



Programación con Java

TAREA 07

Índice.....	1
Introducción.....	2
Programas.....	3
Ejercicio 01:.....	3
Ejercicio 02:.....	4
Ejercicio 03:.....	5
Imágenes ejercicio 03.....	6
Ejercicio 04:.....	8
Webgrafía.....	11

Durante el ejercicio07, aprenderemos a utilizar los ArrayList, arrays “dinámicos” donde puedes ir añadiendo elementos sin definir su tamaño, eliminarlos y se reorganizará sola, etc. y los HashTable, son bibliotecas de datos donde hay un valor que hace referencia a otro, un key que refiere a un value, nosotros utilizaremos HashMap que es similar pero actualizado para las últimas versiones de java.

Ejercicio 01:

En el primer programa solicitaremos la cantidad de alumnos y las notas que tienen cada uno, para después calcular la nota media de cada alumno y mostrarlo por pantalla.

```
import java.util.HashMap;

import javax.swing.JOptionPane;

public class Ejercicio01 {
    Run | Debug
    public static void main(String[] args) {
        HashMap<String, Integer> alum_notas = new HashMap<>();
        StringBuilder nombres_media = new StringBuilder(str: " - - NOTAS MEDIAS DE LOS ALUMNOS - - \n\n");
        int cantAlumnos = Integer.parseInt(JOptionPane.showInputDialog(message: "¿Cuántos alumnos hay que evaluar?"));
        int cantNotas = Integer.parseInt(JOptionPane.showInputDialog(message: "Y cuántas notas tienen cada uno?"));

        nombrarAlumnoyCadena(alum_notas, nombres_media, cantAlumnos, cantNotas);

        JOptionPane.showMessageDialog(parentComponent: null, nombres_media.toString(), title: "Medias de los alumnos", JOptionPane.PLAIN_MESSAGE);
    }

    // Calcular nota media de un alumno
    public static int notaMedia(int cantNotas) {
        int notasSumadas = 0;
        for (int i = 0; i < cantNotas; i++) {
            int nota = Integer.parseInt(JOptionPane.showInputDialog(message: "Ingrese una nota para el alumno"));
            while (!numValido(nota)) {
                nota = Integer.parseInt(JOptionPane.showInputDialog(message: "Por favor, ingrese una nota entre 0 y 10"));
            }
            notasSumadas += nota;
        }
        return notasSumadas / cantNotas;
    }
}
```

main y método para calcular la nota media

```
// Poner nombre al alumno y asignar al HashMap
public static void nombrarAlumnoyCadena(HashMap<String, Integer> alum_notas, StringBuilder nombres_media,
    int cantAlumnos, int cantNotas) {
    int media = 0;
    for (int i = 0; i < cantAlumnos; i++) {
        String nombre = JOptionPane.showInputDialog(message: "¿Cómo se llama el alumno a evaluar?");
        media = notaMedia(cantNotas);
        alum_notas.put(nombre, media);
        nombres_media.append(" - " + nombre + " ==> " + alum_notas.get(nombre) + "\n");
    }
}

// Controlar la nota, números positivos y máximo 10
public static boolean numValido(int nota) {
    if (nota < 0) {
        return false;
    } else if (nota > 10) {
        return false;
    } else {
        return true;
    }
}
}
```

primer método para solicitar los datos del alumno y guardarlo en un hashmap, el segundo método sirve para controlar la nota ingresada

Ejercicio 02:

Para el segundo programa las funciones que queremos que realice es una interfaz gráfica para hacer la compra. Para ello, preguntaremos sobre el nombre del producto que compraremos y su precio, y con dos CheckBox preguntaremos si quiere añadir más artículos a la cesta o si es esencial (para aplicar un IVA menor), después de recopilar toda esta información la mostraremos en una última pantalla como función de ticket.

