

Universidad de las Fuerzas Armadas ESPE

Nombres: Dylan Lopez, Evelyn Yar, Karol Perugachi, Edwin Arequipa.

NRC: 1322

Asignatura: Programación Orientada a Objetos

INFORME

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

- **Modelo**

El modelo maneja los datos y la lógica de la aplicación. En este caso para este ejemplo las clases que representan el modelo son “Producto” e “Inventario”.

- **Clase.-** Producto

Atributos.- Nombre, cantidad y precio.

```
Product.java | InventarioView.java | InventarioControl... | Inventario.java |
1 public class Producto {
2     private String nombre;
3     private int cantidad;
4     private double precio;
5
6     public Producto(String nombre, int cantidad, double precio) {
7         this.nombre = nombre;
8         this.cantidad = cantidad;
9         this.precio = precio;
10    }
11
12    // Getters y Setters
13    public String getNombre() {
14        return nombre;
15    }
16
17    public void setNombre(String nombre) {
18        this.nombre = nombre;
19    }
20
21    public int getCantidad() {
22        return cantidad;
23    }
24
25    public void setCantidad(int cantidad) {
26        this.cantidad = cantidad;
27    }
28
29    public double getPrecio() {
30        return precio;
31    }
32
33    public void setPrecio(double precio) {
34        this.precio = precio;
35    }
36
37    @Override
38    public String toString() {
39        return "Producto: " + nombre + ", Cantidad: " + cantidad + ", Precio: " + precio;
40    }
41 }
```

Atributos: nombre, cantidad, y precio representan la información básica del producto.

Constructor: Se usa para inicializar un producto con sus datos (nombre, cantidad y precio).

Getters y Setters: Son métodos que permiten obtener o modificar los valores de los atributos de forma controlada.

ToString(): Este método convierte el objeto "Producto" en un string que es útil para mostrarlo en la vista.

- Clase.- Inventario

Atributos.- Agregar, Buscar y Eliminar.

```
Main.java | Producto.java | InventarioView.java | InventarioControl... | Inventario.java
1 import java.util.ArrayList;
2 import java.util.List;
3
4 public class Inventario {
5     private List<Producto> productos;
6
7     public Inventario() {
8         productos = new ArrayList<>();
9     }
10
11     public void agregarProducto(Producto producto) {
12         productos.add(producto);
13     }
14
15     public List<Producto> obtenerProductos() {
16         return productos;
17     }
18
19     public Producto buscarProducto(String nombre) {
20         for (Producto p : productos) {
21             if (p.getNombre().equalsIgnoreCase(nombre)) {
22                 return p;
23             }
24         }
25         return null;
26     }
27
28     public void eliminarProducto(String nombre) {
29         productos.removeIf(producto -> producto.getNombre().equalsIgnoreCase(nombre));
30     }
31 }
```

Lista de productos: productos es una lista que contiene todos los productos en el inventario.

AgregarProducto(): Añade un producto a la lista.

ObtenerProductos(): Devuelve todos los productos del inventario.

BuscarProducto(): Busca un producto en la lista por su nombre. Si lo encuentra, lo devuelve; si no, devuelve null.

EliminarProducto(): Elimina un producto de la lista si el nombre coincide

- **Vista**

La vista muestra los datos que el controlador le proporciona en este caso con los datos utilizados.

- **Clase.-** Producto

Atributos.- Nombre, cantidad y precio.

```
Main.java  Producto.java  InventarioView.java  InventarioControlle...  Inventario.java
1  import java.util.List;
2
3  public class InventarioView {
4      public void mostrarProducto(Producto producto) {
5          if (producto != null) {
6              System.out.println(producto.toString());
7          } else {
8              System.out.println("Producto no encontrado.");
9          }
10     }
11
12     public void mostrarInventario(List<Producto> productos) {
13         System.out.println("Inventario de la tienda:");
14         for (Producto p : productos) {
15             System.out.println(p.toString());
16         }
17     }
18 }
19
```

MostrarProducto(): Muestra la información de un producto si existe, o un mensaje de error si no lo encuentra.

MostrarInventario(): Muestra la lista completa de productos en el inventario.

