



UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

Sistema de Gestión de Inventario con MVC.

- Eduardo Anibal Mejia Catillo
- Bryan Roger Camacho Ramirez
- Israel Fernando Portilla Santamaria
- Jenifer Andrea Castro Cuevas
- Julio Emerson Coro Pineida
- Alexis Sebastian Murminacho Cabascango

NRC: 1323

Fecha límite de entrega: 16/02/2025

QUITO-ECUADOR

Link repositorio:

<https://github.com/Andre17C/DeberesAndrea2/tree/2b7674efff75afdf2f5b5cff1407428a2fcd1a2f/GRUPO%209/TRABAJO%20EXPERIMENTAL>

Link del video:

https://drive.google.com/file/d/14_dCdT_pD3fNdKKOzbI5WKqNQBWsdD4R/view?usp=drive_sdk

Introducción

El objetivo de este proyecto es desarrollar un sistema de gestión de inventario utilizando el lenguaje de programación Java. Este sistema permite registrar productos, actualizar inventarios, consultar listas de productos y calcular el valor total del inventario. El proyecto se enfoca en mejorar la eficiencia y precisión en el manejo de inventarios, facilitando la administración y toma de decisiones en cualquier negocio.

Justificación

La gestión de inventarios es una parte esencial de cualquier empresa que maneje productos físicos. La eficiencia en la administración de inventarios puede reducir costos operativos, evitar pérdidas y mejorar el servicio al cliente. Este sistema automatizado se justifica por su capacidad de proporcionar una solución ágil y precisa para la gestión de inventarios, minimizando errores y optimizando el uso de recursos.

Objetivos Específicos

- Desarrollar un sistema que permita registrar productos con su respectivo código, nombre, precio y cantidad.
- Implementar una función para actualizar las cantidades de productos en el inventario. Crear una interfaz de usuario en consola para interactuar con el sistema de inventario. Calcular y mostrar el valor total de todos los productos en el inventario.

Objetivo General

- Desarrollar un sistema de gestión de inventario eficiente y fácil de usar que facilite el control y la administración de productos en una empresa.

Marco Teórico o Estado de la Cuestión

El marco teórico de este proyecto se basa en conceptos de programación orientada a objetos (POO) y estructuras de datos. Se utilizan clases para representar productos y el inventario, y una lista para almacenar los productos. La interfaz de usuario se realiza mediante una consola, utilizando la clase Scanner para la entrada de datos. La programación orientada a objetos permite un diseño modular y reutilizable, mientras que las estructuras de datos eficaces garantizan un acceso rápido y eficiente a la información del inventario.

Metodología

El desarrollo del sistema se llevó a cabo en varias etapas:

Diseño del modelo de datos: Se definieron las clases Producto e Inventario, y sus métodos correspondientes.

Implementación de la lógica del negocio: Se implementaron los métodos para registrar productos, actualizar el inventario, consultar productos y calcular el valor total del inventario.

Diseño de la interfaz de usuario: Se creó la clase VistaConsola para interactuar con el usuario a través de la consola.

Integración y pruebas: Se integraron todas las clases en la clase ControladorInventario y se realizaron pruebas para asegurar el correcto funcionamiento del sistema.

Código

```
import java.util.ArrayList;
import java.util.Scanner;

class Producto {

    private String codigo;

    private String nombre;

    private double precio;

    private int cantidad;

    public Producto(String codigo, String nombre, double precio, int cantidad) {

        this.codigo = codigo;

        this.nombre = nombre;

        this.precio = precio;

        this.cantidad = cantidad;

    }

    public String getCodigo() {

        return codigo;

    }

    public void setCantidad(int cantidad) {

        this.cantidad = cantidad;

    }

    public double getPrecio() {

        return precio;

    }

    public int getCantidad() {
```

```

        return cantidad;
    }

    @Override

    public String toString() {

        return "Codigo: " + codigo + ", Nombre: " + nombre + ", Precio: " +
String.format("%.2f", precio) + ", Cantidad: " + cantidad;

    }
}

class Inventario {

    private ArrayList<Producto> productos;

    public Inventario() {

        productos = new ArrayList<>();
    }

    public void registrarProducto(Producto producto) {

        productos.add(producto);
    }

    public boolean actualizarInventario(String codigo, int nuevaCantidad) {

        for (Producto producto : productos) {

            if (producto.getCodigo().equals(codigo)) {

                producto.setCantidad(nuevaCantidad);

                return true;

            }

        }
    }
}

