

**Universidad de las Fuerzas Armadas - ESPE**  
**MODALIDAD EN LÍNEA**



**Informe de Actividad Experimental N.º 1 Segundo Parcial.**

**Curso:** Programación Orientada a Objetos

**Profesor:** Ing. Luis Enrique Jaramillo Montaña.

**Fecha:** 06/01/2025

**1. Título del Proyecto:**

Caso Práctico: Sistema de Gestión de Inventario con MVC.

**2. Datos de los estudiantes:**

- Jefferson Fernando Jaya Taipe.
- Isaac Sebastian Proaño Panchi
- Nick Mateo Ortiz Córdova
- Nayeli Estefania Vilaña Puga.

**3. NRC:1322**

**4. Fecha de Entrega:**

- 31/01/2025

**Objetivo de la actividad:**

Crear e instaurar un sistema que gestione el inventario de una tienda utilizando el patrón Modelo Vista Controlador (MVC), utilizando programación Orientada a objetos permitiendo al estudiante fomentar competencias en la estructuración de código, la división de responsabilidades que potenciaran la escalabilidad y sostenibilidad del software., esto se visualizará por medio de un diagrama UML y de la simulación de código Java.

**Descripción de la Actividad:**

Implemente un sistema que gestione el inventario de una tienda utilizando el patrón Modelo Vista Controlador (MVC).

**Ejemplo:**

Modelo (Producto y Inventario):

- Registrar Productos: Implementado en la clase Inventario como un método para agregar instancias de Producto a una lista.
- Actualizar Inventario: Método en la clase Inventario que modifica la cantidad de un producto.
- Consultar Productos: Método en la clase Inventario que devuelve la lista de productos.
- Calcular el Valor Total del Inventario: Método en la clase Inventario que itera sobre los productos y calcula el total.

Vista (VistaConsola o GUI):

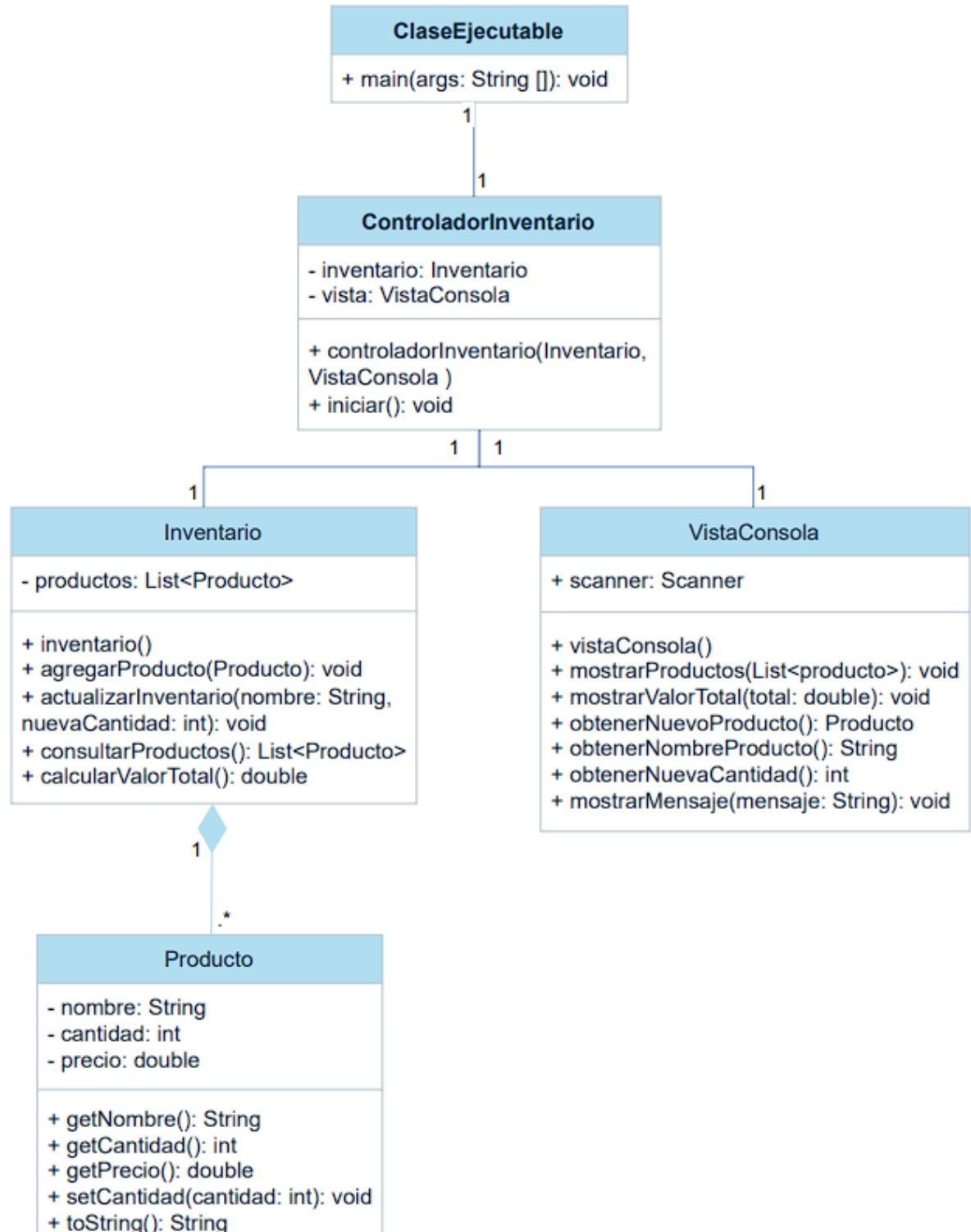
- Mostrar la lista de productos al usuario.
- Presentar el valor total del inventario calculado por el modelo.
- Solicitar datos al usuario para registrar productos o actualizar inventario.