- **Controlador**

- **Clase.-** Inventario Controller / Esta clase gestiona la interacción entre el modelo y la vista.

```

Main.java : Producto.java : InventarioView.java : InventarioControlle... : Inventario.java :
1 public class InventarioController {
2     private Inventario modelo;
3     private InventarioView vista;
4
5     public InventarioController(Inventario modelo, InventarioView vista) {
6         this.modelo = modelo;
7         this.vista = vista;
8     }
9
10    public void agregarProducto(String nombre, int cantidad, double precio) {
11        Producto producto = new Producto(nombre, cantidad, precio);
12        modelo.agregarProducto(producto);
13    }
14
15    public void mostrarInventario() {
16        vista.mostrarInventario(modelo.obtenerProductos());
17    }
18
19    public void buscarProducto(String nombre) {
20        Producto producto = modelo.buscarProducto(nombre);
21        vista.mostrarProducto(producto);
22    }
23
24    public void eliminarProducto(String nombre) {
25        modelo.eliminarProducto(nombre);
26    }
27 }
28

```

Relación modelo-vista: El controlador tiene acceso tanto al modelo “Inventario” como a la vista “InventarioView”.

AgregarProducto(): Crea un nuevo producto y lo agrega al inventario.

MostrarInventario(): Le dice a la vista que muestre la lista de productos usando la información del inventario.

BuscarProducto(): Busca un producto en el inventario y luego usa la vista para mostrar los detalles de ese producto.

EliminarProducto(): Elimina un producto del inventario basado en su nombre.

- **Clase Principal (Main)**

```

Main.java : Producto.java : InventarioView.java : InventarioControlle... : Inventario.java :
1 public class Main {
2     public static void main(String[] args) {
3         // Crear el modelo, la vista y el controlador
4         Inventario modelo = new Inventario();
5         InventarioView vista = new InventarioView();
6         InventarioController controlador = new InventarioController(modelo, vista);
7
8         // Agregar algunos productos
9         controlador.agregarProducto("Laptop HP", 10, 1200.00);
10        controlador.agregarProducto("Teléfono Redmi 13C", 20, 600.00);
11        controlador.agregarProducto("Tablets Samsung", 15, 300.00);
12
13        // Mostrar el inventario
14        controlador.mostrarInventario();
15
16        // Buscar un producto
17        controlador.buscarProducto("Teléfono Redmi 13C");
18
19        // Eliminar un producto
20        controlador.eliminarProducto("Tablet Samsung");
21
22        // Mostrar el inventario actualizado
23        controlador.mostrarInventario();
24    }
25 }
26

```

Inicia el sistema: Crea el modelo “Inventario”, la vista “InventarioView” y el controlador “InventarioController”.

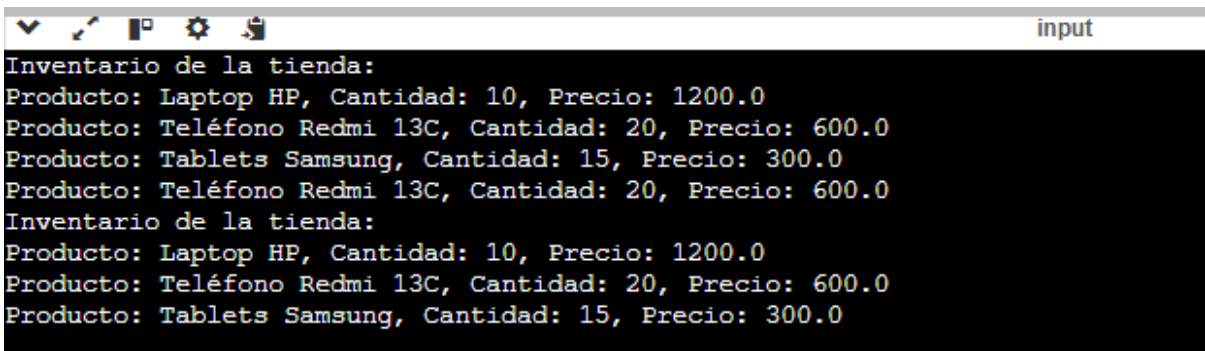
Agregar productos: Usa el controlador para agregar productos al inventario.

Mostrar inventario: Muestra todos los productos en el inventario.

Buscar un producto: Busca un producto por nombre y muestra los detalles.

Eliminar un producto: Elimina un producto del inventario y luego vuelve a mostrar el inventario actualizado.

- **Resultado**



```
input
Inventario de la tienda:
Producto: Laptop HP, Cantidad: 10, Precio: 1200.0
Producto: Teléfono Redmi 13C, Cantidad: 20, Precio: 600.0
Producto: Tablets Samsung, Cantidad: 15, Precio: 300.0
Producto: Teléfono Redmi 13C, Cantidad: 20, Precio: 600.0
Inventario de la tienda:
Producto: Laptop HP, Cantidad: 10, Precio: 1200.0
Producto: Teléfono Redmi 13C, Cantidad: 20, Precio: 600.0
Producto: Tablets Samsung, Cantidad: 15, Precio: 300.0
```