```

```

        return false;
    }

    public ArrayList<Producto> consultarProductos() {

        return productos;
    }

    public double calcularValorTotal() {

        double total = 0;

        for (Producto producto : productos) {

            total += producto.getPrecio() * producto.getCantidad();
        }

        return total;
    }
}

class VistaConsola {

    private Scanner scanner;

    public VistaConsola() {

        scanner = new Scanner(System.in);
    }

    public void mostrarMenu() {

        System.out.println("\n--- Sistema de Gestion de Inventario ---");

        System.out.println("1. Registrar producto");

        System.out.println("2. Actualizar inventario");

        System.out.println("3. Consultar productos");
    }
}

```

```

        System.out.println("4. Calcular valor total del inventario");

        System.out.println("5. Salir");
    }

    public int leerOpcion() {

        System.out.print("Seleccione una opcion: ");

        int opcion = scanner.nextInt();

        scanner.nextLine(); // Limpiar buffer

        return opcion;
    }

    public Producto solicitarDatosProducto() {

        System.out.print("Ingrese el codigo del producto: ");

        String codigo = scanner.nextLine();

        System.out.print("Ingrese el nombre del producto: ");

        String nombre = scanner.nextLine();

        System.out.print("Ingrese el precio del producto: ");

        double precio = scanner.nextDouble();

        System.out.print("Ingrese la cantidad del producto: ");

        int cantidad = scanner.nextInt();

        scanner.nextLine(); // Limpiar buffer

        return new Producto(codigo, nombre, precio, cantidad);
    }

    public String solicitarCodigo() {

```

```
        System.out.print("Ingrese el codigo del producto: ");

        return scanner.nextLine();
    }

    public int solicitarCantidad() {

        System.out.print("Ingrese la nueva cantidad: ");

        return scanner.nextInt();
    }

    public void mostrarProductos(ArrayList<Producto> productos) {

        if (!productos.isEmpty()) {

            System.out.println("\nLista de productos:");

            for (Producto producto : productos) {

                System.out.println(producto);

            }

        } else {

            System.out.println("\nNo hay productos registrados.");

        }

    }

    public void mostrarValorTotal(double valor) {

        System.out.printf("\nEl valor total del inventario es: %.2f\n", valor);

    }

    public void mostrarMensaje(String mensaje) {

        System.out.println(mensaje);
    }
}
```



```

    }
}

class ControladorInventario {

    private Inventario inventario;

    private VistaConsola vista;

    public ControladorInventario() {

        inventario = new Inventario();

        vista = new VistaConsola();

    }

    public void iniciar() {

        while (true) {

            vista.mostrarMenu();

            int opcion = vista.leerOpcion();

            switch (opcion) {

                case 1:

                    Producto producto = vista.solicitarDatosProducto();

                    inventario.registrarProducto(producto);

                    vista.mostrarMensaje("Producto registrado exitosamente.");

                    break;

                case 2:

                    String codigo = vista.solicitarCodigo();

                    int cantidad = vista.solicitarCantidad();

                    if (inventario.actualizarInventario(codigo, cantidad)) {

```

```
        vista.mostrarMensaje("Inventario actualizado exitosamente.");

    } else {

        vista.mostrarMensaje("Producto no encontrado.");

    }

    break;

case 3:

    ArrayList<Producto> productos = inventario.consultarProductos();

    vista.mostrarProductos(productos);

    break;

case 4:

    double valorTotal = inventario.calcularValorTotal();

    vista.mostrarValorTotal(valorTotal);

    break;

case 5:

    vista.mostrarMensaje("Saliendo del sistema. ¡Adios!");

    return;

default:

    vista.mostrarMensaje("Opcion invalida. Intente nuevamente.");

}

}

}

}
```

```
public class Main {  
  
    public static void main(String[] args) {  
  
        ControladorInventario controlador = new ControladorInventario();  
  
        controlador.iniciar();  
  
    }  
  
}
```

