

DOCUMENTACION DE CLASES
SISTEMA DE GESTION DE INVENTARIOS EN
JAVA

DESARROLLADO POR
VLADIMIR ADRIANO CORTES ROJAS

JOSÉ FRANCISCO VITOLA R.

PROGRAMACION ORIENTADA A OBJETOS
(POO)

Descripción del Proyecto

El proyecto es un *Sistema de Gestión de Inventarios* diseñado para administrar productos, categorías, proveedores y usuarios en un entorno empresarial. Este sistema permite realizar operaciones como agregar, modificar, eliminar y buscar productos, gestionar proveedores, generar reportes detallados del inventario y registrar cambios en el sistema. Además, incluye un sistema de autenticación y jerarquización de usuarios, diferenciando entre administradores y usuarios normales.

El propósito principal del proyecto es **facilitar la gestión eficiente de inventarios y proveedores, asegurando un control detallado de los productos y sus movimientos, mientras se mantiene la seguridad mediante un sistema de usuarios autenticados**.

Clases del Proyecto

A continuación, se detalla el funcionamiento y las relaciones de las 19 clases que componen el sistema:

1. InventoryManagementSystem

- Función: Es la clase principal que inicia el programa. Maneja el flujo principal del sistema, incluyendo el inicio de sesión, la carga y el guardado del inventario, y la navegación por los menús.
- Relaciones: Utiliza *UserManagement* para manejar usuarios. Interactúa con *Inventory*, *InventoryReport*, y *InventoryRepository* para gestionar el inventario. Llama a ``MainMenu`` para mostrar el menú principal.

2. MainMenu

- Función: Muestra el menú principal del sistema y procesa las opciones seleccionadas por el usuario.
- Relaciones: Llama a las clases de menú secundarias como *InventoryModificationMenu*, *SearchMenu*, *ReportMenu*, y *SupplierMenu*. Interactúa con *Inventory*, *InventoryReport*, *SupplierManagement*, y *UserManagement*.

3. Inventory

- Función: Administra las operaciones relacionadas con el inventario, como agregar productos, modificar cantidades, eliminar categorías, y buscar productos.
- Relaciones: Utiliza *Producto* para representar los productos. Interactúa con *StockAlert* para verificar niveles bajos de stock. Usa *FileCSV* para importar/exportar inventarios. Se apoya en *InventoryRepository* para cargar y guardar datos.

4. Producto

- Función: Representa un producto en el inventario, con atributos como nombre, categoría, cantidad y precio.
- Relaciones: Es utilizado por *Inventory* para gestionar productos.

5. StockAlert

- Función: Verifica si hay productos con niveles bajos de stock y genera alertas.
- Relaciones: Es llamado por *Inventory* para realizar verificaciones de stock.

6. InventoryReport

- Función: Genera reportes detallados del inventario y registra los cambios realizados en el sistema.
- Relaciones: Es utilizado por *Inventory* y *ReportMenu* para registrar y mostrar cambios.

7. FileCSV

- Función: Permite importar y exportar inventarios en formato CSV.
- Relaciones: Es utilizado por *Inventory* para manejar archivos CSV.

8. InventoryRepository

- Función: Gestiona la persistencia del inventario en un archivo binario.
- Relaciones: Es utilizado por *Inventory* para cargar y guardar datos.

9. Validation

- Función: Proporciona métodos para validar entradas del usuario, como números positivos y cadenas no vacías.
- Relaciones: Es utilizado por múltiples clases para validar datos ingresados por el usuario.

10. UserManagement

- Función: Administra el registro, eliminación y autenticación de usuarios, diferenciando entre administradores y usuarios normales.
- Relaciones: Utiliza *UserRepository* para la persistencia de usuarios. Interactúa con *UserAuthentication* para manejar el inicio de sesión.

11. UserAuthentication

- Función: Maneja el proceso de inicio de sesión de los usuarios.
- Relaciones: Es utilizado por *UserManagement* para autenticar usuarios.

12. UserRepository

- Función: Gestiona la persistencia de usuarios y sus jerarquías en un archivo binario.
- Relaciones: Es utilizado por *UserManagement* para cargar y guardar usuarios.

13. SupplierManagement

- Función: Administra las operaciones relacionadas con proveedores, como agregar, modificar, eliminar y listar proveedores.
- Relaciones: Utiliza *Supplier* para representar proveedores. Interactúa con *SupplierRepository* para la persistencia de datos.

14. Supplier

- Función: Representa un proveedor con atributos como nombre, contacto y productos suministrados.
- Relaciones: Es utilizado por *SupplierManagement* para gestionar proveedores.

15. SupplierRepository

- Función: Gestiona la persistencia de proveedores en un archivo binario.
- Relaciones: Es utilizado por *SupplierManagement* para cargar y guardar proveedores.

16. InventoryModificationMenu

- Función: Muestra el menú de opciones para modificar el inventario, como agregar categorías, productos, y exportar/importar datos.
- Relaciones: Interactúa con *Inventory* y *InventoryReport* para realizar modificaciones.

17. SearchMenu

- Función: Muestra el menú de opciones para buscar productos por nombre o categoría.
- Relaciones: Interactúa con *Inventory* para realizar búsquedas.

18. ReportMenu

- Función: Muestra el menú de opciones para generar reportes y visualizar el historial de cambios.
- Relaciones: Interactúa con *Inventory* y *InventoryReport* para generar reportes.

19. SupplierMenu

- Función: Muestra el menú de opciones para gestionar proveedores.
- Relaciones: Interactúa con *SupplierManagement* para realizar operaciones relacionadas con proveedores.