



**IES Luis Vélez de Guevara**  
**Dpto. de Informática**

**UD01**

-

**Gestión de la información almacenada en  
Ficheros**

# Índice

## Índice de contenido

<b>1. Introducción</b>	<b>3</b>
Objetivo del Proyecto	3
Alcance del Proyecto	3
<b>2. Requisitos del Sistema</b>	<b>3</b>
Requisitos Funcionales	3
Requisitos No Funcionales	3
<b>3. Arquitectura del Sistema</b>	<b>4</b>
Diagrama de Arquitectura	4
Descripción de Componentes	4
<b>4. Diseño del Sistema</b>	<b>5</b>
Diagrama de Clases	5
Descripción de Clases	5
<b>5. Implementación</b>	<b>6</b>
<b>7. Conclusiones y Futuras Mejoras</b>	<b>7</b>
Conclusiones	7
Futuras Mejoras	7

# 1. Introducción

## Objetivo del Proyecto

El objetivo de este proyecto es desarrollar una aplicación de gestión de productos que permita a los usuarios realizar operaciones básicas como agregar, modificar, eliminar y buscar productos en una tienda. Además, la aplicación tiene la capacidad de exportar la lista de productos a un archivo CSV, facilitando así la gestión y análisis de inventarios.

## Alcance del Proyecto

La aplicación está diseñada para ser utilizada en el ámbito de la gestión de productos de una tienda, enfocándose en las operaciones de inventario. No incluye funcionalidades de gestión de usuarios, autenticación o procesos de venta.

# 2. Requisitos del Sistema

## Requisitos Funcionales

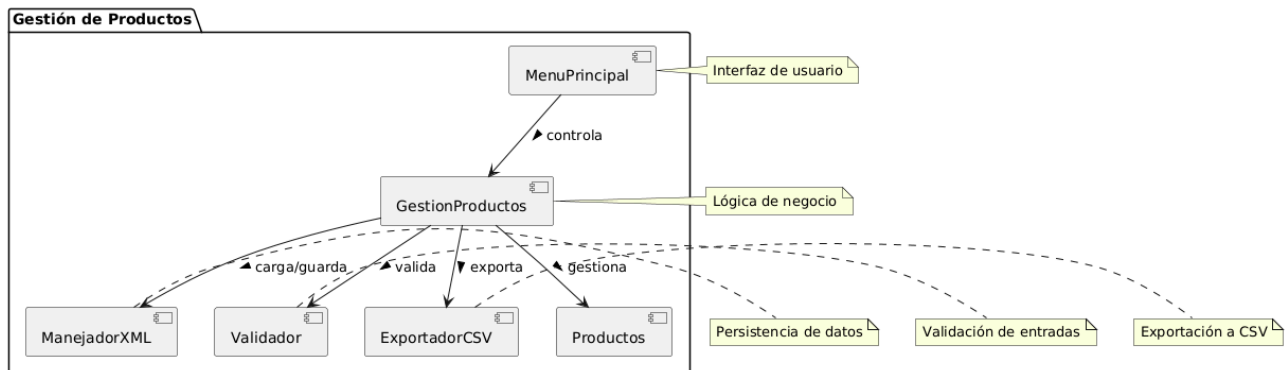
1. **Agregar productos:** Permitir al usuario ingresar nuevos productos en el sistema.
2. **Buscar productos:** Habilitar la búsqueda de productos por nombre.
3. **Modificar productos:** Permitir la modificación de la información de un producto existente.
4. **Eliminar productos:** Facilitar la eliminación de productos de la lista.
5. **Exportar productos a CSV:** Proporcionar la opción de exportar la lista de productos a un archivo en formato CSV.

## Requisitos No Funcionales

- La interfaz de usuario debe ser intuitiva y fácil de navegar.
- La aplicación debe ser capaz de gestionar al menos 1000 productos sin afectar el rendimiento.
- Los mensajes de error y validación deben ser claros y útiles para el usuario.