1ª Parte del código:

- En esta parte, se creará el panel que veremos en donde insertamos la información.

```
import java.util.HashMap;
import java.awt.*;
import javax.swing.*;
import java.text.DecimalFormat;

public class Ejercicio02 {
    public static void main(String[] args) {
        HashMap<String, Double> cesta = new HashMap<>();
        DecimalFormat dosDecimales = new DecimalFormat(pattern:"#.##");
        StringBuilder texto = new StringBuilder(str:"PRECIO DE LOS ARTICULOS\n");
        double precioIVA = 0;
        double totalCompra = 0;

        boolean continuar = true;
        while (continuar) {
            JPanel infoArticulos = new JPanel(new GridLayout(rows:0, cols:2));

            infoArticulos.add(new JLabel(text:"Nombre del producto: "));
            JTextField producto = new JTextField(columns:10);
            infoArticulos.add(producto);

            infoArticulos.add(new JLabel(text:"Precio:"));
            JTextField precio = new JTextField(columns:10);
            infoArticulos.add(precio);

            infoArticulos.add(new JLabel(text:"¿Más artículos?"));
            JCheckBox masArticulos = new JCheckBox();
            infoArticulos.add(masArticulos);

            infoArticulos.add(new JLabel(text:"¿Artículo esencial?"));
            JCheckBox articuloEsencial = new JCheckBox();
            infoArticulos.add(articuloEsencial);

            JOptionPane.showMessageDialog(parentComponent:null, infoArticulos);
        }
    }
}
```

2ª Parte del código:

- Y en la segunda, hacemos el ticket para mostrar la información.

```
String nombreProducto = producto.getText();
double precioProducto = Double.parseDouble(precio.getText());
cesta.put(nombreProducto, precioProducto);

if (articuloEsencial.isSelected()) {
    precioIVA = precioProducto * 1.04;
    String precioIVAFormateado = dosDecimales.format(precioIVA);
    texto.append("Producto: <" + nombreProducto + ">\n    Precio: <" + precioProducto
        + ">\n    IVA aplicado [4%]\n    Precio con el IVA: <" + precioIVAFormateado + ">\n\n");
} else {
    precioIVA = precioProducto * 1.21;
    String precioIVAFormateado = dosDecimales.format(precioIVA);
    texto.append("Producto: <" + nombreProducto + ">\n    Precio: <" + precioProducto
        + ">\n    IVA aplicado [21%]\n    Precio con el IVA: <" + precioIVAFormateado + ">\n\n");
}
totalCompra += precioIVA;

if (!masArticulos.isSelected()) {
    continuar = false;
}

ticket(dosDecimales, texto, totalCompra);

public static void ticket(DecimalFormat dosDecimales, StringBuilder texto, double totalCompra) {
    String laMultas = dosDecimales.format(totalCompra);
    double losBilletes = Double
        .parseDouble(JOptionPane.showInputDialog(parentComponent:null, texto.toString() + "\nTotal a pagar: " + laMultas));
    double cambio = losBilletes - totalCompra;
    String laCalderilla = dosDecimales.format(cambio);
    JOptionPane.showMessageDialog(parentComponent:null,
        "Has pagado el precio de " + laMultas + " con " + losBilletes + "\nTus vueltas son ==> " + laCalderilla);
}
```

Ejercicio 03:

Las funciones a realizar por el tercer programa son básicamente 3; teniendo ya diez artículos en una biblioteca de datos; podremos añadir nuevos productos, habiendo un máximo de artículos por añadir de diez, también podremos listar todos los artículos que hay a través de la terminal y, como última acción, podremos buscar un artículo insertando una entrada de texto.