Controlador (ControladorInventario):

- Coordina entre la Vista y el Modelo.
- Recibe entradas del usuario desde la Vista, las procesa y las envía al Modelo.
- Devuelve la información del Modelo a la Vista para mostrarla al usuario.

## Desarrollo de la Actividad:

### 1. Diagrama UML



### 2. Código de las clases

## 1.\_Inventario.

```
package Modelo;

import java.util.ArrayList;
import java.util.List;
public class Inventario {

    private List<Producto> productos;

    public Inventario () {
        this. productos = new ArrayList<>();
    }

    public void agregarProducto(Producto producto) {
        productos.add(producto);
    }

    public void actualizarInventario(String nombre, int nuevaCantidad) {
        for (Producto producto : productos) {
            if (producto.getNombre().equals(nombre)) {
                producto.setCantidad(nuevaCantidad);
                return;
            }
        }
    }

    public List<Producto> consultarProductos () {
        return productos;
    }

    public double calcularValorTotal() {
        double total = 0;
        for (Producto producto : productos) {
            total += producto.getCantidad() * producto.getPrecio();
        }
        return total;
    }
}
```

## 2.\_Producto.

```
package Modelo;

public class Producto {
    private String nombre;
    private int cantidad;
    private double precio;

    public Producto(String nombre, int cantidad, double precio) {
        this.nombre = nombre;
        this.cantidad = cantidad;
        this.precio = precio;
    }

    public String getNombre () {
        return nombre;
    }

    public int getCantidad () {
        return cantidad;
    }

    public double getPrecio() {
        return precio;
    }

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

    @Override
    public String toString() {
        return "Producto{" +
            "nombre='" + nombre + '\'' +
            ", cantidad=" + cantidad +
            ", precio=" + precio +
            '}';
    }
}
```

## 3.\_ VistaConsola

```
package Vista;

import Modelo.Producto;
import java.util.List;
import java.util.Scanner;
public class VistaConsola {
    // VistaConsola.java

    public Scanner scanner;

    public VistaConsola() {
        this.scanner = new Scanner(System.in);
    }

    public void mostrarProductos(List<Producto> productos) {
        System.out.println("Inventario:");
        for (Producto producto : productos) {
            System.out.println(producto);
        }
    }

    public void mostrarValorTotal(double total) {
        System.out.println("Valor total del inventario: " + total);
    }

    public Producto obtenerNuevoProducto() {
        System.out.print("Ingrese el nombre del producto: ");
        String nombre = scanner.nextLine();
        System.out.print("Ingrese la cantidad: ");
        int cantidad = Integer.parseInt(scanner.nextLine());
        System.out.print("Ingrese el precio: ");
        double precio = Double.parseDouble(scanner.nextLine());
        return new Producto (nombre, cantidad, precio);
    }

    public String obtenerNombreProducto() {
        System.out.print("Ingrese el nombre del producto a actualizar: ");
        return scanner.nextLine();
    }

    public int obtenerNuevaCantidad() {
        System.out.print("Ingrese la nueva cantidad: ");
        return Integer.parseInt(scanner.nextLine());
    }

    public void mostrarMensaje(String mensaje) {
        System.out.println(mensaje);
    }
}
```

