

GUÍA DE EJERCICIOS

Nº 10

ING. LUISINA DE PAULA

**DESARROLLO
WEB FULLSTACK
CON JAVA**

Guías de Ejercicios Nº 10: Collections + Excepciones

1) Collections

- a) Un vendedor de bazar desea un programa para administrar los productos con los que cuenta en stock. Para ello, necesita la creación de una clase Producto que cuente con los siguientes datos: código, nombre, marca, tipo, precio costo, precio venta, cantidad.

A partir de esto, realizar las siguientes acciones en el programa desarrollado:

- ✓ Crear 10 objetos de tipo producto con sus correspondientes valores cargados (por ejemplo, vaso, jarra, mantel, cuchillos, tenedores, etc)
- ✓ Guardar estos objetos creados en un ArrayList.
- ✓ Recorrer el ArrayList y determinar el producto con mayor precio de venta.
- ✓ Recorrer el ArrayList y determinar el producto con menor precio de costo.
- ✓ Borrar el producto que se encuentre en la posición 5 de la ArrayList.
- ✓ Determinar el producto que tenga mayor cantidad en stock. A ese producto, descontarle 3 unidades del stock y actualizar el registro de la lista en base a eso.

- b) Crear una pila (stack) que contenga los números del 1 al 7. A partir de esto realizar las siguientes acciones:

- ✓ Consultar si la pila está vacía y mostrar el resultado por pantalla
- ✓ Mostrar por pantalla el último registro cargado en la pila
- ✓ Borrar el último registro cargado en la pila
- ✓ Mostrar por pantalla el nuevo último registro cargado en la pila
- ✓ Agregar el número 8 a la pila
- ✓ Consultar si se encuentra el número 7 en la pila. Mostrar por pantalla el resultado.

2) Excepciones

Ejercicio Nº 1

- Crear una clase calculadora que tenga los siguientes métodos:
 - ✓ calcularSuma (double num1, double num2)
 - ✓ calcularResta (double num1, double num2)
 - ✓ calcularMultiplicacion (double num1, double num2)
 - ✓ calcularDivision(double num1, double num2)
- Contemplar en el método calcularDivision, la posibilidad de que se produzca una excepción. Para ello utilizar el bloque try/catch,
- Realizar 3 sumas, 1 resta, 2 multiplicaciones y una división. A partir de eso, almacenar los resultados de cada operación en un vector de 6 posiciones y mostrar los resultados por pantalla. Contemplar, para este caso, la posibilidad de un error por desbordamiento utilizando el bloque try/catch.
- Modificar el método calcularDivision para que en lugar de utilizar el bloque try/catch utilice la cláusula “throws”.
- Arrojar una excepción “aritmética” (AritmethicException) en el método main. Utilizar para ello la cláusula “throw”. Luego de esto... ¿Qué ocurre con el programa?