main asignando artículos al HashMap y llamando al método ventana

método ventana con la creación de 3 paneles diferenciados y sus respectivas llamadas

```
import java.awt.*;
import java.awt.event.*;
import java.text.DecimalFormat;
import java.util.*;
import javax.swing.*;
import javax.swing.border.*;

public class Ejercicio03 {
    Run | Debug
    public static void main(String[] args) {
        StringBuilder productosTXT = new StringBuilder(str:"");

        HashMap<String, Integer> productoStock = new HashMap<>();
        productoStock.put(key:"Leche", value:100);
        productoStock.put(key:"Pan", value:150);
        productoStock.put(key:"Huevos", value:200);
        productoStock.put(key:"Arroz", value:80);
        productoStock.put(key:"Pasta", value:120);
        productoStock.put(key:"Atún en lata", value:90);
        productoStock.put(key:"Aceite de oliva", value:110);
        productoStock.put(key:"Queso", value:70);
        productoStock.put(key:"Yogur", value:120);
        productoStock.put(key:"Frutas y verduras", value:200);

        HashMap<String, Double> productoPrecio = new HashMap<>();
        productoPrecio.put(key:"Leche", value:1.50);
        productoPrecio.put(key:"Pan", value:0.60);
        productoPrecio.put(key:"Huevos", value:2.0);
        productoPrecio.put(key:"Arroz", value:1.2);
        productoPrecio.put(key:"Pasta", value:5.25);
        productoPrecio.put(key:"Atún en lata", value:3.30);
        productoPrecio.put(key:"Aceite de oliva", value:12.0);
        productoPrecio.put(key:"Queso", value:2.5);
        productoPrecio.put(key:"Yogur", value:2.25);
        productoPrecio.put(key:"Frutas y verduras", value:15.0);

        ventana(productoStock, productoPrecio, productosTXT);
    }
}
```

```
public static void ventana(HashMap<String, Integer> productoStock, HashMap<String, Double> productoPrecio,
    StringBuilder productosTXT) {
    JFrame frame = new JFrame(title:"Almacen LIDL");
    frame.setSize(width:1000, height:1000);
    frame.setLayout(new BorderLayout());
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    JPanel panelPrincipal = new JPanel(new GridBagLayout());
    GridBagConstraints posicion = new GridBagConstraints();
    posicion.fill = GridBagConstraints.BOTH; // Rellenar vertical y horizontalmente
    posicion.weightx = 1; // Permitir que los componentes se expandan horizontalmente
    posicion.weighty = 1; // Permitir que los componentes se expandan verticalmente

    JPanel tituloLIDL = new JPanel();
    JPanel botones = new JPanel();
    JPanel productos = new JPanel();

    panelTitulo(tituloLIDL);
    panelBotones(frame, botones, productos, productoStock, productoPrecio, productosTXT);
    panelProductos(productoStock, productoPrecio, productos, productosTXT);

    posicion.gridy = 0;
    panelPrincipal.add(tituloLIDL, posicion);

    posicion.gridy = 1;
    panelPrincipal.add(botones, posicion);

    posicion.gridy = 2;
    panelPrincipal.add(productos, posicion);

    frame.add(panelPrincipal, BorderLayout.CENTER);

    frame.pack();
    frame.setLocationRelativeTo(c:null);
    frame.setVisible(b:true);
}
```

Imágenes ejercicio 03

```
public static void panelTitulo(JPanel tituloIDL) {
    tituloIDL.setBackground(Color.BLUE);
    JLabel titulArticulos = new JLabel(text:"ARTICULOS LIDL");
    titulArticulos.setFont(new Font(name:"Arial", Font.BOLD, size:35));
    titulArticulos.setForeground(Color.YELLOW);
    titulArticulos.setAlignmentX(Component.CENTER_ALIGNMENT);
    titulArticulos.setBorder(new EmptyBorder(top:5, left:10, bottom:5, right:10));
    tituloIDL.add(titulArticulos);
}

public static void panelProductos(HashMap<String, Integer> productoStock, HashMap<String, Double> productoPrecio,
    JPanel productos, StringBuilder productosTXT) {
    productos.setBackground(Color.LIGHT_GRAY);
    productos.setLayout(new BoxLayout(productos, BoxLayout.Y_AXIS));
    productos.setBorder(new EmptyBorder(top:5, left:10, bottom:7, right:10));
    DecimalFormat decimales = new DecimalFormat(pattern:"0.00");
    int i = 1;
    for (String producto : productoStock.keySet()) {
        double precio = productoPrecio.get(producto);
        String precioFormateado = decimales.format(precio);
        int stock = productoStock.get(producto);
        JLabel labelProductos = new JLabel(
            "Artículo " + i + " > " + producto + ": " + precioFormateado + "€ / " + stock
            + " unidades en stock");
        labelProductos.setForeground(Color.BLACK);
        productosTXT.append(
            "Artículo: " + producto + "\n\t" + precioFormateado + "€(EUROS)\n\t" + stock
            + " unidades en stock\n");
        productos.add(labelProductos);
        productos.add(Box.createVerticalStrut(height:5)); // Agregar relleno vertical entre los componentes
        i++;
    }
}
```