Análisis de Resultados

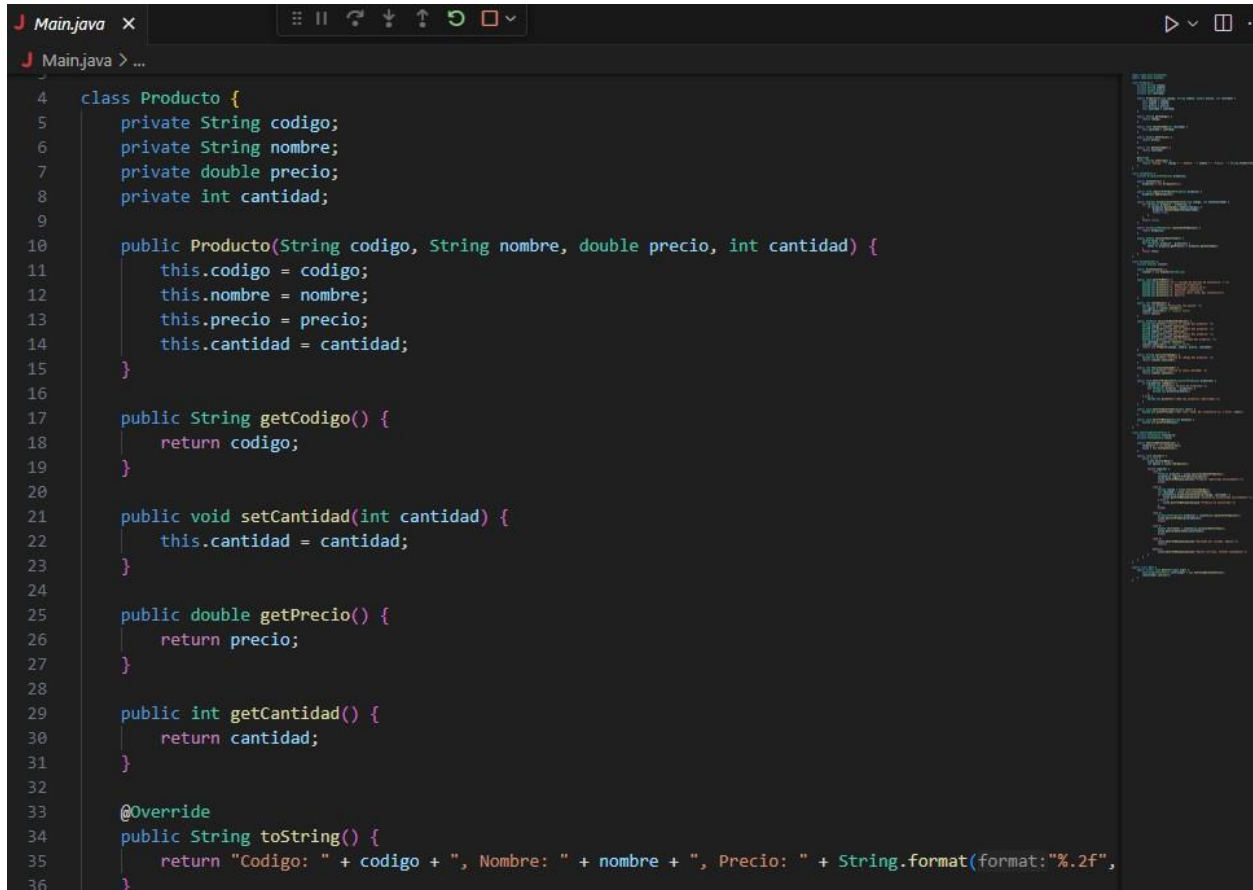
El sistema desarrollado permite realizar todas las funciones esperadas de manera eficiente. Los usuarios pueden registrar nuevos productos, actualizar las cantidades existentes, consultar la lista de productos y calcular el valor total del inventario. La interfaz de usuario es intuitiva y fácil de usar, lo que facilita la adopción del sistema por parte de los usuarios.

