

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE  
ESMERALDAS**

**INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**

Programación Estructura y Funcional

**TEMA:**

Actividad #2: Sistema de Gestión de Inventario Avanzado

**AUTORES:**

Lilian Velásquez Quiñónez, Xavier Ballesteros Cárdenas, Aida Parrales  
Guerrero

**DOCENTE:**

Mgt. Esteban Gonzabay

**NIVEL:**

Segundo semestre

## EXPLICACIÓN DEL CÓDIGO FUENTE “TECNOLOGIX GLOBAL”

El programa inicia importando la clase **Scanner**, luego se declara la clase principal **TecnoLogixGlobal**. Justo al entrar al ejecutar el **main**, se crea un objeto Scanner sc que es el encargado de leer todos los valores proporcionados por el usuario.

A continuación, se definen variables básicas para el funcionamiento del sistema. Primero, maxProductos y diaOperacion, que permiten almacenar cuántos productos se registrarán y qué día de la semana se está realizando la operación. También se declara aplicaTCE, un valor booleano inicializado en false y que más adelante indicará si debe aplicarse la **Tasa de Control Especial (TCE) del 2%**, exclusiva de los días sábado o domingo.

Posteriormente, se inicializan los acumuladores: totalIVA, totalIMC, totalIRT, totalTCE y totalVentas. Estos valores parten en cero y servirán para sumar el total recaudado por cada impuesto y el monto total de ventas del día. También se definen cuatro contadores (cat1, cat2, cat3, cat4) usados para registrar cuántos productos pertenecen a cada categoría. Para identificar el producto con el precio final más alto, se declaran las variables productoMayor y mayorPrecioFinal, que se actualizarán conforme avance el proceso.

Para el desafío extra del sistema, se implementa un ranking de los **tres productos con mayor stock**. Debido a la restricción de no usar arreglos ni listas, se crean variables simples: rnk1, rnk2, rnk3 para los nombres, y stk1, stk2, stk3 para los valores de stock. Estos últimos se inicializan en **-1** para garantizar que cualquier stock real (mínimo 1) los supere durante las comparaciones.

El programa inicia la interacción con el usuario solicitando el número máximo de productos a registrar (maxProductos) y el día de operación (diaOperacion). Si el día ingresado es **6 o 7**, se activa aplicaTCE = true, indicando que el impuesto extra de 2% debe aplicarse. Para evitar problemas al leer cadenas de texto después de números, se limpia el buffer con sc.nextLine().

A partir de aquí comienza un **ciclo for**, que se repetirá tantas veces como productos se registren. Para cada producto, el programa muestra un encabezado con el número de registro y solicita:

1. Nombre del producto
2. Precio base (validado para ser mayor que 0)
3. Categoría (solo valores entre 1 y 4)
4. Riesgo (0 = no, 1 = sí)
5. Stock (al menos 1)

Cada uno de estos campos está protegido con un ciclo **do-while** para asegurar que el usuario no ingrese datos inválidos.

Una vez validados los datos, se procede a calcular los impuestos:

- **IVA (12%)**, excepto para productos de **categoría 3**, que están exentos.
- **IMC (1.5%)**, aplicado siempre.
- **IRT**, solo para productos con riesgo (riesgo = 1).
- **TCE (2%)**, únicamente si el día es sábado o domingo.

Luego se calcula el **precio final**, que es la suma del precio base y todos los impuestos aplicados.

Con el precio final obtenido, se actualizan los acumuladores del sistema: se suman IVA, IMC, IRT y TCE a sus totales correspondientes, y se agrega al total de ventas el valor del precio final

multiplicado por el stock. También se incrementa el contador de la categoría correspondiente para llevar control estadístico.

Para identificar el producto más costoso, se compara el precio final actual con el valor de mayorPrecioFinal. Si es mayor, el programa actualiza tanto este valor como el nombre almacenado en productoMayor.

Luego se actualiza el ranking de los tres mayores stocks. Como no se utilizan estructuras de datos avanzadas, el proceso se realiza mediante comparaciones:

- Si el stock actual es mayor que stk1, todos los puestos se recorren hacia abajo y el producto ocupa el primer lugar.
- Si no supera al primero, pero sí a stk2, se desplaza el segundo al tercer lugar y el actual pasa al segundo.
- Si únicamente supera al tercero, se asigna al tercer puesto.