Creación de dos métodos para:

- poner un título en el primer panel
- añadir todos los productos en el tercer panel

```
public static void panelBotones(JFrame frame, JPanel botones, JPanel productos,
    HashMap<String, Integer> productoStock,
    HashMap<String, Double> productoPrecio, StringBuilder productosTXT) {
    botones.setLayout(new GridLayout(rows:1, cols:3));
    botones.setBackground(Color.GRAY);

    JButton cesta = new JButton(escalarImagen(ruta:"Iconos\\cart-plus.png"));
    modificarBoton(cesta);
    JButton lista = new JButton(escalarImagen(ruta:"Iconos\\list-task.png"));
    modificarBoton(lista);
    JButton lupa = new JButton(escalarImagen(ruta:"Iconos\\search.png"));
    modificarBoton(lupa);

    accionesCesta(frame, cesta, productos, productosTXT, productoStock, productoPrecio);
    accionesLista(lista, productosTXT);
    accionesLupa(lupa, productoPrecio);

    botones.add(cesta);
    botones.add(lista);
    botones.add(lupa);
}

public static void modificarBoton(JButton boton) {
    boton.setContentAreaFilled(b:false);
    boton.setBorderPainted(b:false);
    boton.setAlignmentX(Component.CENTER_ALIGNMENT);
}

public static ImageIcon escalarImagen(String ruta) {
    ImageIcon imagen = new ImageIcon(ruta);
    ImageIcon imagenEscalada = new ImageIcon(imagen.getImage().getScaledInstance(width:25, height:25, Image.SCALE_SMOOTH));
    return imagenEscalada;
}
```

Creación de tres métodos para:

- añadir tres botones en el panel central
- eliminar el fondo y el área que se resalta al pasar por encima, y centrar el botón
- escalar la imagen de la ruta para que sea más pequeña

Imágenes ejercicio 03

```
public static void accionesLista(JButton lista, StringBuilder productosTXT) {
    lista.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            System.out.println(productosTXT.toString());
        }
    });
}

public static void accionesCesta(JFrame frame, JButton cesta, JPanel productos, StringBuilder productosTXT,
    HashMap<String, Integer> productoStock,
    HashMap<String, Double> productoPrecio) {
    cesta.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            String productName = JOptionPane.showInputDialog(parentComponent:null, message:"Ingrese el nombre del nuevo producto:",
                title:"Producto", JOptionPane.PLAIN_MESSAGE);
            if (productName != null && !productName.isEmpty()) {
                double precio = Double
                    .parseDouble(JOptionPane.showInputDialog(parentComponent:null, message:"Ingrese el precio del artículo:",
                        title:"Precio", JOptionPane.PLAIN_MESSAGE));
                int stock = Integer
                    .parseInt(JOptionPane.showInputDialog(parentComponent:null, message:"Ingrese la cantidad en stock:", title:"Stock",
                        JOptionPane.PLAIN_MESSAGE));
                productoStock.put(productName, stock);
                productoPrecio.put(productName, precio);

                actualizarProductos(frame, productoStock, productoPrecio, productos, productosTXT);
            }
        }
    });
}
```

Creación de tres métodos para:

- darle la función de listar los artículos al botón llamado lista
- darle la función de ir agregando nuevos artículos a la biblioteca de datos al botón cesta

```
public static void accionesLupa(JButton lupa, HashMap<String, Double> productoPrecio) {
    lupa.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            DecimalFormat decimales = new DecimalFormat(pattern:"0.00");
            String productoBuscado = JOptionPane.showInputDialog(parentComponent:null, message:"Ingrese el nombre del producto a buscar:",
                title:null, JOptionPane.PLAIN_MESSAGE);
            if (productoBuscado != null && !productoBuscado.isEmpty()) {
                StringBuilder productosCoincidentes = new StringBuilder();
                for (String producto : productoPrecio.keySet()) {
                    if (producto.toLowerCase().contains(productoBuscado.toLowerCase())) {
                        double precio = productoPrecio.get(producto);
                        String precioFormateado = decimales.format(precio);
                        productosCoincidentes
                            .append("\t" + producto + " > " + precioFormateado + "€/u.\n");
                    }
                }
                if (productosCoincidentes.length() > 0) {
                    JOptionPane.showMessageDialog(parentComponent:null, "Productos encontrados:\n" + productosCoincidentes);
                } else {
                    JOptionPane.showMessageDialog(parentComponent:null,
                        "No se encontraron productos que coincidan con: '" + productoBuscado + "'");
                }
            }
        }
    });
}
```

método para la función buscar:

- el botón lupa buscará un artículo a través de una entrada de texto, ya sea que el artículo empieza por el texto ingresado o solo lo contenga

```
public static void actualizarProductos(JFrame frame, HashMap<String, Integer> productoStock,
    HashMap<String, Double> productoPrecio, JPanel productos, StringBuilder productosTXT) {

    frame.setVisible(b:false);
    frame.setSize(width:1000, height:1000);
    productos.removeAll();
    DecimalFormat decimales = new DecimalFormat(pattern:"0.00");
    int i = 1;
    for (String producto : productoStock.keySet()) {
        if (i <= 20) {
            double precio = productoPrecio.get(producto);
            String precioFormateado = decimales.format(precio);
            int stock = productoStock.get(producto);
            JLabel labelProductos = new JLabel(
                "Artículo " + i + " > " + producto + ": " + precioFormateado + "€ / " + stock
                    + " unidades en stock");
            labelProductos.setForeground(Color.BLACK);
            productosTXT.append(
                "Artículo: '" + producto + "'\n\t" + precioFormateado + "€(EUROS)\n\t" + stock
                    + " unidades en stock\n");
            productos.add(labelProductos);
            productos.add(Box.createVerticalStrut(height:5));
            i++;
        } else {
            JOptionPane.showMessageDialog(parentComponent:null, message:"Has llegado al máximo de artículos posible");
        }
    }
    productos.revalidate();
    productos.repaint();

    frame.pack();
    frame.setLocationRelativeTo(c:null);
    frame.setVisible(b:true);
}
```

método para actualizar el panel productos:

- en este método borramos toda la información del panel productos para volver a rellenarlo, pero con nuevos valores, y revalidamos y repintamos el panel. Al principio del método agrandamos el frame y al terminarlo lo ajustamos con *“.pack()”* para reajustar el frame al nuevo tamaño.

Compilación y ejecución de un programa en Java

Ejercicio 04:

En el cuarto programa lo que haremos será juntar los dos programas anteriores para generar una interfaz GUI para poder gestionar el flujo de acciones en un almacén.

(En estas imágenes la explicación será muy breve porque se repiten gran parte de las cosas)

main:

- El único cambio en el main es que cuando llamamos al método ventana le metemos más variables

(De igual manera el único cambio en este, es en la llamada al método panelBotones)

(Los métodos panelTitulo y panelProductos, siguen igual)

```
public static void main(String[] args) {
    StringBuilder productosTXT = new StringBuilder(str:"");
    StringBuilder texto = new StringBuilder(str:"PRECIO DE LOS ARTICULOS\n");
    DecimalFormat dosDecimales = new DecimalFormat(pattern:"#.##");
    ArrayList<String> arrayProductos = new ArrayList<>();
    double totalCompra = 0;
    double totalProducto = 0;

    HashMap<String, Integer> cesta = new HashMap<>();

    HashMap<String, Integer> productoStock = new HashMap<>();
    productoStock.put(key:"Leche", value:100);
    productoStock.put(key:"Pan", value:150);
    productoStock.put(key:"Huevos", value:200);
    productoStock.put(key:"Arroz", value:80);
    productoStock.put(key:"Pasta", value:120);
    productoStock.put(key:"Atún en lata", value:90);
    productoStock.put(key:"Aceite de oliva", value:110);
    productoStock.put(key:"Queso", value:70);
    productoStock.put(key:"Yogur", value:120);
    productoStock.put(key:"Frutas y verduras", value:200);

    HashMap<String, Double> productoPrecio = new HashMap<>();
    productoPrecio.put(key:"Leche", value:1.50);
    productoPrecio.put(key:"Pan", value:0.60);
    productoPrecio.put(key:"Huevos", value:2.0);
    productoPrecio.put(key:"Arroz", value:1.2);
    productoPrecio.put(key:"Pasta", value:5.25);
    productoPrecio.put(key:"Atún en lata", value:3.30);
    productoPrecio.put(key:"Aceite de oliva", value:12.0);
    productoPrecio.put(key:"Queso", value:2.5);
    productoPrecio.put(key:"Yogur", value:2.25);
    productoPrecio.put(key:"Frutas y verduras", value:15.0);

    ventana(productoStock, productoPrecio, productosTXT, arrayProductos, cesta, dosDecimales, texto, totalCompra,
            totalProducto);
}
```