Conclusiones

El sistema de gestión de inventario desarrollado cumple con los objetivos planteados, ofreciendo una solución eficiente y fácil de usar para la administración de inventarios. La programación orientada a objetos y las estructuras de datos utilizadas han permitido un diseño modular y reutilizable, facilitando la futura ampliación y mejora del sistema.

Anexos

Muestra del código y compilación



The image shows a screenshot of an IDE with a dark theme. The main editor window displays the source code for a Java class named 'Producto'. The code is as follows:

```
4  class Producto {
5      private String codigo;
6      private String nombre;
7      private double precio;
8      private int cantidad;
9
10     public Producto(String codigo, String nombre, double precio, int cantidad) {
11         this.codigo = codigo;
12         this.nombre = nombre;
13         this.precio = precio;
14         this.cantidad = cantidad;
15     }
16
17     public String getCodigo() {
18         return codigo;
19     }
20
21     public void setCantidad(int cantidad) {
22         this.cantidad = cantidad;
23     }
24
25     public double getPrecio() {
26         return precio;
27     }
28
29     public int getCantidad() {
30         return cantidad;
31     }
32
33     @Override
34     public String toString() {
35         return "Codigo: " + codigo + ", Nombre: " + nombre + ", Precio: " + String.format(format:"%.2f",
36     }
```

The IDE interface includes a top bar with a file explorer, a toolbar with icons for running, debugging, and other actions, and a right-hand pane showing a project structure or output window.

```

39 class Inventario {
40     private ArrayList<Producto> productos;
41
42     public Inventario() {
43         productos = new ArrayList<>();
44     }
45
46     public void registrarProducto(Producto producto) {
47         productos.add(producto);
48     }
49
50     public boolean actualizarInventario(String codigo, int nuevaCantidad) {
51         for (Producto producto : productos) {
52             if (producto.getCodigo().equals(codigo)) {
53                 producto.setCantidad(nuevaCantidad);
54                 return true;
55             }
56         }
57         return false;
58     }
59
60     public ArrayList<Producto> consultarProductos() {
61         return productos;
62     }
63
64     public double calcularValorTotal() {
65         double total = 0;
66         for (Producto producto : productos) {
67             total += producto.getPrecio() * producto.getCantidad();
68         }
69         return total;
70     }
71 }
72

```

```

73 class VistaConsola {
74     private Scanner scanner;
75
76     public VistaConsola() {
77         scanner = new Scanner(System.in);
78     }
79
80     public void mostrarMenu() {
81         System.out.println(x:"\n--- Sistema de Gestión de Inventario ---");
82         System.out.println(x:"1. Registrar producto");
83         System.out.println(x:"2. Actualizar inventario");
84         System.out.println(x:"3. Consultar productos");
85         System.out.println(x:"4. Calcular valor total del inventario");
86         System.out.println(x:"5. Salir");
87     }
88
89     public int leerOpcion() {
90         System.out.print(s:"Seleccione una opción: ");
91         int opcion = scanner.nextInt();
92         scanner.nextLine(); // Limpiar buffer
93         return opcion;
94     }
95
96     public Producto solicitarDatosProducto() {
97         System.out.print(s:"Ingrese el código del producto: ");
98         String codigo = scanner.nextLine();
99         System.out.print(s:"Ingrese el nombre del producto: ");
100        String nombre = scanner.nextLine();
101        System.out.print(s:"Ingrese el precio del producto: ");
102        double precio = scanner.nextDouble();
103        System.out.print(s:"Ingrese la cantidad del producto: ");
104        int cantidad = scanner.nextInt();
105        scanner.nextLine(); // Limpiar buffer
106        return new Producto(codigo, nombre, precio, cantidad);

