

Manual Técnico

Inventario de Tienda

Katherin Yasmin Miranda Hernández

12/09/2025

Introducción

El sistema de inventario de tienda fue desarrollado como un proyecto de uso académico. Esta diseñado para la gestión de productos, registrar ventas y generar reportes en PDF.

Este manual técnico describe los aspectos internos de la aplicación, como las clases principales, la librería utilizada y los pasos necesarios para su ejecución.

Requisitos del sistema

Hardware:

- Procesador Core i3 en adelante
- 8gb de memoria RAM

Software:

- Lenguaje: Java
- Librerías: iTextPDF
- IDE: NetBeans (recomendado)

Estructura del código

InventarioTienda (clase principal)

```
int opcion = 0;
do{
    System.out.println("--Menu--");
    System.out.println("1. Agregar producto");
    System.out.println("2. Buscar producto");
    System.out.println("3. Eliminar producto");
    System.out.println("4. Registrar venta");
    System.out.println("5. Mostrar inventario");
    System.out.println("6. Generar reporte stock PDF");
    System.out.println("7. Generar reporte de ventas PDF");
    System.out.println("8. Mostrar bitacora");
    System.out.println("9. Ver datos estudiante");
    System.out.println("10. Salir");
    System.out.println("Seleccione una opcion: ");
```

Controla todo el programa y es la clase que contiene el menú

Primero solicita los datos del usuario y del estudiante, luego carga productos al inventario

Muestra el menú, como se muestra en la imagen, al seleccionar una opción se guarda en la bitácora.

ReportePDF

```
// Clase ReportePDF de la librería iText
class ReportePDF{
    //generar el reporte en pdf del stock
    public static void generarReporteStock(Inventario inventario, String usuario) {
        LocalDateTime fechaHora = LocalDateTime.now();
        DateTimeFormatter formato = DateTimeFormatter.ofPattern("dd_MM_yyyy_HH_mm_ss");
        String nombreArchivo = formato.format(fechaHora) + "_Stock.pdf";
        Document doc = new Document();
    }
}
```

Su función es generar los reportes en formato PDF utilizando la librería iText

Los métodos que tiene esta clase son:

- generarReporteStock(Inventario, String): este método genera un PDF y muestra los productos que existan en el inventario en forma de tabla.
- generarReporteVentas(String): este método genera un PDF con todas las ventas que se hayan registrado

Bitácora

```
// mostrar lo que hay en la bitacora
public static void mostrarBitacora() {
    try {
        Scanner lector = new Scanner(new File("bitacora.txt"));
        System.out.println("===== BITACORA =====");
        while (lector.hasNextLine()) {
            String linea = lector.nextLine();
            System.out.println(linea);
        }
        lector.close();
    } catch (Exception e) {
        System.out.println("No hay bitacora o no se pudo leer");
    }
}
}
```

La clase bitácora registra cada acción que el usuario realice.

Los métodos que contiene esta clase son:

- registrarAccion(String acción, String resultado, String usuario): este método guarda la acción en la bitácora
- mostrarBitacora(): este método lee y muestra el contenido de la bitácora en consola.

Venta

```
class Venta{
    //metodo estatico para registrar una venta
    public static void registrarVenta(Inventario inventario, String codigoVenta, int cantidadVenta, String usuario){
        Producto productoVendido = inventario.obtenerProducto(codigoVenta);
        if (productoVendido == null){
            System.out.println("No existe el producto");
            Bitacora.registrarAccion("Registrar venta", "Fallida", usuario);
            return;
        }
    }
}
```

La clase venta es la que se encarga de manejar las ventas realizadas

Métodos:

- registrarVenta(Inventario, String codigoVenta, int cantidadVenta, String usuario):
registra la venta del producto, realizando los siguientes pasos
 - Busca el producto en el inventario
 - Verifica si la cantidad que se desea vender es suficiente con la equivalente en stock
 - Resta a la cantidad que existía
 - Guarda la acción en la bitácora

Inventario

```
class Inventario{
    //aqui guardo los productos
    public Producto[] listaProductos = new Producto[50];
    public int totalProductos = 0;

    //metodo para agregar un producto
    public boolean agregarProducto(Producto nuevoProducto){
        if (totalProductos >= 50){
            System.out.println("Inventario lleno");
            return false;
        }
        //verificacion de duplicado (recorrido)
        int i = 0;
        while (i < totalProductos) {
            Producto productoActual = listaProductos[i];
            if (productoActual != null) {
                if (productoActual.codigoProducto != null) {
                    if (productoActual.codigoProducto.equalsIgnoreCase(nuevoProducto.codigoProducto)) {
                        System.out.println("Ya existe un producto con ese codigo");
                        return false;
                    }
                }
            }
            i++;
        }
        listaProductos[totalProductos] = nuevoProducto;
        totalProductos++;
        System.out.println("Producto agregado");
        return true;
    }
}
```

Esta es la clase inventario, gestiona la lista de productos existentes

Atributos:

- Producto[] listaProductos: es un vector que permite registrar hasta un máximo de 50 productos
- Int totalProductos

Métodos:

- agregarProducto(Producto): agrega un producto
- buscarProducto(String): busca el producto por código, nombre o categoría
- eliminarProducto(String): elimina un producto en base al código
- mostrarInventario(): muestra todos los productos en el inventario
- obtenerProducto(String): devuelve un producto buscado por código

Producto

```
class Producto {
    //atributos del producto
    public String nombreProducto;
    public String categoriaProducto;
    public int cantidadProducto;
    public String codigoProducto;
    public double precioProducto;

    //constructor con datos
    public Producto(String nombre, String categoria, double precio, int cantidad, String codigo){
        nombreProducto = nombre;
        categoriaProducto = categoria;
        precioProducto = precio;
        cantidadProducto = cantidad;
        codigoProducto = codigo;
    }
}
```

Esta es la clase producto, aquí se almacena la información del producto

Atributos:

- String nombreProducto
- String categoriaProducto
- Int cantidadProducto
- String codigoProducto
- Double precioProducto

Métodos:

- Producto(String nombre, String cadena, double precio, int cantidad, String código): constructor que inicializa un producto con sus datos
- mostrarProducto(): imprime en consola la información del producto.

Diagrama de flujo



