

Universidad de San Carlos de Guatemala

Facultad de Ingeniería Escuela de Ciencias y Sistemas

Laboratorio Introducción a la Programación y Computación 1

Sección: D

Segundo Semestre 2025

Ing. Herman Veliz

Aux. Lesther López

Proyecto 1

Manual Tecnico

Carlos Didiere Cabrera Rodriguez

202401939

Introducción

Este manual técnico describe detalladamente el funcionamiento del sistema de gestión de inventario y ventas desarrollado como parte del Proyecto 1. El propósito del manual es proporcionar a los desarrolladores una guía técnica completa sobre cómo opera el sistema, incluyendo la lógica detrás de cada módulo y función.

Funcionamiento General del Programa

El programa ofrece un sistema interactivo basado en menú para manejar un inventario de productos. Permite registrar nuevos productos asegurando unicidad en los códigos, buscar productos por distintos criterios, realizar ventas verificando disponibilidad, eliminar productos, generar reportes en PDF de inventario y ventas, y visualizar los registros de acciones realizadas mediante una bitácora.

Requerimientos:

1. Sistema Operativo: El programa debería poder ejecutarse en la mayoría de los sistemas operativos, incluidos:

1.2. Windows (7, 8, 10, 11)

1.3. macOS (10.x o versiones posteriores)

1.4. Linux (cualquier distribución moderna)

2. Requisitos de Software: Necesitarás el JDK instalado en la máquina. Este es el entorno de desarrollo necesario para compilar y ejecutar programas Java.

2.1. Versión recomendada: JDK 8 o superior. El programa debería funcionar en versiones más recientes de Java (como JDK 11 o 17) sin problemas, pero en general, JDK 8 es una buena opción de compatibilidad.

3. Un IDE o editor de texto (como IntelliJ IDEA, Eclipse, o incluso VS Code con las extensiones de Java).

Descripción de métodos

1. Registro de nuevo producto Este módulo solicita los siguientes datos del usuario: código único, nombre, categoría, precio y stock. Antes de registrar el producto, el sistema verifica que el código no esté repetido recorriendo el arreglo ``inventario[]``. Si el código es válido y el resto de los datos son correctos ($\text{precio} > 0$ y $\text{stock} \geq 0$), se crea una nueva instancia de ``Producto`` y se agrega al arreglo. También se incrementa el contador ``totalProductos``.
2. Búsqueda de producto Este método permite buscar un producto mediante nombre, categoría o código exacto. El valor ingresado por el usuario se compara con cada producto del arreglo usando ``equalsIgnoreCase()`` (para el código) y ``contains()`` para los demás atributos. Si encuentra coincidencia, muestra los datos del producto usando ``mostrarProducto()`` y registra la búsqueda en la bitácora.
3. Venta de producto Permite al usuario realizar una venta. Se pide el código del producto y la cantidad. Se verifica que el código coincida exactamente (``equalsIgnoreCase``). Luego, se valida si existe suficiente stock. Si es así, se descuenta la cantidad del inventario, se calcula el total de venta (``cantidad * precio``) y se crea un objeto ``Venta``, almacenándolo en el arreglo ``ventas[]``. También se registra la operación en la bitácora.
4. Eliminación de producto Este módulo solicita un código de producto exacto a eliminar. Se recorre el arreglo ``inventario[]`` buscando coincidencia con ``equalsIgnoreCase``. Si se encuentra, se elimina desplazando los elementos siguientes una posición a la izquierda. Se actualiza ``totalProductos`` y se registra en la bitácora.
5. Mostrar Bitácora Este método imprime todas las entradas de la bitácora. Para cada acción almacenada en el arreglo ``acciones[]``, muestra el nombre del usuario, la acción realizada, el resultado (exitosa o fallida) y la fecha/hora.
6. Generación de Reportes PDF Permite generar archivos PDF usando la clase ``ReportePDF``. Pregunta al usuario si desea generar el reporte de Stock o de Ventas. En ambos casos, se llama a los métodos ``generarReporteStock()`` o

``generarReporteVentas()`` con el nombre de archivo generado dinámicamente basado en la fecha y hora actual. El nombre de archivo se genera con ``LocalDateTime.now()`` y ``DateTimeFormatter``.

7. Mostrar información del desarrollador Este caso imprime en consola los datos personales del programador: nombre completo, carné y curso. Es útil como identificador del autor del proyecto.

Método `mostrarProducto()` en clase `Producto` Este método imprime en consola todos los atributos de un producto: nombre, código, categoría, precio y stock. Es llamado durante la búsqueda o visualización de inventario.

Método `registrar()` en clase `Bitacora` Recibe tres parámetros: acción (String), éxito (booleano) y nombreUsuario (String). Crea un objeto de acción y lo agrega al arreglo de bitácora si hay espacio. Almacena también la fecha/hora actual usando ``LocalDateTime.now()``.

Métodos `generarReporteStock()` y `generarReporteVentas()` en `ReportePDF` Ambos métodos reciben como parámetros el arreglo correspondiente (inventario o ventas), su tamaño, y el nombre del archivo. Usan ``com.lowagie.text.Document`` y ``PdfWriter.getInstance()`` para crear un archivo PDF. Iteran sobre los elementos y agregan sus datos al documento en formato legible para el usuario.