```

```

106     return new Producto(codigo, nombre, precio, cantidad);
107 }
108
109 public String solicitarCodigo() {
110     System.out.print(s:"Ingrese el código del producto: ");
111     return scanner.nextLine();
112 }
113
114 public int solicitarCantidad() {
115     System.out.print(s:"Ingrese la nueva cantidad: ");
116     return scanner.nextInt();
117 }
118
119 public void mostrarProductos(ArrayList<Producto> productos) {
120     if (!productos.isEmpty()) {
121         System.out.println(x:"\nLista de productos:");
122         for (Producto producto : productos) {
123             System.out.println(producto);
124         }
125     } else {
126         System.out.println(x:"\nNo hay productos registrados.");
127     }
128 }
129
130 public void mostrarValorTotal(double valor) {
131     System.out.printf(format: "\nEl valor total del inventario es: %.2f\n", valor);
132 }
133
134 public void mostrarMensaje(String mensaje) {
135     System.out.println(mensaje);
136 }
137 }

```

```

139 class ControladorInventario {
140     private Inventario inventario;
141     private VistaConsola vista;
142
143     public ControladorInventario() {
144         inventario = new Inventario();
145         vista = new VistaConsola();
146
147     }
148
149     public void iniciar() {
150         while (true) {
151             vista.mostrarMenu();
152             int opcion = vista.leerOpcion();
153
154             switch (opcion) {
155                 case 1:
156                     Producto producto = vista.solicitarDatosProducto();
157                     inventario.registrarProducto(producto);
158                     vista.mostrarMensaje(mensaje: "Producto registrado exitosamente.");
159                     break;
160
161                 case 2:
162                     String codigo = vista.solicitarCodigo();
163                     int cantidad = vista.solicitarCantidad();
164                     if (inventario.actualizarInventario(codigo, cantidad)) {
165                         vista.mostrarMensaje(mensaje: "Inventario actualizado exitosamente.");
166                     } else {
167                         vista.mostrarMensaje(mensaje: "Producto no encontrado.");
168                     }
169                     break;
170
171                 case 3:
172                     ArrayList<Producto> productos = inventario.consultarProductos();
173                     vista.mostrarProductos(productos);
174             }
175         }
176     }
177 }

```

```

172         vista.mostrarProductos(productos);
173     }
174     break;
175     case 4:
176         double valorTotal = inventario.calcularValorTotal();
177         vista.mostrarValorTotal(valorTotal);
178         break;
179
180     case 5:
181         vista.mostrarMensaje(mensaje:"Saliendo del sistema. ¡Adiós!");
182         return;
183
184     default:
185         vista.mostrarMensaje(mensaje:"Opción inválida. Intente nuevamente.");
186     }
187 }
188 }
189 }
190
191 public class Main {
192     Run | Debug
193     public static void main(String[] args) {
194         ControladorInventario controlador = new ControladorInventario();
195         controlador.iniciar();
196     }
197 }

```

Producto registrado exitosamente.

--- Sistema de Gestión de Inventario ---

1. Registrar producto
2. Actualizar inventario
3. Consultar productos
4. Calcular valor total del inventario
5. Salir

Seleccione una opción: 1

Ingrese el código del producto: 123463

Ingrese el nombre del producto: Powerade

Ingrese el precio del producto: 1,05

Ingrese la cantidad del producto: 25

Producto registrado exitosamente.

