

Manual Técnico

Sistema De gestión de inventarios

Estructura del proyecto

Inventario/

→ src/

- Main.java # Clase principal con el menú del sistema
- Producto.java # Gestión de productos (CRUD básico)
- Venta.java # Registro de ventas en archivo de texto
- Reportes.java # Generación de reportes en PDF y bitácora

→ librerías/

- itextpdf-5.5.12.jar # Librería externa para exportar reportes en PDF

Main.java: punto de entrada del sistema, contiene el menú principal y el flujo de control.

Producto.java: permite agregar, buscar y eliminar productos del inventario.

Venta.java: gestiona la venta de productos, descuenta stock y registra en ventas.txt.

Reportes.java: genera reportes de stock y ventas en PDF utilizando **iTextPDF**, además de mantener una bitácora de acciones.

librerías/: contiene dependencias externas necesarias (en este caso iTextPDF).

2. Requisitos de Software y Hardware

Software necesario

- **Java JDK 17 o superior** (para compilar y ejecutar el proyecto).
- **Editor/IDE recomendado:**
 - IntelliJ IDEA, Eclipse o NetBeans.
- **Sistema Operativo compatible:**
 - Windows, Linux o macOS.
- **Librerías externas:**
 - itextpdf-5.5.12.jar (para la generación de reportes en PDF).

Hardware mínimo recomendado

- **Procesador:** Intel i3 o equivalente.
- **Memoria RAM:** 4 GB mínimo (8 GB recomendado).
- **Almacenamiento:** 500 MB libres para archivos generados (PDF y TXT).

- **Pantalla:** resolución mínima 1366x768.
 - **Periféricos:** teclado para ingresar datos.
-

3. Descripción de los Métodos Principales

◆ Clase Main

- **main(String[] args)** → Inicia el programa, muestra el menú principal y controla la ejecución de opciones.
 - **mostrarMenuInventario()** → Muestra las opciones disponibles al usuario.
 - **mostrarDatosEstudiante()** → Imprime información del autor del proyecto.
-

◆ Clase Producto

- **Agregarproducto(Scanner teclado)**
Permite ingresar un nuevo producto (validando código único, precio y cantidad positiva).
 - **Buscarproducto(Scanner teclado)**
Búsqueda por código, nombre o categoría dentro del inventario.
 - **Eliminarproducto(Scanner teclado)**
Elimina un producto existente desplazando los elementos en el arreglo.
 - **getInventario(), getContador()**
Métodos de apoyo para obtener la lista de productos y la cantidad actual.
-

◆ Clase Venta

- **Registrarventa(Scanner teclado)**
 - Solicita un código de producto.
 - Verifica existencia y stock disponible.
 - Calcula el total de la venta.
 - Descuenta del inventario.
 - Registra la transacción en el archivo ventas.txt con fecha y hora.
-

◆ Clase Reportes

- **Generarreportes(Scanner teclado)**
Muestra un submenú para generar:
 - **Reporte de Stock (PDF)**
 - **Reporte de Ventas (PDF)**
- **generarReporteStock()**
 - Extrae la información de Producto[] inventario.
 - Genera un PDF con tabla de productos.
- **generarReporteVentas()**

- Extrae la información de Ventas[] inventario.
 - Genera un PDF con la información de ventas registradas.
 - **registrarAccion(tipoAccion, estado, usuarioActual)**
 - Guarda acciones en la **bitácora** (máx. 200 registros).
 - **bitácora()**
 - Muestra en consola las acciones realizadas (ej. generación de reportes, errores, etc.).
-

4. Flujo General del Sistema

1. El usuario ingresa su nombre de usuario
 2. El usuario ingresa al menú principal (Main.java).
 3. Puede **gestionar productos** (agregar, buscar, eliminar).
 4. Puede **registrar ventas** que se guardan en ventas.txt.
 5. Puede **generar reportes en PDF** de stock y ventas (con iTextPDF).
 6. El sistema registra cada acción en la **bitácora de auditoría**.
-

5. Mantenimiento del Sistema

- **Agregar funcionalidades nuevas:**
 - Extender clases Producto o Venta para más campos.
 - Incluir reportes adicionales en Reportes.java.
- **Actualizar librerías:**
 - Sustituir iTextPDF por una versión más reciente o compatible. Como la versión iTextPDF7
- **Respaldo de archivos:**
 - Realizar copias periódicas de ventas.txt y de los reportes generados.