Compilación y ejecución de un programa en Java

panelBotones:

- El cambio en este método es un nuevo botón para efectuar la compra
-

```
public static void panelBotones(JFrame frame, JPanel botones, JPanel productos,
    HashMap<String, Integer> productoStock,
    HashMap<String, Double> productoPrecio, StringBuilder productosTXT, ArrayList<String> arrayProductos,
    HashMap<String, Integer> cesta,
    DecimalFormat dosDecimales, StringBuilder texto, double totalCompra, double totalProducto) {
    botones.setLayout(new GridLayout(rows:0, cols:4));
    botones.setBackground(Color.GRAY);

    JButton anadir = new JButton(escalarImagen(ruta:"Iconos\\cart-plus.png"));
    modificarBoton(anadir);
    JButton comprar = new JButton(escalarImagen(ruta:"Iconos\\cart-check.png"));
    modificarBoton(comprar);
    JButton lista = new JButton(escalarImagen(ruta:"Iconos\\list-task.png"));
    modificarBoton(lista);
    JButton lupa = new JButton(escalarImagen(ruta:"Iconos\\search.png"));
    modificarBoton(lupa);

    accionesAnadir(frame, anadir, productos, productosTXT, productoStock, productoPrecio);
    accionesLista(lista, productosTXT);
    accionesLupa(lupa, productoPrecio);
    accionesComprar(frame, productos, productosTXT, productoStock, comprar, arrayProductos, productoPrecio, cesta,
        dosDecimales, texto, totalCompra, totalProducto);

    botones.add(comprar);
    botones.add(anadir);
    botones.add(lista);
    botones.add(lupa);
}
```

(A la vez que este, creamos un nuevo método para añadir acciones a este botón)

```
public static void accionesComprar(JFrame frame, JPanel productos, StringBuilder productosTXT,
    HashMap<String, Integer> productoStock, JButton comprar, ArrayList<String> arrayProductos,
    HashMap<String, Double> productoPrecio, HashMap<String, Integer> cesta, DecimalFormat dosDecimales,
    StringBuilder texto, double totalCompra, double totalProducto) {
    comprar.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            for (String producto : productoStock.keySet()) {
                arrayProductos.add(producto);
            }
            panelCrearTicket(arrayProductos, productoStock, productoPrecio, cesta, dosDecimales, texto, totalCompra,
                totalProducto);
            actualizarProductos(frame, productoStock, productoPrecio, productos, productosTXT);
        }
    });
}
```

Modificación Ejercicio02:

- Hemos modificado el ejercicio02 para que el nombre del producto sea un ComboBox
-

```
public static void panelCrearTicket(ArrayList<String> arrayProductos, HashMap<String, Integer> productoStock,
    HashMap<String, Double> productoPrecio, HashMap<String, Integer> cesta, DecimalFormat dosDecimales,
    StringBuilder texto, double totalCompra, double totalProducto) {
    boolean continuar = true;
    String eliminar = "";
    while (continuar) {
        JPanel infoArticulos = new JPanel(new GridLayout(rows:0, cols:2));

        infoArticulos.add(new JLabel(text:"Nombre del producto: "));
        JComboBox<String> seleccionable = crearSeleccionable(arrayProductos);
        infoArticulos.add(seleccionable);

        infoArticulos.add(new JLabel(text:"Cantidad:"));
        JTextField cantidad = new JTextField(columns:10);
        infoArticulos.add(cantidad);

        infoArticulos.add(new JLabel(text:"¿Más artículos?"));
        JCheckBox masArticulos = new JCheckBox();
        infoArticulos.add(masArticulos);

        infoArticulos.add(new JLabel(text:"¿Artículo esencial?"));
        JCheckBox articuloEsencial = new JCheckBox();
        infoArticulos.add(articuloEsencial);

        JOptionPane.showMessageDialog(parentComponent:null, infoArticulos, title:"Titulo", JOptionPane.PLAIN_MESSAGE);
    }
}
```

