	1.00	100
asds	10.25	100
Atún enlatado 3uds	2.55	100
Barra Pan	1.20	105
Bombones 50 ud.	4.10	100
Caldo de Pollo 1L	2.50	100
Cerveza 50cl	0.50	100
Champú	1.25	100
Desodorante Axe	1.75	100
Detergente líquido	5.99	100
Galletas 500g	1.79	100
Ketchup Heinz	2.00	100
Lavavajilas mano	0.75	100
Lavavajillas Finish 18...	6.79	100
Mayonesa 400g	1.25	100
Pasa de dientes Colg...	1.75	100
Patas de pulpo cocid...	5.25	100

Volver

En esta ventana podremos ver todos los artículos y sus correspondientes características, para volver a la ventana principal haremos clic en “Volver”.

REF. TICKET: 1 **NUEVO TICKET**

Fecha Ticket: / /

Artículo:

Cantidad:

--LISTA DE ARTÍCULOS--

Precio Total: 0€

Añadir **Eliminar**

Cancelar **Guardar**

Para registrar tickets, tendremos que rellenar los correspondientes campos. Se ha tenido en cuenta los siguientes errores por parte del usuario:

- El usuario introduce una fecha incorrecta o inexistente.
- El usuario intenta eliminar un artículo de la lista que no ha seleccionado.
- El usuario intenta agregar un artículo a la lista que ya está agregado. Obligamos así a que tenga que eliminar el artículo previamente creado si quisiera modificar la cantidad.
- El usuario introduce un ticket sin artículo. Los tickets deben de tener al menos un artículo.

El total del ticket se irá actualizando conforme vayamos añadiendo artículos a la lista.

REF. TICKET: 1

NUEVO TICKET

Fecha Ticket:

Día

 1 /

Mes

 Enero /

Año

 2019

Artículo: Aceite Girasol

Cantidad: 1

--LISTA DE ARTÍCULOS--

Añadir

Eliminar

Aceite Girasol -- 1 -- 0,92€

Precio Total: 0,92€

Cancelar

Guardar

REF. TICKET: 1

NUEVO TICKET

Fecha Ticket:

Día

 1 /

Mes

 Enero /

Año

 2019

Artículo: Aceite Girasol

Cantidad: 1

--LISTA DE ARTÍCULOS--

Añadir

Eliminar

Aceite Girasol -- 1 -- 0,92€

Elemento a eliminar no seleccionado.

Precio Total: 0,92€

Cancelar

Guardar

REF. TICKET: 1

NUEVO TICKET

Fecha Ticket:

Día

Mes

Año

1

 /

Febrero

 /

2019

Artículo:

Aceite Girasol

Cantidad:

1

--LISTA DE ARTÍCULOS--

Añadir

Eliminar

Aceite Girasol -- 1 -- 0,92€

El artículo que está agregando ya existe.

Precio Total: 0,92€

Cancelar

Guardar

REF. TICKET: 1

NUEVO TICKET

Fecha Ticket:

Día

Mes

Año

31

 /

Febrero

 /

2019

Artículo:

Aceite Girasol

Cantidad:

1

--LISTA DE ARTÍCULOS--

Añadir

Eliminar

Aceite Girasol -- 1 -- 0,92€

La fecha elegida no es correcta.

Precio Total: 0,92€

Cancelar

Guardar

REF. TICKET: 3 **NUEVO TICKET**

Fecha Ticket: Día / Mes / Año

Artículo:

Cantidad:

--LISTA DE ARTÍCULOS--

Debe introducir al menos un artículo en el ticket.