### 3. Arquitectura del Sistema

#### Diagrama de Arquitectura



#### Descripción de Componentes

**MenuPrincipal**: Clase que gestiona la interfaz de usuario y controla el flujo del menú principal.

**GestionProductos**: Clase responsable de la lógica de negocio relacionada con la gestión de productos, incluyendo la adición, búsqueda, modificación y eliminación de productos.

**ManejadorXML**: Clase que utiliza JAXB para cargar y guardar la lista de productos desde/hacia un archivo XML, facilitando la persistencia de datos.

**Validador**: Clase que asegura la validez de la entrada del usuario, validando precios, cantidades y opciones de menú.

**ExportadorCSV**: Clase que maneja la exportación de la lista de productos a un archivo CSV, permitiendo a los usuarios exportar y analizar los datos de inventario.

## 4. Diseño del Sistema

### Diagrama de Clases



### Descripción de Clases

- **Producto:** Representa un producto en el inventario, con atributos como nombre, precio y cantidad.
- **Productos:** Contiene una lista de objetos Producto y métodos para gestionar esta lista, como agregar y eliminar productos.
- **GestionProductos:** Proporciona métodos para las operaciones de gestión de productos, interactuando directamente con la lista de productos.
- **ManejadorXML:** Utiliza JAXB para la serialización y deserialización de productos en formato XML, permitiendo la carga y el guardado de datos.
- **Validador:** Contiene métodos para validar las entradas del usuario, asegurando que sean correctas antes de procesarlas.

- **ExportadorCSV**: Clase que maneja la exportación de la lista de productos a un archivo CSV, permitiendo a los usuarios exportar y analizar los datos de inventario.
- **MenuOpciones (Enum)**: Representa las opciones del menú principal que el usuario puede seleccionar.
  - **Opciones Definidas**:

AGREGAR\_PRODUCTO: Permite al usuario agregar un nuevo producto.

BUSCAR\_PRODUCTO: Permite al usuario buscar un producto existente.

MODIFICAR\_PRODUCTO: Permite al usuario modificar la información de un producto.

ELIMINAR\_PRODUCTO: Permite al usuario eliminar un producto de la lista.

EXPORTAR\_CSV: Permite al usuario exportar la lista de productos a un archivo CSV.

SALIR: Permite al usuario salir de la aplicación.
  - **Importancia**: Al utilizar un enum, se mejora la legibilidad del código y se minimiza el riesgo de errores al manejar opciones del menú, ya que se utilizan constantes en lugar de cadenas de texto sueltas.

## 5. Implementación

Los métodos implementados en la clase GestionProductos permiten realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre la lista de productos. Por ejemplo, el método agregarProducto valida la entrada del usuario, crea un nuevo objeto Producto y lo agrega a la lista. El método buscarProducto recorre la lista de productos y retorna el producto que coincide con el nombre ingresado.

### Estrategia de Manejo de Errores

Se implementan validaciones en todas las entradas del usuario, utilizando la clase Validador para asegurar que los datos ingresados sean correctos. En caso de errores, se muestran mensajes claros para guiar al usuario sobre cómo corregir la entrada.

### Uso de Bibliotecas Externas

- **Jakarta XML Binding (JAXB)**: Utilizada para el manejo de la serialización y deserialización de productos en formato XML.

## **7. Conclusiones y Futuras Mejoras**

### **Conclusiones**

Este proyecto ha permitido desarrollar una comprensión profunda de la gestión de datos en aplicaciones Java, así como la utilización de técnicas de validación de entrada y manejo de archivos XML y CSV.

### **Futuras Mejoras**

- Implementar un sistema de gestión de usuarios para proporcionar autenticación y autorización.
- Añadir una interfaz gráfica de usuario (GUI) para mejorar la experiencia del usuario.
- Incluir la funcionalidad de búsqueda avanzada por diferentes atributos (precio, cantidad, etc.).