

Universidad San Carlos de Guatemala

Introduccion a la Computación y

Programacion

PROYECTO 1

Nombre: Pablo Gabriel Ordoñez Escobar
Carné: 202407178
Seccion: "B"
Fecha: 15/08/2025

Requisitos para hacer uso del programa:

- Sistema operativo: Windows, Linux o macOS con Java instalado.
- Versión de Java: JDK 17 o superior.
- Ejecución: Se puede correr desde NetBeans o desde consola.
- 10 MB para archivos de datos

Estructuras de datos:

Array de Productos:

```
productos = new Producto[100]; //Tamaño inicial  
cantidadProductos = 0;
```

Array de Ventas:

```
ventas = new Venta[100]; //Capacidad inicial de 100 ventas  
cantidadVentas = 0; //Inicia contador en 0
```

Descripción de clases y métodos:

Clase Producto:

Funcion: Representa un producto de la tienda con todos sus atributos.

Atributos:

- String nombre – Nombre del producto
- String categoría – Categoría (camisas, pantalones, etc)
- Double precio – Precio Unitario
- Int cantStock – Cantidad disponible
- String codigo – Codigo unico identificador

```
String nombre;  
String categoria;  
double precio;  
int CantStock;  
String codigo;
```

Métodos:

- Producto(String, String, double, int, String) – Constructor

```
public Producto(String nombre, String categoria, double precio, int CantStock, String codigo){  
    this.nombre = nombre;  
    this.categoria = categoria;  
    this.precio = precio;  
    this.CantStock = CantStock;  
    this.codigo = codigo;  
}
```

- Void mostrarInfo() – Muestra informacion del producto en consola

```
//Mostrar informacion del producto
public void mostrarInfo(){
    System.out.println("Codigo: "+ codigo);
    System.out.println("Nombre: "+ nombre);
    System.out.println("Categoria: "+ categoria);
    System.out.println("Precio: "+ precio);
    System.out.println("Stock: "+ CantStock);
    System.out.println("-----");
}
```

Clase Inventario:

Función: Gestiona el inventario de productos con operaciones CRUD

Métodos Públicos:

- *void agregarProducto(Scanner entrada)*
 - Funcion: Agrega un nuevo producto al inventario
 - Validaciones: Código único, precios positivos, stock no negativo
 - Persistencia: Guarda automáticamente en archivo
- *Producto buscarPorCodigo(String código)*
 - Función: Busca producto por código único
 - Retorno: Objeto Producto o null si no existe
- *void buscarProductos(Scanner entrada)*
 - Función: Búsqueda por código, nombre o categoría
 - Interfaz: Menú interactivo con usuario
- *void eliminarProducto(Scanner entrada)*
 - Función: Elimina producto con confirmación
 - Validación: Verifica existencia del producto
- *Producto obtenerProducto(int índice)*
 - Función: Acceso seguro a productos del array
 - Validación: Verifica índices válidos
- *Int obtenerCantidadProductos()*
 - Función: Retorna cantidad actual de productos

Métodos Privados:

- *void guardarEnArchivo()*
 - Función: Persiste inventario en archivo texto
 - Formato: *nombre|categoria|precio|stock|codigo*
- *void cargarDesdeArchivo()*
 - Función: Carga inventario desde archivo al iniciar
 - Tolerancia: Maneja archivo inexistente o corrupto

Clase Venta:

Función: Representa una transacción de venta.

Atributos:

- String codigoProducto - Código del producto vendido
- int cantidadVendida - Cantidad vendida
- String fechaHora - Fecha y hora de la transacción
- double total - Total de la venta

Métodos:

- Venta(String, int, double) - Constructor con fecha automática
- Venta(String, int, double, String) - Constructor con fecha específica
- void mostrarInfo() - Muestra información de la venta

Clase GestorVentas:

Funcion: Gestiona el proceso de ventas y transacciones

Métodos Públicos:

- void registrarVenta(Scanner scanner)
 - Función: Registra nueva venta
 - Validaciones: Stock suficiente, producto existe
 - Efectos: Actualiza inventario y persiste venta
- Venta obtenerVenta(int indice)
 - Función: Acceso seguro a ventas del array
- int obtenerCantidadVentas() - Retorna cantidad de ventas

Métodos Privados:

- void guardarVenta(Venta venta)
 - Función: Persiste venta en archivo texto
 - Formato: codigoProducto|cantidad|total|fechaHora
- void cargarVentas()
 - Función: Carga historial de ventas al iniciar

Clase bitácora:

Función: Registro de auditoría de acciones del sistema

Métodos Estáticos:

- void registrarAccion(String tipoAccion, String resultado, String usuario)
 - Función: Registra acción en bitácora
 - Formato: fechaHora|tipoAccion|resultado|usuario
 - Persistencia: Append en archivo texto
- void mostrarBitacora()
 - Función: Muestra contenido completo de la bitácora

Clase GeneradorReportes:

Función: Genera reportes en formato texto legible

Métodos Estáticos:

- *void generarReporteStock(Inventario inventario)*
 - Función: Genera reporte completo de inventario
 - Nombre archivo: DD_MM_YYYY_HH_mm_ss_Stock.txt
- *void generarReporteVentas(GestorVentas gestorVentas)*
 - Función: Genera reporte histórico de ventas
 - Nombre archivo: DD_MM_YYYY_HH_mm_ss_Venta.txt

Clase Main:

Función: Punto de entrada y menú principal de la aplicación

Métodos:

- *void main(String[] args)* - Método principal
- *void mostrarMenu()* - Muestra menú de opciones

Flujo Principal:

1. Inicializa inventario y gestor de ventas
2. Carga datos desde archivos
3. Muestra menú interactivo
4. Procesa opciones del usuario
5. Registra acciones en bitácora
6. Persiste datos al finalizar

Archivos de datos:

- *inventario.txt* - Productos y stock
- *ventas.txt* - Historial de ventas
- *bitacora.txt* - Registro de acciones del sistema