Precio Total: 0€

REF. TICKET: 1 **NUEVO TICKET**

Fecha Ticket: Día / Mes / Año

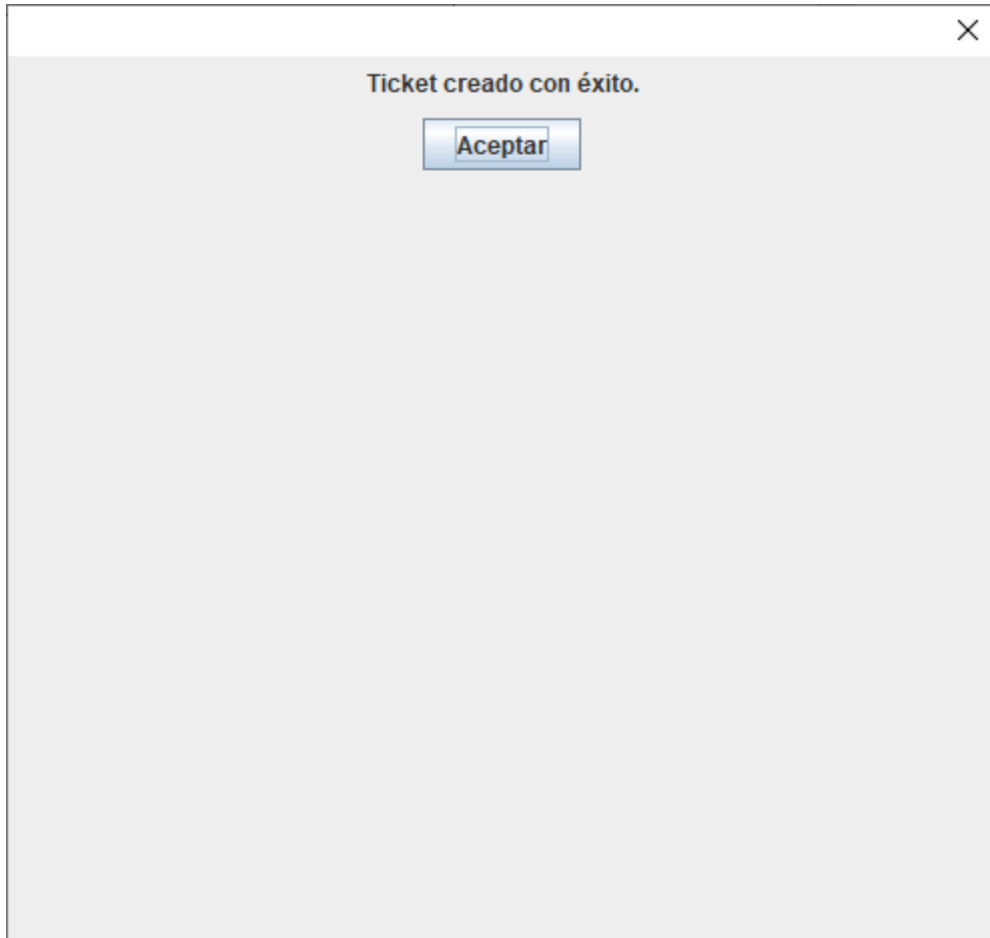
Artículo:

Cantidad:

--LISTA DE ARTÍCULOS--

Aceite Girasol -- 1 -- 0,92€
Atún enlatado 3uds -- 3 -- 7,65€
Caldo de Pollo 1L -- 3 -- 7,50€
Champú -- 5 -- 6,25€
Mayonesa 400g -- 2 -- 2,50€
Queso Oveja Viejo 1Kg -- 1 -- 11,95€

Precio Total: 36,77€



Una vez creado un ticket con éxito, la referencia del ticket habrá cambiado y será el siguiente número sin registrar en la base de datos. Por defecto el primer ticket de nuestra base de datos será el 1. La referencia del ticket será su identificador único.

A continuación, crearemos otro ticket a modo de prueba.

REF. TICKET: 2

NUEVO TICKET

Fecha Ticket:

Día

1

/

Mes

Enero

/

Año

2019

Artículo:

Aceite Girasol

Cantidad:

1

--LISTA DE ARTÍCULOS--

Añadir

Eliminar

Precio Total: 0€

Cancelar

Guardar

REF. TICKET: 2

NUEVO TICKET

Fecha Ticket:

Día

1

/

Mes

Enero

/

Año

2019

Artículo:

Vino Blanco 75cl

Cantidad:

4

--LISTA DE ARTÍCULOS--

Añadir

Eliminar

Caldo de Pollo 1L -- 5 -- 12,50€

Potito varios tipos -- 3 -- 3,30€

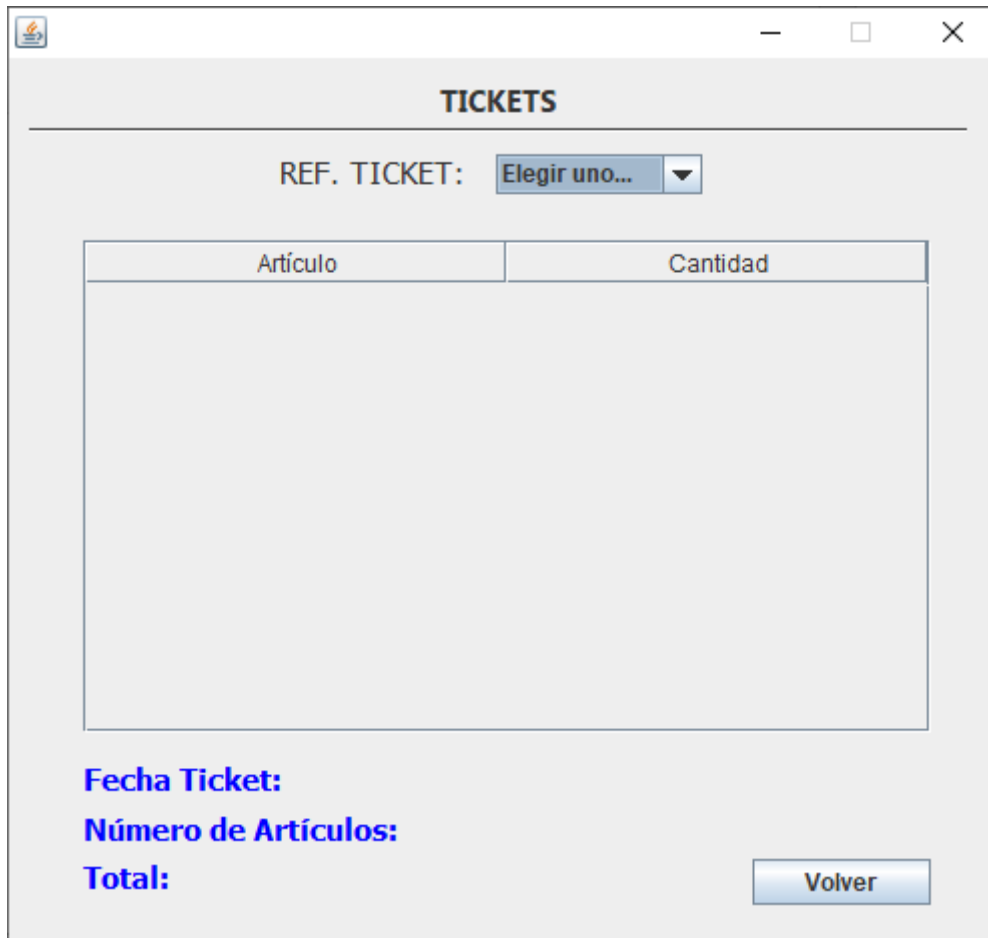
Refresco naranja -- 8 -- 12,00€

Vino Blanco 75cl -- 4 -- 14,00€

Precio Total: 41,80€

Cancelar

Guardar



TICKETS


REF. TICKET:

Artículo	Cantidad
----------	----------

Fecha Ticket:
Número de Artículos:
Total:

Por último, en esta ventana podremos ver todos los tickets que tenemos registrado, así como los artículos que contiene y su cantidad. También mostramos las características que generamos con cada ticket (fecha, número de artículos y el precio total).

Para ello seleccionamos la referencia de un ticket y automáticamente se nos cargará una tabla en la que veremos todas las características del ticket seleccionado.

 — □ ×

TICKETS


REF. TICKET:

Elegir uno...
Elegir uno...

Artículo	Cantidad
1	
2	

Fecha Ticket:
Número de Artículos:
Total:

Volver

 — □ ×

TICKETS


REF. TICKET:

1

Artículo	Cantidad
Atún enlatado 3uds	3
Queso Oveja Viejo 1Kg	1
Aceite Girasol	1
Caldo de Pollo 1L	3
Mayonesa 400g	2
Champú	5

Fecha Ticket: 27/02/2019
Num. Artículos: 6
Total: 36.77€

Volver

 — □ ×

TICKETS

REF. TICKET:

Artículo	Cantidad
Caldo de Pollo 1L	5
Refresco naranja	8
Vino Blanco 75cl	4
Potito varios tipos	3

Fecha Ticket: 01/01/2019

Num. Artículos: 4

Total: 41.80€

Volver

BIBLIOGRAFÍA

- Apuntes Desarrollo de Interfaces. DAM. Tema 4. Grupo Studium. 2019.