De esta manera se imita un ranking sin usar arreglos ni listas.

Al finalizar cada iteración, se imprime un **reporte individual** del producto con todos sus datos e impuestos calculados, y nuevamente se limpia el buffer para evitar errores en la siguiente lectura de texto.

Una vez que concluye el ciclo y se han registrado todos los productos, el programa identifica la **categoría más utilizada**, comparando los valores de cat1, cat2, cat3 y cat4. Esta comparación se realiza mediante condicionales simples.

Finalmente, el programa presenta un **reporte general del día**, que incluye:

- Total de productos registrados
- Total recaudado por cada impuesto
- Monto total de ventas
- Categoría con mayor cantidad de productos
- Producto con el mayor precio final
- Ranking de los tres productos con mayor stock

Al concluir el reporte, el programa cierra el objeto Scanner utilizando sc.close(), completando así su ejecución.

Este diseño cumple con todas las restricciones: no utiliza arreglos, listas, clases adicionales ni programación orientada a objetos; se basa únicamente en lectura de datos, secuencias, ciclos y condicionales.

## Descripción general

El programa es una aplicación gráfica sencilla que permite registrar productos con sus datos, calcula impuestos y genera reportes, todo mediante ventanas de diálogo (JOptionPane). Incluye validaciones de entrada y muestra mensajes de error y confirmación para mejorar la experiencia.

## Flujo y estructura

### Configuración inicial:

Pide al usuario ingresar el límite máximo de productos a registrar.

Valida que el límite sea mayor a 0, mostrando error si no.

Pide el día de operación (1 a 7, representando días de la semana).

Valida que el día sea válido, mostrando error si no.

Muestra una ventana de confirmación para empezar el registro o cancelar la operación.

### Variables y acumuladores:

Inicializa variables para acumular impuestos (IVA, IMC, IRT, TCE), monto total de ventas y contar productos por categoría.

Variables para rastrear el producto con mayor precio final y, adicionalmente, para un "Desafío Extra 1" que registra el producto con mayor stock y mayor precio.

### Bucle de registro de productos (según el límite ingresado):

Para cada producto, muestra ventanas para ingresar:

Nombre

Precio base (validado > 0)

Categoría (1 a 4, con nombres específicos)

Riesgo tecnológico (0 o 1)

Stock (mínimo 1)

Cada entrada se valida; si es incorrecta, muestra mensaje de error y pide nuevamente.

Calcula impuestos aplicables según categoría, riesgo y día:

IVA (12%, excepto categoría 3 que es exenta)

IMC (1.5%)

IRT (3.7% si tiene riesgo tecnológico)

TCE (2% si es fin de semana, día 6 o 7)

Calcula el precio final sumando base e impuestos.

Muestra un reporte con toda esta info para el producto actual.

Actualiza acumuladores y contadores.

Actualiza registros para el producto con mayor precio, y para el ranking de stock y precio.

Después del registro:

Determina qué categoría tiene más productos registrados.

Construye y muestra un reporte final que incluye:

Total acumulado de impuestos recaudados.

Total acumulado de ventas.

Categoría con más productos.

Producto con mayor precio final.

Ranking extra: producto con mayor stock y producto con mayor precio final.

Cierre:

Muestra mensaje de despedida indicando que la operación finalizó correctamente.

Resumen de funcionalidades importantes

Validaciones robustas: asegura que los datos ingresados estén en rangos válidos, mostrando errores y reintentando.

Interacción amigable: usa íconos, títulos y emojis en los cuadros de diálogo para mejorar la experiencia.

Cálculos financieros: aplica impuestos específicos y calcula el precio final por producto.

Reportes inmediatos y finales: informa al usuario producto por producto y al final del día con datos resumidos.

Ranking adicional: identifica productos con mayor stock y precio para reportes extra.

Esta estructura modular y ordenada hace que el programa sea fácil de entender, usar y mantener, con ventanas intuitivas que guían al usuario paso a paso.