#### 4.\_ ControladorInventario

```
Package Controlador;

import Modelo.Inventario;
import Modelo.Producto;
import Vista.VistaConsola;
import java.util.List;
public class ControladorInventario {

    private Inventario inventario;
    private VistaConsola vista;

    public ControladorInventario(Inventario inventario, VistaConsola vista) {
        this.inventario = inventario;
        this.vista = vista;
    }

    public void iniciar() {
        boolean continuar = true;
        while (continuar) {
            // Mostrar productos y valor total
            List<Producto> productos = inventario.consultarProductos();
            vista.mostrarProductos(productos);
            vista.mostrarValorTotal(inventario.calcularValorTotal());

            // Menú de opciones
            System.out.println("1. Agregar producto");
            System.out.println("2. Actualizar inventario");
            System.out.println("3. Salir");
            System.out.print("Seleccione una opción: ");
            int opcion = Integer.parseInt(vista.scanner.nextLine());

            switch (opcion) {
                case 1:
                    // Agregar nuevo producto
                    Producto nuevoProducto = vista.obtenerNuevoProducto();
                    inventario.agregarProducto(nuevoProducto);
                    vista.mostrarMensaje("Producto agregado.");
                    break;
                case 2:
                    // Actualizar inventario
                    String nombreProducto = vista.obtenerNombreProducto();
                    int nuevaCantidad = vista.obtenerNuevaCantidad();
                    inventario.actualizarInventario(nombreProducto,
nuevaCantidad);
                    vista.mostrarMensaje("Inventario actualizado.");
                    break;
                case 3:
                    // Salir
                    continuar = false;
                    vista.mostrarMensaje("Saliendo del sistema...");
                    break;
                default:
```

```

        vista.mostrarMensaje("Opción no válida. Intente de
nuevo.");
    }
}
}
}
}

```

// Clase ejecutable

```

package Modelo;

import Controlador.ControladorInventario;
import Vista.VistaConsola;

public class ClaseEjecutable {
    // Main.java

    public static void main(String[] args) {
        // Crear el modelo (Inventario)
        Inventario inventario = new Inventario();

        // Crear la vista (VistaConsola)
        VistaConsola vista = new VistaConsola();

        // Crear el controlador (ControladorInventario)
        ControladorInventario controlador = new
ControladorInventario(inventario, vista);

        // Iniciar la aplicación
        controlador.iniciar();
    }
}

```



### 3. Ejecución del Programa

```
1. Agregar producto
2. Actualizar inventario
3. Salir
Seleccione una opción: 1
Ingrese el nombre del producto: MANZANAS
Ingrese la cantidad: 5
Ingrese el precio: 2.00
Producto agregado.
Inventario:
Producto{nombre='MANZANAS', cantidad=5, precio=2.0}
Valor total del inventario: 10.0
1. Agregar producto
```

#### Resumen de Resolución:

En la actividad se realizó un sistema que gestione el inventario de una tienda utilizando el patrón Modelo Vista Controlador (MVC), utilizando Programación Orientada a Objetos, para lograr el objetivo se realizó el diagrama UML que plantea los moldes de las clases que se solicitaron en el ejemplo con sus respectivos atributos y métodos, representando la base para la implementación del código que fue el paso siguiente dividiendo la aplicación en tres componentes principales modelo que se estableció la clase producto la cual guardara información como nombre, precio, cantidad. La clase inventario esto administrara el inventario facilitando la anotación de nuevos productos la clase vista aquí se diseñó una interfaz de consola que permitirá al usuario visualizar la lista de productos, clase controlador funciona como un enlace entre vista y el modelo esto manejará datos del usuario y modelo luego transmite los datos a la vista para que sean mostrados.

#### Conclusión:

La implementación de la creación de este sistema a través del patrón MVC, que nos ayudará a simplificar la extensión y conservación del código además potenciará la estructura la práctica nos ayudará a tener una creación de un inventario, así como nos solito en este trabajo experimental más organizado y nos permitirá cambiar o ampliar cada elemento de manera independiente sin afectar el resto del sistema.

**Referencias:**

Romero, Y. F., & González, Y. D. (2012). Patrón modelo-vista-controlador. *Revista Telem@tica*, 11(1), 47-57.

Bascón Pantoja, E. (2004). El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing. *Acta Nova*, 2(4), 493-507.

Fernández Romero, Y., & Díaz González, Y. (2012). Patrón Modelo-Vista-Controlador. *Telem@tica (La Habana)*, 11(1), 47-57.