--- Sistema de Gestión de Inventario ---

1. Registrar producto
2. Actualizar inventario
3. Consultar productos
4. Calcular valor total del inventario
5. Salir

Seleccione una opción: 4

El valor total del inventario es: 113,25

--- Sistema de Gestión de Inventario ---

1. Registrar producto
2. Actualizar inventario
3. Consultar productos
4. Calcular valor total del inventario
5. Salir

Seleccione una opción: 3

Lista de productos:

Codigo: 123456, Nombre: Arroz, Precio: 0,60, Cantidad: 40

Codigo: 123465, Nombre: Atun, Precio: 1,00, Cantidad: 30

Codigo: 123464, Nombre: Sardina, Precio: 1,10, Cantidad: 30

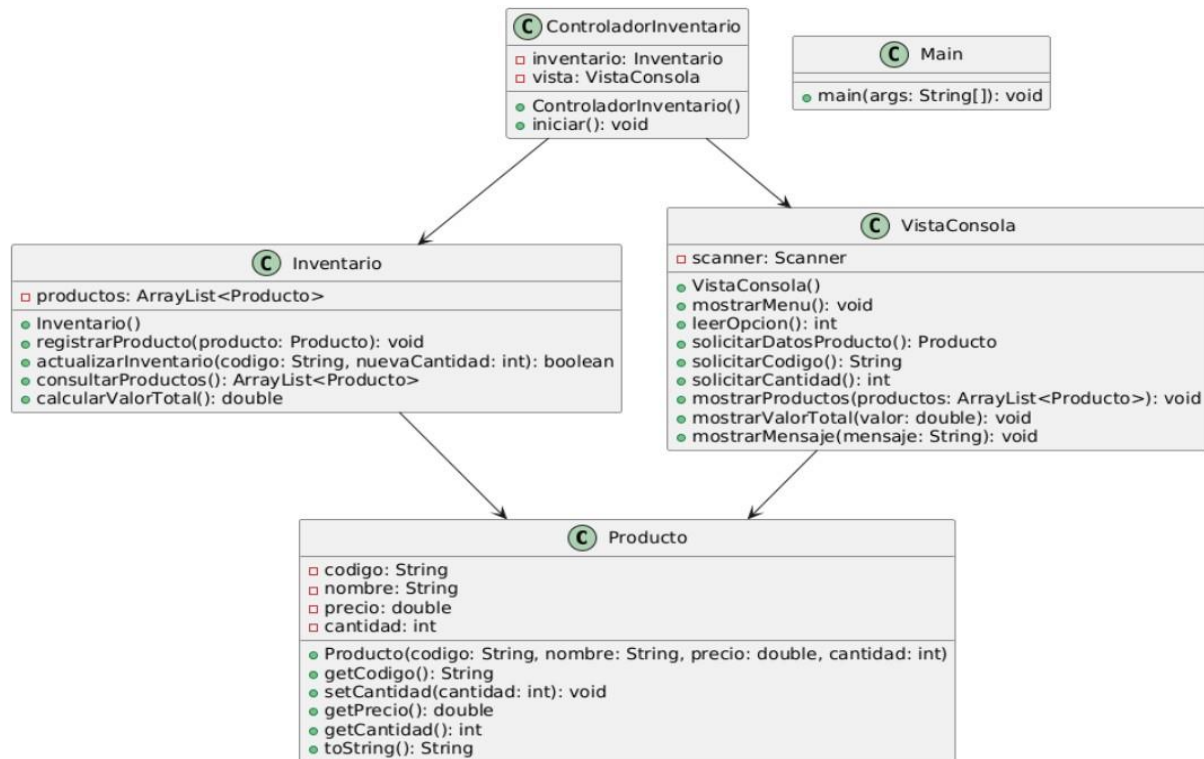
Codigo: 123463, Nombre: Powerade, Precio: 1,05, Cantidad: 25

Run: ...

Run: Main

powershell

UML



Bibliografias:

Paradigma De La Programación Orientada, E., & En, O. (n.d.). POO y MVC en PHP.

<https://compuzoft.com/edukcol/storage/actividades/3.pdf>

Steve Burbeck. Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller (MVC).

<http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>.

<https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>

Justificación

Eduardo Anibal Mejia Catillo, Bryan Roger Camacho Ramirez y Alexis Sebastian Murminacho Cabascango no participaron del video dado que tuvieron motivos de trabajo que atender, pero si participaron activamente en el desarrollo del programa y su implementación.