Compilación y ejecución de un programa en Java

Recogemos los valores del nombre del producto y su stock para buscarlo y asignarle un precio

Cuando el CheckBox “masArticulos” **no** esté seleccionado cerraremos el bucle. Y cuando “articuloEsencial” lo esté se aplicará un IVA del 4%, después se suma y se muestra en “ticket”

```
eliminar = (String) seleccionable.getSelectedItem();

int cantidadProducto = Integer.parseInt(cantidad.getText());
cesta.put(eliminar, cantidadProducto);

double precioProducto = 0;
double precioIVA = 0;
for (String nomProducto : cesta.keySet()) {
    for (String productListista : productoPrecio.keySet()) {
        if (productoLista.equals(nomProducto)) {
            precioProducto = productoPrecio.get(productoLista);
        }
    }
}

// no esta del todo terminado, faltan posibilidades
int stockProducto = 0;
int stockFinal = 0;
for (String nomProducto : cesta.keySet()) {
    for (String productListista : productoStock.keySet()) {
        if (productoLista.equals(nomProducto)) {
            stockProducto = productoStock.get(productoLista);

            stockFinal = stockProducto - cantidadProducto;

            productoStock.put(productoLista, stockFinal);
        }
    }
}
```

```
if (!masArticulos.isSelected()) {
    continuar = false;
} else if (eliminar.equals(anObject:"- Seleccionar producto -") || cantidad.getText().isEmpty()
    || cantidad.getText().equals(anObject:"0")) {
    JOptionPane.showMessageDialog(parentComponent:null, "Ingrese un producto o un precio " +
        "superior a 0", title:"Ingrese valores",
        JOptionPane.WARNING_MESSAGE);
} else {
    arrayProductos.remove(eliminar);
}

if (articuloEsencial.isSelected()) {
    precioIVA = precioProducto * 1.04;
    totalProducto = precioIVA * cantidadProducto;

    String precioIVAFormateado = dosDecimales.format(precioIVA);
    String totalFormateado = dosDecimales.format(totalProducto);
    texto.append("Producto: " + eliminar + "\n    Precio/unidad: <" + precioProducto
        + "> / <" + cantidadProducto + ">\n    IVA aplicado [4%] => PrecioIVA/unidad ["
        + precioIVAFormateado + "]\n    Precio total: <" + totalFormateado + ">\n\n");
} else {
    precioIVA = precioProducto * 1.21;
    totalProducto = precioIVA * cantidadProducto;

    String precioIVAFormateado = dosDecimales.format(precioIVA);
    String totalFormateado = dosDecimales.format(totalProducto);
    texto.append("Producto: " + eliminar + "\n    Precio/unidad: <" + precioProducto
        + "> / <" + cantidadProducto + ">\n    IVA aplicado [21%] => PrecioIVA/unidad ["
        + precioIVAFormateado + "]\n    Precio total: <" + totalFormateado + ">\n\n");
}

totalCompra += totalProducto;
}
ticket(dosDecimales, texto, totalCompra);
}
```

crearSeleccionable:

- Hemos creado un método para crear el seleccionable que se muestra en el “panelTicket”

```
public static JComboBox<String> crearSeleccionable(ArrayList<String> arrayProductos) {
    String[] productos = new String[(arrayProductos.size() + 1)];
    productos[0] = "- Seleccionar producto -";
    int i = 1;
    for (String producto : arrayProductos) {
        productos[i] = producto;
        i++;
    }
    JComboBox<String> seleccionable = new JComboBox<>(productos);

    return seleccionable;
}
```

- <https://docs.oracle.com/en/java/javase/21/docs/api/index.html>
- El copiloto de confianza como ayuda siempre, importante:
<https://chat.openai.com/>